

681.3(075)

1-74

БАЖЕНОВ В.А., ВЕНГЕРСЬКИЙ П.С., ГАРВОНА В.С.,
ГОРЛАЧ В.М., ДУДЗЯНИЙ І.М., КОРКУНА М.Д.,
КРАВЧУК С.О., ЛЕВЧЕНКО О.М., ЛІЗУНОВ П.П.,
РЕЗНИКОВ А.С., ШОНІН В.О.

ІНФОРМАТИКА

Комп'ютерна техніка

Комп'ютерні технології



ПІДРУЧНИК



ВИДАВНИЦТВО
“КАРАВЕЛА”

Львівський національний університет ім. І. Франка
Національний технічний університет України
«Київський політехнічний інститут»
Київський національний університет будівництва і архітектури
Національний університет «Львівська політехніка»

ІНФОРМАТИКА.

Комп'ютерна техніка.

Комп'ютерні технології

Затверджено
Міністерством освіти і науки України
як підручник
для студентів вищих навчальних закладів

Четверте видання

Київ «Каравела» 2012

УДК 004(075.8)
ББК 32.97я73
І 74

Гриф надано
Міністерством освіти і науки України
(лист № 1/11-1833 від 07.05.2008 р.)

Авторський колектив:

В.А. Баженов, П.П. Лізунов (розд. 1, 3); **А.С. Резніков** (розд. 3, 4);
С.О. Кравчук, В.О. Шонін (розд. 1, 2, 13, 14); **І.М. Дудзяний** (розд. 5);
О.М. Левченко (розд. 6); **В.М. Горлач** (розд. 7, 12); **М.Д. Коркуна**
(розд. 8); **В.С. Гарвона** (розд. 9, 11); **П.С. Венгерський** (розд. 10)

Наукова редакція:

д.ф.-м.н., проф. **Г.А. Шинкаренко**,
к.т.н., проф. **О.В. Шишов**

Рецензенти:

д.т.н., проф. **Ю.В. Веружський**
к.ф.-м.н., доц. **А.В. Костенко**

І 74 Інформатика. Комп'ютерна техніка. Комп'ютерні технології: Підручник. 4-те вид. – К.: Каравела, 2012. – 496 с.

ISBN 966-8019-05-9

Підручник містить основи концепції та методів інформатики, які реалізовані в сучасному апаратному та програмному забезпеченні комп'ютерів. Розглянуто роботу операційних систем Windows XP та Linux. В доступній формі описано принципи роботи з програмами сімейства Microsoft Office: Word, Excel, Access, PowerPoint. Описано роботу з програмами оптичного розпізнавання, перекладу та перевірки правопису тексту. Окремий розділ присвячено питанням комп'ютерної графіки. Значну увагу приділено програмуванню в середовищі Object Pascal. Також розглянуто основи мережних технологій, роботу в глобальній мережі Internet та Web-технології.

Для студентів вищих навчальних закладів гуманітарних, соціально-економічних, інженерно-технічних, медичних, педагогічних та природничих спеціальностей, аспірантів та викладачів, а також усіх тих, хто вирішив зробити досягнення інформаційних технологій повноправним елементом свого арсеналу професійних інтересів.

УДК 004(075.8)
ББК 32.97я73

© Баженов В.А., Венгерський П.С.,
Горлач В.М. та ін., 2012
© Видавництво «Каравела», 2012

ISBN 966-8019-05-9

ЗМІСТ

Передмова	8
1. ОСНОВИ ІНФОРМАТИКИ.....	9
1.1. Основні положення інформатики як галузі наукового знання	9
1.2. Історія розвитку інформатики та комп'ютерної техніки	10
1.3. Логічна структура комп'ютера.....	16
1.4. Комп'ютерне представлення даних	17
2. ОСНОВИ КОМП'ЮТЕРНОЇ ТЕХНІКИ	19
2.1. Принципи компонування персональних комп'ютерів	19
2.2. Материнська плата	22
2.3. Центральний процесор	26
2.4. Внутрішня пам'ять	31
2.5. Зовнішня пам'ять.....	34
2.6. Монітор та відеокарта.....	40
2.7. Аудіосистема комп'ютера.....	47
2.8. Пристрої введення.....	48
2.8.1. Клавіатура	48
2.8.2. Маніпулятори.....	49
2.8.3. Сканери	53
2.8.4. Цифрові фотокамери.....	55
2.8.5. Відеокамери	57
2.9. Пристрої виведення.....	58
2.9.1. Принтери.....	58
2.9.2. Плотери	63
2.9.3. Проектори	63
2.10. Допоміжні пристрої комп'ютера.....	64
3. ОПЕРАЦІЙНА СИСТЕМА Windows XP	69
3.1. Загальні положення.....	69
3.2. Робочий стіл та його елементи.....	71
3.3. Вікна Windows	74
3.4. Головне меню Windows.....	79
3.5. Файли та папки Windows XP, основні прийоми роботи	81
3.6. Параметри налагодження вигляду об'єктів ОС.....	104
3.7. Стандартне програмне забезпечення.....	115
4. ОПЕРАЦІЙНА СИСТЕМА Linux.....	117
4.1. Можливості та переваги Linux.....	117
4.2. Дистрибутиви Linux	119

4.3. Перше знайомство з Linux.....	119
4.4. Робота з файлами та каталогами в Linux через командний рядок.....	124
4.5. Графічна оболонка KDE.....	130
5. ОСНОВИ ПРОГРАМУВАННЯ МОВОЮ Object Pascal.....	139
5.1. Основні конструкції мови.....	139
5.2. Галуження і цикли.....	147
5.3. Модульне програмування.....	152
5.4. Структуровані типи даних.....	157
5.5. Об'єктно-орієнтоване програмування.....	166
5.6. Програмування віконних застосунів.....	172
6. ТЕКСТОВИЙ ПРОЦЕСОР Microsoft Word.....	183
6.1. Інтерфейс програми Word.....	184
6.2. Робота з документами.....	185
6.3. Введення і редагування тексту.....	192
6.4. Форматування тексту.....	195
6.4.1. Форматування символів.....	196
6.4.2. Форматування абзаців.....	198
6.4.3. Форматування списків.....	204
6.5. Пошук і заміна фрагментів тексту.....	208
6.6. Використання мовних функцій Word.....	211
6.7. Таблиці.....	213
6.7.1. Створення таблиць.....	213
6.7.2. Робота з елементами таблиці.....	215
6.7.3. Робота із вмістом таблиці.....	219
6.8. Вставлення та форматування ілюстрацій.....	221
6.9. Математичні формули.....	225
6.10. Основи верстання. Вставлення тексту.....	228
6.11. Формування зовнішнього вигляду документа.....	229
6.11.1. Оформлення сторінок.....	229
6.11.2. Розділи.....	232
6.11.3. Колонки.....	233
6.11.4. Нумерація сторінок і колонтитули.....	234
6.12. Робота зі стилями.....	236
6.12.1. Вбудовані стилі документа.....	236
6.12.2. Створення та модифікація стилів.....	239
6.13. Шаблони.....	241
6.13.1. Шаблон документа.....	241
6.13.2. Створення і модифікація шаблонів.....	243
6.14. Робота з ілюстраціями та ілюстративними засобами.....	244
6.14.1. Ілюстрації як незалежні зображення.....	244

6.14.2. Текстові рамки	247
6.14.3. Ілюстративні засоби програми Word	249
6.14.4. Текстові спецефекти.....	251
6.15. Виноски	252
6.16. Зміст документа.....	253
6.17. Макрокоманди	255
6.18. Друкування документа.....	257
7. ТАБЛИЧНИЙ ПРОЦЕСОР Microsoft Excel	259
7.1. Основні відомості.....	259
7.2. Початок роботи з Microsoft Excel.....	260
7.3. Побудова таблиць	267
7.3.1. Способи адресації даних	267
7.3.2. Уведення, редагування та опрацювання даних	269
7.3.3. Робота з клітинками таблиці	274
7.3.4. Форматування клітинок.....	276
7.4. Бази даних в Excel	282
7.4.1. Створення списків. Використання форм.....	282
7.4.2. Використання фільтра.....	286
7.4.3. Сортування даних.....	287
7.4.4. Використання команди Підсумки	287
7.4.5. Зведені таблиці	290
7.4.6. Формати файлів баз даних.....	293
7.5. Імпорт даних з текстових файлів	293
7.6. Ділова графіка.....	296
7.7. Друкування аркуша	302
7.8. Макрокоманди	307
Додатки.....	308
8. РОБОТА З БАЗАМИ ДАНИХ. Microsoft Access	321
8.1. Створення бази даних	322
8.2. Таблиці.....	323
8.2.1. Створення таблиць за допомогою майстра Table Wizard	324
8.2.2. Створення таблиць в режимі Design View (Режим конструктора)	326
8.2.3. Властивості полів	331
8.2.4. Ключові поля	339
8.2.5. Режим таблиць.....	340
8.2.6. Введення зовнішніх таблиць	341
8.2.7. Міжтабличні зв'язки	342
8.3. Запити QBE	344
8.3.1. Запити на вибірку	345
8.3.2. Запити з обчислювальними полями.....	346

8.3.3. Запити для впорядкування записів в таблиці	347
8.3.4. Групувальні запити	348
8.3.5. Запити з умовами вибору (параметрами)	349
8.3.6. Запити на створення таблиці	351
8.3.7. Перехресні запити	353
8.3.8. Запити вилучення	355
8.3.9. Запити на оновлення записів в таблицях	356
8.3.10. Запити на додавання записів	357
8.3.11. Запити, що здійснюють вибірку записів, що повторюються	358
8.3.12. Запити, що здійснюють вибірку записів, які не мають підпорядкованих	361
8.3.13. Властивості запитів	363
9. СТВОРЕННЯ ПРЕЗЕНТАЦІЙ. Microsoft PowerPoint	367
9.1. Поняття презентації	367
9.2. Початок роботи з Microsoft PowerPoint	367
9.3. Головне вікно PowerPoint	371
9.4. Підготовка презентації	372
9.5. Приклад створення презентації	376
9.6. Режим сортувальника. Демонстрація	379
10. КОМП'ЮТЕРНА ГРАФІКА	383
10.1. Основні визначення, поняття та пристрої комп'ютерної графіки	383
10.2. Растровий графічний редактор Adobe Photoshop	389
10.2.1. Вікно програми. Робота з документами	389
10.2.2. Панелі інструментів	392
10.2.3. Створення виділених областей	396
10.2.4. Робота з виділеними областями	400
10.2.5. Малювання	404
10.2.6. Редагування зображень	408
10.2.7. Робота з контурами	411
10.2.8. Фільтри	414
10.3. Векторний графічний редактор CorelDRAW	416
10.3.1. Інтерфейс CorelDraw	416
10.3.2. Огляд панелі інструментів	418
10.3.3. Друкування документа	420
10.3.4. Малювання об'єктів	421
10.3.5. Робота з об'єктами	429
10.3.6. Заповнення	434
10.3.7. Контури об'єктів	440
10.3.8. Робота з текстом	441

11. ОПРАЦЮВАННЯ ТЕКСТОВОЇ ІНФОРМАЦІЇ: РОЗПІЗНАВАННЯ, РЕДАГУВАННЯ, ПЕРЕКЛАД.....	443
11.1. Система оптичного розпізнавання тексту FineReader.....	443
11.1.1. Робоче вікно ABBYY FineReader	443
11.1.2. Сканування зображень	446
11.1.3. Сегментування зображень	447
11.1.4. Процес розпізнавання	449
11.1.5. Редагування розпізнаного тексту	450
11.2. Системи перевірки правопису та перекладу ProLing Office, Pragma	453
12. ОСНОВИ КОМП'ЮТЕРНИХ МЕРЕЖ.....	457
12.1. Мережні сервіси	458
12.2. Середовища передавання.....	459
12.3. Мережні топології	460
12.4. Методи доступу до середовища передавання.....	461
12.5. Методи передавання даних.....	462
12.6. Принципи взаємодії відкритих систем. Модель OSI	463
12.7. Мережні операційні системи.....	467
13. ГЛОБАЛЬНА МЕРЕЖА INTERNET	469
13.1. Сервіси Internet	469
13.2. Доступ до віддаленого комп'ютера в Інтернет. Протокол Telnet.....	470
13.3. Передача файлів через Інтернет. Протокол FTP.....	471
13.4. Електронна пошта в Інтернет.....	473
13.5. Служба новин в Інтернет.....	475
13.5.1. Способи розповсюдження новин в Інтернет.....	475
13.5.2. Програмні засоби супроводу списків розсилки.....	477
13.5.3. Програмні засоби телеконференцій UseNet.....	478
13.6. Служба Web.....	480
13.7. Інтерактивні служби Інтернет	484
13.7.1. Обмін повідомленнями	484
13.7.2. IP-телефонія	484
14. ПРОЕКТУВАННЯ WEB-СТОРИНОК. ВВЕДЕННЯ В HTML... 	487
14.1. Мови розмітки документів	487
14.2. Технологія і засоби проектування Web-сторінок	490
14.3. Введення в HTML (XHTML).....	491
Список літератури	495

ПЕРЕДМОВА

Комп'ютер та Інтернет було визнано одними з найбільш значних досягнень людства у ХХ ст. За цими двома термінами інформатики стоїть фантастичний здобуток сучасної науки, технологій та бізнесу, який не лише докорінно реорганізував структуру основних сфер діяльності людини, а й змінив її світогляд та спосіб життя.

Ми живемо в світі інформаційних технологій. Багато людей сьогодні бояться відстати від життя і залишитися за бортом інформаційного суспільства. Мета даного підручника – запропонувати швидкий, компетентний та впорядкований шлях до подолання страху перед комп'ютерними технологіями. У ньому фахово та доступно подано матеріал, необхідний для успішної роботи з комп'ютером. Розглянуто роботу операційних систем Windows XP та Linux. Описано принципи роботи з програмними продуктами для роботи в середовищі Windows: текстовим процесором Microsoft Word, табличним процесором Microsoft Excel, програмою для роботи з базами даних Microsoft Access, програмою для створення презентацій Microsoft PowerPoint, програмою оптичного розпізнавання тексту ABBYY FineReader, засобами перевірки правопису та перекладу ProLing Office, Pragma. Окремий розділ присвячено питанням комп'ютерної графіки: розглянуто роботу з редактором растрової графіки Adobe Photoshop та редактором векторної графіки CorelDRAW. Значну увагу приділено програмуванню в середовищі Object Pascal. Розглянуто основи мережних технологій, роботу в глобальній мережі Internet та Web-технології.

Підручник призначений для студентів вищих навчальних закладів гуманітарних, соціально-економічних, інженерно-технічних, медичних, педагогічних та природничих спеціальностей, аспірантів та викладачів, а також усіх тих, хто вирішив зробити досягнення інформаційних технологій повноправним елементом свого арсеналу професійних інтересів.

Авторський колектив: д.т.н., проф. **В.А. Баженов**, д.т.н., проф. **П.П. Лізунов**, **А.С. Резніков**, **В.С. Гарвона** (Київський національний університет будівництва і архітектури); к.т.н., доц. **С.О. Кравчук**, к.т.н., доц. **В.О. Шонін** (Національний технічний університет України «Київський політехнічний інститут»); к.ф.-м.н., доц. **П.С. Венгерський**, к.ф.-м.н., доц. **В.М. Горлач**, к.ф.-м.н., доц. **І.М. Дудзяний**, к.ф.-м.н., доц. **М.Д. Коркуна** (Львівський національний університет ім. І. Франка); к.т.н., доц. **О.М. Левченко** (Національний університет «Львівська політехніка»).

Наукова редакція: д.ф.-м.н., проф. **Г. А. Шинкаренко** (Львівський національний університет ім. І. Франка); к.т.н., проф. **О.В. Шишов** (Київський національний університет будівництва і архітектури).

1. ОСНОВИ ІНФОРМАТИКИ

📖 План викладу матеріалу:

1. Основні положення інформатики як галузі наукового знання.
2. Історія розвитку інформатики та комп'ютерної техніки.
3. Логічна структура комп'ютера.
4. Комп'ютерне представлення даних.

↪ Ключові терміни розділу

▼ <i>Інформатика</i>	▼ <i>Комп'ютер</i>
▼ <i>Комп'ютерна техніка</i>	▼ <i>Логічна схема комп'ютера</i>
▼ <i>Інформація</i>	▼ <i>Біт</i>
▼ <i>Інформаційне суспільство</i>	▼ <i>Байт</i>

1.1. Основні положення інформатики як галузі наукового знання

Інформатика – це галузь науково-технічної діяльності людини, яка вивчає структуру і загальні властивості інформації, а також закономірності та методи її створення, зберігання, пошуку, оброблення, представлення і використання в різних сферах людської діяльності.

Термін «інформатика» походить від франц. *informatique* (інформація) та *automatique* (автоматика). Інформатика як наука базується на обчислювальній техніці, дає знання про методи створення та функціонування в соціальному середовищі інформаційних систем різного класу і призначення, що являють собою нову технологію збору, обробки, передачі інформації, яка використовує електронні обчислювальні машини (ЕОМ), охоплює всі види суспільної діяльності – виробництво, управління, науку, освіту, проектні розробки, торгівлю, медицину, криміналістику, охорону навколишнього середовища і т.ін., а також побут.

Комп'ютерна техніка розглядається як сукупність технічних засобів (ЕОМ, пристроїв, приладів), призначених для автоматизації процесів обробки інформації, математичних обчислень, розв'язку задач, що потребують великого обсягу обчислень в автоматизованих системах управління, обліку, плануванні, статистиці, для оцінки та прогнозування, прийняття рішень, обробки даних експериментів, в інформаційно-пошукових системах і т.ін.

➤ Як самостійна наука інформатика дійсна тоді, коли для явища, що вивчається, побудована так звана інформаційна модель. Інформатика дає загальні методологічні принципи побудови таких моделей, але сама побудова і обґрунтування моделі є завданням окремої науки. На відміну від

кібернетики, інформатика займається інформаційним, технічним, програмним наповненням кібернетичних систем, технологією їх розробки та машинного використання. Тому в інформатиці виділяються змістова, організаційна, технічна, алгоритмічна і програмна галузі.

Інформація – це відомості про об'єкти і явища навколишнього середовища, їх параметри, властивості та стан.

В інформатиці ці відомості мають назву даних, оскільки очікується їх подання в ЕОМ для подальшого використання або обробки. При розв'язанні обчислювальних задач даними є числа. При розв'язанні задач, пов'язаних з інформаційним пошуком, редагуванням, перекладом, плануванням і т.п., в ЕОМ подається, зберігається та обробляється нечислова інформація: тексти, малюнки, графіки, різноманітні списки і таблиці. Вони і є даними для ЕОМ. Для машинізованої обробки дані подаються у вигляді величин або їх сукупності. Світовий парк обчислювальних машин зайнятий обробкою даних в набагато більшому обсязі, ніж обчисленнями, а створення і експлуатація інформаційних ресурсів більш ніж на 90% виконується ЕОМ.

Інформація в сучасному суспільстві стала важливим економічним, соціальним і політичним ресурсом. Саме тому особливе значення набувають організації, підприємства і системи, зайняті збором, обробкою та передачею інформації. До них належать органи науково-технічної інформації, автоматизовані системи управління, інформаційно-обчислювальні центри, засоби і системи зв'язку, телебачення, радіомовлення та ін. Весь цей складний комплекс у цілому складає інформаційну інфраструктуру суспільства.

Інформаційне суспільство – це суспільство, до всіх сфер діяльності членів якого входять комп'ютери та інші засоби збереження, обробки і передачі інформації, що дозволяють із великою швидкістю виконувати обчислення і переробляти будь-яку інформацію (цифрову, текстову, графічну), моделювати реальні процеси, явища, події.

Для життя в інформаційному суспільстві, використання комп'ютера як знаряддя інтелектуальної праці людині необхідна системна сукупність знань і вмінь, яка містить практичні навички спілкування з ЕОМ, розуміння і знання загальних принципів побудови і функціонування ЕОМ, принципів дії основних елементів ЕОМ, поняття про програму як форму подання алгоритму для ЕОМ та вміння використовувати сучасні прикладні програми.

1.2. Історія розвитку інформатики та комп'ютерної техніки

Незважаючи на бурхливий технічний прогрес, закладення фундаменту комп'ютерної революції відбувалося повільно і далеко не просто. Початком цього процесу можна вважати винахід *рахівниці* понад 1 500 років

тому у країнах Середземномор'я. Рахівниці виявилися дуже ефективним інструментом і незабаром поширилися по всьому світу.

Наступним після рахівниць пристроєм, що полегшує множення і ділення чисел та деякі інші розрахунки, стала *логарифмічна лінійка*, винайдена наприкінці 1620-х років.

Перший *механічний рахунковий пристрій* створив у 40-х роках XVII ст. видатний французький математик, фізик, письменник і філософ *Блез Паскаль* (на його честь названо одну з найпоширеніших сучасних мов програмування). Підсумовувальна машина Паскаля – «паскалина» являла собою скриню з численними шестеренками.

Першу машину, що дозволяла легко виконувати розрахунки, множення і ділення, – *механічний калькулятор*, винайшов у 1673 р. у Німеччині *Готфрід-Вільгельм Лейбніц*. Надалі конструкцію механічного калькулятора видозмінювали і доповнювали вчені та винахідники різних країн (зокрема, ручне обертання каретки в *електромеханічному калькуляторі* було замінено на привод від вбудованого в цей калькулятор електродвигуна).

З усіх винахідників минулих часів, які зробили той чи той внесок у розвиток обчислювальної техніки, найближче до створення комп'ютера в сучасному його розумінні підійшов англієць *Чарльз Беббідж*. У 1822 р. Беббідж опублікував наукову статтю з описом машини, здатної розраховувати і друкувати великі математичні таблиці. Того ж року він побудував пробну модель своєї *різницевої машини*, що складалася із шестерень і валиків, які обертали вручну за допомогою спеціального важеля. Протягом наступного десятиліття Беббідж, невтомно працюючи над своїм винаходом, намагався її реалізувати на практиці. Однак, продовжуючи працювати в цьому напрямі, він прийшов до ідеї створення ще більш потужної машини, яку назвав аналітичною.

Аналітична машина Беббіджа на відміну від своєї попередниці мала не просто розв'язувати математичні задачі одного типу, але й виконувати різноманітні обчислювальні операції відповідно до інструкцій, задаваних оператором. Аналітична машина повинна була мати такі компоненти, як «млин» і «склад» (за сучасною термінологією – арифметичний пристрій і пам'ять), що складаються з механічних важельців і шестерень. Інструкції, чи команди вводилися в аналітичну машину за допомогою перфокарт (аркушів картону з пробитими в них отворами), уперше використаних у 1804 р. французьким інженером *Жозефом Марі Жаккардом* для керування роботою ткацьких верстатів.

Однією з тих, хто розумів, як працює машина і які потенційні галузі її застосування, була *графиня Лавлейс*, уроджена Огаста Ада Байрон, єдина законна донька поета лорда Байрона (на її честь названо одну з мов програмування – *Ada*). Графиня, маючи неабиякі математичні та літературні здібності, всіляко сприяла здійсненню проекту Беббіджа.

Лише через 19 років після смерті Беббіджа один із принципів, що лежить в основі ідеї аналітичної машини, – використання перфокарт – знайшов втілення в діючому пристрої. Це – *статистичний табулятор*, побудований американцем *Германом Холлерітом* з метою прискорити оброблення результатів перепису населення США в 1890 р. Після успішного використання табулятора Холлеріт організував фірму з виробництва табуляторних машин (*Tabulating Machine Company*). З роками підприємство зазнало чимало змін – злиттів і перейменувань. Остання така зміна відбулася в 1924 р., за 5 років до смерті Холлеріта, коли він створив Міжнародну корпорацію машин для бізнесу (*IBM – International Business Machines Corporation*).

Ще одним чинником, що сприяв появі сучасного комп'ютера, стали праці з *двійкової системи числення*. Одним з перших, хто зацікавився двійковою системою, був німецький учений Готфрід-Вільгельм Лейбніц. У своїй праці «Мистецтво складання комбінацій» (1666 р.) він заклав основи формальної двійкової логіки. Але основну лепту у дослідження двійкової системи числення вніс англійський математик Джордж Буль. У праці «Дослідження законів мислення» (1854 р.) він винайшов своєрідну алгебру – систему позначень і правил, застосовну до різних об'єктів – від чисел і букв до речень (цю алгебру потім назвали на його честь *булевою алгеброю*). Користуючись цією системою, Буль міг закодувати висловлення, твердження, істинність чи хибність яких потрібно було довести за допомогою символів своєї мови, а потім маніпулювати цими символами як двійковими числами.

У 1936 р. випускник одного з американських університетів *Клод Шеннон* довів, що якщо побудувати електричні кола відповідно до принципів булевої алгебри, то вони могли б виражати логічні висловлення, визначати істинність тверджень, а також виконувати складні обчислення, і впритул наблизився до теоретичних основ побудови комп'ютера.

Ще троє дослідників – двоє в США (*Джон Атанасофф* і *Джордж Стібіц*) і один у Німеччині (*Конрад Цузе*) – розвивали одні й ті самі ідеї майже одночасно. Першу грубу модель обчислювальної машини на електричних схемах побудував Атанасофф (1939 р.). У 1937 р. Джордж Стібіц склав першу електромеханічну схему, що виконує операцію двійкового додавання (тепер двійковий суматор, як і раніше, залишається одним з основних компонентів будь-якого цифрового комп'ютера).

Не маючи ніякого уявлення про працю Беббіджа і Буля, Конрад Цузе в Берліні почав розробляти універсальну обчислювальну машину, подібну до аналітичної машини Беббіджа. Перший варіант машини, названої *Z1*, був створений у 1938 р. Дані в машину вводилися з клавіатури, а результат виводився на панелі з безліччю маленьких лампочок. У другому варіанті машини *Z2* дані вводилися за допомогою перфорованої фотоплівки.

На замовлення командування військово-морського флоту та за фінансової і технічної підтримки фірми *IBM* гарвардський математик *Говард*

Ейкен приступив до розроблення машини, в основу якої лягли ідеї Беббіджа і технологія XX ст. Як перемикальні пристрої в машині Ейкена застосовували прості електромеханічні реле (причому використовувалася десяткова система числення); інструкції оброблення даних були записані на перфострічці, а дані вводилися в машину у вигляді десяткових чисел, закодованих на перфокартах фірми *IBM*. Машину, названу *Марк-1*, уперше успішно випробувано на початку 1943 р. Машина *Марк-1*, завдовжки майже 17 м і заввишки понад 2,5 м містила близько 750 тис. деталей, з'єднаних проводами загальною довжиною близько 800 км. Машину стали використовувати для виконання складних балістичних розрахунків.

Однак машина *Марк-1* застаріла ще до того, як була побудована. У 1941 р., майже за два роки до того, як *Марк-1* почала працювати, Конрад Цузе побудував діючий комп'ютер – програмно-керований пристрій, оснований на двійковій системі числення. Машина *Z3* була значно меншою від машини Ейкена, а її виробництво набагато дешевшим. Як машина *Z3*, так і її «спадкоємець» *Z4* використовувалися для розрахунків, пов'язаних з конструюванням літаків і ракет.

Для пошуку способів розшифрування секретних німецьких кодів британська розвідка збрала групу вчених з представників різних спеціальностей – від інженерів до професорів літератури. Входив у цю групу і математик *Алан Тьюрінг*. Ще в 1936 р. у віці 24 років він у своїй праці описав абстрактний механічний пристрій – «універсальну машину», яка мала справлятися з будь-якою припустимою, теоретично розв'язною задачею – математичною чи логічною. Деякі ідеї Тьюрінга в кінцевому підсумку знайшли відображення в реальних машинах, побудованих групою вчених. Спочатку вдалося створити кілька дешифраторів на основі електромеханічних перемикачів. Однак наприкінці 1943 р. було створено набагато потужнішу машину, яка замість електромеханічних реле містила близько 2 000 електронних вакуумних ламп. Цю машину назвали *Колос*.

Водночас у США на потребу воєнного часу появився пристрій, що за принципами роботи і застосування був теоретично наближений до «універсальної машини» Тьюрінга. Машина *Еніак* (*ENIAC – Electronic Numerical Integrator and Computer* – електронний цифровий інтегратор і комп'ютер), як *Марк-1* Говарда Ейкена, також призначалася для вирішення завдань балістики. Головним консультантом проекту був *Джон У. Мочлі*, головним конструктором – *Дж. Преспер Екерт*. Передбачалося, що конструкція машини буде містити 17 468 ламп. Таку кількість ламп почасти зумовлено тим, що *Еніак* мала працювати з десятковими числами. Наприкінці 1945 р. *Еніак* було нарешті складено.

Однак не встигла *Еніак* поступити в експлуатацію, як Мочлі й Екерт уже працювали на замовлення військових над новим комп'ютером. Головний недолік комп'ютера *Еніак* – апаратна реалізація програм за допомогою

електронних схем. Наступна модель – машина *Едвак* – створена на початку 1951 р. (*EDVAC* від *Electronic Discrete Automatic Variable Computer* – електронний комп'ютер з автоматичними дискретними змінними), була більш гнучкою. Її внутрішня пам'ять містила не тільки дані, але і програму в спеціальних пристроях (заповнених ртуттю трубка), названих *лініями затримки*. Істотно й те, що *Едвак* кодувала дані вже в двійковій системі.

Серед слухачів курсу лекцій про електронні комп'ютери, які читали Мочлі й Екерт у процесі реалізації проекту *Едвак*, був англійський дослідник *Моріс Уїлкс*. Повернувшись у Кембріджський університет, він у 1949 р. завершив створення першого у світі комп'ютера з програмами, збережуваними в пам'яті. Комп'ютер одержав назву *Едсак* (*EDSAC* – від *Electronic Delay Storage Automatic Calculator* – електронний автоматичний калькулятор з пам'яттю на лініях затримки).

Епоха масового виробництва комп'ютерів почалася з випуску першого комерційного комп'ютера *LEO* (*Lyons' Electronic Office* – електронний офіс *Lyons*), що використовувався для розрахунку зарплати працівникам чайних магазинів, які належали фірмі *Lyons*.

Якісно новий етап у проектуванні комп'ютерів настав, коли фірма *IBM* запустила свою відому серію машин – *IBM/360*. Машини цієї серії мали різну продуктивність, різний набір пристроїв і призначалися для розв'язання різних задач, однак їх побудовано за єдиними принципами, що істотно полегшувало модернізацію комп'ютерів і обмін програмами між ними.

У колишньому СРСР розроблення комп'ютерів (їх було названо електронно-обчислювальними машинами (ЕОМ)) розпочали наприкінці 40-х років. У 1950 р. в Інституті електротехніки АН УРСР у Києві було випробувано першу вітчизняну ЕОМ на електронних лампах – малу електронну обчислювальну машину (МЕОМ), яку спроектувала група вчених та інженерів під керівництвом *академіка С.А. Лебедєва*. У 1952 р. під його керівництвом було створено велику електронну обчислювальну машину (ВЕОМ), що після модернізації в 1954 р. мала швидкодію 10 000 операцій/с.

Перші комп'ютери через надмірну вартість і технічну складність працювали лише в строго заданих умовах експлуатації і тому безпосередній доступ до них мав лише технічний обслуговуючий персонал та оператори. Користувачі віддавали операторам свої набиті на перфокартах програми і дані, а потім годинами, а то і цілодобово очікували результатів.

Поява *режиму поділу часу* процесора між декількома користувачами і мінікомп'ютерів – машин нового класу (в середині 60-х років) забезпечила одночасно декільком користувачам безпосередній доступ до ресурсів комп'ютера і спільне використання програм та даних.

Хоча ПК досить швидко завоював світ, його створення тривало довго.

Перший ПК *Альмайр-8800* фірми *MITS* (*Micro Instrumentation and Telemetry Systems* – мікроапаратура і системи телеметрії), побудований на

основі мікропроцесора *Intel-8008*, з'явився на початку 1975 р. Ємність пам'яті цього комп'ютера становила усього 256 байт, він не мав ні клавіатури, ні дисплея. Користувачі вводили програми і дані в двійковій формі, клацаючи набором маленьких ключів, що могли займати два положення – *вгору* і *вниз*; результати зчитувалися також у двійкових кодах – за світлими і темними лампочками (рис. 1.1).



Рис. 1.1. Перший ПК Altair-8800

Ці комп'ютери тривалий час не мали ніякого програмного забезпечення і їх купували здебільшого ентузіасти-програмісти. Одну з перших програм для ПК *Altair-8800*, що реалізувала популярну мову *Basic*, написали майбутні засновники фірми *Microsoft* Білл Гейтс і Поль Аллен.

Комерційний успіх ПК *Altair-8800* сприяв розвитку інших компаній, що посіли тепер чільне місце у виробництві ПК.

Однак найбільший успіх на першому етапі масового поширення ПК мала заснована Стівом Джобсом і Стефаном Возняком компанія *Apple*. Перший ПК Джобса і Возняка (*Apple*), зібраний у гаражі, не мав успіху, однак другий, названий *Apple-2*, з'явився в 1977 р. Цей комп'ютер, що мав систему кольорової графіки, забезпечив компанії в перший же рік рівень продажу 2,7 млн дол.

У 1981 р. фірма *IBM* вийшла на ринок ПК, випустивши свій перший ПК – *IBM PC (Personal Computer* – персональний комп'ютер) (рис. 1.2). Ця модель комп'ютера, а також моделі *IBM PC XT (eXtended Technology* – розширена технологія) і *IBM PC AT (Advanced Technology* – передова технологія) поклали початок дійсно масовому використанню ПК.

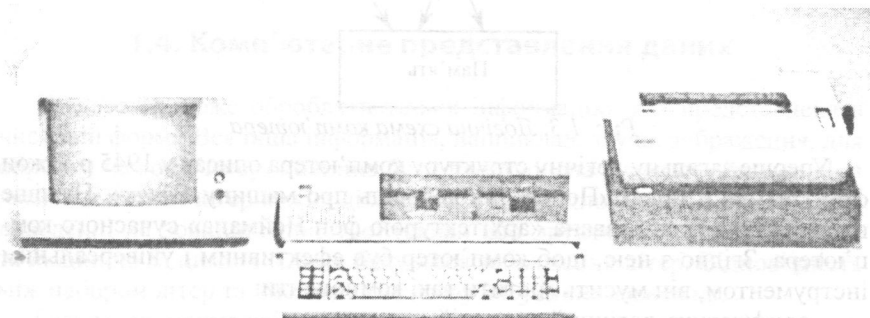


Рис. 1.2. Модель IBM PC

Усі попередні ПК задовольняли здебільшого інтереси вузького кола аматорів і професіоналів. Персональний комп'ютер *IBM* виявився саме таким, тобто з гнучкою системою, яка легко адаптується і здатна використовувати різноманітні апаратні й програмні засоби інших виробників.

Для створення *IBM PC* був використаний модульний принцип побудови ПК із можливістю заміни окремих компонентів комп'ютера у разі їх несправності чи модернізації. Крім того, у процесі розвитку програмних і апаратних засобів був запроваджений *принцип сумісності «зверху-вниз»*: усі нові пристрої і програми мають бути сумісними, тобто забезпечувати здатність наступних версій обслуговувати раніше створені програми.

Фірма *IBM* незабаром стала витіснятися іншими виробниками ПК. Ці комп'ютери, названі *клонами* (близнюками) комп'ютерів *IBM PC* (*IBM-сумісними ПК*), тепер складають вагомую частку ринку ПК у світі.

1.3. Логічна структура комп'ютера

Логічну схему комп'ютера зображено на рис. 1.3 (суцільні лінії – керувальні зв'язки, переривчасті – інформаційні).

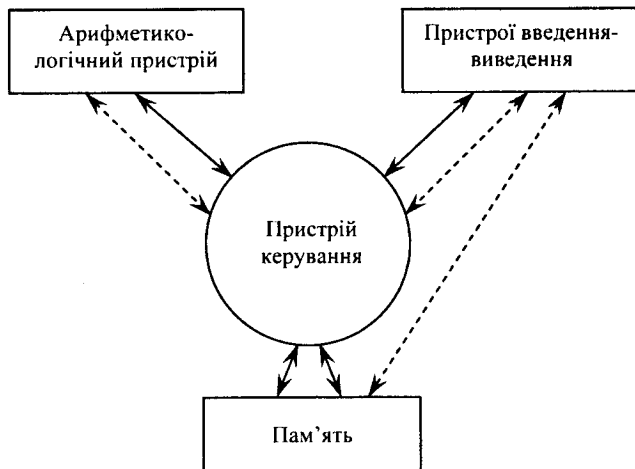


Рис. 1.3. Логічна схема комп'ютера

Уперше загальну логічну структуру комп'ютера описав у 1945 р. Джон фон Нейман у праці «Попередня доповідь про машину *Едвак*». Пізніше ця структура була названа «архітектурою фон Неймана» сучасного комп'ютера. Згідно з нею, щоб комп'ютер був ефективним і універсальним інструментом, він мусить містити такі компоненти:

- арифметико-логічний пристрій;
- пристрій керування;

- запам'ятовувальний пристрій;
- пристрої введення-виведення інформації.

Арифметико-логічний пристрій виконує арифметичні та логічні перетворення даних, що надходять до нього.

Пристрій керування керує процесом оброблення інформації, посылаючи всім іншим пристроям сигнали про виконання тих чи інших дій.

Сукупність арифметико-логічного пристрою та пристрою керування називають *процесором*.

Пам'ять зберігає інформацію, передану з інших пристроїв (зокрема, пристроїв уведення), і видає інформацію іншим пристроям комп'ютера, включаючи пристрої виведення.

Пам'ять комп'ютера складається з великої кількості пронумерованих комірок, в кожній з яких можуть знаходитись або дані, що обробляються, або інструкції програм. Спочатку за допомогою якогось зовнішнього пристрою в пам'ять комп'ютера вводиться програма. Пристрій управління зчитує вміст комірки пам'яті, в якій знаходиться перша інструкція (команда) програми, і організує її виконання. Як правило, після виконання команди пристрій управління починає виконувати команду з комірки пам'яті, що знаходиться безпосередньо за коміркою, в якій міститься виконана команда. Але цей порядок можна змінити за допомогою команд передачі управління (переходу), що дозволяє створювати складні програми.

Пристрої введення і виведення служать для введення даних у машину, виведення результатів і, в разі потреби, для керування процесом оброблення інформації.

Більшість сучасних комп'ютерів в основних рисах відповідають принципам, викладеним фон Нейманом, хоча їх схеми трохи відрізняються від наведеної вище. Зокрема, арифметико-логічний пристрій і пристрій керування, як правило, об'єднуються в один пристрій – центральний процесор. Окрім того, багато швидкодіючих комп'ютерів здійснюють паралельну обробку даних на декількох процесорах.

1.4. Комп'ютерне представлення даних

Комп'ютер може обробляти тільки інформацію, яка представлена в числовій формі. Вся інша інформація, наприклад, звуки, зображення, для обробки на комп'ютері повинна бути перетворена в числову форму. При введенні в комп'ютер кожна літера кодується відповідним числом, а при виведенні на зовнішні пристрої (екран або друкування) для сприйняття людиною за цими числами будується зображення літер. Відповідність між набором літер та числами має назву *кодування символів*.

Окрім елементи двійкового коду, що набувають значення 0 чи 1, називають розрядами чи *бітами*.

У старих комп'ютерах, призначених для обчислювальних задач, мінімальною одиницею інформації, доступною для оброблення, була *комірка*. Кількість розрядів у комірці орієнтовано на подання чисел і вона різна у різних комп'ютерах (24 біт, 48 біт і т. д.). Однак такий великий розмір комірки був незручний для подання символів, оскільки для подання символів інформації достатньо 5...8 біт. Це дає можливість подати від 32 до 256 символів. Тому мінімальною одиницею інформації, що обробляється в сучасному комп'ютері, є *байт*, який складається з восьми двійкових розрядів (бітів). Кожен байт, розміщений у пам'яті комп'ютера, має свою адресу, що визначає його місцезнаходження і задається відповідним кодом. Адреси пам'яті починаються з нуля для першого байта і послідовно збільшуються на одиницю для кожного наступного біта.

Похідні одиниці від байта – кілобайт (2^{10} байт) – кбайт; мегабайт (2^{20} байт) – Мбайт; гігабайт (2^{30} байт) – Гбайт; терабайт (2^{40} байт) – Тбайт і петабайт (2^{50} байт) – Пбайт.

Для подання чисел використовують один чи декілька послідовно розміщених байтів. Групи байтів утворюють двійкові слова, що, у свою чергу, можуть бути як фіксованої, так і змінної довжини.

Формати даних *фіксованої довжини* (*пісслово, слово і подвійне слово*) складаються відповідно з одного, двох і чотирьох послідовно розміщених байтів. Звернення до цих даних виконується за адресою крайнього лівого байта числа, що для слова має бути кратним числу 2, а для подвійного слова – числу 4.

Формат даних *змінної довжини* складається з групи послідовно розміщених байтів від 1 до 256. Адресація таких даних виконується, як і у форматах фіксованої довжини, за адресою найлівішого байта.

Залежно від характеру інформації використовують формати даних як фіксованої, так і змінної довжини. Так, у форматах даних фіксованої довжини зазвичай подають двійкові числа, команди і деякі логічні дані, а у форматах змінної довжини – десяткові числа і алфавітно-цифрова інформація..

Запитання до самоперевірки

1. Що вивчає інформатика? Зв'язок інформатики з іншими науками.
2. Дайте визначення інформації та інформаційного суспільства.
3. Ким і коли створено перший механічний обчислювальний пристрій?
4. Які компоненти мала аналітична машина Беббіджа і які технічні засоби використовувалися для її реалізації?
5. Чому праці з двійкової системи числення Лейбніца і Буля стали теоретичною основою створення комп'ютера?
6. Які обчислювальні пристрої можна назвати першими комп'ютерами?
7. Які основні принципи закладено в архітектурі фон Неймана?
8. Чому в комп'ютерах використовують двійкову систему числення?

2. ОСНОВИ КОМП'ЮТЕРНОЇ ТЕХНІКИ

📖 План викладу матеріалу

1. Принципи компонування персональних комп'ютерів.
2. Материнська плата.
3. Центральний процесор.
4. Внутрішня пам'ять.
5. Зовнішня пам'ять.
6. Монітор та відеокарта.
7. Аудіосистема комп'ютера.
8. Пристрої введення.
9. Пристрої виведення.
10. Допоміжні пристрої комп'ютера.

↔ Ключові терміни розділу

✓ Материнська плата	✓ Клавіатура
✓ Шина	✓ Маніпулятор "мишка"
✓ Центральний процесор	✓ Сканер
✓ Пам'ять	✓ Цифрова фотокамера
✓ Оперативна пам'ять	✓ Відеокамера
✓ Гнучкий магнітний диск	✓ Принтер
✓ Вінчестер	✓ Плотер
✓ Компакт-диск	✓ Проектор
✓ Flash-пам'ять	✓ Корпус системного блоку
✓ Відеосистема комп'ютера	✓ Блок живлення
✓ Монітор	✓ Система охолодження
✓ Відеокарта	✓ Мережевий фільтр
✓ Аудіосистема комп'ютера	✓ UPS

2.1. Принципи компонування персональних комп'ютерів

Незважаючи на те, що існують комп'ютери різного класу і різного призначення, їхні основні компоненти і принципи компонування несуттєво відрізняються між собою. Тому компоненти комп'ютера надалі будемо розглядати стосовно ІВМ-сумісних ПК. У них закладена можливість вдосконалення його окремих частин та використання нових пристроїв. Фактично ІВМ РС став стандартом персонального комп'ютера.

На основній електронній платі комп'ютера IBM PC (системній, або материнській, платі) розміщені блоки, які здійснюють обробку інформації (обчислення). Схеми, що керують рештою пристроїв комп'ютера – монітором, дисками, принтерами тощо, реалізовані на окремих платах, які встановлюються в стандартні роз'язтя на системній платі – слоти. До цих електронних схем підводиться електроживлення з єдиного блока живлення, а для зручності та надійності все це розміщується в системному блоці.

Завдяки відкритій архітектурі IBM PC користувачі можуть самостійно розширювати можливості своїх комп'ютерів, під'єднуючи відповідні пристрої у вільні роз'язтя на системній платі.

Функціонування комп'ютера потребує такого мінімального набору пристроїв:

- материнська плата (motherboard);
- центральний процесор (processor);
- пам'ять (storage);
- стандартний пристрій виведення – дисплей (display) чи монітор (monitor);
- пристрій, що з'єднує дисплей з комп'ютером – відеокарта (video card) або відеоконтролер (video controller);
- стандартний пристрій введення – клавіатура (keyboard);
- блок електроживлення (power supply).

Оскільки в сучасних операційних системах із графічним інтерфейсом багато операцій виконуються за допомогою «мишки» чи аналогічних їй пристроїв, то один з цих пристроїв також входить у мінімальний склад комп'ютера.

Материнська плата, яку називають також системною платою (system board) чи головною платою (main board), є пристроєм, що керує внутрішніми зв'язками і взаємодіє із зовнішніми пристроями. На материнській платі розміщуються всі основні елементи комп'ютера, лінії з'єднання і роз'язтя для підключення інших пристроїв.

Центральний процесор безпосередньо опрацьовує інформацію і виконує функції керування роботою всього комп'ютера відповідно до заданої програми (у деяких пристроях сучасного комп'ютера можуть використовуватися свої процесори).

Пам'ять сучасного комп'ютера має ієрархічну структуру. Можна виділити три рівні пам'яті комп'ютера: реєстрову, оперативну і зовнішню.

Реєстрова пам'ять – найбільш швидкодіюча, але найменша за ємністю пам'ять (зазвичай декілька сот байтів). Реєстрова пам'ять містить деякі системні дані (наприклад, адресу наступної команди, яку буде виконувати процесор), вихідні дані і результати виконання деяких команд процесора, а також часто використовувані в програмі дані.

Оперативна пам'ять має значно більшу ємність (від декількох десятків мегабайтів до декількох гігабайтів) і використовується для тимчасового зберігання програм та даних під час їх оброблення. Так, для запуску програми вона попередньо завантажується з пристрою зовнішньої пам'яті і тільки потім починає виконуватися.

Пристрої реєстрової і оперативної пам'яті є *енергозалежними*, тобто у разі вимикання комп'ютера дані, що знаходяться на цих пристроях, губляться. Пристрої *зовнішньої пам'яті* є *енергонезалежними* і призначені для тривалого зберігання даних. Цей рівень пам'яті найменш швидкодіючий і об'єднує велику групу пристроїв, що істотно відрізняються як за обсягом, так і за швидкістю доступу до даних. Пристрої зовнішньої пам'яті можуть бути *зі знімними носіями даних* (наприклад, дисководи гнучких дисків і дисководи *CD-ROM*). Носії даних інших пристроїв жорстко зафіксовані, наприклад у дисководах жорстких дисків (хоча існують знімні дисководи жорстких дисків). Ємність і швидкодія пристроїв зі знімними носіями зазвичай нижчі ємності і швидкодії пристроїв з фіксованими носіями.

Для збільшення швидкості доступу до даних у комп'ютері використовується буферна і кеш-пам'ять.

Буферну пам'ять використовують для збільшення швидкодії комп'ютера у разі обміну із зовнішніми пристроями. Оскільки швидкість доступу до зовнішнього пристрою значно нижча від швидкості оброблення даних у процесорі, процесор простоює, очікуючи закінчення введення-виведення. Щоб цього не відбувалося, фрагмент даних для читання попередньо зчитується з пристрою в буферну пам'ять. Для записування даних фрагмент даних спочатку передається в буферну пам'ять, а потім з буферної пам'яті на пристрій. Збільшення швидкодії відбувається за рахунок того, що обмін між буферною пам'яттю і зовнішнім пристроєм виконується без участі процесора (він у цей час може виконувати інші процеси). Буферна пам'ять реалізується або апаратно в електронній схемі пристрою введення-виведення, або програмно в оперативній пам'яті комп'ютера.

Кеш-пам'ять (від *cache* – притулок, склад), крім функції буферизації, зберігає найчастіше використовувані дані або ті дані, до яких недавно відбувалося звернення. Ці дані поміщають у більш швидкодіючу пам'ять. Так, доцільно таблиці розміщення файлів на дисках помістити в оперативну пам'ять, оскільки під час оброблення даних звернення до цієї таблиці відбувається досить часто.

Дисплей і відеоконтролер, утворюючи разом *відеосистему комп'ютера*, забезпечують виведення даних для користувача в текстовому і графічному вигляді.

Клавіатура призначена для введення текстової інформації та для переміщення курсора по екрану в повноекранних і графічних прикладних задачах.

Блок живлення перетворює напруги електромережі в напруги живлення, використовувани в блоках і пристроях комп'ютера. Він містить проводи з роз'язтями для підключення різних пристроїв.

Поряд з основним набором пристроїв комп'ютер, залежно від складу розв'язуваних на ньому задач, містить також додаткові пристрої: принтери, сканери, звукові карти, акустичні системи, мікрофони, модеми і т. ін.

Усі моделі комп'ютера (від найстарішої до найсучаснішої) мають однаковий *блоковий* принцип компонування.

Основний блок комп'ютера – *системний блок*. Він містить блок живлення, кріпильні елементи для материнської (системної) плати, електронних плат і дисководів.

Пристрої комп'ютера, розміщені в системному блоці, називають *внутрішніми пристроями*. *Зовнішні пристрої* підключаються до комп'ютера двома способами:

- безпосередньо через роз'язтя материнської плати;
- через роз'язтя плат (карт), усталених у материнську плату.

Деякі пристрої (наприклад, модеми чи жорсткі диски) мають як зовнішні, так і внутрішні варіанти виконання. Крім того, різні класи ПК мають також різне конструктивне виконання. Так, у настільному комп'ютері (*desktop computer*) дисплей, клавіатура і «мишка» – зовнішні пристрої, а портативні комп'ютери (*notebooks*) являють собою один системний блок, у який вбудовано і дисплей, і клавіатуру, і аналог «мишки».

Нижче розглядаються основні пристрої комп'ютера та найбільш поширені додаткові пристрої, використовувани в комп'ютері для виконання проектних і науково-інженерних робіт, а також пристрої оброблення і введення-виведення звукових та відеоданих.

2.2. Материнська плата

Материнська плата – це основний пристрій комп'ютера, що передусім визначає його продуктивність.

Материнська плата містить такі основні компоненти:

- шини;
- базову систему введення-виведення *BIOS (Basic Input/Output System)*;
- кеш-пам'ять;
- системні ресурси.

Шини (bus) – це сукупності ліній (провідників на материнській платі), по яких паралельно й одночасно обмінюються даними компоненти і пристрої комп'ютера. Шину призначено для обміну даними між двома і більше пристроями. Шину, що сполучає тільки два пристрої, називають *портом*.

Шина має власну архітектуру, що містить такі компоненти:

- контролер шини;

- лінії для обміну даними (шини даних);
- лінії для адресації даних (шини адреси);
- лінії для керування даними (шини керування).

Контролер шини керує процесом обміну даними і службовими сигналами.

По *шині даних* відбувається обмін даними між процесором, зовнішніми пристроями й оперативною пам'яттю.

Процес обміну даними можливий лише в тому разі, коли відомі відправник і одержувач цих даних. Кожний компонент комп'ютера, кожний регістр введення-виведення та комірка оперативної пам'яті мають свою адресу і входять у загальний адресний простір комп'ютера. Для адресації до якого-небудь пристрою комп'ютера використовується *шина адреси*, по якій передається унікальний ідентифікаційний код (адреса).

Для того щоб дані були записані (прочитані) у підключені до шини регістри пристроїв, адреси яких зазначені на шині адреси, потрібні службові сигнали: записування-зчитування, готовності до приймання-передавання даних, підтвердження приймання даних і т. ін. Усі ці сигнали передаються по *шині керування*.

На материнській платі вбудовано такі шини:

- системна шина, чи шина процесора – для пересилання даних процесору й одержання даних від процесора;
- шина кеш-пам'яті – для обміну інформацією між процесором і кеш-пам'яттю;
- шина пам'яті – для обміну інформацією між оперативною пам'яттю і процесором;
- шини введення-виведення (локальні та стандартні шини).

Локальна шина введення-виведення – це швидкісна шина, призначена для обміну інформацією між периферійними швидкодіючими пристроями (відеокартами, мережевими картами й ін.) і системною шиною.

Стандартна шина введення-виведення використовується для підключення до шин повільніших пристроїв («мишки», клавіатури, модемів та ін.).

Шини утворюють ієрархію, верхнім рівнем якої є системна шина.

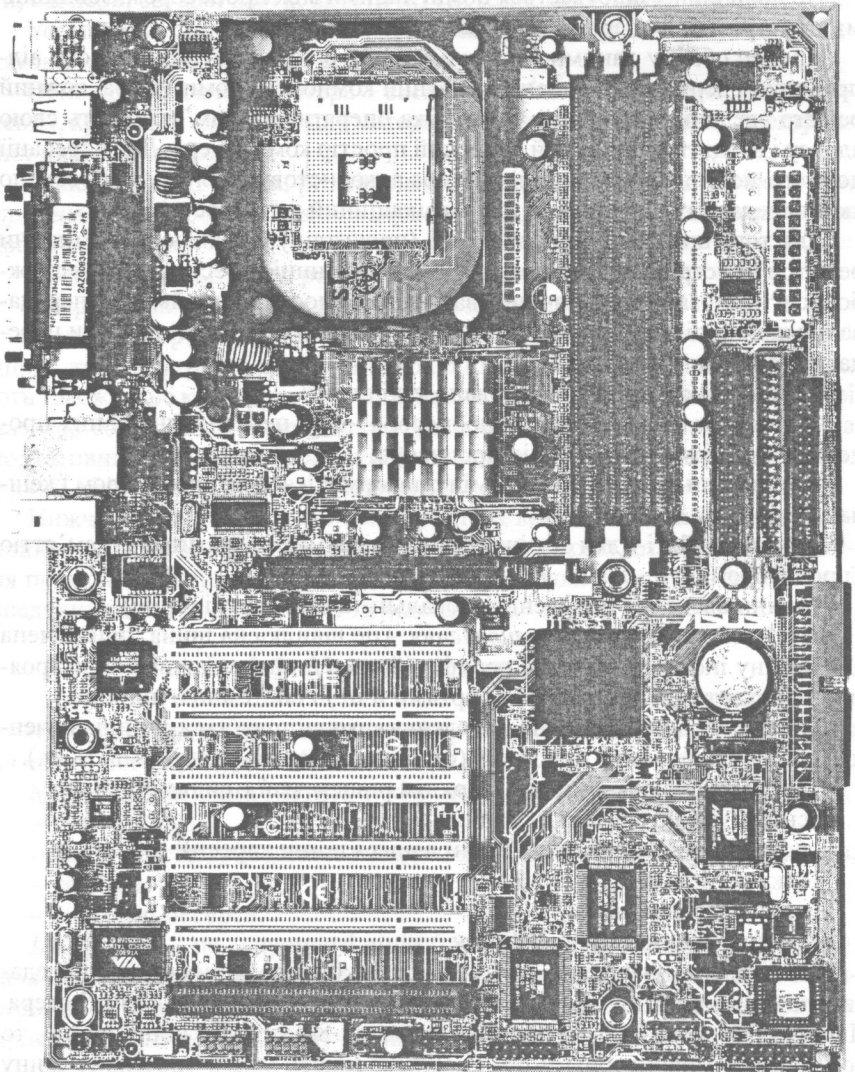
Основні характеристики шини:

- розрядність (окремо для шини даних і шини адреси);
- тактова частота;
- пропускна здатність;
- стандарт шинного інтерфейсу.

Чим вища *розрядність шини даних*, тим більше даних можна передавати за визначений проміжок часу і тим вища продуктивність комп'ютера. Якщо перші IBM-сумісні комп'ютери мали 8-розрядну шину даних, то комп'ютери з процесорами сім'ї *Pentium* мають уже 64-розрядну шину даних.

Тактова частота шини і кількість переданих за один такт бітів визначають пропускну здатність, що дорівнює кількості бітів інформації, переданих по шині за секунду.

Загальний вигляд та основні компоненти материнської плати показано на рис. 2.1, а, б.



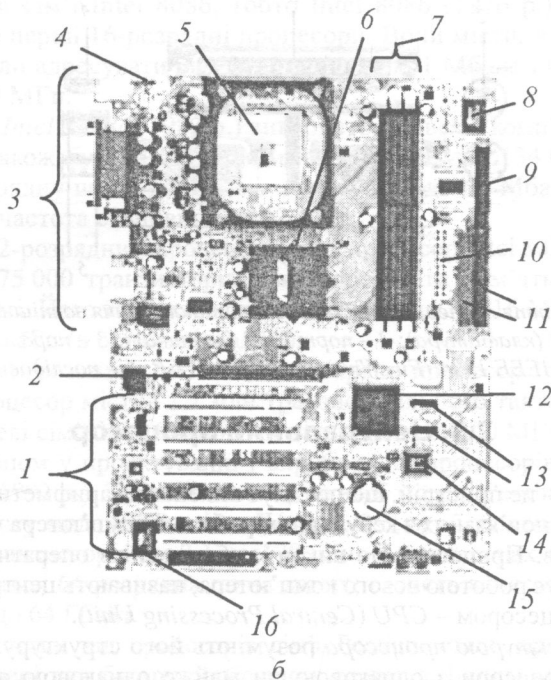


Рис. 2.1, а) загальний вигляд материнської плати;

б) основні компоненти материнської плати: 1 – розняття PCI;

2 – розняття AGP; 3 – розняття зовнішніх пристроїв; 4 – двопозиційний перемикач; 5 – гніздо центрального процесора; 6 – головний (північний) міст;

7 – розняття оперативної пам'яті; 8 – контролер Super I/O;

9 – розняття Floppy Disk; 10 – розняття електроживлення; 11 – розняття

ATA; 12 – міст ATA/PCI/USB (південний міст); 13 – розняття Serial ATA;

14 – BIOS; 15 – батареїка; 16 – контролер IEEE 1394 (FireWire)

На материнську плату зазвичай виводять такі розняття:

- гніздо для підключення процесора;
- розняття (слоти) для установлення модулів оперативної пам'яті;
- розняття для підключення нагромаджувачів жорстких дисків, дисководів CD і DVD по інтерфейсах ATA і Serial ATA;
- розняття для підключення дисководу гнучких дисків;
- розняття (слот) для підключення відеокарти по інтерфейсу AGP;
- розняття (слоти) для установлення карт розширення по інтерфейсу PCI та ISA (у старих платах);
- розняття для підключення зовнішніх пристроїв (рис. 2.2);
- розняття для підключення електроживлення.

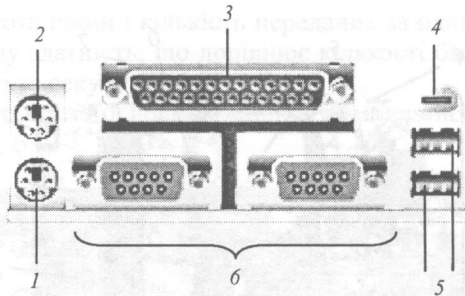


Рис. 2.2. Розняття материнської плати для підключення зовнішніх пристроїв: 1 – порт PS/2 (клавіатура); 2 – порт PS/2 («мишка»); 3 – паралельний порт; 4 – порт IEEE 1394 (FireWire); 5 – порти USB; 6 – послідовні порти

2.3. Центральний процесор

Процесор – це пристрій, що поєднує виконання арифметичних і логічних операцій, пов'язаних з керуванням роботою комп'ютера чи його окремих пристроїв. Процесор, що виконує збережені в оперативній пам'яті команди і керує роботою всього комп'ютера, називають центральним, або головним процесором – CPU (*Central Processing Unit*).

Під архітектурою процесора розуміють його структуру і склад компонентів. Процесори з однаковою чи майже однаковою архітектурою утворюють сім'ї процесорів.

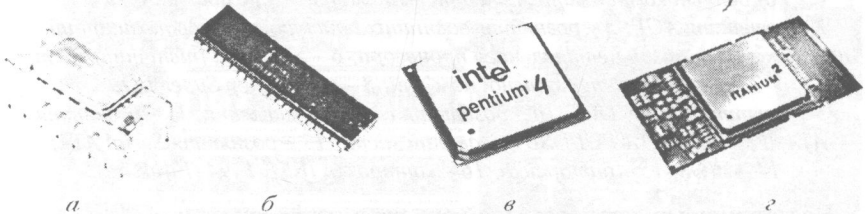


Рис. 2.3. Процесори фірми Intel: а – Intel 4004; б – Intel 8080; в – Intel Pentium 4; г – Intel Itanium 2

Перший процесор Intel – Intel 4004 (1971 р.) був чотирирозрядним процесором з тактовою частотою 400...800 кГц (рис. 2.3, а). Він містив 2300 транзисторів і адресував пам'ять ємністю до 640 байт. Цей процесор використовувався в калькуляторах.

Процесор Intel 8080 (1974 р.) був уже 8-розрядним з тактовою частотою 2 МГц і містив 6 000 транзисторів (рис. 2.3, б). Він міг адресувати пам'ять ємністю до 64 кбайт, мав 8-розрядну шину даних і використовувався в першому ПК Altair-8800.

Процесори сім'ї Intel 8086, тобто Intel 8086 (1976 р.) і Intel 8088 (1979 р.) – це перші 16-розрядні процесори. Вони містили 29 000 транзисторів, могли адресувати пам'ять ємністю до 1 Мбайт і мали тактову частоту 5...10 МГц.

Процесор *Intel 80286* (1982 р.) використовувався в комп'ютерах IBM PC AT. Він також був 16-розрядним, але містив уже 134 000 транзисторів, 16-розрядну шину даних, міг адресувати до 64 Мбайт пам'яті, а його тактова частота становила до 12 МГц.

Першим 32-розрядним процесором був процесор Intel 80386 (1985 р.). Він містив 275 000 транзисторів і міг адресувати пам'ять ємністю до 4 Гбайт пам'яті. Його тактова частота змінювалася від 16 МГц (у першій моделі) до 33 МГц (в останній моделі).

У результаті розвитку Intel 80386 створено процесор Intel 80486DX (1989 р.). Процесор містив 1,2 млн транзисторів; його тактову частоту в останній моделі сім'ї Intel 80486DX4 підвищено до 100 МГц.

Новим етапом у проектуванні і виробництві процесорів став процесор Pentium (1993 р.). Тактова частота першої моделі Pentium становила 66 МГц; частоту моделей Pentium IV (рис. 2.3, в) збільшено до 3,4 ГГц, а кількість транзисторів – з 3,1 до 178 млн. У процесорах сім'ї Pentium використовується 64-розрядна шина даних; ємність адресованої пам'яті збільшилася до 64 Гбайт.

Відгалуженнями від процесорів сім'ї Pentium стали процесори сім'ї Xeon, призначені для багатопроцесорних серверів. Іншим відгалуженням процесорів сім'ї Pentium стали процесори сім'ї Celeron, що є спрощеним варіантом Pentium. Крім цього, в 2003 р. фірма *Intel* розробила для портативних комп'ютерів процесор Pentium M зі зниженим споживанням енергії.

Перший 64-розрядний процесор фірми *Intel* – Itanium з'явився в 2001 р. Він містив 25 млн транзисторів і працював з тактовою частотою 733 МГц. Адресна шина цього процесора також стала 64-розрядною, що дозволило адресувати пам'ять ємністю до 1 024 Тбайт. Розрядність шини даних у цьому процесорі збільшилася до 128 байт. Модель Itanium 2 (рис. 2.3, з) містить 410 млн транзисторів і працює з тактовою частотою 1,5 ГГц.

Якщо перший процесор Intel 4004 виконано з використанням 10-мікронної технології, то останні моделі процесорів Intel уже виготовляли за 0,09-мікронною технологією.

Процесори AMD – це простіші і дешевші аналоги (клони) процесорів фірми *Intel*. Так, 32-розрядні процесори AMD Athlon XP і AMD Athlon MP є аналогами відповідно Pentium 4 і Xeon, а 64-розрядні процесори AMD Athlon 64 FX (рис. 2.4, а) і AMD Opteron – аналогами Itanium 2.

32-розрядні процесори, які виготовляє корпорація *VIA Technologies* за 0,13-мікронною технологією, відрізняються малими розмірами і низьким енергоспоживанням.



Рис. 2.4. Процесори – клони Intel: а – AMD Athlon 64 FX; б – VIA C3; в – VIA Eden-N (порівняно з Pentium M і одноцентовою монетою); г – процесор UltraSPARC IV; д – процесор PowerPC

Процесор VIA C3 (спадкоємець процесорів фірми *Cyrix* – колишнього конкурента фірми *Intel*), тактова частота якого до 1,4 ГГц (рис. 2.4, б), процесор VIA Antaur, тактова частота якого близько 1,2 ГГц, призначені для використання в невеликих настільних і портативних комп'ютерах з підвищеними вимогами до захисту оброблення даних.

Процесор VIA Eden-N – найменший за розмірами (15×15 мм) клон процесора Intel (рис. 2.4, в). Він має тактову частоту 1 ГГц і призначений для материнської плати VIA Nano-iTX з форм-фактором 12×12 см, що передбачається використовувати в портативних комп'ютерах.

Сім'ю процесорів UltraSPARC (що містять процесори серії UltraSPARC IV, UltraSPARC III і UltraSPARC II) застосовують для комп'ютерів, виготовлених фірмою *Sun* (рис. 2.4, г), а сім'ю процесорів PowerPC – для комп'ютерів серії Macintosh фірми *Apple* та комп'ютерів фірми *IBM* (рис. 2.4, д).

Останнім часом набули поширення *багатопроцесорні комп'ютери*, тобто комп'ютери, які містять кілька процесорів. Функціонування багатопроцесорної системи потребує виконання таких умов:

- материнська плата має підтримувати кілька процесорів, тобто мати додаткові роз'єкти для установа процесорів і відповідний набір мікросхем;
- процесор має працювати в багатопроцесорній системі;
- операційна система має підтримувати декілька процесорів (такими операційними системами є Windows NT/XPi і Unix).

Багатопроцесорна система найбільш ефективна у випадках, коли вона використовується багатозадачними операційними системами і прикладними програмами, створеними за допомогою спеціальних засобів, що дозволяють виконувати паралельне оброблення даних.

У процесі одночасної роботи декількох процесорів операційна система розподіляє різні задачі між процесорами. Існують два режими роботи багатопроцесорних систем – асиметричний і симетричний.

У режимі асиметричного оброблення один процесор виконує тільки задачі операційної системи, а решта процесорів реалізують прикладні програми.

У режимі симетричного мультиоброблення – SMP (*Symmetric Multi-Processing*) задачі операційної системи і користувацькі прикладні задачі може виконувати будь-який процесор залежно від його завантаження. Цей режим більш продуктивний і тому його використовують для більшості багатопроцесорних систем.

Можливості процесора визначають за такими основними характеристиками:

- ступенем інтеграції чи безпосередньо зв'язаною з нею технологією виробництва;
- внутрішньою і зовнішньою розрядністю оброблених даних;
- частотою тактового генератора для процесора;
- продуктивністю процесора;
- ємністю пам'яті, до якої може адресуватися процесор;
- смністю кеш-пам'яті;
- підтримуваною частотою системної шини;
- набором команд;
- напругою електроживлення і споживаною потужністю;
- розняттями підключення.

Ступінь інтеграції будь-якої мікросхеми, зокрема процесора, показує, скільки транзисторів уміщується в ній. Ця характеристика для процесорів Intel змінилася від десятків тисяч до сотень мільйонів транзисторів.

Одна з основних характеристик процесора – кількість бітів, яку він може обробляти одночасно – *внутрішня розрядність*. Внутрішня розрядність процесорів Pentium становить 32, а процесорів Itanium і процесорів Sun UltraSparc та IBM Power 4 – 64. Зовнішню розрядність процесора визначають за розрядністю системної шини. Зовнішня розрядність процесорів сім'ї Pentium становить 64, а процесорів Itanium – 128.

Найважливіші характеристики процесора – *частота тактового генератора* і залежний від неї такт роботи процесора – *машинний такт*. Чим коротший машинний такт, тим вища продуктивність процесора, що виражається кількістю виконаних команд (операцій) за одиницю часу.

Продуктивність процесорів сучасних комп'ютерів становить від декількох мільйонів до декількох мільярдів операцій за секунду, при цьому значення машинного такту становить 10...100 нс, а частота тактового генератора перевищила 3 ГГц. Продуктивність процесора визначається не тільки тактовою частотою, але й кількістю команд, виконуваних за один такт. Продуктивність процесора зазвичай визначають одиницею виміру *MIPS (Million Instructions Per Second* – мільйон операцій за секунду). Однак визначити значення цієї характеристики окремо для процесора дуже важко, тому вимірюють продуктивність усього комп'ютера.

Ємність пам'яті, до якої може адресуватися процесор, визначається шириною адресної шини. Ширина адресної шини процесорів перших IBM-сумісних комп'ютерів становила 20 біт, а максимальна ємність пам'яті – відповідно 1 Мбайт. Максимальна ємність пам'яті сучасних комп'ютерів дорівнює 64 Гбайт (за ширини адресної шини 36 біт). Ширина адресної шини процесорів Itanium становить 64, а максимальна ємність адресованої пам'яті – відповідно 2^{64} байт.

Істотний вплив на продуктивність процесора справляють *кількість рівнів ієрархії кеш-пам'яті* і *ємність кеш-пам'яті кожного рівня*.

Підтримувана частота системної шини (деякі процесори підтримують кілька частот) визначає швидкість обміну даними між процесором та іншими пристроями (насамперед оперативною пам'яттю). Натепер найбільш використовувані частоти 400, 533 і 800 МГц.

Важливою характеристикою процесора є *використовуваний набір команд*. Так, якщо процесор-клон AMD чи VIA не підтримує розширення мультимедійних команд SSE2 і SSE3, то програми, що використовують ці розширення, не будуть працювати з таким процесором.

Напруга електроживлення і споживана потужність є істотними для моделей процесорів, призначених для портативних комп'ютерів, оскільки вони визначають час автономної роботи комп'ютера (на батарейках). Так, модель Itanium 2 з низькою напругою електроживлення (1,5 В) споживає 62 Вт (звичайний процесор Pentium II споживав 130 Вт), моделі Mobil Pentium 4 мають напруги живлення 1,2...1,55 В і споживану потужність 2...3 Вт, а моделі Pentium M – напруги живлення 1,18 та 1 В і відповідно споживану потужність – 1 і 0,5 Вт. Для зниження споживання електроенергії процесором під час його бездіяльності використовується «сплячий» режим зі зниженою напругою електроживлення.

Для процесорів, починаючи з Intel 80486, через збільшення споживаної потужності процесора виникла потреба в охолоджувальній системі у вигляді вентилятора для поліпшення відведення тепла від спеціальних ребристих пластин. Сучасні моделі процесорів мають систему стеження за температурою та її регулюванням.

Центральний процесор підключається до системної шини і схеми електроживлення материнської плати.

Існують два *способи підключення процесора до материнської плати*.

За першим способом процесор вставляють у гніздо материнської плати за допомогою багатоконтактних роз'єктів (див. рис. 2.3). Так підключають процесори Pentium IV, Itanium 2, AMD і VIA. Наприклад, щоб підключити Pentium IV, використовують 478-контактне гніздо (Socket 478) зі спеціальним притискним важелем, що забезпечує міцне кріплення процесора в гнізді, а для підключення Itanium 2 підходить 611-контактне гніздо.

Другим способом передбачено розміщення процесора (наприклад, Pentium II) з вентилятором на спеціальній карті, що вставляється в слот розширення материнської плати.

2.4. Внутрішня пам'ять

До внутрішньої пам'яті комп'ютера належать:

- оперативна, чи системна пам'ять (RAM);
- пам'ять BIOS (ROM BIOS);
- пам'ять пристроїв комп'ютера.

За способом зберігання даних пам'ять поділяють на тимчасову енергозалежну і постійну енергонезалежну пам'ять.

В *оперативну пам'ять* завантажуються як системні програми, зокрема модулі операційної системи, так і прикладні програми користувачів. Будь-яку програму можна виконувати тільки з оперативної пам'яті. Історично оперативну пам'ять називають *пам'яттю з довільним доступом RAM (Random Access Memory)* на відміну від зовнішньої пам'яті з послідовним доступом *SAM (Serial Access Memory)*, реалізованої на магнітній стрічці.

Оперативна пам'ять – це *енергозалежна пам'ять*, тобто після вимкнення електроживлення її вміст губиться. Оскільки в момент увімкнення комп'ютера оперативна пам'ять порожня, комп'ютер починає завантажуватися з програми, розміщеної в *енергонезалежній пам'яті BIOS*.

Елементами внутрішньої пам'яті комп'ютера є *комірки*. Ємність комірки пам'яті кратна одному байту. Кожній комірці пам'яті привласнюється своя адреса чи номер.

Повна адреса комірки даних містить два компоненти – адресу рядка (row address) – другу половину адреси й адресу стовпця (column address) – першу половину адреси. Внутрішня пам'ять реалізується в одній чи декількох мікросхемах (чипах), кожна з яких має матричну структуру.

Керування доступом до пам'яті виконує *контролер пам'яті*, реалізований або в окремій мікросхемі, або в одній з мікросхем набору, на якому контролер пам'яті реалізований у мікросхемі головного моста.

Існує два різновиди енергозалежної пам'яті: динамічна і статична.

Динамічну оперативну пам'ять DRAM (*Dynamic RAM*) побудовано на мікросхемах, що потребують періодичного відновлення (регенерації) інформації в них, щоб уникнути втрат.

Запам'ятовувальний елемент DRAM – це конденсатор, що може знаходитися в зарядженому чи розрядженому стані. Зарядженому стану конденсатора відповідає одиниця, розрядженому – нуль. Крім того, до складу запам'ятовувального елемента входить транзистор, що зчитує значення, які зберігаються в конденсаторі.

У реальному конденсаторі існує струм витоку, тому дані в оперативній пам'яті дуже швидко набувають значення 0, оскільки ємність конденсатора дуже мала і він дуже швидко розряджається.

Відновлення (регенерація) даних відбувається під час виконання операцій зчитування чи запитування даних. Але, оскільки частота звернення до різних ділянок пам'яті різна, не можна гарантувати, що до конкретних даних відбудеться звернення до того, як вони «зіпсуються». Тому в комп'ютері реалізовано спеціальну схему регенерації, яка через визначені проміжки часу (зазвичай декілька мілісекунд) зчитує увесь вміст динамічної пам'яті. Під час регенерації центральний процесор знаходиться в стані очікування.

Запам'ятовувальний елемент у *статичній пам'яті з довільним доступом SRAM (Static RAM)* виконано на транзисторах (від 4 до 6 транзисторів), що забезпечують майже необмежений час зберігання даних за дуже малих витрат електроенергії. Крім того, швидкість зчитування-записування даних у статичній пам'яті вища, ніж у динамічній.

Однак статична пам'ять має два істотні недоліки:

– за однакового обсягу даних статична пам'ять у чотири і більше разів перевищує динамічну пам'ять;

– статична пам'ять технологічно більш складна й обходиться значно дорожче, ніж динамічна пам'ять такої самої ємності.

Тому оперативну пам'ять комп'ютера реалізовано на основі динамічної пам'яті та схеми регенерації її вмісту. Статичну пам'ять використовують там, де ємність пам'яті невелика і потрібний швидкий доступ до даних, тобто в кеш-пам'яті. Пам'ять CMOS також реалізовано як статичну пам'ять. Крім того, статичну пам'ять використовують у мікроконтролерах різних електронних пристроїв.

Постійна пам'ять *NVRAM (Non Volatile RAM)* не має потреби в електроживленні для зберігання даних.

Існують такі типи постійної (енергонезалежної) пам'яті:

- пам'ять тільки для зчитування – *ROM (Read Only Memory)*;
- програмована ROM – *PROM (Programmable ROM)*;
- програмована ROM, що стирається, – *EPROM (Erasable Programmable ROM)*;

- PROM, що стирається електричним способом, – *EEPROM (Electrically Erasable PROM)*;
- Flash-пам'ять;
- фероелектрична RAM – *FRAM (Ferroelectric RAM)*;
- магнітна RAM – *MRAM (Magnetic RAM)*.

Так само, як в енергозалежній пам'яті, біти в постійній пам'яті зберігаються в комірках матриці, але замість транзисторів і конденсатора використовуються елементи, що можуть знаходитися в одному з двох станів: у стані «1», якщо пропускають струм; у стані «0», якщо не пропускають.

Основні характеристики елементів пам'яті такі:

- тип пам'яті;
- глибина адресного простору, розрядність і сміність;
- максимальна пропускна здатність.

Тип пам'яті (наприклад, DDR SDRAM, QDR-II SRAM чи Flash-пам'ять) однозначно визначає як вид пам'яті (динамічна, статична чи постійна), так і її основні можливості.

Залежно від форм-фактора розрізняють такі види модулів пам'яті:

- SIMM-модулі;
- DIMM-модулі;
- SODIMM-модулі;
- RIMM-модулі.

Перші модулі пам'яті назвали однорядними модулями пам'яті – *SIMM-модулями (Single In-line Memory Module)*, оскільки електричні контакти зі слотом з обох боків карти попарно з'єднувалися між собою, тобто фактично були однобічними. Перші модулі були 30-контактними, потім стали випускати 72-контактні SIMM (рис. 2.5, а).

На відміну від модулів SIMM, дворядні модулі пам'яті – модулі *DIMM (Dual In-line Memory Modules)* мають електрично незалежні контакти по обидва боки роз'язттів. Найпоширеніші 168-контактні 64-розрядні модулі DIMM (рис. 2.5, б) мають по 84 контакти з кожного боку і два ключі. Тепер більшість материнських плат обладнано слотами для модулів DIMM. Модуль DIMM DDR (рис. 2.5, в) має такий самий форм-фактор, але на відміну від модуля DIMM містить 184 контакти й один ключ.

Модуль DIMM малого розміру ($6,67 \times 2,54$ см) – модуль *SODIMM (Small Outline DIMM)* (рис. 2.5, з) призначено для портативних комп'ютерів; він містить 144 контакти. Його модифікація SODIMM DDR (рис. 2.5, д) має 200 контактів та інше розміщення ключа.

Модуль пам'яті фірми *Rambus* – модуль *RIMM (Rambus In-line Memory Module)* (рис. 2.5, е) має такі самі розміри, що й DIMM-модулі, але містить 232 контакти і має інше розміщення ключа. Оскільки зі збільшенням ступеня мініатюризації мікросхеми пам'яті стали сильно нагріватися, їх почали укладати в спеціальні тепловідвідні корпуси (рис. 2.5, е).

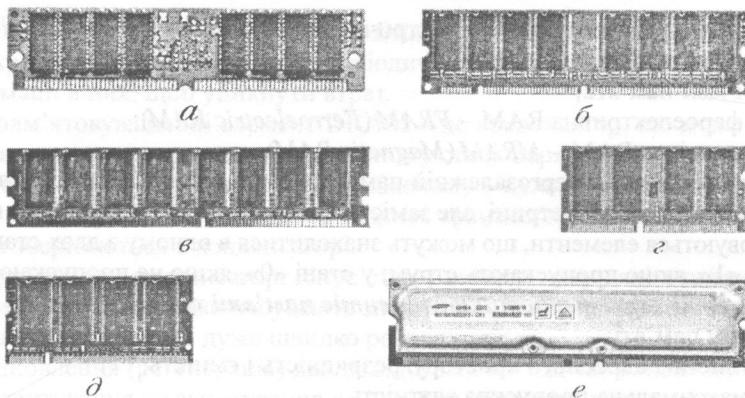


Рис. 2.5. Форм-фактори модулів пам'яті: а – модуль SIMM; б – модуль DIMM; в – модуль DIMM DDR; з – модуль SODIMM; д – модуль SODIMM DDR; е – модуль RIMM

Через високу сукупну електричну ємність сучасних модулів пам'яті великої власної ємності час їх зарядження стає неприпустимо великим, що призводить до втрати тактів. Щоб уникнути цього, деякі модулі забезпечуються спеціальною мікросхемою – буферною пам'яттю, яка зберігає дані, що надійшли відносно швидко. Такі модулі зазвичай містять у своїй назві слово *Registered*, оскільки буферна пам'ять реалізується як регістрова пам'ять. Модулі, що не мають буферної пам'яті, іноді позначають як *Unbuffered*. Слід зазначити, що Registered DIMM переважно несумісні зі звичайними DIMM, тому ці два типи DIMM мають різне положення ключів.

2.5. Зовнішня пам'ять

Зовнішня пам'ять чи зовнішні запам'ятовувальні пристрої використовуються для довгострокового зберігання інформації.

Зовнішні запам'ятовувальні пристрої можна класифікувати за такими ознаками:

- способом зберігання і доступу до даних (магнітоелектричні чи магнітні, магнітооптичні, оптичні, електричні);
- видом носія даних (дискета, жорсткий диск, магнітна стрічка, магнітооптичний диск, компакт-диск, DVD-диск і Flash-пам'ять);
- режимом доступу до пам'яті (пам'ять з довільним доступом RAM (дискета, усі види дисків і Flash-пам'ять, пам'ять з послідовним доступом – SAM (магнітна стрічка));
- типом носіїв даних – знімними носіями (дискета, компакт-диск, DVD-диск, знімні жорсткі диски, магнітна стрічка), незнімними носіями (жорсткий диск, магнітооптичний диск);

– можливістю перезаписування даних на перезаписувані носії (дискета, жорсткий і магнітооптичний диски, диски CD-RW, DVD-RW, DVD-RAM і DVD+RW, магнітна стрічка і Flash-пам'ять), носії з однократним записом («тільки для зчитування») (диски CD-ROM, CD-R, DVD-ROM, DVD-R і DVD+R);

– типом пристрою зберігання даних: внутрішні (вбудовуються в системний блок) і зовнішні (підключаються до системного блоку через один з портів чи за допомогою плати розширення).

Для зберігання даних на магнітних пристроях використовують *феромагнетиками* – речовини, що намагнічуються за допомогою постійного магніту чи електромагніту і зберігають стан намагніченості після видалення магніту. Шар магнітного матеріалу наносять на *немагнітну основу*, яку для магнітних стрічок і гнучких дисків виготовляють з різних пластмас, а для жорстких дисків використовують круглі алюмінієві пластини. Спочатку як магнітний матеріал використовували оксид заліза, а тепер – мікропорошкові магнітні матеріали, які наносять на основу тонкою плівкою (кілька мікрометрів) за допомогою гальванопластики або напилювання. Поверх магнітного матеріалу наносять тонкий *захисний шар та мастильний шар*.

Для зчитування даних використовують головки зчитування, а для записування – записувальні головки. Для зчитування і записування даних на дискети і магнітні стрічки (і на старі магнітні диски) використовують загальну *феритову головку зчитування-записування*, що являє собою залізне осердя з намотаним на нього дротом, тобто електромагніт.

Дисководи FDD (Floppy Disk Drives – дисководи гнучких магнітних дисків) використовують дискети розміром 3,5 дюйма (часто використовується запис 3,5").

Дисководи жорстких магнітних дисків. Сьогодні випускаються декілька десятків типів дисководів жорстких дисків – *HDD (Hard Disk Drives)*, які називають також *вінчестерними дисками* чи просто *вінчестерами*. Однак принципи функціонування більшості дисководів однакові.

Основні елементи конструкції типового дисководу жорстких дисків (рис. 2.6):

- магнітні диски;
- головки зчитування-записування;
- механізм приводу головок;
- двигун приводу дисків;
- гермоблок;
- друкована плата з електронною схемою керування;
- – роз'язтя.

У дисководах на жорстких дисках для кожної сторони диска передбачено свою головку зчитування-записування. Усі головки змонтовано на загальному рухомому каркасі і тому вони переміщуються одночасно.

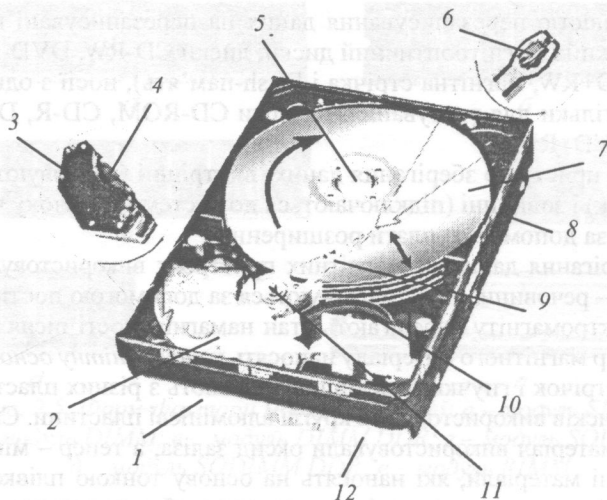


Рис. 2.6. Основні елементи дисководу жорстких дисків:

- 1 – роз'язтя інтерфейсу; 2 – кожух магніту; 3 – магніт; 4 – рухома катушка; 5 – двигун дисків; 6 – головки читування-записування;
 7 – диски; 8 – корпус; 9 – важіль головки; 10 – шина контролера;
 11 – вісь обертання; 12 – роз'язтя електроживлення

Коли дисковод вимкнено, головки торкаються дисків. Під час розкручування дисків зростає аеродинамічний тиск на головки, внаслідок чого вони відриваються від робочих поверхонь. За допомогою механізму приводу головки переміщуються від центра дисків до їх країв і встановлюються на заданий циліндр. Двигун приводу дисків, що обертає магнітні диски, з'єднаний з пакетом дисків безпосередньо (без передатних шестерень чи шківів). Швидкість обертання двигуна сучасних дисководів коливається від 3 600 до 7 200 об/хв, а в деяких моделях досягає 10 000 об/хв і більше.

Сучасні дисководи на жорстких дисках мають функцію автоматичного паркування магнітних головок, тобто, коли вимикають комп'ютер, головки встановлюються на визначений, найчастіше останній, циліндр.

У гермоблоці розміщено всі механічні частини дисководу. Насправді корпус дисководу не цілком герметичний, а має щілину, яку закрито мікрофільтром з метою захисту матеріалу дисків від пилу і частинок бруду. Через цю щілину вирівнюється також тиск повітря усередині дисководу і зовні. Ще один фільтр видаляє частинки, що утворюються в результаті роботи механічних частин диска.

На друкованій платі монтується електронні контролери двигуна і приводу головок, схема оброблення команд дисководу, схема для обміну

даними з інтерфейсом. Крім того, на друкованій платі сучасних дискондодів розміщено кеш-пам'ять дисководів, яка зберігає попереджувальні дані, що ймовірно знадобляться процесору.

На жорстких дисках більшості дискондодів є кілька роз'язттів для підключення до системи подачі електроживлення та інтерфейсного кабелю.

Перший жорсткий диск (1956 р.) мав ємність 5 Мбайт. Діапазон форматонаних ємностей сучасних жорстких дисків – до 3 Тбайт.

Пристрої зчитування-записування даних на магнітну стрічку (*tape recorder*), на відміну від дискондодів гнучких і жорстких дисків та CD і DVD, не набули поширення. Проте ці пристрої є обов'язковим компонентом потужних комп'ютерів, у яких зберігаються великі обсяги спільно використовуваних даних, що в разі їх псування чи руйнування потребують великих витрат на відновлення.

Магнітооптичні диски. На відміну від магнітних дисків і дискет дані на магнітооптичний диск записуються по спіралі, тобто диск має одну доріжку, розбиту на сектори однакового розміру.

Усі магнітооптичні диски знімні. Магнітооптичні диски бувають однобічними і двобічними (двобічні диски складаються з двох однобічних, склеєних між собою підкладками).

Записування даних у магнітооптичному блоці виконується не по бітах чи байтах, а по блоках (заввичай розмір блоку дорівнює 512 байт) протягом двох проходжень. Під час проходження лазерного променя розігрівається весь блок, а під час проходження магнітної головки дані перезаписуються у всьому блоці. Тому операція записування в магнітооптичному дисководі виконується в два рази довше, ніж операція зчитування.

Існує також різновид магнітооптичних дисків – *диски WORM* (від *Write Once, Read Many* – писати один раз, читати багато разів). Дані на ці диски можна записати тільки один раз (їх заввичай використовують для архівування даних, які не передбачається змінювати).

Оптичні диски. Для зберігання двійкових даних оптичні носії використовують зміну відображення світла від матеріалу носія. Так, повне чи майже повне відображення світла можна зіставити з одиницею, а відсутність відбитого від носія світла (чи його сильне ослаблення) – з нулем. Як оптичну зовнішню пам'ять комп'ютера використовують два види носіїв: компакт-диски і DVD-диски.

Компакт-диски, названі також *дисками CD* (*Compact Disk* – компакт-диск) використовують для записування аудіоданих у цифровому вигляді та програють на спеціальних пристроях – плеєрах CD.

Аналогічно *диски DVD* (*Digital Video Disk* – цифровий відеодиск) використовують для записування відеоданих (здебільшого фільмів), вони зчитуються на плеєрах DVD, підключених до телевізора.

Поступово спочатку диски CD, а потім і DVD-диски стали використовувати в комп'ютерах, причому не тільки для відтворення аудіо- і відеоданих, але і як зовнішню пам'ять для зберігання текстових та графічних даних, програм і т. ін. У зв'язку з цим змінилося і розшифрування скорочення DVD, що тепер трактується як *Digital Versatile Disk* – універсальний цифровий диск.

Систему для зчитування даних з оптичних дисків у спрощеному вигляді показано на рис. 2.7.

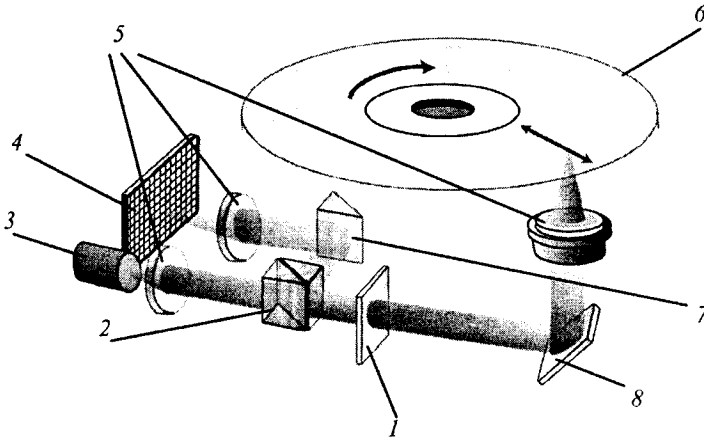


Рис. 2.7. Зчитування даних з оптичного диска:

- 1 – поляризаційний фільтр; 2 – поляризаційно-розділова призма;
 3 – лазер; 4 – фотодетектор; 5 – оптичні лінзи;
 6 – оптичний диск; 7 – відхилювальна призма; 8 – дзеркало

Світло від лазера проходить через поляризаційно-розділову призму, потрапляє на поляризаційний фільтр (при цьому світло поляризується у визначеній площині), а потім фокусується на поверхні оптичного диска. Якщо промінь лазера потрапляє на плоску поверхню (lands) диска CD-ROM чи DVD-ROM, світло відбивається майже цілком. Якщо ж світло потрапляє в поглиблення (pits), то велика частина світла розсіюється. Аналогічно для оптичних дисків з одноразовим записуванням, якщо світло потрапляє на відбивну поверхню, воно відбивається майже повністю, а в тих точках, де диск «пропалювався», відбиття світла майже нульове. Оптичні диски з багаторазовим записуванням погано відбивають світло в тих точках, де є реєструвальний шар, що знаходиться в аморфному стані, і краще – там, де він кристалічний (однак відбивна здатність у цьому разі гірша, ніж дисків CD-ROM, DVD-ROM, CD-R, DVD-R і DVD+R, тому на старих дисководів перезаписувані диски можуть не зчитуватися).

DVD-диски можуть бути з *одношаровим* чи *двошаровим записуванням*, а також *однобічними* і *двобічними* (два однобічні диски склеєні між собою). При цьому товщина диска не змінюється (це досягається за рахунок зменшення товщини захисного шару).

Для DVD-дисків розміром 120 мм визначено такі значення ємностей:

- DVD-5 (тип A) – однобічний одношаровий диск ємністю 4,7 Гбайт;
- DVD-9 (тип C) – однобічний двошаровий диск ємністю 8,5 Гбайт;
- DVD-10 (тип B) – двобічний одношаровий диск ємністю 9,4 Гбайт;
- DVD-18 (тип D) – двобічний двошаровий диск ємністю 17 Гбайт.

Незважаючи на те, що ємність DVD-диска наводиться в гігабайтах (один гігабайт дорівнює 2^{30} байт), реально її вимірюють у мільярдах байтів (10^9 байт). Тому реальна ємність DVD-диска, виражена в гігабайтах, менша від зазначених вище. Так, ємність диска 4,7 Гбайт насправді становить $4,7 \cdot 10^9$ байт = 4,3 Гбайт, а ємність диска 17 Гбайт – 15,9 Гбайт.

Усі дисководи оптичних дисків мають форм-фактор 5,25 дюйма і сумісні за розмірами дисків, тобто можуть зчитувати як 120-міліметрові, так і 80-міліметрові диски, а також оптичні візитні картки (для зчитування дисків останніх двох типів у лотку дисковода передбачено поглиблення).

Швидкість передавання даних – це максимальна швидкість, за якої виконується обмін даних між дисководом і комп'ютером. Це найважливіша характеристика оптичного дисковода, що майже завжди наводиться разом з назвою моделі, причому її задають не кількістю мегабайтів за секунду, як для інших пристроїв зовнішньої пам'яті, а коефіцієнтом збільшення відносно базової швидкості.

Диски, виконані методом гарячого штампування (CD-ROM і DVD-ROM), забезпечують до 10 000 циклів безпомилкового зчитування даних і можуть зберігати дані від 50 до 300 років (за дотримання умов зберігання). Однократно записувані диски (CD-R, DVD-R і DVD+R) можуть зберігати дані від 20–30 до 250 років (залежно від використовуваного барвника), а диски CD-RW, DVD-RW і DVD+RW – від 25 до 100 років. Диски DVD-RAM можна перезаписувати до 100 000 разів, а тривалість зберігання даних на них за різними оцінками може досягати 30 років.

Flash-пам'ять. Останнім часом Flash-пам'ять стали використовувати не тільки як внутрішню пам'ять, але і як зовнішню, підключену до комп'ютера по інтерфейсу USB (1.1 чи 2.0). Перевага цього інтерфейсу полягає в тому, що пристрій можна вмикати і вимикати без перезавантаження комп'ютера. Деякі пристрої мають додаткову можливість підключення до радіомережі по інтерфейсу IEEE 802.11b.

▶ Пристрій Flash-пам'яті містить такі компоненти (рис. 2.8):

- одну чи кілька мікросхем Flash-пам'яті в корпусі TQPF;
- контролер, що керує доступом до мікросхем Flash-пам'яті;
- адаптер USB.

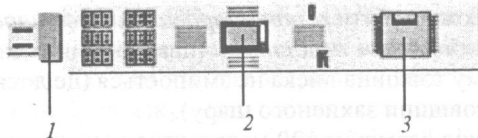


Рис. 2.8. Структура модуля Flash-пам'яті: 1 – адаптер USB; 2 – контролер пам'яті; 3 – мікросхема Flash-пам'яті

Пристрої Flash-пам'яті мають невеликі розміри і найрізноманітніші форми корпусів (пластмасові чи алюмінієві) (рис. 2.9). Звичайний пристрій має індикатор, що загоряється під час доступу до даних, та перемикач, що дозволяє чи забороняє записування у пам'ять. Крім того, деякі пристрої мають перемикач, що дозволяє чи забороняє зчитування даних без використання пароля. Пристрій Flash-пам'яті можна використовувати як завантажувальний пристрій.



Рис. 2.9. Зразки пристроїв зовнішньої Flash-пам'яті

Основні характеристики зовнішньої Flash-пам'яті: ємність, швидкість передавання даних і надійність. Пристрої ємністю 2 і 4 Гбайт зображені на рис. 2.9 угорі праворуч. Максимальна швидкість передавання даних для пристроїв Flash-пам'яті досягає 1,5 Мбайт/с (USB 1.1) і 50...60 Мбайт/с (для USB 2.0). Пристрої Flash-пам'яті мають такі показники надійності: час зберігання даних – до 10 років, кількість циклів перезаписування – до 1 млн.

2.6. Монітор та відеокарта

Два основні компоненти відеосистеми ПК – дисплей чи монітор (ці два слова використовують як синоніми) і відеокарта.

Дисплей (*display*) або монітор (*monitor*) – обов'язковий пристрій ПК, призначений для виведення чорно-білих або кольорових рухомих зображень. Виведені на екран монітора зображення є растровими, тобто становлять прямокутну матрицю точок, названих пікселями. За такого

подання враховують обмежену здатність ока, яка, починаючи з деякої відстані, сприймає дві близько розміщені точки як одну. Тобто матриця на екрані монітора сприймається як цілісне зображення.

Рухоме зображення на моніторах комп'ютерів передається як послідовність нерухомих зображень (кадрів), що швидко змінюються і відображають окремі фази руху. За досить високої частоти змінювання кадрів окремі кадри перестають сприйматися, і рух на екрані стає плавним.

Тепер для ПК використовують *три основні типи моніторів*:

- CRT-монітори (рис. 2.10, а);
- LCD-монітори (рис. 2.10, б);
- плазмові монітори (рис. 2.10, в).

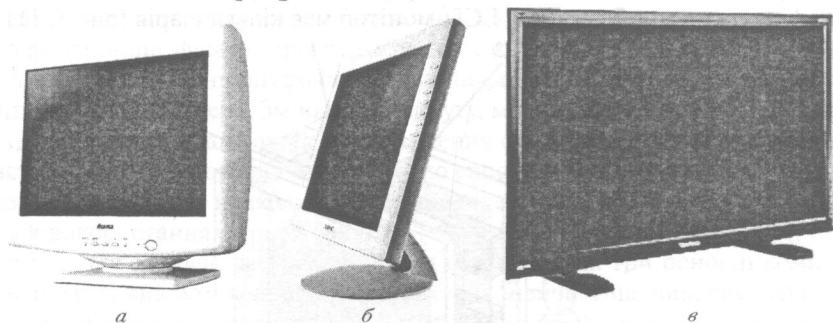


Рис. 2.10. Типи моніторів: а – CRT-монітор;
б – LCD-монітор; в – плазмовий монітор

CRT-монітори. Найпоширеніші натеper – монітори на основі електронно-променевої трубки – *CRT-монітори* (*Cathode Ray Tube* – катодна променева трубка).

CRT-монітори містять такі основні компоненти: електронно-променеву трубку, електронну схему керування, кнопки керування, роз'єкти або з'єднувальні кабелі, корпус.

Електронно-променева трубка – це скляна трубка, що нагадує велику колбу з пробкою без повітря усередині. У вузькій частині трубки розміщено *електронну гармату*, що складається з катода, нагрівника і фокусувальної системи. Широка частина є *екраном*, на який виводиться зображення. Внутрішню частину екрана покрито зернистою речовиною, що випускає світло під час бомбардування його зарядженими частинками (такі речовини називають *люмінофорами*). Різні люмінофори випускають кольори різної довжини хвилі (червоний, зелений, синій) або білий колір.

Електронна гармата випромінює під час нагрівання пучок електронів, що проходять через *модулятор*, який регулює інтенсивність цього пучка електронів і зумовлену цим яскравість зображення. *Відхильна система* дозволяє змінювати напрям пучка.

Монітор має роз'язття або кабель для підключення до відеокарти, що знаходиться в системному блоці, та кабель електроживлення.

Рідкокристалічні монітори. *Рідкокристалічний монітор*, чи *LCD-монітор* (*Liquid Crystal Display*) містить такі самі компоненти, що й CRT-монітор, однак формують пікселі зображення не пучки електронів, а *рідкі кристали*. Ці речовини названі так тому, що вони зазвичай знаходяться в рідинному стані, але при цьому мають властивості кристалічних тіл. Під впливом електрики молекули рідких кристалів, що мають довгасту форму, можуть змінювати свою орієнтацію і внаслідок цього змінювати властивості світлового променя, що проходить крізь них.

Екран LCD-монітора являє собою масив малих сегментів (пікселів) для формування зображення. LCD-монітор має кілька шарів (рис. 2.11).

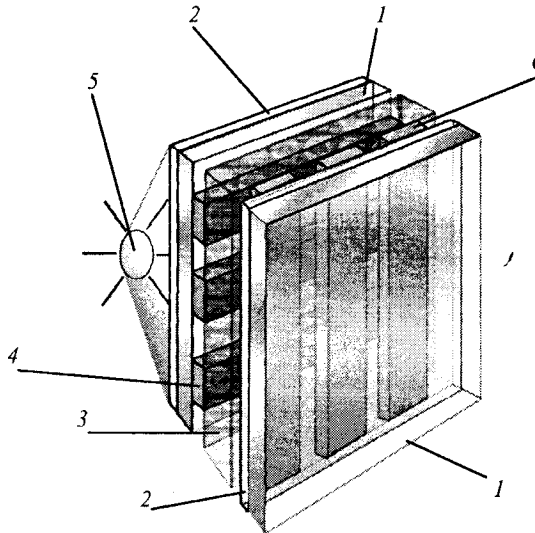


Рис. 2.11. LCD-монітор: 1 – скляна пластина; 2 – фільтр-поляризатор; 3 – рідкі кристали; 4 – горизонтальний електрод; 5 – джерело світла; 6 – вертикальний електрод

Ключову функцію деяких LCD-моніторів виконують дві плоскі панелі, зроблені з позбавленого від натрію і дуже чистого скляного матеріалу, названого субстратом чи підкладкою, що містить між собою тонкий шар рідких кристалів. Тому LCD-монітори, а також плазмові монітори, часто називають *плоскопанельними моніторами* (*flat panel monitors*).

Борозенки-електроди на кожній панелі розміщено паралельно (але перпендикулярно між панелями). Рідкі кристали в комірках можуть за допомогою електродів змінювати свою орієнтацію; такі комірки названо *закругленими нематичними* (*twisted nematic*) (від грецьк. пета – голка).

Рідкокристалічна панель освітлюється джерелом світла (залежно від місцезнаходження джерела рідкокристалічні панелі працюють на відображення чи на проходження світла). Оскільки джерело світла дає неполяризоване випромінювання, перший (внутрішній) фільтр-поляризатор пропускає світло тільки в одному напрямі поляризації. Напрямок поляризації другого зовнішнього фільтра-поляризатора повернено на 90° відносно напрямку поляризації першого фільтра. Коли до електродів якого-небудь пікселя прикладено напругу, спіраль рідких кристалів розправляється і не змінює напрямку поляризації світла, що проходить уздовж неї. У цьому разі світло затримується зовнішнім поляризаційним фільтром, і піксель має чорний колір. Якщо зняти напругу, спіраль закрутиться так, що кристали, які знаходяться на її кінцях, ляжуть у борозенки. Світло, пронизавши внутрішній поляризаційний фільтр, проходить уздовж спіралі, змінює свою поляризацію на 90° і тому пропускається зовнішнім фільтром, тобто формується світлий (білий) піксель. Змінюючи напругу, можна одержати сірі відтінки.

Для виведення кольорового зображення світло має виникати в задній частині LCD-монітора, щоб можна було спостерігати зображення високої якості, навіть якщо навколишнє середовище не є світлим. Колір створюється використанням трьох фільтрів (червоного, зеленого і синього), що виділяють з джерела випромінювання білого світла три основні компоненти. Щоб вивести кольорове зображення, найчастіше використовують таку властивість рідкокристалічної комірки: зі змінюванням напруженості електричного поля кут повороту площини поляризації випромінювання змінюється по-різному для компонентів світла з різною довжиною хвилі. Отже, випромінювання заданої довжини хвилі світла може відбиватися (чи поглинатися), тобто створювати заданий колір.

Технологія функціонування LCD-моніторів не може забезпечити швидку зміну даних на екрані. Через досить велику електричну ємність комірок напруга на них не може змінюватися досить швидко, тому картинка відновлюється повільно. Мала швидкість змінювання прозорості кристалів не дозволяє правильно відображати рухомі зображення. Монітори такої технології назвали *моніторами з пасивною матрицею (passive matrix)*. Незважаючи на застосування технологій поліпшення контрастності через збільшення кута повороту площини поляризації світла з 90° до 270° (у технології Super Twisted Nematic), ці монітори тепер майже не випускають.

Для *моніторів з активною матрицею (active matrix)* використовують окремі керувальні елементи (транзистори) для кожної комірки екрана, що компенсують вплив ємності комірок і дозволяють значно зменшувати час змінювання їх прозорості. Оскільки транзистори, розміщені на задній частині панелі, мають пропускати світло, їх реалізовано в пластикових плівках за технологією *TFT (Thin Film Transistor – тонкоплівковий транзистор)*. Монітори з використанням технології TFT називають *TFT-моніторами*.

Плазмові монітори. У плазмових моніторах формування зображення базується на використанні електричного розряду в плазмі.

Комірки плазмових екранів, як і рідкокристалічних, створюються накладанням один на одного двох скляних пластин, на які нанесено перпендикулярно одна до одної борозенки-електроди (рис. 2.12, *а*). Але простір між скляними пластинами заповнюється не рідкими кристалами, а інертним газом (наприклад, аргоном чи неоном) або їх сумішшю. Під дією високочастотної напруги, подаваної на електроди, у прилеглий до електрода газовій ділянці виникає електричний розряд (рис. 2.12, *б*). Плазма газового розряду випромінює світло в ультрафіолетовому діапазоні, що спричиняє світіння частинок люмінофора. Кожен піксель плазмового екрана містить три комірки з червоним, зеленим і синім люмінофорами. Інтенсивність кольору в кожній комірці залежить від величини розряду.

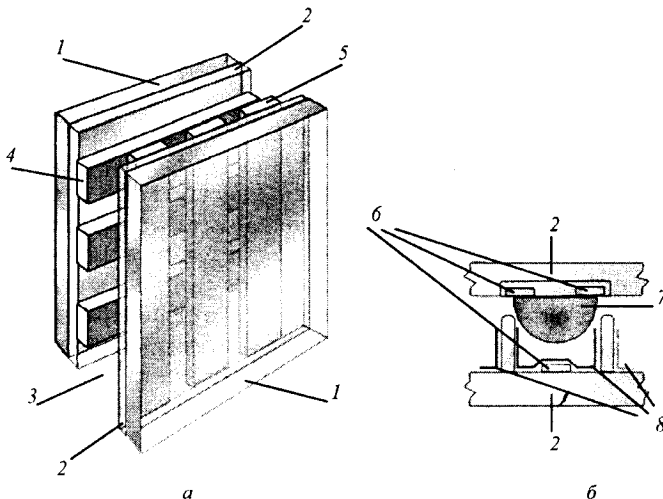


Рис. 2.12. Плазмовий монітор: *а* – структура екрана; *б* – розряд у комірці:
 1 – скляна пластина; 2 – ізолятор; 3 – аргонеонеова суміш; 4 – горизонтальний електрод; 5 – вертикальний електрод; 6 – електроди; 7 – розряд; 8 – люмінофори

Основними характеристиками усіх моніторів є:

- розмір екрана по діагоналі;
- роздільна здатність;
- частота регенерації;
- відповідність стандартам ТСО.

Діагоналлю екрана монітора, як і телевізора, називають відстань між лівим нижнім і правим верхнім кутами екрана. Цю відстань вимірюють у дюймах. Однак для CRT-моніторів під діагоналлю екрана виробники зазвичай вказують розмір діагоналі електронно-променевої трубки.

Оскільки трубку укладено в корпус, розмір екрана стає трохи меншим від розміру трубки (деякі виробники вказують обидва розміри). Так, для монітора 17" (43,18 см) дійсний розмір екрана становить 40,55 см. Рідкокристалічні чи плазмові монітори чорного поля майже не мають, тому реальний розмір екрана цих типів моніторів, якщо розмір діагоналі екрана однаковий із CRT-монітором, більший.

Стандартні монітори ПК – це монітори з діагоналями 15" і 17". Для професійної роботи з графікою використовують монітори з діагоналями 19", 21" чи 22".

Найбільші розміри (70" і більше) мають плазмові монітори.

LCD-монітори і плазмові монітори мають одну роздільну здатність, названу «рідною» (*native*), що відповідає максимальній фізичній роздільній здатності здатності CRT-моніторів. Ця роздільна здатність визначається кількістю пікселів, що для LCD-монітора і плазмових моніторів фіксована. Наприклад, якщо такий монітор має роздільну здатність 1024×768 пікселів, то це означає, що на кожному з 768 рядків розміщено 1024 електродів, тобто пікселів. При цьому є можливість використовувати і більш низьку роздільну здатність.

Для сумісної роботи відеокарти і монітора потрібне тимчасове узгодження переданих елементів зображення – *синхронізація*. Відеокарта формує два сигнали синхронізації, один з яких формує *горизонтальну частоту рядків* (у кілогерцах), а другий – *вертикальну частоту повторення кадрів* (у герцах). Частоту повторення кадрів часто називають *частотою регенерації*. Око людини сприймає зміну зображень як рухоме зображення в тому разі, коли частота зміни зображень не нижча за 20...25 Гц (тому в телебаченні використовують саме ці частоти). Однак для роботи з монітором на близькій відстані до екрана таких частот недостатньо.

Для CRT-моніторів час світіння люмінофорних елементів дуже малий, тому електронний промінь має проходити через кожний елемент люмінофорного шару досить часто, щоб не було помітно мерехтіння зображення. Якщо частота такого обходу екрана стає меншою за 70 Гц, то інерційності зорового сприйняття буде недостатньо для того, щоб зображення не мерехтіло, тому мінімально безпечною частотою кадрів вважається 75 Гц. Дослідження показали, що якщо частота вертикального розгорнення вища за 110 Гц, людина вже не помічатиме ніякого мерехтіння зображення.

Яскравість окремого елемента екрана LCD-монітора є незмінною протягом усього інтервалу часу між відновленнями картинки. Однак це є і недоліком, оскільки часта зміна зображення (наприклад, під час перегляду фільмів чи відеоігор) позначається на інерційності перемикання пікселів з одного стану в другий, що призводить до спотворення зображень. Частота регенерації LCD-моніторів становить 20...50 Гц, тому для виконання мультимедійних прикладних задач краще використовувати CRT-монітори.

Істотні характеристики для роботи з моніторами – *ергономічні характеристики* і *характеристики безпеки*. Ці характеристики для пристроїв комп'ютера визначаються стандартами TCO, розробленими *Шведською конфедерацією професійних службовців (The Swedish Confederation of Professional Employees)*. З 1998 р. усі роботи зі стандартизації виконуються компанією *TCO Development*, спеціально створеною конфедерацією для цієї мети. Стандарти TCO визначають вимоги до моніторів за такими напрямками: ергономіка (ergonomics), випромінювання (emissions), енергія (energy) і екологія (ecology). Відповідність монітора стандарту TCO засвідчується наклейкою з логотипом TCO.

Відеокарти. *Відеокарта*, чи *відеоадаптер* здійснює безпосереднє керування монітором. Вона передає керувальні сигнали електронній схемі монітора, тобто контролює процес формування зображення на екрані.

Конструктивно відеокарта являє собою плату, що вставляється в слот розширення PCI чи AGP.

Зображення, виведене на екран монітора, – це растрове зображення. Однак вихідними даними зображення для виведення можуть бути як растрові, так і векторні зображення. Тому майже всі сучасні відеокарти належать до комбінованих пристроїв, що, крім своєї головної функції – формування сигналів, відповідно до яких монітор виводить на екран растрове зображення, виконують графічні операції з побудови двовимірних (2D) і тривимірних (3D) векторних зображень і перетворюють їх у растрове зображення (для виведення на екран).

Основні компоненти сучасної відеокарти такі:

- Video BIOS;
- набір мікросхем;
- відеопам'ять;
- RAMDAC;
- роз'язтя підключення до шини і зовнішніх пристроїв.

Операції із зображеннями на екрані монітора виконуються за допомогою спеціальних команд. Набір таких команд міститься в постійному запам'ятовувальному пристрої (Flash-пам'яті – Video BIOS). Сучасні відеокарти підтримують також стандарт Plug&Play, тому в Video BIOS містяться відомості про модель, виробника і параметри відеокарти.

Набір мікросхем (chipset) визначає можливості відеокарти. Зазвичай в його склад входять 64-розрядний чи 128-розрядний спеціалізований суперконвеєрний і суперскалярний процесори та набір апаратних засобів як для оброблення відеозображень різних форматів, так і для оброблення векторної графіки.

Відеопам'ять являє собою пам'ять довільного доступу (RAM). Для відеокарт без графічного прискорювача використовували такі самі моделі динамічної пам'яті, що й для оперативної пам'яті, або моделі пам'яті,

спеціально розроблені для зберігання відеоданих, наприклад, VRAM (Video RAM), WRAM (Window RAM), SGRAM (Synchronous Graphics RAM) і MDRAM (Multi-bank RAM). Ці моделі відрізняються від моделей оперативної пам'яті підвищеною пропускнуою здатністю, потрібною для швидкого оброблення зображень.

У цифро-аналоговому перетворювачі RAM – *RAMDAC (RAM Digital-Analogue Converter)* цифрові дані відеопам'яті (коди кольорових пікселів) перетворюються в аналогову форму і передаються на монітор. Деякі відеокарти підтримують також стандарт DVI і можуть передавати цифрові дані з відеопам'яті (в обхід RAMDAC) прямо на монітор.

Розняття відеокарти можуть, крім розняттів для підключення монітора, містити також розняття для введення-виведення відеозображень у різних телевізійних форматах, підключення зовнішнього DVD-плеєра й інших пристроїв.

Ємність відеопам'яті сучасних відеокарт становить 128...256 Мбайт (ємність перших відеокарт дорівнювала 128 кбайт).

Додаткові можливості – підключення до відеокарти декількох моніторів (зазвичай від 2 до 4), а також підтримання відеокартою багатоекранного режиму (multi-screen mode), коли на екран монітора може виводитися кілька незалежних вікон.

2.7. Аудіосистема комп'ютера

У середині комп'ютера звукові дані подаються в дискретному (цифровому) вигляді. Перетворення звуку з аналогової форми в цифрову (на вході) і обернене перетворення (на виході) виконує основний компонент звукової системи – *звукова карта*. Для введення аналогового звуку використовують *мікрофон*, а для виведення – *акустичну систему*. Ці три компоненти й утворюють базову звукову систему комп'ютера. Крім них, за професійної роботи зі звуком можна використовувати магнітофони, CD-плеєри, DVD-плеєри, музичні клавіатури, синтезатори й інші пристрої.

Розняття для *вхідних аналогових інтерфейсів* дозволяють підключати (до аналогового входу) такі пристрої:

- мікрофон (розняття Mic In);
- магнітофон чи аудіоплеєр (розняття Line In);
- CD-дисковод чи DVD-дисковод (для виведення через звукову карту аудіоданих з цих пристроїв) – розняття CD In;
- TV-тюнер чи інший пристрій оброблення відеоданих (також для виведення аудіоданих з цих пристроїв) (розняття Aux In).

Сигнали з розняттів CD In і Aux In подаються прямо на аналоговий вихід звукової карти.

Вихідні аналогові інтерфейси зазвичай реалізовано за допомогою двох роз'язтів – Line Out1 і Line Out2, до яких можна підключати як акустичну систему комп'ютера, так і інші приймачі звуку (наприклад, магнітофон). До виходу Line Out1 можна підключати і головні телефони (навушники).

Усі вхідні та вихідні аналогові інтерфейси технічно реалізовані за допомогою роз'язття Mini Jack зі штекером діаметром 3,5 мм.

Мікрофон. Мікрофон – це пристрій, що сприймає звукові сигнали і перетворює їх в електричні сигнали, напруга чи сила струму в яких пропорційна амплітуді звукового сигналу, а частота відповідає частоті коливань звукового сигналу.

За способом перетворення звуку в електричні сигнали існують такі *типи мікрофонів*: динамічні, конденсаторні, електретні, п'єзокристалічні, стрічкові.

Акустична система. Акустична система комп'ютера виконує відтворення звуку. Хоча існує кілька технологій відтворення звуку (наприклад, електростатична технологія чи планарна магнітна технологія), у комп'ютерах, телевизорах та інших пристроях використовують таку саму технологію, що й у динамічних мікрофонах, але звукові хвилі генеруються у зворотному напрямі у разі зміни напруги електричного струму.

Акустичні системи зазвичай підключають до комп'ютера *по аналоговому інтерфейсу* Line Out1 звукової карти (у разі підключення за допомогою стереофонічного штекера Mini Jack) чи до виходів Line Out1 і Line Out2 за допомогою монофонічних штекерів.

2.8. Пристрої введення

2.8.1. Клавіатура

Клавіатуру (keyboard) призначено для введення в комп'ютер інформації від користувача.

Розміщення клавіш на клавіатурі (зони клавіатури) показано на рис. 2.13.

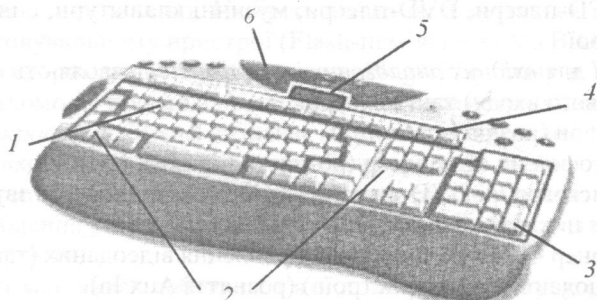


Рис. 2.13. Зони клавіатури: 1 – алфавітно-цифрові клавіші; 2 – керувальні клавіші; 3 – цифрові клавіші; 4 – функціональні клавіші; 5 – індикатори; 6 – додаткові клавіші

Зона 1 містить клавіші букв, цифр і розділових знаків. Найпоширеніший стандарт розміщення клавіш – QWERTY (ЙЦУКЕН) – за першими п'ятьма буквами першого ряду буквених клавіш.

Зона 2 містить клавіші керування курсором: →, ↓, ↑, ←, Home, End, Page Up, Page Down, видалення попереднього символу Back (чи ←) чи наступного символу Delete (Del), клавішу введення (Enter чи Return), клавішу регістра букв Shift, клавішу фіксації регістра букв Caps Lock, клавішу табуляції Tab, дві керувальні клавіші Ctrl (від Control), дві клавіші вибору Alt (від Alternate), клавішу переходу Esc (від Escape), клавішу виведення вмісту екрана Print Screen, клавішу блокування прокручування Scroll Lock і клавішу переривання Break.

Зону 3 можна використовувати в двох режимах за допомогою клавіші NumLock: натисканням – вводити дані, що містять тільки цифри і знаки арифметичних дій, відпусканням – дублювання клавішами зони керування курсором.

Зона 4 (функціональні клавіші) містить клавіші F1–F12, дія яких визначається виконуваною програмою.

Зона 5 містить індикатори стану клавіш Caps Lock, Num Lock і Scroll Lock.

Зона 6 містить клавіші, кнопки й індикатори, склад яких залежить від моделі клавіатури (цієї зони в багатьох моделях клавіатури немає).

Під комп'ютерною клавіатурою знаходяться грати (матриця) з провідників. Кожна клавіша розміщена над перетинанням двох провідників і в разі натискання замикає електричне коло. Якщо натиснуто яку-небудь клавішу, що замкнула ланцюг, схема декодування, запам'ятавши номер рядка, починає опитувати стовпці, причому стан натискання цієї клавіші фіксується доти, доки клавішу не буде відпущено, а опитування клавіатури продовжуватиметься, тобто в цьому разі можна зафіксувати одночасне натискання двох чи більше клавіш. У пам'яті ROM контролера клавіатури (зазвичай ємністю 8 кбайт) зберігається таблиця відповідності номера рядка і стовпця натиснутої клавіші її коду. Якщо натиснуто більше однієї клавіші, то також шукається відповідність клавіш визначеному коду. Свої коди мають праві і ліві клавіші – Shift, Alt чи Ctrl, а також комбінація цих клавіш з якою-небудь іншою клавішею. Так, якщо одночасно з клавішею, що відповідає букві q, буде натиснуто клавішу Shift, то буде знайдено код букви Q. Однобайтові чи двобайтові коди окремих клавіш і комбінацію клавіш називають *скан-кодами*. Для перших 128 символів скан-код збігається з кодом ASCII.

2.8.2. Маніпулятори

Існують такі основні *типи маніпуляторів*:

– «мишка» (рис. 2.14, а);

- трекбол (рис. 2.14, б);
- джойстик (рис. 2.14, в);
- сенсорна панель (рис. 2.14, з і д);
- сенсорний екран (рис. 2.14, е);
- диджитайзер (рис. 2.14, ж);
- графічні планшети (рис. 2.14, и і к).

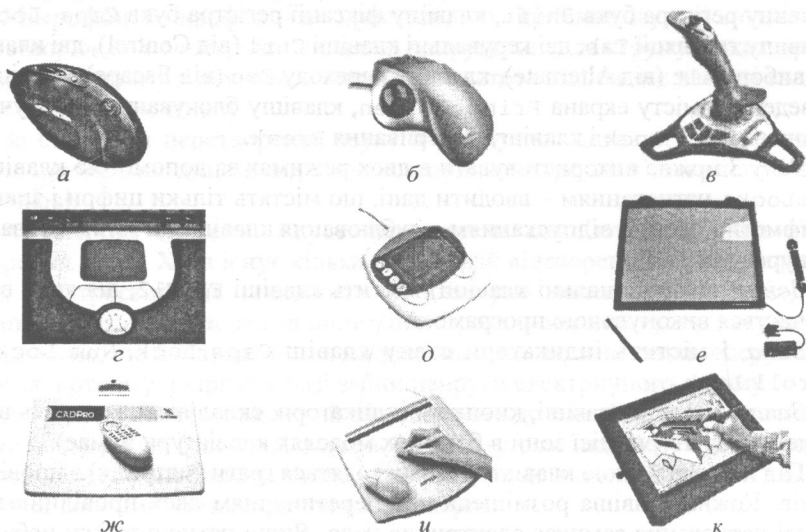


Рис. 2.14. Основні типи маніпуляторів:

- а – «мишка»; б – трекбол; в – джойстик; г і д – сенсорні панелі;
 е – сенсорний екран; ж – диджитайзер; и і к – графічні планшети

«Мишка». Маніпулятор «мишка» одержала свою назву завдяки формі та принципу роботи. Переміщення «мишки» по поверхні столу чи килимка програмою керування «мишкою» (драйвером «мишки») перетворюється в набір імпульсів, що передаються комп'ютеру. У разі натискання однієї чи більше клавiш «мишки» код клавiші також передається комп'ютеру.

Основний робочий орган оптико-механічної «мишки» (рис. 2.15, а) – металева (для ваги) куля, покрита гумою. Під час переміщення маніпулятора по поверхні столу куля обертається і передає обертання двом бічним валикам. Вісь обертання одного з валиків вертикальна, а другого – горизонтальна. На цих осях встановлено диски з растровими отворами, що обертаються між двома пластмасовими цоколями. На першому цоколі знаходиться джерело світла в інфрачервоному діапазоні, а на другому – фоточутливий елемент (фотодіод, фоторезистор чи фототранзистор). Цей елемент дуже точно визначає, де знаходиться джерело світла: перед отвором чи за пластмасовою перегородкою диска. Оскільки таких растрових дисків два,

то порядок висвітлення фоточутливих елементів визначає напрям переміщення «мишки», а частота імпульсів, що надходять, – швидкість.

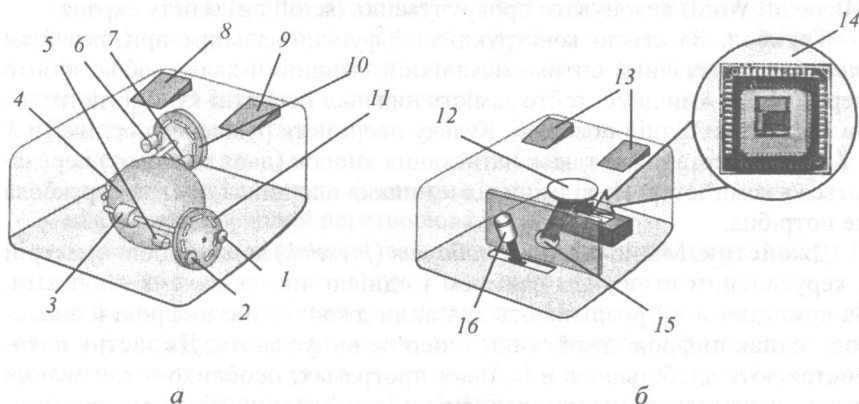


Рис. 2.15. Типи «мишок»: а – оптико-механічна; б – оптична:
 1, 2 – фотодетектори; 3 – валики; 4 – кулька; 5, 6, 7 – джерела світла; 8, 9 – кнопки; 10, 11 – диски з прорізами; 12 – дзеркало;
 13 – кнопки; 14 – елемент CMOS; 15 – лінза; 16 – LED

Оптична «мишка» (рис. 2.15, б) на відміну від оптико-механічної «мишки» не має рухомих частин. «Мишка» містить діод, що випромінює світло – LED (Light-Emitting Diode). Цей діод випромінює червоний колір в імпульсному режимі (приблизно 1500 імпульсів/с). Після відображення від будь-якої поверхні світло попадає на світлочутливий елемент, виконаний на основі CMOS. Такі ж елементи використовуються і в пам'яті CMOS, і в цифрових фотоапаратах, і відеокамерах. Елемент CMOS посилає кожне отримане зображення поверхні в цифровий сигнальний процесор DSP, що порівнює отримане зображення з попереднім і визначає (на основі цього порівняння) напрям і швидкість переміщення «мишки». Потім ці дані пересилаються в комп'ютер, який кілька сотень раз за секунду переміщає курсор, що створює враження плавності переміщення. Оптична «мишка» не потребує килимка і може працювати майже на будь-якій поверхні (за винятком відбивних поверхонь, наприклад, дзеркала).

Якщо раніше «мишку» підмикали до комп'ютера по інтерфейсу послідовного порту, то тепер для цього використовують роз'єтня PS/2 (таке саме, як і для під'єднання клавіатури), однак майже у всіх нових моделях «мишок» використовують інтерфейс USB. Існують також «мишки» з шинним інтерфейсом (Bus Mouse), що підключаються до власної карти розширення. Існують і безпроводні мишки: з передавачем інфрачервоного випромінювання, радіомишка (яка передає дані за допомогою радіосигналу), і мишки з використанням інтерфейсу Bluetooth.

Найпоширеніший додатковий елемент «мишки» – коліщатко, що дозволяє в деяких прикладних задачах (наприклад, у текстовому редакторі Microsoft Word) виконувати прокручування (scrolling) вмісту екрана.

Трекбол. За своєю конструкцією і функціональним призначенням *трекбол* аналогічний оптико-механічній «мишці» і являє собою нібито перевернуту «мишку», тобто замість нижньої поверхні кулька міститься на верхній чи бічній поверхні. Кульку обертають рукою, а координати X і Y цього обертання, а також натискання кнопок (двох чи трьох) передаються в комп'ютер. На відміну від «мишки» площинка для руху трекбола не потрібна.

Джойстик. Маніпулятор *джойстик* (*joystick*) являє собою пристрій з керувальним стрижнем-важелем і однією чи декількома кнопками. За принципом дії розрізняють два види джойстиків: цифрові й аналогові, однак цифрові джойстики тепер не випускають. Джойстик використовують здебільшого в ігрових програмах, особливо в так званих іграх-симуляторах (наприклад, імітація керування літаками чи автомашинами).

Сенсорна панель. *Сенсорна панель* (*touch pad*) – це пристрій, альтернативний «мишці» чи трекболу. Переміщенням пальця по поверхні панелі можна переміщувати курсор на екрані монітора. Сенсорні панелі зазвичай використовуються замість «мишки» у блокнотних та настільних комп'ютерах.

Сенсорний екран. *Сенсорний екран* (*touch screen*) – це екран, що реагує на натискання пальцем чи стилем (пластмасовим стрижнем із закругленим наконечником). На відміну від «мишки», трекбола, джойстика чи сенсорної панелі, у яких точка екрана, яку потрібно виокремити, визначається переміщеннями курсора, залежними, у свою чергу, від переміщення маніпулятора, сенсорний екран дозволяє виділити потрібну точку просто натисканням пальцем чи стилем у цій точці екрана. Контролер сенсорного екрана визначає координати точки дотику і передає їх операційній системі комп'ютера. Сенсорні екрани використовуються не тільки в комп'ютерах, але й в ігрових автоматах і електронних кіосках.

Диджитайзери. *Диджитайзер*, як і «мишка», указує позицію на екрані монітора. Однак на відміну від «мишки», що переміщує курсор з деякої позиції в заданому напрямі, диджитайзер відображає точну позицію (координати) обраної точки на екрані. Диджитайзери використовують головним чином у картографії, будівництві й архітектурі для одержання на комп'ютері точної копії з паперового носія (іноді великих розмірів).

Графічні планшети. Графічні планшети – різновид диджитайзерів, що реалізовані за своїми технологіями і мають свою сферу використання (створення рисунків, а також коректування рисунків і фотографій тощо).

2.8.3. Сканери

Сканер – це пристрій введення, що перетворює зображення в його цифрову форму (по точках) і передає цей образ у комп'ютер. Зображення, що вводиться, може бути текстом, малюнком, фотографією і навіть тривимірним об'єктом невеликої висоти.

Основні компоненти сканера:

- джерело світла;
- світлочутливі елементи (сенсори);
- алфавітно-цифровий перетворювач.

Джерело світла спрямовує світловий потік на сканований об'єкт (у деяких типах сканерів використовується природне освітлення).

Світловий потік, відбиваючись від об'єкта (непрозорого) чи проходячи через об'єкт, потрапляє на *сенсори*, що перетворюють значення інтенсивності падаючого на них світла в аналогові значення електричної напруги.

Сенсор може просто фіксувати наявність і відсутність світла в певній точці об'єкта (чорно-біле сканування). Якщо сенсор фіксує загальну інтенсивність падаючого на нього світла, то таке сканування називають «сірим». Для відображення кольору об'єкта використовують модель RGB. У цьому разі для вимірювання інтенсивності кожного кольорового компонента (червоного, зеленого і синього) застосовують окремий сенсор.

Алфавітно-цифровий перетворювач, виконаний на одній чи декількох мікросхемах, одержує на вході дані від сенсорів і перетворює їх у цифрову форму.

Сканери розрізняються за способом переміщення зчитувального пристрою і об'єкта один відносно одного:

- ручні;
- барабанні;
- планшетні;
- рулонні;
- роликові;
- плівкові (сканери слайдів);
- проєкційні.

У *ручних сканерах (handheld scanners)* переміщення пристрою відносно паперу здійснює сам користувач (рис. 2.16, а, б, в і г).

У *барабанних сканерах (drum scanners)* оригінал закріплюється на поверхні прозорого циліндра (барабана), стійкість якого забезпечується масивною основою (рис. 2.16, д). Барабан обертається з великою швидкістю (300...1350 об/хв), а сканувальний датчик, який знаходиться поруч з ним, зчитує зображення по пікселях.

У *планшетних сканерах (flatbed scanners)* сканувальна головка переміщується відносно паперу за допомогою крокового двигуна (рис. 2.16, е, ж).



Рис. 2.16. Типи сканерів: а, б, в, г – ручні; д – барабанний; е, ж – планшетні; и – рулонний; к – роликівий; л – сканер слайдів; м – проєкційний

Сканувальна головка рулонних сканерів (*sheetfed scanners*) стоїть на місці, а папір переміщується відносно неї за допомогою протяжного механізму (рис. 2.16, и). Основна перевага рулонних сканерів – мала витрата електроенергії і невеликі габарити дозволяють їм працювати на акумуляторних батареях; їх зазвичай використовують разом з портативними комп'ютерами.

У роликівих сканерах (*rollfed scanners*) папір також переміщується відносно головки за допомогою роликів. Такі сканери використовують для сканування зображень великого формату (рис. 2.16, к).

Сканери слайдів (*slide scanners*) або плівкові сканери (*film scanners*) забезпечують цифрацію слайдів чи кадрів плівки (рис. 2.16, л).

У проєкційних сканерах (*overhead scanners*) зчитувальна частина переміщується за допомогою мікромеханізму, який спрямовує «око» сканера на кожну лінію оригіналу (у більш сучасних моделях), або містить матрицю елементів CCD (рис. 2.16, м). Основою сканера може бути LCD-монітор або плоска панель (іноді з проєкційною лампою).

Роздільну здатність сканера вимірюють кількістю точок на дюйм – dpi (dots per inch). Для сканерів визначено оптичну роздільну здатність,

що дорівнює кількості елементів на лінійці фотодетекторів (роздільна здатність по горизонталі). Крім того, для сканерів визначено також роздільну здатність по вертикалі (у разі переміщення оригіналу відносно механізму сканування). Існує також дійсна роздільна здатність, яка визначається за допомогою сканування спеціального штрихового зразка.

Ручні сканери можуть підтримувати кілька різних значень оптичної роздільної здатності (у діапазоні 100...800 dpi з кроком 100 dpi). Барабанні сканери мають найвищі значення роздільної здатності: оптичну роздільну здатність (по горизонталі) – до 12000 dpi, логічну – до 64000 dpi. Планшетні сканери за технологією CCD середнього рівня мають оптичну роздільну здатність 600×1200 dpi, а професійні моделі – 1200×2400 dpi. Логічна роздільна здатність цих сканерів може досягати 19200 dpi. Ручонні сканери мають оптичну роздільну здатність до 600 dpi по горизонталі і таку саму логічну роздільну здатність, а роликові сканери – відповідно 500 і 2400 dpi. Оптична роздільна здатність плівкових сканерів може досягати 6000×6000 dpi, а проєкційна – визначається за кількістю елементів у матриці CCD і досягає значення 1360×1024 dpi.

2.8.4. Цифрові фотокамери

Результатом знімання за допомогою *цифрової фотокамери* є оцифровані кадри, що зберігаються в пам'яті фотокамери. Отримані кадри можна перенести або в комп'ютер для подальшого оброблення, або в спеціальний принтер, який зазвичай називають *фотопринтером*.

Основні компоненти цифрової фотокамери (рис. 2.17):

- оптична система;
- сенсори й аналого-цифровий перетворювач;
- блок керування;
- пам'ять для зберігання фотознімків;
- пристрої індикації і керування;
- система електроживлення.

Оптична система цифрової камери містить:

- об'єктив, діафрагму і затвор;
- систему автофокусування;
- систему зміни фокусної відстані;
- видошукач;
- імпульсний спалах.

Лінзи (lenses) *об'єктива* фотокамери фокусують світловий потік на сенсорах. За лінзами об'єктива у фотокамері розміщено *діафрагму* (diaphragm), що регулює кількість світла, яке потрапляє на сервер. За діафрагмою може розміщуватися *затвор* (shutter) – шторка, що відкривається тільки в момент знімання. Для цифрових фотокамер використовують електронний затвор, що вмикає чи вимикає сенсор для приймання світлового потоку.

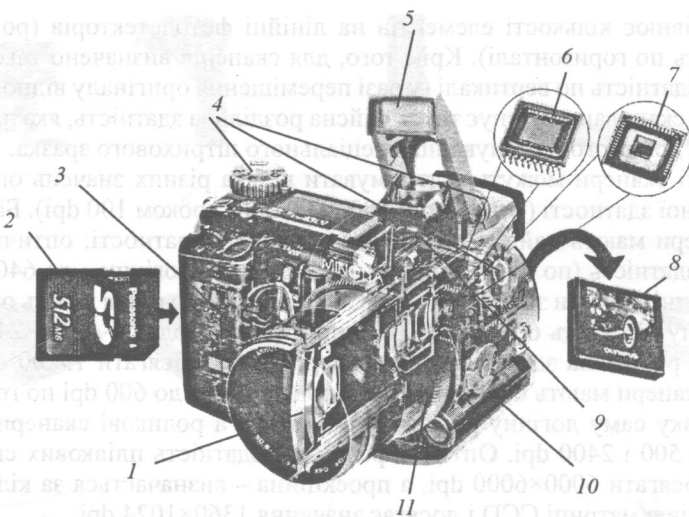


Рис. 2.17. Компоненти цифрової фотокамери: 1 – оптична система; 2 – Flash-пам'ять; 3 – плата АЦП і блоку керування; 4 – пристрої індикації і керування; 5 – спалах; 6 – матриця CCD; 7 – матриця CMOS; 8 – LCD; 9 – шина матриці АЦП; 10 – батарейки; 11 – матриця сенсорів

Діафрагма і затвор разом визначають витримку (*exposure*) знімка – кількість світла і тривалість його впливу на сенсор. Витримка зазвичай визначається автоматично за освітленістю центральної ділянки знімка, однак у деяких професійних камерах кадр розбивається на декілька ділянок, для кожної з яких визначається значення освітленості, а потім за середнім значенням освітленості визначається витримка знімка.

Багато цифрових фотокамер мають механізм автоматичного фокусування (*autofocus system*). Цей механізм використовують і в деяких моделях плівкових фотокамер.

Поряд з об'єктивами, у яких фокусна відстань фіксована, у фотокамері можуть використовуватися й об'єктиви зі змінною фокусною відстанню (*трансфокатори*). Оскільки зі збільшенням фокусної відстані об'єкт ніби збільшується, такі лінзи називають збільшувальними лінзами (*зоом*). Деякі фотокамери мають декілька різних об'єктивів з фіксованими чи змінними фокусними відстанями. Крім оптичного збільшення (за допомогою лінз), можна використовувати і цифрове збільшення кадру.

Видошукач (*viewfinder*) використовують для напрямлення фотокамери на об'єкт знімання. У цифрових фотокамерах використовують один із трьох видів видошукачів: прямих (оптичний), дзеркальний та електронний.

Імпульсний спалах (*flash*) дозволяє одержати короткочасне освітлення об'єкта знімання з високою яскравістю.

Важлива характеристика цифрової фотокамери – *кількість пікселів* (окремих елементів) у сенсорі. Цифрова фотографія поки що поступається плівковій, у якій для плівки ISO 100 за щільності 200 зерен на міліметр загальна кількість зерен дорівнює 34 500 000.

Сенсор цифрових фотокамер має фіксований розмір, названий *форм-фактором*. Найпоширеніші форм-фактори сенсорів – 1/3, 1/1,8 і 2/3 дюйма по діагоналі.

Тип сенсора (CCD чи CMOS) визначає якість знімків і ціну цифрової камери. Елементи CCD більш чутливі до світла і точніше визначають його інтенсивність. Крім того, у них майже немає паразитного – *темного струму*. Кількість пікселів у сенсорах CCD більша, ніж у сенсорах CMOS, однак цей діапазон постійно скорочується. Водночас елементи CMOS споживають значно менше електроенергії і дешевші за масового виробництва. Оскільки сенсори CCD забезпечують кращу якість знімків, їх використовують у професійних цифрових фотокамерах, а сенсори CMOS – у фотокамерах масового попиту.

2.8.5. Відеокамери

Відеокамери (*VCR – Video Camera Recorders*) виконують знімання послідовності кадрів із заданою кількістю кадрів за секунду – *fps (frames per second)*. Основні компоненти відеокамери:

- оптична система;
- пам'ять для зберігання результатів знімання;
- звукова підсистема.

Оптична система відеокамери містить:

- об'єктив, діафрагму і затвор;
- систему зміни фокусної відстані;
- систему автофокусування;
- систему стабілізації зображення;
- вищошукач.

Об'єктив, діафрагма, затвор, система зміни фокусної відстані відеокамери і система автофокусування функціонують так само, як і відповідні системи фотокамери.

Для зберігання результатів знімання в деяких моделях цифрових відеокамер як альтернативний пристрій можна використовувати і Flash-пам'ять. Для зменшення обсягу збережуваних даних зазвичай застосовують один з методів ущільнення відеофайлів (MPEG-2 чи MPEG-4).

Звукова підсистема відеокамери включає до свого складу мікрофон чи мікрофони (або вбудовані в камеру, або зовнішні мікрофони, які вставляються в спеціальні гнізда, або мікрофони, підімкнені як окремі пристрої). Блок керування виконує синхронне записування звукового супроводу і (для цифрових відеокамер) перетворення аудіоданих у цифрову форму.

Різновидом цифрових відеокамер є Web-камери.

Web-камера (Web camera чи просто *Webcam*) містить такі самі компоненти, що й звичайна цифрова відеокамера, але програми, що постачаються з нею, призначені для передавання результатів знімання в мережу Internet. Залежно від спеціалізації існують кілька типів Web-камер. У Web-камерах, використовуваних у системах слідкування й охоронних системах, зняті відеофрагменти зберігаються в Flash-пам'яті (з ущільненням даних в одному з відеоформатів), причому, якщо пам'ять заповнюється повністю, то попередній вміст стирається. Збережені в Web-камері відеодані доступні як Web-сторінки. Підключена до мережі Web-камера одержує адресу, за якою можна звертатися до її Web-сторінок з будь-якого Web-браузера мережі. Деякі Web-камери оснащені детекторами руху, причому окремі камери починають записувати тільки у разі спрацювання детектора руху.

2.9. Пристрої виведення

2.9.1. Принтери

Принципи друкування і типи принтерів. Друкувальні пристрої комп'ютерів чи *принтери (printers)* використовують для одержання «твердої» копії (hard copy) файлів у пам'яті комп'ютера. Основний вид носія для одержання «твердої» копії – паперові аркуші різних форматів. Однак деякі типи сучасних принтерів дозволяють вивід на інші носії (наприклад, прозорі плівки).

Використовують такі основні типи принтерів:

- матричні принтери – звичайні (рис. 2.18, *a*) і лінійні (рядкові) (рис. 2.18, *б*);
- струминні і світлодіодні принтери – звичайні (рис. 2.18, *в*) і портативні (мобільні) (рис. 2.18, *г*);
- лазерні принтери – чорно-білі (рис. 2.18, *д*) і кольорові (рис. 2.18, *е*);
- твердочорнильні принтери (рис. 2.18, *ж*);
- сублімаційні принтери (рис. 2.18, *и*);
- воскові принтери (рис. 2.18, *к*);
- автохромні принтери (рис. 2.18, *л*).

Матричний принтер працює за контактною технологією, усі інші типи принтерів – *безконтактні*, тобто механізм друкування безпосередньо не торкається носія. Усі типи принтерів друкують текст не по символах, як друкарська машинка, а по точках, тобто так, як виконується виведення на екран монітора (для позначення найменшого елемента виведення на принтери використовують саме термін *точка* – *dot*, а не термін *піксель*).

Більшість матричних принтерів друкують тільки в чорно-білому режимі і частково емулюють «сірий» режим. До окремих моделей матричних принтерів включено додаткову можливість друкувати з використанням тільки чотирьох кольорів: чорного, червоного, синього і жовтого. Інші типи принтерів забезпечують чорно-білий, «сірий» і кольоровий режими друку.

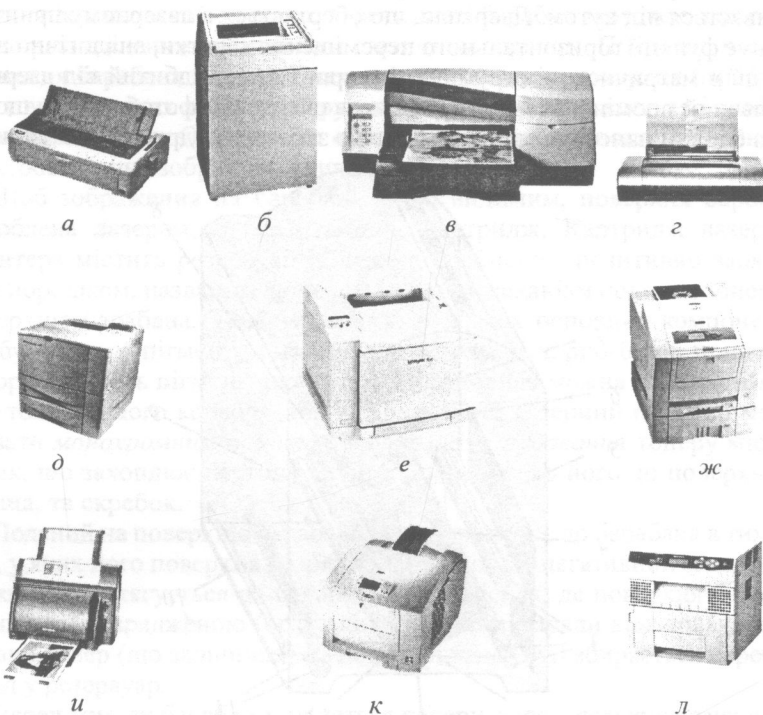


Рис. 2.18. Основні типи принтерів: а – матричний; б – лінійний (рядковий) матричний; в – струминний; г – портативний (мобільний) струминний; д – чорно-білий лазерний; е – кольоровий лазерний; ж – твердочорнильний; и – сублімаційний; к – восковий; л – автохромний

Струминні принтери (*inkjet printers* – чорнильно-струминні принтери), так само, як і матричні, виводять зображення по точках, але точки утворюються за допомогою дуже маленьких крапель рідкого чорнила.

Лазерні принтери (*laser printers*) для фіксації точок на папері застосовують промінь лазера.

Чорно-білий лазерний принтер містить такі основні компоненти (рис. 2.19): блок лазерного сканування, блок керування, фотобарабан і блок нанесення тонера на поверхню паперу, блок протягання паперу.

Блок лазерного сканування принтера містить імпульсний лазер і обертове багатогранне дзеркало. Блок керування посиляє в лазер сигнали, що змушують його генерувати імпульси в тих місцях сторінки, що містять чорні точки. Сам лазер залишається нерухомим, тому, щоб розгорнути генеровані імпульси в рядок, використовується обертове багатогранне дзеркало. Промінь світла від лазера потрапляє на поверхню дзеркала і відбивається під кутом. Дзеркало, що обертається в лазерному принтері, виконує функції горизонтального переміщення каретки, аналогічно подібній дії в матричному і струминному принтерах. Відбитий від дзеркала імпульсний промінь світла потрапляє на обертовий фотобарабан, що разом із блоком нанесення тонеру на папір забезпечує друкування даних.

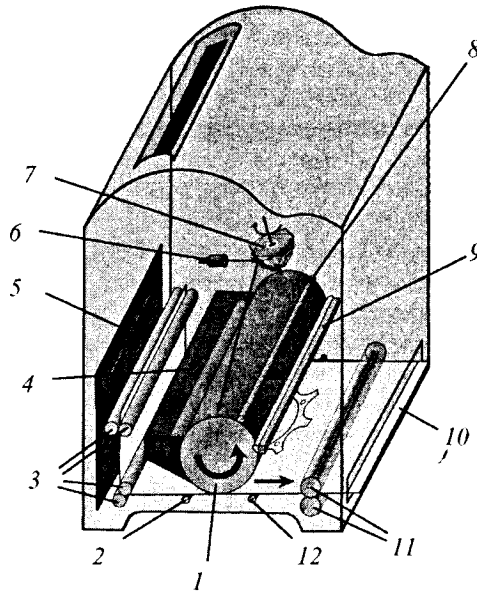


Рис. 2.19. Основні компоненти лазерного принтера: 1 – фотобарабан; 2 – коронувальний передавальний ролик; 3 – валики; 4 – картридж з тонером; 5 – блок керування; 6 – лазер; 7 – відхильне дзеркало; 8 – зарядовий коронувальний ролик; 9 – розрядна лампа; 10 – папір; 11 – грубка; 12 – коронувальний очищувальний ролик

Блок нанесення тонера на папір містить такі компоненти:

- зарядний коронувальний ролик (чи провід);
- картридж із тонером;
- передавальний коронувальний ролик (чи провід);
- коронувальний очищувальний ролик (чи провід);
- грубку.

Фотобарабан (photodrum) являє собою металевий циліндр з діаметром приблизно 7,5 см, покритий тонкою плівкою світлопровідного напівпровідника. До того, як на барабан потраплять промені від лазера, його поверхня проходить під *зарядовим коронувальним роликком чи провідом (charge corona roller or wire)*. На цей ролик чи провід подається висока напруга, що зумовлює виникнення довкола нього світної іонізованої ділянки, названої *короною*. У результаті барабан заряджається рівномірно розподіленим позитивним статичним зарядом. Потім під час проходження барабана під променем світла від лазера поверхня барабана в тих точках, у яких лазер згенерував імпульси, розряджається і здобуває негативний заряд. Отже, лазер «малює» на барабані поки що невидиме електростатичне зображення виведених даних.

Щоб зображення на барабані стало видимим, поверхня барабана, оброблена лазером, проходить повз картридж. Картридж лазерного принтера містить резервуар із дуже подрібненим позитивно зарядженим порошком, названим *тонером (toner)*, і механізм подання тонеру на поверхню барабана. Тонер складається з двох основних компонентів: фарбувального пігменту і частини пластику. У чорно-білих принтерах використовують пігмент чорного кольору, однак можна використовувати і тонер іншого кольору, тому чорно-білий лазерний принтер можна назвати *монохромним принтером*. Механізм подавання тонеру містить валик, що захоплює частину тонеру і переносить його до поверхні барабана, та скребок.

Поданий на поверхню барабана тонер прилипає до барабана в тих місцях, у яких його поверхня розряджена, тобто має негативний заряд, і, навпаки, не притягується до барабана в тих місцях, де поверхня барабана залишилася зарядженою (оскільки однойменні заряди відштовхуються). Зайвий тонер (що залишився на поверхні барабана) збирається скребком назад у резервуар.

Перед тим, як барабан торкнеться паперу, *коронувальний валик чи передавальний провід (transfer corona roller or wire)* переносить на папір негативний електростатичний заряд. Цей заряд більший за величиною від негативного заряду на барабані, тому, коли барабан торкається паперу, тонер прилипає до нього, точно повторюючи зафіксоване на барабані зображення чи текст.

Для того щоб папір, у свою чергу, не прилипав до барабана він відразу після захоплення тонеру розряджається за допомогою *очищувального коронувального валика чи проводу (detach corona roller or wire)*.

І, нарешті, папір протягається через *зрубку (fuser)* – два валики з нагрівальними елементами всередині. Під впливом високої температури (180...200° С) і тиску частинки пластику в тонері розплавляються і фіксують зображення чи текст на поверхні паперу.

Після того, як тонер перенесено на папір, поверхня барабана проходить повз *розрядну лампу*, світло якої стирає вже непотрібне електростатичне зображення. Цикл друкування повторюється доти, доки всі виведені дані будуть перенесені на поверхню паперу.

Таку схему роботи лазерного принтера називають *друкуванням «білим по чорному»*. Перед початком друкування «чорним по білому» барабан заряджається негативно. Потім лазер заряджає барабан позитивним зарядом у тих місцях, які він висвітлює. У цьому разі тонер заряджається негативним зарядом, а передавальний коронувальний провід чи валик повідомляють паперу позитивний заряд.

Блок протягання паперу в лазерному принтері, так само, як і в струминному, забезпечує переміщення паперу із вхідного лотка у вихідний.

Блок протягання паперу кольорових лазерних і світлодіодних принтерів забезпечує подавання паперу послідовно для чотирьох блоків нанесення тонеру на папір (для чотирьох базових кольорів – ціанового, пурпурного, жовтого і чорного). Кожний з чотирьох блоків використовує свій власний блок лазерного сканування чи світлодіодну лінійку з фокусувальною лінзою. Блок керування кольорового лазерного чи світлодіодного принтера розділяє виведені на друк дані на базові кольорні компоненти відповідно до використовуваних алгоритмів розмивання і згладжування, а також забезпечує точну синхронізацію протягання паперу і роботу чотирьох блоків формування електростатичного зображення разом із блоками нанесення тонеру на папір.

Твердочорнильні принтери. Чорнило для *твердочорнильних принтерів (solid ink printers)* розробила за спеціальною технологією фірма *Xerox*. При кімнатній температурі це чорнило зберігається у твердому стані у вигляді невеликих брикетів. Якщо тверде чорнило нагріти до певної температури, то воно становиться рідким (переходить у рідку фазу), але в процесі друкування дуже швидко знову набуває твердого стану (фази). Тому твердочорнильні принтери іноді називають принтерами *зі зміною фази (phase-changed printers)*.

Сублімаційні (dye-sublimation printers) і воскові принтери (wax printers), так само, як і твердочорнильні, є термічними принтерами, тобто використовують для нанесення кольорового зображення на папір нагрівання барвників, що знаходяться у твердому стані.

Барвник (dye), використовуваний у сублімаційних принтерах, під час нагрівання переходить із твердого стану в газоподібний без проміжного рідкого стану (такий процес називають сублімацією). Воскові принтери використовують рідке чорнило, що утворюється у процесі нагрівання твердого барвника на основі воску і застигає на папері.

Автохромні принтери, які розробила фірма *Fuji*, відрізняються від інших типів принтерів тим, що, крім спеціального паперу, для друкування

не потрібно ніяких інших видаткових матеріалів (чорнила, тонеру чи барвника). Нижній шар спеціального паперу є основою, на яку наносяться три шари, що містять мікрокапсули відповідно сіанового, пурпурного і жовтого кольорів. Ці мікрокапсули «проявляються» на папері під час нагрівання паперу. Зверху папір покрито тонким захисним теплостійким шаром.

Основна характеристика, що визначає якість друкування принтера, – роздільна здатність, вимірювана у кількості точок друкованого зображення на дюйм – dpi (dots per inch) чи на міліметр.

Суттєвою характеристикою принтера є *швидкість друкування*. Швидкість друкування *матричних принтерів* вимірюється у символах за секунду – cps (characters per second). Швидкість друкування інших типів принтерів вказується в кількості друкованих сторінок за хвилину – ppm (pages per minute) чи як час друкування однієї сторінки (у секундах).

2.9.2. Плотери

Плотери (plotters), як і принтери, призначені для одержання твердих копій даних. Назва пристрою походить від слова plot (креслити), і дійсно перший тип плотерів – *перові плотери (pen plotters)* призначалися для виведення креслень, зображень чи текстів на папір за допомогою друкарського вузла (на відміну від принтера, що друкує по точках). У друкарських вузлах використовуються спеціальні фломастери, чорнильні і кулькові ручки, а також інші пристрої, що забезпечують різну ширину ліній, насиченість, колірну палітру й інші параметри.

Крім способу виведення, плотери відрізняються від принтерів ще й тим, що *дані виводяться на носії великого формату: ISO A2 (42×59,4 см), A1 (59,4×84 см) і A0 (84×111,8 см)*. У США використовують набір форматів ANSI, у якому формат C (43,2×55,9 см) відповідає формату A2, формат D (55,9×86,3 см) – формату A1, а формат E (86,3×111,8 см) – формату A0. Деякі моделі плотерів можуть виводити дані на носії ще більших форматів, ніж A0.

2.9.3. Проектори

Розміри CRT і LCD-моніторів не дозволяють показувати зображення на екрані великій кількості людей, наприклад, під час проведення лекцій чи презентацій. Вирішити цю проблему можна застосуванням проекторів.

Існує два типи проекторів зображень: дзеркальні та діапроектори.

У *дзеркальних проекторах – overhead projectors* (проекторах, розміщених над головою) світло, що проходить крізь прозорий оригінал чи відбите від непрозорого оригіналу, проектується дзеркалом на великий екран. Для виведення зображень з комп'ютера замість прозорого оригіналу використовують панель LCD, на яку подаються сигнали з виходу відеокарти (рис. 2.20).

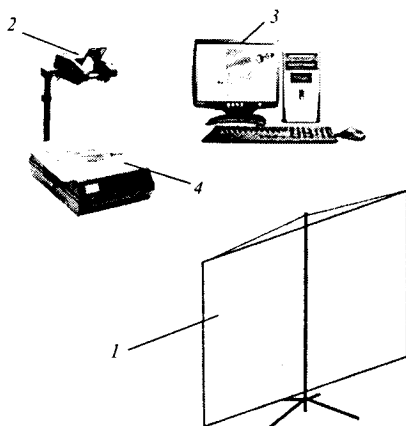


Рис. 2.20. Використання панелі LCD із дзеркальним проектором для демонстрації виведеного комп'ютером зображення:

1 – екран; 2 – проектор; 3 – монітор; 4 – екран

Панелі LCD дедалі більше витісняються мультимедійними проекторами, що дуже подібні до звичайних діапроекторів для перегляду слайдів, тому їх часто називають діапроекторами. Існує кілька типів таких проекторів:

- проектори LCD;
- полісиліконові проектори;
- проектори DLP.

Основний елемент *проектора LCD* – мініатюрний рідкокристалічний екран, виконаний на основі технології TFT. Світло від проекційної лампи великої потужності потрапляє на лінзу-конденсор, що перетворює світловий потік з розбіжного в паралельний і після проходження крізь об'єктив зображення проектується на екран.

Незважаючи на використання ламп великої потужності, яскравість зображення в проекторах LCD невелика, оскільки світло істотно послаблюється під час проходження через панель TFT. Якщо збільшувати потужність лампи, то виникає загроза перегрівання панелі. Щоб запобігти цьому недоліку проекторів LCD, були розроблені полісиліконові проектори, що дають більш яскраве зображення. У цих проекторах світло проходить не крізь одну, а через три монохромні (і, відповідно, більш тонкі) панелі TFT. До кожної з цих панелей від зовнішнього джерела підводиться свій колірний компонент зображення (червоний, зелений чи синій).

2.10. Допоміжні пристрої комп'ютера

До допоміжних пристроїв комп'ютера належать:

- корпус системного блоку;
- блок електроживлення;

— система охолодження;

— пристрої захисту від порушень роботи електроживлення.

Корпус системного блоку (System Block Case) (рис. 2.21) призначено для розміщення материнської плати, блоку електроживлення, дискководів зовнішньої пам'яті всіх типів і деяких інших пристроїв. Корпус системного блоку може мати найрізноманітніші форми і розміри, однак його здебільшого виконують у двох модифікаціях: горизонтальній – настільній (desktop) і більш поширеній вертикальній – «вежа» (tower).

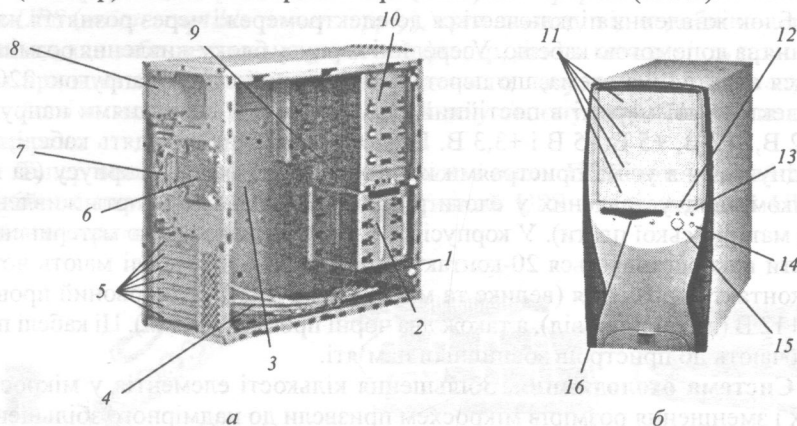


Рис. 2.21. Компоненти корпусу: а – вигляд збоку, б – вигляд спереду:

1 – динамік; 2 – відсік для пристроїв з форм-фактором 3,5"; 3 – місце кріплення материнської плати; 4 – з'єднувальні проводи; 5 – отвори для роз'яттів карт у слотах розширення материнської плати; 6 – отвір для вмикання вентилятора; 7 – отвори для роз'яттів інтерфейсів материнської плати; 8 – отвір для вмикання електроживлення; 9 – відсік для дискОВОДА гнучких дисків; 10 – відсік для пристроїв з форм-фактором 5,25"; 11 – місце для монтажу пристроїв у відсіку 5,25"; 12 – індикатор увімкнення комп'ютера; 13 – індикатор звернення для дискОВОДА жорстких дисків; 14 – кнопка Reset; 15 – кнопка Power; 16 – отвір для вставляння дискет

Корпус містить гнізда з пазами для кріплення материнської плати, джерела живлення, внутрішніх пристроїв зовнішньої пам'яті (дискководів магнітних, магнітооптичних і оптичних дисків, а також нагромаджувачів на магнітній стрічці). Для кріплення пристроїв зовнішньої пам'яті зазвичай передбачають два відсіки: 5,25-дюймовий і 3,5-дюймовий. На задній стінці корпусу передбачено отвори для підключення блоку живлення і вентилятора, а також планку з отворами для роз'яттів інтерфейсів материнської плати й отвори для роз'яттів карт, що вставляються в слот розширення материнської плати (у процесі складання комп'ютера захисні планки, що закривають ці отвори, вилучають).

На передній панелі корпусу виведено кнопку вмикання-вимикання комп'ютера (*Power*), кнопку перевантаження (*Reset*), індикатор вмикання дисководу та індикатор звернення до дисководу жорстких дисків.

Блок живлення (*power supply*) забезпечує електричною енергією усі компоненти комп'ютера, розміщені в системному блоці: материнську плату, нагромаджувачі, плати розширення, клавіатуру й інші пристрої, що підключаються до портів чи плат і не мають автономного живлення.

Блок живлення підключається до електромережі через роз'язтя живлення за допомогою кабелю. Усередині корпусу блоку живлення розміщується електронна схема, що перетворює змінний струм напругою 220 В з електричної мережі в постійний струм з такими значеннями напруги: +12 В, -12 В, +5 В, -5 В і +3,3 В. Із блоку живлення виходять кабелі, що з'єднуються з усіма пристроями комп'ютера системного корпусу (за винятком карт, уставлених у слоти розширення, що одержують живлення від материнської плати). У корпусі *ATX* для підключення до материнської плати використовується 20-контактне роз'язтя. Інші кабелі мають чотиріконтактне роз'язтя (велике та мале) і напруги +5 В (червоний провід) та +12 В (жовтий провід), а також два чорні проводи (земля). Ці кабелі підключають до пристроїв зовнішньої пам'яті.

Система охолодження. Збільшення кількості елементів у мікросхемах і зменшення розмірів мікросхем призвели до надмірного збільшення загального виділення тепла в системному блоці. Для зниження температури окремих компонентів усередині всього системного блоку застосовують здебільшого *пасивне охолодження* за допомогою радіаторів і теплових трубок та *активне охолодження* вентиляторами.

Основні пристрої в системному корпусі, які потрібно охолоджувати, – це центральний процесор, модулі оперативної пам'яті, північний міст, відеокарта і дисководи жорстких дисків.

Процесор охолоджується системою, що складається з радіатора і вентилятора. Ця система розміщена над процесором і кріпиться до нього за допомогою спеціального термостійкого клею. У материнських платах *ATX* вентилятор підключається до спеціального роз'язтя плати.

Модулі оперативної пам'яті містяться у спеціальних тепловідвідних корпусах, що виконують роль радіаторів.

Північний міст охолоджує радіатор, розміщений над мікросхемою.

Відеокарти зазвичай містять один чи два вентилятори.

Жорсткі диски охолоджуються або нагрівальними трубками, або за допомогою одного чи двох вентиляторів, розміщених у відсіку зовнішньої пам'яті корпусу.

Поряд з повітряними системами охолодження починають використовувати і *системи водяного охолодження*.

Пристрої захисту від порушень роботи електроживлення. Основні типи пристроїв захисту – мережеві фільтри і неперервні джерела живлення (рис. 2.22).

Мережеві фільтри. *Мережеві фільтри*, названі також пристроями захисту від стрибків (*surge protectors*) чи придушення стрибків (*surge suppressors*), призначені передусім для захисту електронного устаткування від стрибків напруги та видалення шумових завад з мережі живлення.

Неперервні джерела живлення (UPS). *Неперервні джерела живлення UPS (uninterruptible power supply)*, крім захисту від стрибків напруги і завад, що забезпечують мережевий фільтр, дають можливість пристроям працювати нормально протягом деякого часу в разі спаду напруги чи його вимкнення. Для цього UPS має у своєму складі акумуляторну батарею.

Тепер використовують чотири *основні типи UPS*: автономний UPS, лінійно-інтерактивний UPS, інтерактивний UPS, інтерактивний UPS з дельта-перетворенням.

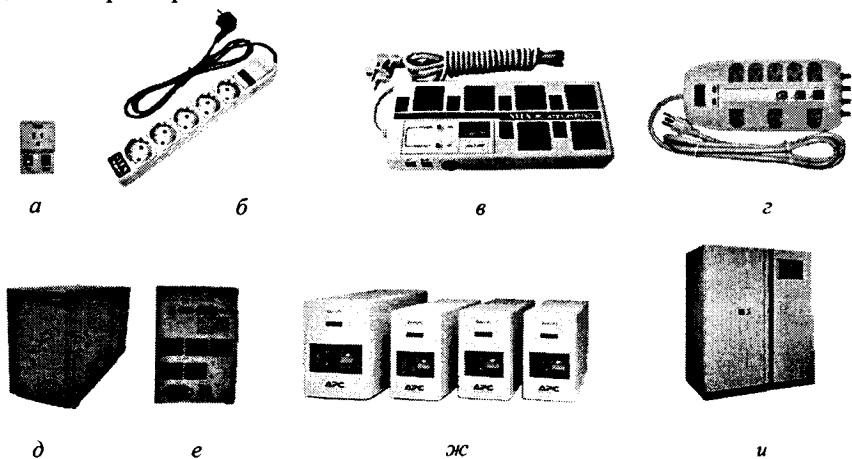


Рис. 2.22. Пристрої захисту від порушень електроживлення: а – мережевий фільтр для одного пристрою; б – мережевий фільтр із загальним вимикачем; в – мережевий фільтр з індивідуальними вимикачами і роз'язтями для телефонної лінії; г – мережевий фільтр із роз'язтями для телефонної лінії і коаксіальних кабелів; д – UPS; е – UPS – вигляд ззаду; ж – сім'я UPS; и – UPS для сервера

Тип пристрою UPS визначає його ціну, діапазон потужностей і сферу використання.

Автономні UPS найдешевші, однак мають істотні недоліки, основні з яких – мала потужність зарядного блоку, погана стабілізація вихідного сигналу і ненульовий час переходу.

Більш швидкий перехід до аварійного стану і крашу стабілізацію вихідного сигналу забезпечують *лінійно-інтерактивні UPS*. Це дозволяє використовувати ці UPS в електромережах низької якості.

Інтерактивні UPS дорожчі і їх застосовують тоді, коли устаткування потребує якісного електроживлення. Недоліком цього типу UPS є виділення тепла. Цей тип UPS здебільшого використовують для захисту серверів.

Інтерактивний UPS з дельта-перетворенням призначено для запобігання надлишковому виділенню тепла.

Час переходу в режим автономної роботи для автономних UPS дорівнює 10...20 мс, для лінійно-інтерактивних UPS – 2...10 мс. Інтерактивні UPS і UPS з дельта-перетворенням мають нульовий час переходу.

Час зарядження батарей для UPS становить зазвичай 2...12 год.

Усі сучасні моделі UPS взаємодіють з комп'ютером. У нормальному режимі програми забезпечують моніторинг UPS – слідкування за його параметрами і настроювання параметрів, а в аварійному – розсилання повідомлень користувачам і закриття (*shutdown*) комп'ютера через певний час.

Заяпитання до самоперевірки

1. Які ви знаєте найважливіші характеристики процесорів?
2. Як функціонує внутрішня пам'ять?
3. Які фізичні принципи використовуються для записування і зчитування даних з магнітних пристроїв?
4. Наведіть основні характеристики дисководів на жорстких дисках.
5. Які технології і режими використовують для записування даних на оптичні диски? Як виконується зчитування даних з оптичного диска?
6. Які основні компоненти містять пристрої Flash-пам'яті?
7. Опишіть принцип дії CRT-монітора.
8. Опишіть принцип дії LCD-монітора.
9. Опишіть принцип дії плазмового монітора.
10. Які основні функції виконує відеокарта комп'ютера?
11. Які основні технології фіксації і натискання клавіш використовуються в клавіатурі? Наведіть стислий опис кожної технології.
12. Які існують типи «мишок»? Які основні характеристики «мишки»?
13. Які основні технології використовуються в сканерах?
14. Принцип дії та основні характеристики цифрової фотокамери.
15. Як відбувається процес друкування в лазерному принтері?
16. Які типи плотерів використовуються сьогодні?
17. Назвіть основні типи діапроекторів та принципи їх функціонування.
18. Які основні типи й характеристики блоку живлення комп'ютера?
19. Які основні типи пристроїв охолодження використовуються тепер?
20. Які типи пристроїв захисту існують для боротьби із порушеннями системи електроживлення комп'ютера?

3. ОПЕРАЦІЙНА СИСТЕМА WINDOWS XP

📖 План викладу матеріалу

1. Загальні положення.
2. Робочий стіл та його елементи.
3. Вікна Windows.
4. Головне меню Windows.
5. Файли та папки Windows XP, основні прийоми роботи.
6. Файлові системи Windows XP.
7. Параметри налагодження об'єктів середовища операційної системи.
8. Стандартне програмне забезпечення.

↔ Ключові терміни розділу

✓ <i>Операційна система</i>	✓ <i>Файл</i>
✓ <i>Піктограми</i>	✓ <i>Папка</i>
✓ <i>Контекстне меню</i>	✓ <i>Ярлик</i>
✓ <i>Drag & Drop</i>	✓ <i>Пошук файлів</i>
✓ <i>Панель задач</i>	✓ <i>Навігація по папках</i>
✓ <i>Вікна</i>	✓ <i>Буфер обміну</i>
✓ <i>Файлова система</i>	✓ <i>Шрифт</i>

3.1. Загальні положення

Операційна система (далі ОС) – це комплекс програм, який забезпечує управління апаратними ресурсами комп'ютера, організовує роботу з файлами, а також реалізує інтерфейс взаємодії користувача з персональним комп'ютером.

Windows XP – достатньо нова ОС, де, окрім вже звичного для ОС серії **Windows** дружнього інтерфейсу, значної уваги до мультимедіа-додадків, забезпечення роботи в мережі **Internet**, велику увагу приділено безпеці даних та забезпеченню відмовостійкості ОС. З огляду на прагнення розробника охопити якомога більший ринок, **Windows XP** була випущена у трьох різних версіях:

- **Windows XP Professional**, яка призначена для корпоративного використання. Насправді ця версія ОС може успішно працювати на домашньому комп'ютері (навіть на невід'язаному до мережі), тільки при цьому певні можливості ОС (зокрема сервіси, що забезпечують роботи з мережевими модемами) не будуть задіяні;

- **Windows XP Home Edition**, яка придатна для користування на домашніх ПК. В цій версії наголос зроблено перш за все на ті сервіси, які буде використовувати рядовий користувач для розваги та організації вільного часу. Звідси й розширені можливості по роботі з мультимедіа та **Internet**;

- **Windows XP 64-Bit Edition**, яка призначена для роботи на комп'ютерах, які підтримують 64-бітну обробку. Вони застосовуються там, де необхідна оперативна пам'ять великих розмірів (роботи з тривимірною анімаційною графікою, звуком тощо).

Windows XP має досить багато нових можливостей, які відрізняють її від **Windows 2000**, яка побудована на ядрі більш старої ОС **Windows NT**, а від більш старих попередників (**Windows 95/98/Me**) **Windows XP** відрізняється докорінно, як новими можливостями, так і ядром ОС. Нижче наведені основні нові можливості ОС **Windows XP**:

- забезпечення безпеки даних. Присутній сервіс **Центр забезпечення безпеки**, за допомогою якого можна перевірити стан антивірусного програмного забезпечення та брандмауера **Windows**;

- брандмауер **Windows** дозволяє захистити комп'ютер від різноманітних загроз безпеці – з боку вірусів або несанкціонованого доступу до персонального комп'ютера;

- дистанційне керування робочим столом;

- наявність сервісу **Защита файлов Windows**, який допоможе уникнути заміни або знищення системних файлів;

- відновлення ОС після збою, що досягається завдяки наявності програми **Восстановление системы**, яка слідкує за змінами в ОС та періодично створює точки відновлення, за допомогою яких можна після збою повернути систему до попереднього стану;

- в дистрибутив ОС включена велика кількість драйверів пристроїв, що забезпечує роботу різноманітного периферійного обладнання без додаткового встановлення потрібного програмного забезпечення;

- включення в дистрибутив мультимедійної бібліотеки **DirectX 9**;

- швидке перемикання користувачів, а також можливість продовження роботи програм, запущених в сеансі одного користувача, після переходу до сеансу іншого користувача;

- використання технології **Dualview**, яка забезпечує можливість підключення до пересувного комп'ютера другого монітора для того, щоб на різних екранах переглядати різні програми;

- використання технології **ClearType**. Для відображення екранних шрифтів дозволяє домогтися чіткого відображення тексту на екрані монітора;

- сортування файлів у вікні папок, а також перегляд графічних файлів у вигляді ескізів та діафілму;

- перенесення за допомогою **Мастера переноса файлів і параметрів** параметрів настройки ОС на інший ПК без виконання копітких дій по їх налаштуванню на цьому комп'ютері;
- збереження вмісту оперативної пам'яті комп'ютера в режимі **Hibernate Mode** після зупинки його роботи. Це досягається тим, що вміст оперативної пам'яті зберігається на жорсткому диску й, відповідно, після увімкнення комп'ютера усі запущені на час вимкнення додатки знову будуть відкритими;
- відображення вмісту **Панелі управління** за категоріями.

Окрім наведених нововведень, потрібно згадати також, що не дивлячись на достатньо великий спектр нових можливостей ОС, філософія роботи в ній залишилася та ж сама, а принципи організації інтерфейсу ідентичні. Про елементи інтерфейсу ми далі і поговоримо.

3.2. Робочий стіл та його елементи


Після завантаження **Windows XP** і введення імені користувача та паролю, як і в минулих версіях, перед користувачем з'являється **Робочий стіл** (рис. 3.1).





Рис. 3.1

На робочому столі, як видно з рисунка, наявні наступні елементи інтерфейсу.

Піктограми – графічні зображення стандартного розміру 32×32 пікселі (може бути й 64×64 або 16×16 та інші), де **піксель** – це елементарний елемент прямокутної форми растрового зображення. Піктограми є візуальним представленням файла або папки, а за їх виглядом часто можна ідентифікувати тип файла. Серед стандартних піктограм, наявних на робочому столі, відзначимо наступні:

- **Мой комп'ютер**  – централізований засіб доступу як до інформації, що зберігається на комп'ютері, до програмних файлів, так і до налаштувань ОС. Саме з нього користувач часто починає навігацію по дисках та папках. Для того, щоб відкрити засіб **Мой комп'ютер**, необхідно двічі клацнути лівою кнопкою мишки на піктограмі **Мой комп'ютер**, після чого відкриється відповідне вікно;

- **Корзина**  – виходячи з назви, це місце (папка), де зберігаються вилучені файли або папки до їх остаточного видалення;

- **Мои документи**  – папка, в якій користувачу пропонується зберігати свої основні документи. Оскільки власне право кожного зберігати свої файли де йому зручно, в ОС передбачена можливість змінити знаходження папки **Мои документи**. Для цього необхідно клацнути правою кнопкою мишки на піктограмі **Мои документи**. У контекстному меню, яке з'явиться, обрати, один раз клацнувши лівою кнопкою мишки, команду **Свойства** (*контекстне меню* – це меню, що з'являється після того, як клацнути правою кнопкою мишки на певному об'єкті – вміст меню залежить від типу об'єкта). У вікні **Свойства: Мои документи** (рис. 3.2) необхідно на вкладці **Папка назначения** в полі **Папка** вказати шлях до нової папки **Мои документи** ввівши його з клавіатури, або скориставшись кнопкою **Переместить....** По закінченні всіх цих дій натисніть кнопку **ОК** (яка в усіх програмах поряд з кнопкою **Применить** означає затвердження внесених змін) для збереження внесених змін або кнопку **Отмена** для їх скасування.

Панель задач, на якій розташовуються тимчасові кнопки вікон, що відкриті в ОС. Натисканням на таку кнопку можна або згорнути вікно (якщо воно розгорнуте), або розгорнути (якщо вікно згорнуте).

System Tray – це область в крайній правій частині **Панелі задач** (тобто в нижньому правому куті екрана), в якій деякі програми, які на даний момент запущені в ОС, розміщують свої піктограми. Зокрема, у **System Tray** часто розміщуються піктограми запущеного антивірусного програмного забезпечення, а також тут можна побачити поточний (встановлений

в ОС) час. У Windows XP з'явилася можливість за бажанням ховати вміст System Tray, для чого на межі між Языковой панелью та System Tray була додана кнопка, що або приховує значки System Tray, або, навпаки, їх показує, якщо вони вже приховані.

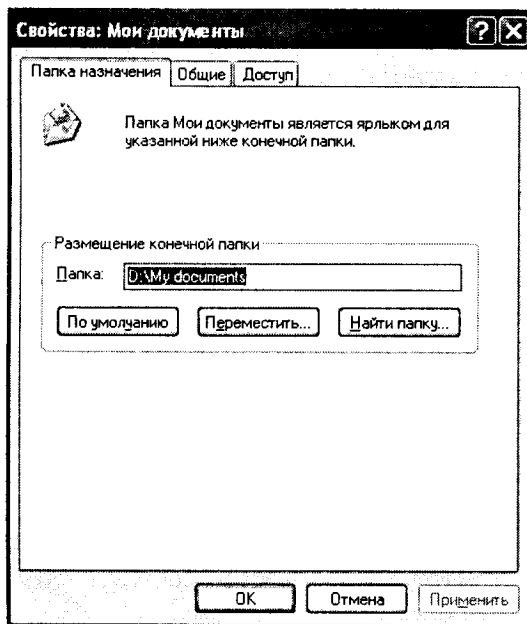


Рис. 3.2

Языковая панель – відображає поточну розкладку клавіатури в ОС. За бажанням її можна розгорнути, після чого вона опиниться нагорі екрану.

Панель быстрого запуска – виходячи з назви, призначена для швидкого доступу до певних файлів або папок, ярлики яких можна додати до **Панели быстрого запуска** простим перетягуванням (*Drag & Drop* – натискання лівої кнопки мишки на об'єкті та переміщення об'єкта в потрібне місце при натиснутій кнопці мишки). Для відображення **Панели быстрого запуска** необхідно виконати наступну послідовність дій:

1. Клацнути правою кнопкою мишки у вільному місці **Панели задач** та у контекстному меню, що з'явиться на екрані, обрати команду **Свойства**;
2. У вікні, що з'явиться на екрані (**Свойства Панели задач и меню «Пуск»**), на вкладці **Панель задач** необхідно встановити прапорець **Отображать панель быстрого запуска**;
3. Натиснути кнопку **ОК**.

Кнопка **Пуск** використовується для входу у **Главное меню**.

3.3. Вікна Windows

Легкість засвоєння нових програмних продуктів, працюючих під Windows, забезпечується, окрім відповідно організованого інтерфейсу в самих програмних продуктах, також схожістю будови вікон різних програм, наявністю стандартних елементів та однаковими принципами організації наповнення меню додатків.

3.3.1. Типи вікон Windows XP

Можна назвати наступні типи вікон Windows.

Вікна програм. На відміну від інших, вікна даного типу мають рядок меню та можуть містити та підпорядковувати собі вторинні вікна.

Вторинні вікна. Створюються програмами під час їх роботи. Прикладом вторинних вікон будуть вікна книг, які створюються при роботі з редактором електронних таблиць Microsoft Excel (рис. 3.3)

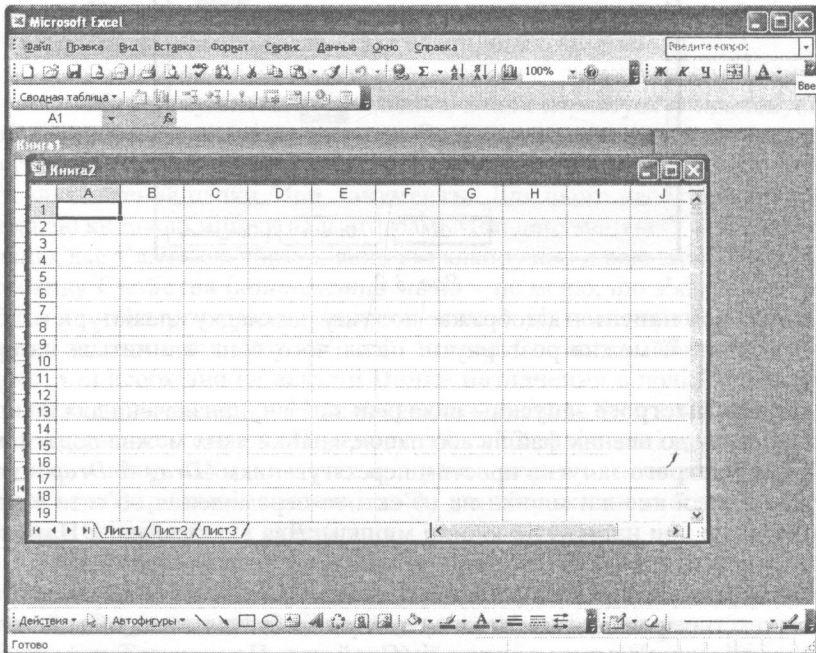


Рис. 3.3





Вікна запитів (допоміжні вікна). Виводяться завжди поверх інших вікон. Часто подальша робота з програмою неможлива доти, доки користувач не закриє вікно запити.

Вікна запитів відрізняються за характером повідомлення (важливістю повідомлення), який можна часто визначити за змістом графічного зображення, що розташовується в області вікна:

- ✘ – таку позначку має вікно з повідомленням про критичну помилку в системі або в програмі.
- ⚠ – цю позначку має вікно з важливим інформаційним повідомленням (попередженням). Часто вікно з такою позначкою повідомляє про неправильні дії користувача, які не можуть призвести до критичних наслідків.
- ℹ – таку позначку має вікно з простим інформативним повідомленням.
- ❓ – таку позначку має вікно, яке ставить користувачу запитання.

3.3.2. Структура вікна папки Windows XP

Стандартне вікно папки **Windows XP** показано на рис. 3.4, проте схожі елементи є також в вікнах програм інших розробників. Отже, загалом вікно складається з таких елементів.

Рядок заголовку. Відображає назву вікна, яка у випадку папки є ім'ям папки, а також у правій частині містить кнопки згортання , розгортання на весь екран  (або навпаки – згортання до первісних розмірів, якщо вікно розгорнуте на весь екран ) та закриття  вікна.

Рядок заголовку Рядок меню Панель Обычные кнопки Область відображення вмісту папки

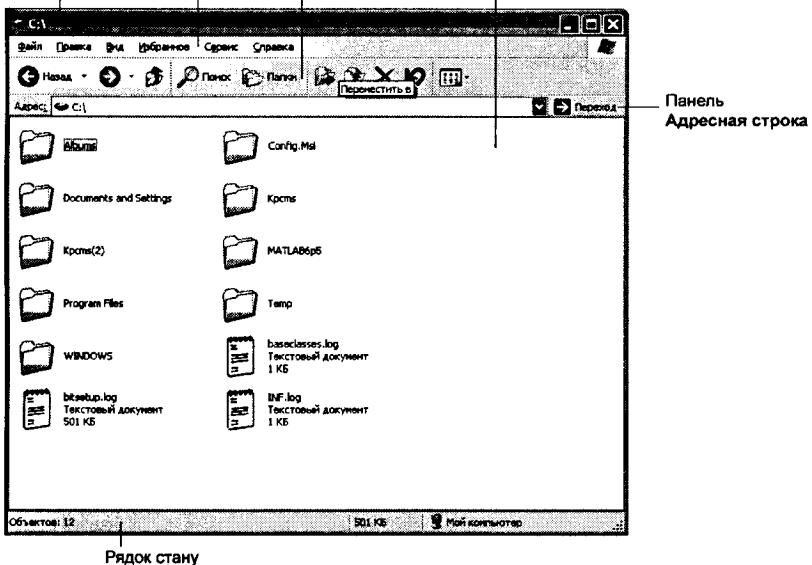


Рис. 3.4

Рядок меню. Містить кнопки меню, натискаючи на які, користувач заходить у відповідне меню. Як самі меню, так і команди в них обираються натисканням лівої кнопки мишки. Якщо праворуч від назви команди меню знаходиться чорна стрілка, то на таку команду меню достатньо просто навести курсор мишки, після чого відкриється підменю. Якщо назва команди закінчується трьома крапками (наприклад, **Свойства папки...**), то це означає, що після вибору такої команди на екрані обов'язково з'явиться ще вікно. У випадку, коли ім'я команди меню не містить трьох крапок, команда безпосередньо виконує певні дії.

Приклад меню з командами різного типу наведений на рис. 3.5,а. Ми бачимо, що після вибору команди **Панели инструментов**, поряд з назвою якої наявна чорна стрілка, з'являється підменю (рис. 3.5,б). Прапорці, які стоять ліворуч від назв пунктів меню, вказують на те, що даний пункт є просто опцією, встановивши біля якої прапорець, ми її активуємо. На рис. 3.5,а чорна точка біля пункту меню вказує на те, що серед групи команд меню, які знаходяться між двома роздільними лініями, одночасно може бути відмічена тільки одна команда.

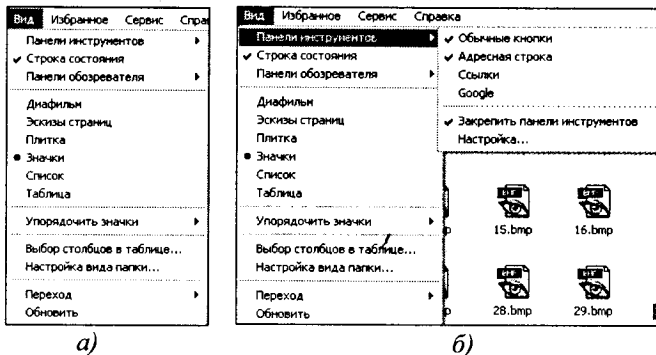


Рис. 3.5

Панелі інструментів. Це група елементів, на кожному з яких знаходяться кнопки. Наприклад, у вікні папки ми за замовчуванням можемо бачити панелі інструментів **Обычные кнопки** та **Адресная строка**. Всі кнопки, як і команди меню, обираються одиничним клацанням лівою кнопкою мишки. За бажанням панелі інструментів у вікні папки можна прибрати або, навпаки, показати додаткові. Для цього необхідно виконати наступні дії:

- зайти у меню **Вид**;
- обрати команду **Панели инструментов**;
- в підменю встановити прапорець біля панелі інструментів, яку необхідно показати, або зняти прапорець, якщо панель інструментів необхідно забрати (встановлення та зняття прапорця також проводимо, клацнувши один раз лівою кнопкою мишки на необхідній команді меню).

Область відображення вмісту папки. В цій області знаходяться піктограми всіх файлів та папок, які присутні в поточній папці (*поточна папка* – це папка, в якій на даний момент працює користувач).

В області відображення вмісту папки, якщо вміст не вміщується у вікно з поточними розмірами, з'являються смуги прокрутки – горизонтальна або вертикальна чи обидві відразу, залежно від розмірів вікна та режиму відображення вмісту папки. Користуватись ними просто – достатньо навести курсор мишки на повзунок смуги прокрутки, натиснути ліву кнопку мишки та, утримуючи її, перетягнути його у потрібне положення.

Рядок стану. В рядку стану відображається інформація про кількість виділених об'єктів (якщо вони є), сумарний об'єм виділених файлів або сумарний об'єм всіх файлів у відкритій папці (якщо жодного з об'єктів не виділено). На рис. 3.6,а показаний рядок стану при відсутності виділених об'єктів: ліворуч показана загальна кількість об'єктів, посередині – загальний розмір всіх файлів. На рис. 3.6,б показаний рядок стану при виділеному файлі: ліворуч показані параметри виділеного файлу, а замість сумарного розміру всіх файлів у поточній папці – розмір виділеного файлу.

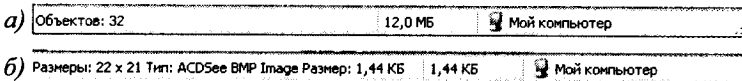


Рис. 3.6

3.3.3. Операції з вікнами

В **Windows XP** можна робити всі ті операції з вікнами, що й у попередніх версіях. Ми вже зазначили, як можна закрити вікно, згорнути його або розгорнути, коли розглядали рядок заголовку в попередньому розділі. Тепер необхідно розглянути інші основні операції, подані нижче.

Переміщення вікна.

Виконується методом **Drag & Drop**. Для переміщення вікна необхідно встановити курсор мишки на рядок заголовка вікна, натиснути ліву кнопку мишки та, утримуючи її в такому стані, перетягнути мишкою вікно у потрібне місце екрану, після чого відпустити кнопку мишки.

Зміна геометричних розмірів вікна.

Для зміни розмірів вікна необхідно навести на межу вікна курсор мишки. Коли курсор мишки набуде вигляд подвійної стрілки, необхідно натиснути ліву кнопку мишки та, утримуючи її натиснутою, встановити потрібні розміри вікна.

Впорядкування вікон на робочому столі.

Іноді виникає потреба розташувати вікна за певною системою. Звісно, робити це простим переміщенням нераціонально, тому у **Windows XP** передбачена можливість впорядкування (сортування) вікон. Для цього необхідно клацнути правою кнопкою мишки у вільному місці **Панелі задач**

та у контекстному меню, яке з'явиться на екрані, обрати одну з команд, що відповідають режимам впорядкування (Окна каскадом (рис. 3.7), Окна сверху-вниз або Окна слева-направо (рис. 3.8)).

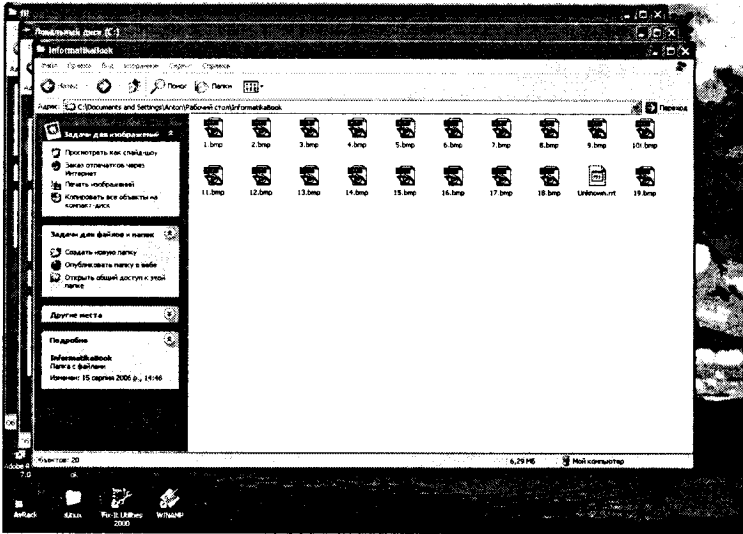


Рис. 3.7

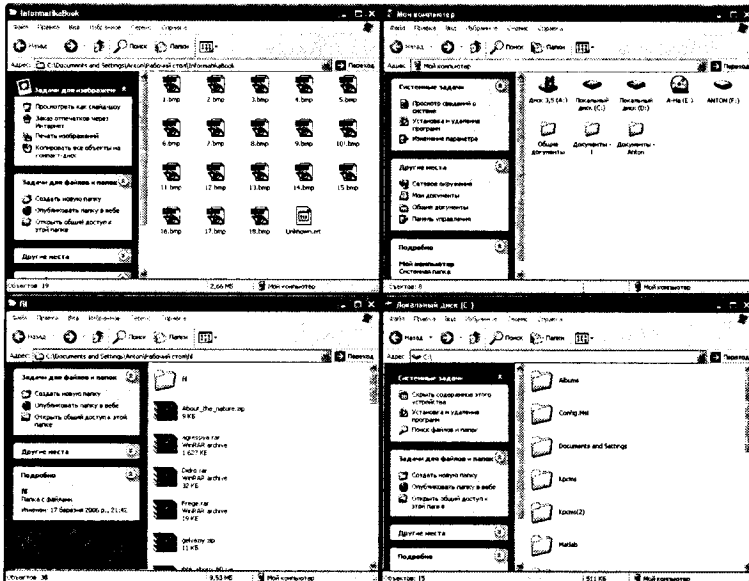


Рис. 3.8

Згортання всіх відкритих вікон

Часто для швидкого доступу до вмісту **Рабочего стола** необхідно відразу згорнути усі вікна. Для цього необхідно клацнути лівою кнопкою мишки у вільному місці **Панели задач** та в контекстному меню обрати команду **Показать рабочий стол**. Інакше це можна зробити, натиснувши на кнопку **Свернуть все окна** на **Панели быстрого запуска**

Зворотну операцію, тобто розгортання всіх вікон відразу, можна зробити викликом контекстного меню **Панели задач** та обравши команду **Отменить Свернуть все**.

3.4. Головне меню Windows

Главное меню або меню **Пуск** надає швидкий доступ до найбільш часто вживаних програм або сервісів. У **Windows XP** користувач має змогу зробити вибір між двома стилями меню **Пуск** – класичним (рис. 3.9,а) та властивим **Windows XP** (рис. 3.9,б).

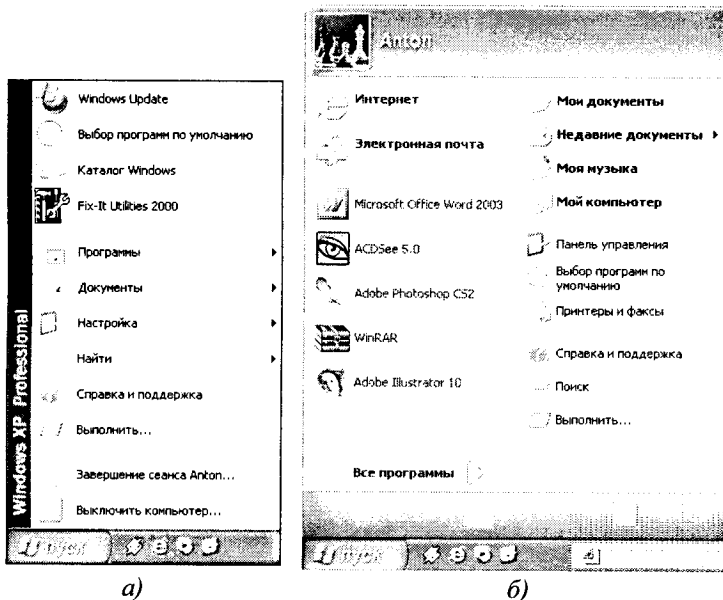


Рис. 3.9

Перемикатися між наведеними двома режимами відображення меню **Пуск** можна з вкладки **Меню «Пуск»** вікна **Свойства Панели задач и меню «Пуск»**, яке можна викликати клацнувши правою кнопкою мишки у вільній області **Панели задач** та обравши у контекстному меню команду **Свойства**.

В обох режимах відображення меню **Пуск** воно має дві області: список, що знаходиться вище роздільної лінії та має назву **Список закріплених елементів**, а нижче знаходиться **Список наиболее часто используемых программ**. При класичному вигляді (рис. 3.9,а) меню **Пуск** до **Списка закріплених елементів** не можна додати елемент або, навпаки, видалити його. При звичайному вигляді меню (рис. 3.9,б) будь-яку програму можна додати до цього списку, якщо клацнути на піктограмі програми правою кнопкою мишки та у контекстному меню обрати команду **Закрепить в меню Пуск**. Для того, щоб видалити програму з фіксованого списку, необхідно клацнути правою кнопкою мишки на піктограмі програми у цьому списку та у контекстному меню обрати команду **Изъять из меню Пуск**.

До **Списка наиболее часто используемых программ** програми додаються в міру їх встановлення. Додавати та вилучати програми з цього списку можна й за бажанням користувача (це стосується обох режимів відображення меню **Пуск**) таким же чином, як можна копіювати та видаляти файли та каталоги (див. нижче). Наведемо основні елементи меню **Пуск** в класичному вигляді та їх призначення.

Программы – містить групи програм, встановлені користувачем на комп'ютері. Надає швидкий доступ до встановлених на комп'ютері програм.

Документы – містить 15 останніх файлів (текстових документів, графічних файлів тощо), які були переглянуті користувачем під час роботи з ОС. В цьому розділі зберігаються не самі документи, а тільки ярлики до них. Оскільки **ярлик** – це файл, що містить посилання на інший об'єкт файлової системи (папка, файл, диск), то видалення ярлика не видаляє сам об'єкт, на який він посилається. Тому видалення ярликів останніх документів з розділу **Документы** не видаляє самі файли, на які ці ярлики посилаються.

Настройка – надає швидкий доступ до основних налаштувань ОС.

Найти – надає швидкий доступ до засобів пошуку об'єктів як на локальних дисках, так і в мережі.

Справка и поддержка – запускає довідкову систему **Windows XP**.

Выполнить... – запускає діалогове вікно для введення ім'я програми, яку необхідно швидко запустити, або шляху до папки, яку необхідно відкрити, або для введення адреси мережевого комп'ютера для того, щоб отримати доступ до файлів, які на ньому зберігаються.

Завершение сеанса (Им'я користувача)... – припиняє сеанс роботи поточного користувача та дає змогу заново ввести ім'я та пароль для входу під ім'ям іншого користувача.

Выключить компьютер... – відкриває вікно **Выключить компьютер** (рис. 3.10), у якому можна обрати одну з наступних подальших дій:

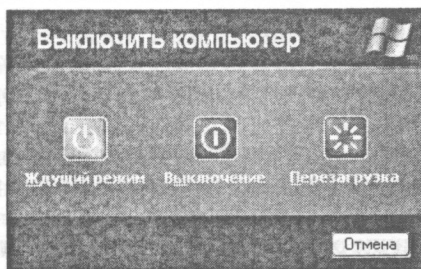


Рис. 3.10

- **Ждущий режим:** стан, в якому комп'ютер споживає менше енергії, але залишається готовим для негайної роботи. Дані при цьому не записуються на жорсткий диск та при вимкненні живлення будуть втрачені;
- **Выключение:** вимикає комп'ютер;
- **Перезагрузка:** перезавантажує комп'ютер. Перезавантаження часто необхідне під час встановлення програмного забезпечення, хоча у випадку з **Windows XP** його доводиться робити не так часто, як, наприклад, при встановленні програм під **Windows 98**;
- **Отмена:** скасовує виклик вікна.

3.5. Файли та папки Windows XP, основні прийоми роботи

Файли, папки – це об'єкти файлової системи і саме з ними працює рядовий користувач під час своєї роботи з ОС, тому знання операцій з файлами та папками, особливостей файлових систем є необхідним для роботи з ПК. Але спочатку потрібно розібратися, що є файловою системою.

3.5.1. Файлові системи Windows XP

Файлова система – це спосіб організації файлів і папок на диску. **Windows XP** підтримує три файлові системи для жорстких дисків:

1. **FAT (File Allocation Table) або FAT16.** Досить стара файлова система, яка використовувалась ще у всіх версіях **MS-DOS** та перших версіях **Windows 95**. Для зберігання файлу у **FAT** може використовуватися один або декілька кластерів (кластер – мінімальний обсяг дискового простору, в якому може бути розміщений файл). Кожному кластеру диска в таблиці **FAT** відповідає окремий запис, який або вказує на наступний кластер файлу, або містить мітку кінця файлу. У складі кожного каталогу зберігаються імена файлів, які входять до даного каталогу, а також вказівник на перший кластер цього файлу. Окрім цього, в каталозі зберігається інформація про дату створення файлу, його розмір та атрибути.

Недоліками FAT є:

- максимальний розмір дискового тому – 2 Гб, що фактично робить FAT несучасною з огляду на поширені розміри жорстких дисків (від 60 Гбайт);
- оскільки в FAT для зберігання файлових атрибутів використовується тільки 1 байт, то в цій файловій системі не можна зберігати дані про права доступу до файлу та його власника;
- відомості про фізичне розміщення файлів зберігаються в одному місці – таблиці розміщення FAT, що призводить до зменшення надійності зберігання інформації та швидкості доступу до файлу.

2. Починаючи з пізніх модифікацій **Windows 95** з'явилася нова, вдосконалена файлова система **FAT32**, основна відмінність якої від **FAT16** полягає в тому, що записи, які відповідають дисковим кластерам, таблиця розміщення файлів ідентифікує за допомогою 32-розрядних чисел (на відміну від **FAT16**, де для цього використовувались 16-ти розрядні числа), що дозволяє збільшити максимально можливий розмір дискового тому до 2 Тбайт. Зрештою, ця файлова система не відрізняється від **FAT16** та має малу швидкість роботи з великими за обсягом каталогами. Загалом, файлові системи FAT мають такі переваги та недоліки.

Переваги FAT:

- не вимагає великого обсягу оперативної пам'яті для роботи;
- забезпечує швидкий доступ до інформації, що знаходиться на невеликих за обсягом вінчестерах (жорстких дисках);
- швидко працює з малими та середніми за обсягом каталогами.

Недоліки FAT:

- втрата швидкодії при великому ступені фрагментованості дискового простору;
- повільна робота з великими за обсягом каталогами та проблеми з доступом до великих файлів (≈10% від обсягу жорсткого диска);
- неможливість зберігання даних про права доступу та ім'я власника, а, відповідно, низьке забезпечення безпеки зберігання даних.

Підсумовуючи вищесказане, зробимо висновок, що системи **FAT16** та **FAT32** вже застарілі та використовуються нині переважно або на комп'ютерах старих поколінь, або якщо користувачу за якоїсь причини потрібно працювати у **Windows 95/98**.

3. Файлова система **NTFS (New Technology File System)**. Також достатньо стара файлова система, яка була розроблена ще для **Windows NT**, проте зараз є основною файловою системою для сімейства ОС **Windows NT/2000**, а також **Windows XP**. Розмір одного дискового тому у файловій системі **NTFS** теоретично може бути будь-яким, а якщо точніше, то до 16 Екзабайт (1 Екзабайт дорівнює 1 073 741 824 Гігабайт). Це означає, що при швидкості запису на диск в 1 Мбайт за секунду 1 Екзабайт буде

записуватися 300 тисяч років. На практиці розмір дискового тому файлової системи NTFS буде обмежуватись розмірами встановленого у системі жорсткого диска.

Кожен елемент файлової системи NTFS являє собою файл, а найголовніший файл має назву MFT (Master File Table) – загальна таблиця файлів, та є централізованим каталогом всіх інших файлів диска та себе самого. MFT поділений на записи однакового розміру (зазвичай 1 Кбайт), а кожний запис відповідає певному файлу. Каталог у файловій системі NTFS являє собою специфічний файл, що зберігає посилання на інші файли і каталоги та містить інформацію про ім'я файла, базові атрибути та посилання на елемент MFT, який вже надає повну інформацію про елемент каталога.

NTFS є файловою системою, що забезпечує високу відмовостійкість (що ми вже визначали, коли підкреслювали особливості Windows XP) за допомогою підтримки NTFS механізму транзакцій. Транзакція – це сукупність дій по передачі інформації від джерела до приймача, яка сприймається як єдине ціле та здійснюється повністю або не здійснюється зовсім. Це означає, що під час копіювання даних на диск та настанні критичної помилки скасований буде весь процес, а на диску не залишиться неповних даних, які могли б залишитися після незакінченого копіювання.

Ще однією відмінністю файлової системи NTFS є наявність у її файлів атрибуту **Сжатый**, що досягається підтримкою NTFS вбудованої підтримки стиснення дисків.

Файлова система NTFS також підтримує задання прав доступу для користувачів завдяки тому, що на відміну від FAT16/32 здатна зберігати інформацію про такі атрибути файла, як права доступу та власника. Загалом, основними позитивними рисами файлової системи NTFS є:

- підтримка максимально можливих розмірів дискових томів;
- швидка робота з файлами великого розміру та великими за обсягом каталогами;
- швидка робота з великими за обсягом жорсткими дисками;
- забезпечення безпеки зберігання інформації на комп'ютері;
- забезпечення захисту інформації від несанкціонованого доступу та підтримка можливості надання кожному користувачеві індивідуальних прав доступу;

Серед негативних рис файлової системи NTFS необхідно відмітити:

- великі вимоги до оперативної пам'яті комп'ютера;
- вимогливість до дискового обладнання, тобто не рекомендовано використовувати NTFS на повільних старих жорстких дисках;
- від жорсткого диска забирається 12% місця під MFT-зону.

Загалом можна зробити висновок, що файлову систему FAT можна використовувати на комп'ютерах зі старим устаткуванням та при роботі з невеликими жорсткими дисками, а також там, де не потрібний високий

ступінь безпеки даних та немає необхідності організації окремого сеансу доступу для кожного користувача. Файлову систему **NTFS** можна використовувати на потужних сучасних комп'ютерах, де вона, на відміну від **FAT**, покаже більшу швидкість роботи, а також на комп'ютерах, де необхідно забезпечити високий ступінь захисту інформації як від критичної помилки, так і від несанкціонованого доступу. В таблиці 3.1 наведена порівняльна характеристика трьох описаних файлових систем.

Таблиця 3.1
Файлові системи, які підтримує Windows XP

Характеристика	FAT	NTFS
Ім'я файла	«Вісім плюс три» символи ASCII (дозволяється один відокремлювач – крапка);	255 символів Unicode (дозволяється декілька відокремлювачів)
Атрибути	Тільки декілька один-бітових прапорців	Вся інформація, включаючи дані, які обробляються як атрибути файла
Загальна риса	Простота	Швидкість, відмовостійкість та безпека даних
Вбудовані засоби безпеки	Немає	Є
Максимальний розмір дискового тому	2 Гбайт у FAT16 та 2 Тбайти у FAT32	Обмежений фізичним розміром жорсткого диску
Операційні системи	Windows 95/ NT/98/2000/XP	Windows NT/2000/XP

Отже, ми зазначили основні риси трьох файлових систем **Windows XP**, проте необхідно згадати ще деякі нюанси. Зокрема, можливо перетворити файлову систему **FAT** на **NTFS**, але навпаки це можна зробити тільки після попереднього форматування дискового тому, що звісно призведе до повного знищення всіх даних на відповідному диску. Якщо на комп'ютері з якихось причин необхідна наявність двох операційних систем, наприклад **Windows XP** та **Windows 98**, а також необхідно забезпечити доступ до певного дискового тому з обох операційних систем, то необхідно форматувати цей том у файлову систему **FAT32**, оскільки якщо відформатувати том у файловій системі **NTFS**, то **Windows 98** його не «побачить». Мінімальний рекомендований розмір диска для використання файлової системи **NTFS** дорівнює 10 Гбайтам. Для дисків меншого розміру краще використовувати файлову систему **FAT32**.

Окрім того, що **Windows XP** під час свого встановлення надає можливість відформатувати диск з певною файловою системою, серед додаткового програмного устаткування для роботи з логічними розділами

жорсткого диска (тобто областями дискового простору, на які поділений фізичний жорсткий диск) можна виділити програмний продукт **Power Quest Partition Magic**, головне вікно якого показане на рис. 3.11.

На рисунку видно, що фізичний диск був розділений користувачем на два томи (C: та D:), перший з яких було відформатовано з файловою системою NTFS та встановлено **Windows XP**. Другий розділ (розділ, де знаходяться документи) було відформатовано з файловою системою **FAT32** з очевидним «прицілом» на майбутнє встановлення ОС **Windows 98** (яка, звісно, не «побачить» розділ, відформатований користувачем у **NTFS**). Серед можливостей програмного продукту **PowerQuest Partition Magic** відзначимо наступні:

- робота з файловими системами не тільки **Windows**, а й **Linux**;
- форматування розділів;
- переміщення розділів;
- розбиття диска на розділи;
- робота як з **Windows**, так і з завантажувальних дискет (тобто з таких, з яких може виконуватися завантаження комп'ютера з метою запуску спеціальних службових програм), які можна створити безпосередньо з програми;
- створення розділів на основі нерозміченої ділянки жорсткого диска;
- вилучення розділів.

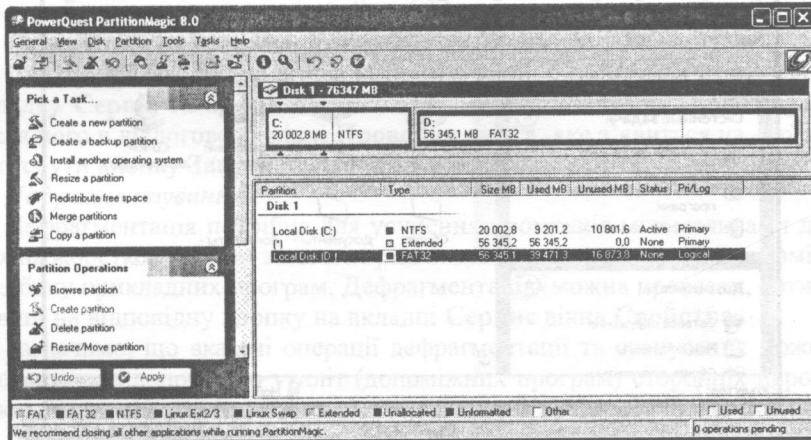


Рис. 3.11

3.5.2. Операції з об'єктами файлової системи

Розглянувши основні властивості файлових систем, які підтримує **Windows XP**, можна переходити до об'єктів файлової системи, з якими

безпосередньо працює користувач під час роботи з комп'ютером – з файлами, папками, дисками.

Робота з дисками

Після відкриття вікна **Мой компьютер** (рис. 3.12) ми бачимо піктограми дисководів для магнітних та оптичних дисків, логічних розділів жорсткого диска (на рисунку це **Локальный диск (C:)** та **Локальный диск (D:)**).

Відразу зазначимо, що наявність в системі двох томів не свідчить про таку ж кількість фізичних пристроїв (жорстких дисків) у комп'ютері, хоча може бути й таке. В даному випадку два розділи – це логічні розділи (частини дискового простору) одного фізичного пристрою. Кожен з цих розділів може бути підданий наступним основним діям.

Форматування

Необхідно зазначити, що системний розділ (на якому встановлена **Windows XP**) не можна відформатувати безпосередньо з **Windows**. Щодо форматування ми вже згадували про **PowerQuest Partition Magic**, хоча це можна робити й засобами **Windows XP**, зокрема клацнувши правую кнопку мишки на піктограмі відповідного розділу та в контекстному меню обравши команду **Форматировать...** (таким же чином форматується дискета). Нагадаємо, що при форматуванні всі дані на диску будуть знищені.

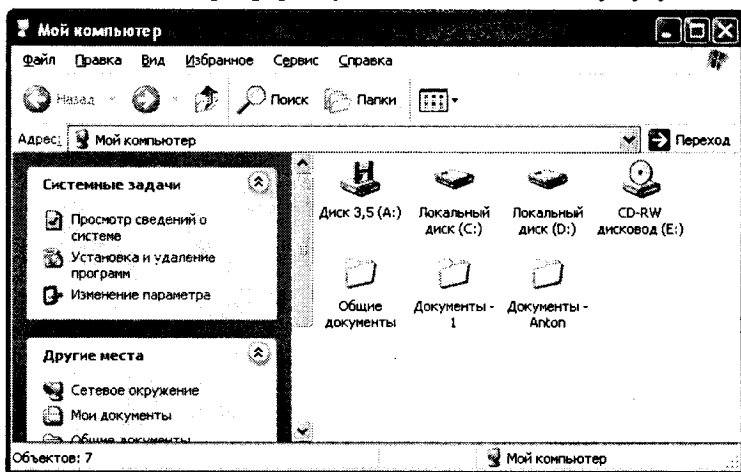


Рис. 3.12

Перегляд на предмет кількості зайнятого або вільного місця

Для цього необхідно клацнути на потрібному розділі правою кнопкою мишки та в контекстному меню обрати команду **Свойства...** Після виконання даної команди на екрані з'явиться вікно **Свойства** (рис. 3.13), де на вкладці **Общие** користувач має змогу переглянути інформацію про кількість вільного та зайнятого місця на даному логічному розділі.

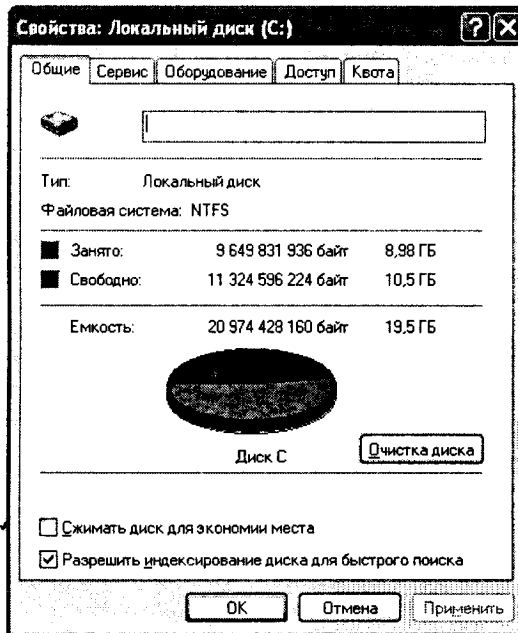


Рис. 3.13

Сканування на предмет наявності помилок

Для цього необхідно в діалоговому вікні **Свойства** активізувати вкладку **Сервис** та на ній натиснути кнопку **Выполнить проверку...**, після чого в діалоговому вікні **Проверка диска**, яке з'явиться на екрані, натиснути кнопку **Запуск** для початку операції.

Дефрагментування

Дефрагментація потрібна для усунення проміжків між масивами даних на жорсткому диску з метою пришвидшення звернення до його вмісту з боку прикладних програм. Дефрагментацію можна провести, натиснувши на відповідну кнопку на вкладці **Сервис** вікна **Свойства**.

Зазначимо, що вказані операції дефрагментації та сканування можна проводити за допомогою утиліт (допоміжних програм) сторонніх виробників, наприклад, програм з комплексу **Norton Utilities** або **Fix-It Utilities**.

Операції з файлами та папками

Файл – це поіменований набір даних, збережений на комп'ютерному носії. **Папка** – це поіменований простір для збереження інформації на носії, який може містити в собі інші папки та файли. Оскільки для того, щоб проводити з файлом (або папкою) певні дії, необхідна наявність самого файлу (папки), то розпочнемо розгляд операцій з цими об'єктами з операції створення.

Створення файлу та папки

Для створення папки та файлу необхідно виконати наступні дії.

1. Відкрити вікно папки, в якій необхідно створити об'єкт.
2. Клацнути у вільному місці області відображення папки правою кнопкою мишки та у контекстному меню, що з'явиться після цього на екрані, обрати команду **Создать**.
3. В підменю обрати команду **Папку**, якщо необхідно створити папку, або обрати команду з назвою файлу відповідного типу, щоб створити файл даного типу (наприклад, текстовий документ). Тип файлу визначає розширення, тобто частина імені файлу, що знаходиться після крайньої правої крапки (наприклад, файл з розширенням **doc** – документ **Microsoft Word**, а з розширенням **psd** – графічний малюнок **Adobe Photoshop**).

Після виконання наведеної послідовності команд буде створена папка або файл (в залежності від того, що було обрано) та відразу буде запропоновано ввести ім'я об'єкта. Можна не вводити ім'я, зафіксувавши ім'я за замовчуванням натисканням клавіші **Enter**, або ввести з клавіатури нове ім'я (бажано латинськими літерами) унікаючі символів \ / * " ? < > : |, які не можна використовувати в іменах файлів та папок. Якщо, ввівши нове ім'я, користувач, не натиснувши ще клавішу **Enter**, вирішив залишити старе, то необхідно натиснути клавішу **Esc**.

Після того, як ми створили файл або папку, ми тепер можемо її копіювати, переміщувати, видаляти, проте для цього необхідно знати, яким чином можна виділяти файл (папки) або їх групи.

Методи виділення файлів та папок

Файли, папки можна виділяти поодиноці, суміжними та несуміжними групами. Для виділення файлів або папок поодиноці достатньо навести курсор мишки на об'єкт (на його піктограму) та клацнути лівою кнопкою мишки.


Для виділення груп суміжних об'єктів (файлів та папок) існує простий спосіб виділення їх рамкою. Для цього необхідно виконати такі дії:

1. Вказати першу вершину прямокутної рамки. Для цього потрібно навести курсор мишки у вільному місці області відображення вмісту папки в тому місці, де на думку користувача повинна знаходитись перша вершина рамки, так, щоб рамка охоплювала всі потрібні об'єкти.
2. Натиснути ліву кнопку мишки та утримуючи її натиснутою, пересувати курсор у бік групи об'єктів, обводячи їх рамкою. З-під курсора «виростає» синя рамка, якою необхідно хоча б частково обвести всі об'єкти, які потрібно виділити. По мірі того, як рамка буде перекивати кожен новий об'єкт, він буде фарбуватись в інший колір, що вказує на те, що даний об'єкт буде виділений.
3. Після закінчення виділення рамкою всіх об'єктів відпустити ліву кнопку мишки.

Існує також інший спосіб виділення суміжних об'єктів. Можна просто клацати лівою кнопкою мишки на папках та файлах одночасно утримуючи натиснутою клавішу **Shift**. Зазначимо, що якщо користувач клацне мишкою при натиснутій клавіші **Shift** на, наприклад, першому та п'ятому файлі зі списку, то автоматично будуть виділені всі файли з першого по п'ятий включно.

Часто виникає необхідність виділення несуміжних файлів. Для такого виділення достатньо клацати лівою кнопкою мишки на піктограмах потрібних файлів або папок, утримуючи при цьому натиснутою клавішу **Ctrl**.

Відкриття папок і файлів та запуск файлів

Для того, щоб відкрити файл або папку, необхідно двічі клацнути лівою кнопкою мишки на піктограмі відповідного файлу (папки). Папка буде відкрита, а файл буде відкритий тим додатком, якому відповідають файли даного типу. Якщо файл має невизначений тип, його піктограма має вигляд , а після спроби відкриття такого файлу з'являється вікно, зображене на рис. 3.14, в якому можна вказати, де шукати програму, яка «розуміє» файли подібного типу, з переліку програм, встановлених в системі або в Інтернеті. Потрібно зазначити, що дане вікно може бути викликане для файла будь-якого типу, для чого необхідно клацнути правою кнопкою мишки на піктограмі файла та у контекстному меню виконати команду **Открыть с помощью...**

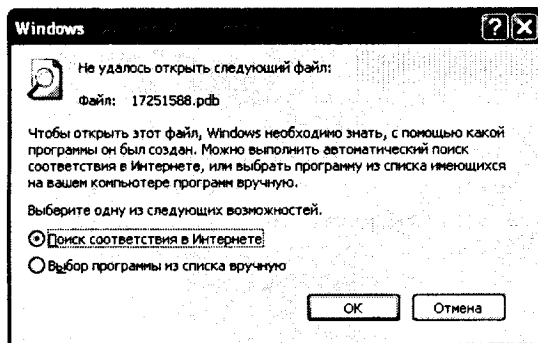


Рис. 3.14

Якщо необхідно відкрити відразу декілька файлів одного типу, то раціонально їх виділити, а потім натиснути клавішу **Enter**.

Для відкриття папок, дисків та запуску виконавчих файлів програм (тобто таких файлів, які самі по собі запускають певну програму, виконавчим файлом якої вони є та мають розширення **exe**) можна також скористатись діалоговим вікном **Запуск программы** (рис. 3.15), яке можна викликати, обравши в меню **Пуск** команду **Выполнить...**

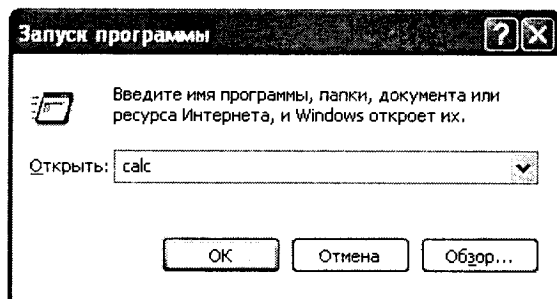


Рис. 3.15

Для запуску виконавчого файлу або відкриття папки або диску необхідно ввести в поле **Открыть:** повний шлях до потрібного об'єкта та натиснути кнопку **ОК** (наприклад, ввівши шлях `C:\MATLAB6p5\bin\win32\matlab.exe`, ми запустимо програму **Matlab**, виконавчим файлом якої є `matlab.exe`). **Повний шлях до файлів** – це послідовність об'єктів, послідовно відкриваючи які користувач зможе зайти у папку, що містить даний файл, відокремлених символом `\`. Починається повний шлях з мітки диска (наприклад `C:`), на якому знаходиться файл, а закінчується ім'ям файлу.

Існує певна група стандартних програм, для запуску яких не потрібно вводити повний шлях до виконавчого файлу цих програм, а досить ввести тільки коротке ім'я файлу без розширення. На рис. 3.15 після натискання кнопки **ОК** буде запущена програма **Калькулятор**, виконавчим файлом якої є `calc.exe`.

Поняття ярлика, створення ярликів

Ярлик – файл з розширенням `lnk` або `pif`, який містить тільки посилання на певний об'єкт. Таким чином, відкриваючи ярлик, ми відкриваємо об'єкт, на який ярлик посилається. Ярлики розміщуються переважно на **Робочем столі**, **Панелі быстрого запуска**, в розділі **Программы** меню **Пуск** для швидкого доступу до тих об'єктів, на які вони посилаються. Існує декілька способів створити ярлик.

1. Виконати таку послідовність дій:

- відкрити вікно папки, де необхідно мати ярлик, та відкрити вікно папки, де знаходиться папка або файл, ярлик якої необхідно створити;
- навести на об'єкт, до якого створюється ярлик, курсор мишки, натиснути праву кнопку мишки;
- утримуючи кнопку мишки натиснутою, перетягнути курсор в область відображення вмісту тієї папки, де потрібен ярлик;
- відпустити кнопку мишки;
- на екрані з'явиться контекстне меню, де необхідно обрати команду **Создать ярлики**.

2. Перетягувати об'єкт, до якого потрібно створити ярлик, в ту папку, де необхідно мати ярлик, тримаючи натиснутою ліву кнопку мишки та клавішу **Alt**. При перетягуванні біля курсора з'явиться чорна вигнута стрілка, що вказує на те, що після відпускання кнопки мишки буде створений ярлик до об'єкта там, куди було перетягнуто піктограму об'єкта.
3. Виконати такі дії:
 - відкрити папку, де необхідно створити ярлик, та у вільному місці області відображення (або, у випадку з **Робочим столом**, на його вільному місці) клацнути правою кнопкою мишки;
 - у контекстному меню, яке з'явиться на екрані, потрібно обрати команду **Создать** та в підменю команду **Ярлык**, після чого на екрані з'явиться вікно **Создание ярлыка** (рис. 3.16, а);
 - у вікні **Создание ярлыка** натиснути кнопку **Обзор...**;
 - у вікні **Обзор папок** (рис. 3.16, б) обрати потрібний об'єкт, до якого необхідно створити ярлик та натиснути кнопку **ОК**;
 - знову опинившись у вікні **Создание ярлыка**, натиснути на кнопку **Далее >** та ввести ім'я ярлика;
 - після введення імені, натиснути на кнопку **Готово**, яка з'явиться на місці кнопки **Далее >**.

Зазвичай візуально піктограма ярлика відрізняється від піктограми іншого файлу наявністю чорної вигнутої стрілки в нижньому лівому куті.

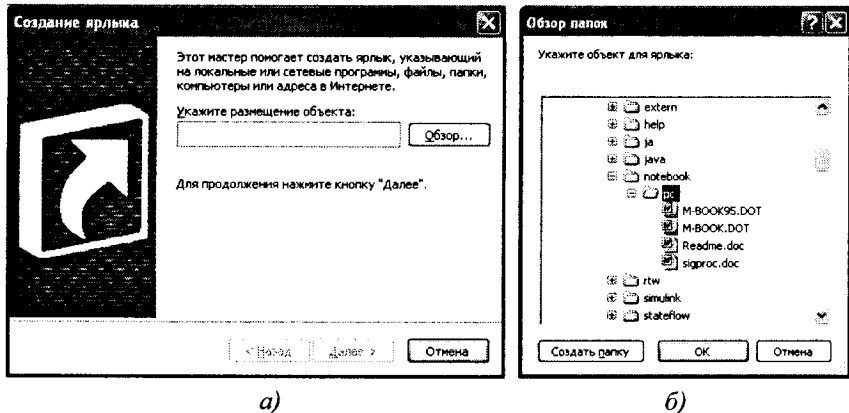


Рис. 3.16

► Потрібно пам'ятати, що видаляючи ярлик, ми не видаляємо сам об'єкт, на який ярлик посилається. Першими двома способами можна створити ярлики до кількох об'єктів відразу, для чого потрібно спочатку виділити групу об'єктів, а потім повторити вказані дії в п.1 або 2.

Копіювання та переміщення файлів та папок за допомогою буфера обміну

Буфер обміну – область оперативної пам'яті, яка використовується для обміну інформацією між додатками **Windows** та може бути використана при виконанні операцій копіювання та переміщення файлів або папок. Загалом, методика копіювання або переміщення будь-якого об'єкта (файлу, текстової інформації, графіки тощо) за допомогою буфера обміну полягає в тому, щоб помістити спочатку об'єкт до буфера обміну, а потім, відкривши об'єкт-приймач, зробити вставку вмісту буфера обміну. Що ж стосується безпосередньо копіювання та переміщення за допомогою буфера обміну, то це можна зробити в такій послідовності.

1. Виділяємо об'єкт (групу об'єктів).

2. Розміщуємо об'єкт в буфері обміну, натиснувши комбінацію клавіш **Ctrl+C** (при копіюванні) або **Ctrl+X** (при переміщенні).

Якщо користувач не пам'ятає комбінацій клавіш, він може помістити об'єкт до буфера обміну, клацнувши правою кнопкою мишки на об'єкті та у контекстному меню обравши команду **Копировать** (при копіюванні) або **Вырезать** (при переміщенні). При цьому перший пункт стає необов'язковим, якщо потрібно скопіювати тільки один файл або папку, а не їх групу.

3. Відкриваємо папку, куди необхідно скопіювати або перемістити об'єкт.

4. Натискаємо комбінацію клавіш **Ctrl+V** або, клацнувши правою кнопкою у вільному місці області відображення вмісту папки, в контекстному меню обираємо команду **Вставить**.

Після виконання цих операцій в потрібне місце буде скопійований або переміщений об'єкт (залежно від того, яку команду обрано в п.2).

Інші способи копіювання та переміщення

Серед інших способів копіювання або переміщення можна здійснити простим перетягуванням об'єкта до потрібної папки. Для цього необхідно зробити наступне:

- відкрити папку, яка містить потрібний об'єкт (групу об'єктів) та папку-приймач;
- навести курсор на об'єкт;
- натиснути ліву кнопку мишки та, утримуючи її в такому стані, перетягнути курсор (а, відповідно, й піктограму об'єкта) до папки-приймача.

При перетягуванні файла або папки потрібно пам'ятати важливий нюанс. Якщо папка, що містить об'єкт, з яким проводиться операція, та папка-приймач знаходиться на одному логічному розділі, то буде виконана операція переміщення. Якщо ж ці дві папки знаходяться на різних логічних розділах, то буде виконана операція копіювання.

Щоб не виникло плутанини, таке перетягування можна робити при натиснутій клавіші **Ctrl** при копіюванні (в такому випадку біля піктограми об'єкта з'являється позначка +) та **Shift** при переміщенні.

Наведені способи копіювання та переміщення дозволяють обрати користувачу найбільш раціональний та зручний спосіб в залежності від ситуації.

Переименовання файлів або папок

Переименовання файла або папки можна виконати одним з наступних способів.

1. Клацнути правою кнопкою мишки на файлі або папці та у контекстному меню обрати команду **Переименовать**, після чого ви отримуете можливість ввести ім'я файла. Нагадаємо, що в імені файла або папки не можна використовувати символи \ / * " ? < > |.
2. Виділити об'єкт, через деякий час ще раз клацнути лівою кнопкою мишки на назві об'єкта, після чого можна також ввести нове ім'я.
3. Виділити об'єкт, натиснути клавішу **F2** та ввести нове ім'я.

Видалення файлів або папок

Для того, щоб видалити файл, папку або групу цих об'єктів, необхідно їх виділити та натиснути клавішу **Del**. Після цього на екрані з'явиться діалогове вікно **Подтверждение удаления файла** (або папки) (рис. 3.17,а), в якому для видалення файлу потрібно натиснути кнопку **Да**.

Файли видаляються до папки **Корзина Windows XP**, з якої потім при бажанні видалені файли можна поновити в тій папці, з якої користувач їх видаляв. Для цього необхідно виконати наступні дії:

- відкрити папку **Корзина**;
- клацнути правою кнопкою мишки на потрібному об'єкті або групі об'єктів (перед цим їх виділивши);
- у контекстному меню обрати команду **Восстановить**.

Видалити файли або папки з папки **Корзина** можна як і з будь-якої звичайної папки, проте слід пам'ятати, що видалені з **Корзини** об'єкти часто поновити неможливо.

Для того, щоб відразу видалити файли або папки, обминаючи **Корзину**, необхідно:

- виділити потрібні об'єкти;
- натиснути комбінацію клавіш **Shift+Del**;
- у вікні (рис. 3.17,б) натиснути кнопку **Да**.

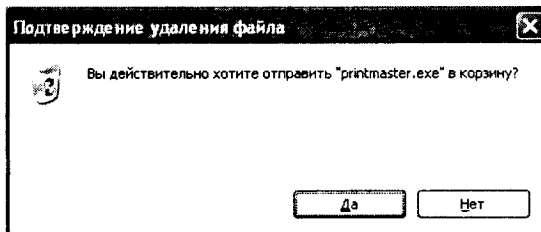


Рис. 3.17,а

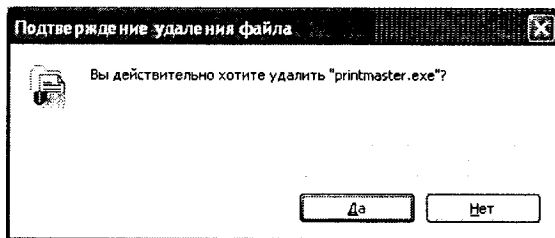



Рис. 3.17,6

Видалення файлів або папок можна також проводити за допомогою команди **Удалить** з контекстного меню, яке можна викликати, клацнувши правою кнопкою мишки на відповідному об'єкті. При цьому видаляти файли, оминаючи **Корзину**, можна при натиснутій клавіші **Shift**.

Пошук файлів

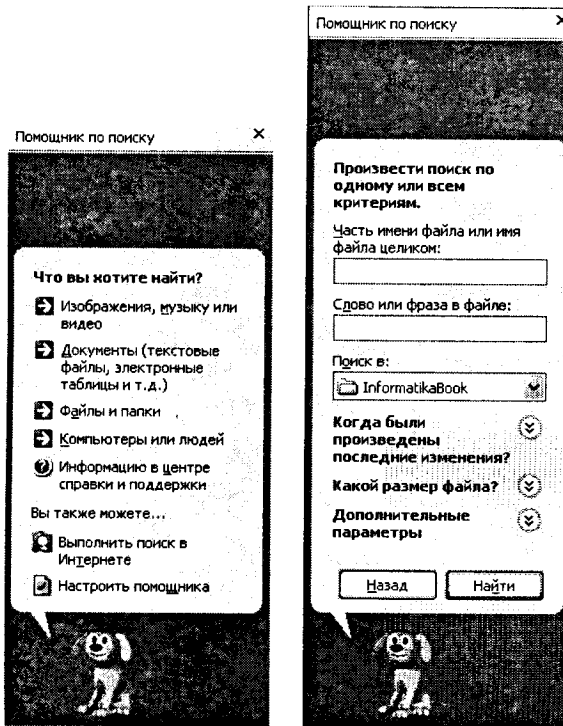
Пошук файлів – це операція, згідно з якою потрібно шукати файли за певними критеріями (за ім'ям файлу або тільки за відомою частиною імені, або за датою створення тощо). Проводити пошук можна наступним чином.

1. Виконати команду **Найти** → **Файлы и папки...** з меню **Пуск**, або натиснути кнопку  на панелі інструментів **Обычные кнопки** у вікні будь-якої відкритої папки.
2. Після цього на екрані з'явиться панель інструментів **Помощник по поиску** (рис. 3.18,а), де необхідно обрати тип об'єктів, які потрібно шукати. Якщо користувач обирає пункт **Файлы и папки** (найбільш узагальнений), панель інструментів змінить свій вигляд на зображений на рис. 3.18,б.
3. Користувачу пропонується ввести критерії пошуку: дату останньої внесеної зміни, місце пошуку (поле **Поиск в**), ім'я файлу, розмір або додаткові параметри (враховувати чи ні реєстр, чи шукати в системних папках тощо).
4. Після введення всіх даних, потрібних для пошукової системи, натиснути кнопку **Найти** для початку пошуку або кнопку **Назад**, якщо було вирішено вибрати інший тип об'єктів, які потрібно шукати.

Оскільки часто потрібно шукати файл не за повним відомим ім'ям, а за типом чи навіть за першою буквою в імені, то виникає завдання пошуку файлів за маскою, згідно з якою потрібно задати таку маску (шаблон імені файла), яка б водночас і могла представляти той об'єкт, який ми шукаємо, а водночас не була надто загальною, щоб результатом такого пошуку було якомога менше об'єктів (а серед них і потрібний). Маска задається як за допомогою звичайних літер алфавіту, так і за допомогою деяких спеціальних символів, а саме:

? означає один будь-який символ в імені файла;

* означає будь-яку кількість будь-яких символів в імені файла.



а)

б)

Рис. 3.18

Нижче в таблиці 3.2 наведені приклади деяких масок.

Таблиця 3.2

Приклади масок пошуку

Маска	Що шукаємо
*.doc	Всі файли з розширенням doc
Document.*	Файли будь-якого типу з ім'ям Document
????*.exe	Файли з розширенням exe та з чотирма будь-якими символами в імені.
????.*	Файли будь-якого типу з чотирма будь-якими символами в імені
*T.bmp	Файли з розширенням bmp та в імені яких остання буква T
T???*.txt	Файли з розширенням txt , ім'я яких починається на букву T та складається з чотирьох символів
.	Всі файли та папки
??T*.ht?	Файли, в імені яких третя літера від початку T , а розширення складається з трьох символів, причому перші два з них h та t , а останній невідомий.

Маску пошуку потрібно ввести у поле **Часть имени файла или имя файла целиком**, після чого натиснути кнопку **Найти**. Результати пошуку відображаються праворуч від панелі інструментів **Помощник по поиску** (рис. 3.19).

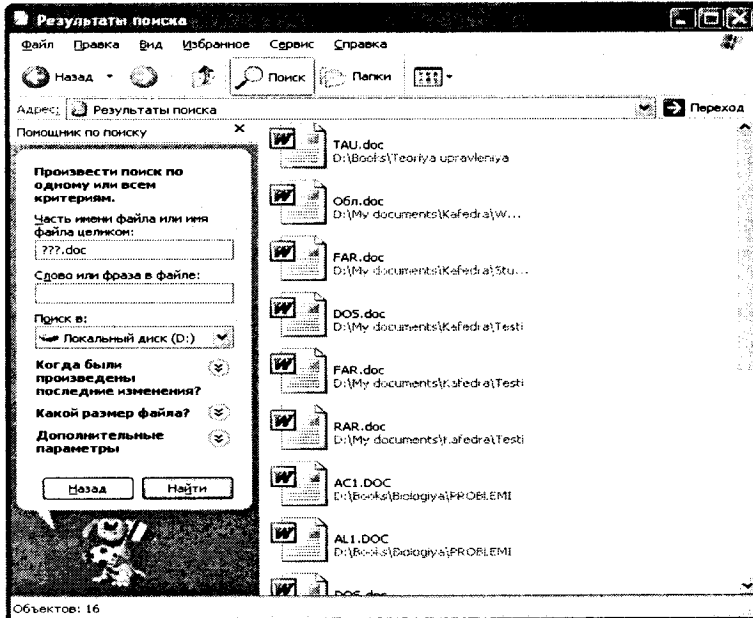


Рис. 3.19

У вікні результатів пошуку можна також проводити всі вищезгадані операції з об'єктами, зокрема, видаляти знайдені файли, змінювати їх ім'я, копіювати та перемішувати. Не дозволяється створювати файли та папки. Якщо необхідно відкрити папку, яка містить знайдений об'єкт, то необхідно клацнути правою кнопкою мишки на потрібному знайденому об'єкті та у контекстному меню обрати команду **Открыть содержащую объект папку**.

Перегляд властивостей файла, зміна атрибутів

Під властивостями файла мається на увазі розмір файлу, дата створення, додаток, якому відповідає даний файл, повний шлях до файла. Для перегляду властивостей файлів і зміни атрибутів файла потрібно або клацнути на файлі правою кнопкою мишки та у контекстному меню обрати команду **Свойства**, або виділити файл та натиснути комбінацію клавіш **Alt+Enter**. В обох випадках на екрані з'явиться діалогове вікно **Свойства** (рис.3.20), на вкладці **Общие** якого можна дізнатись основні властивості файла.

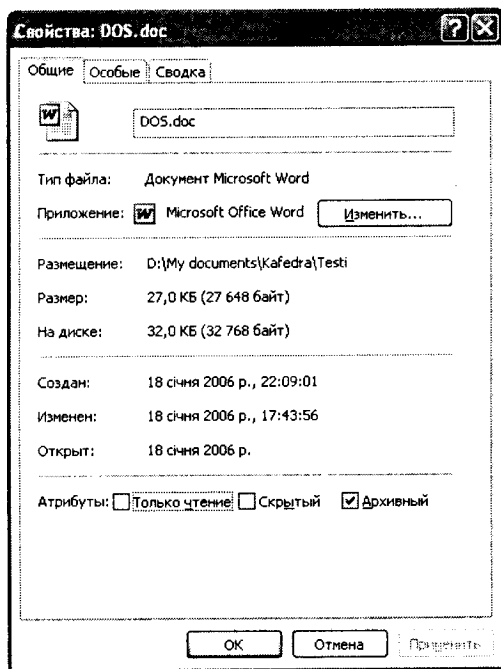


Рис. 3.20

За допомогою діалогового вікна **Свойства** можна також перейменувати файл, для чого необхідно ввести нове ім'я у верхнє текстовє поле, після чого натиснути кнопку **ОК**, якщо необхідно зберегти зміни та закрити вікно, або клацнути кнопку **Применить**, якщо необхідно зберегти зміни, але продовжувати роботу з діалоговим вікном (це загальна відмінність кнопки **ОК** від кнопки **Применить**).

Параметр **Приложение** показує додаток, якому відповідає файл даного типу в операційній системі. Натиснувши на кнопку **Изменить...** отримаємо можливість встановити відповідність файлів даного типу іншому додатку.

Потрібно розрізняти два розміри файлу, які зазначені в діалоговому вікні. Параметр **Размер** показує теоретичний розмір файлу, а параметр **На диске** показує фактичний обсяг дискового простору, який займає файл. Для користувачів більш важливий саме другий параметр.

В нижній частині діалогового вікна **Свойства** знаходяться атрибути файлу, задати які можна за допомогою прапорця ліворуч від назви відповідного атрибута. Встановлення атрибута **Только чтение** означає, що зміни в даному файлі не можна буде зберегти в цьому ж файлі доти, доки даний атрибут встановлений. При спробі збереження змін в файлі, який має встановлений атрибут **Только чтение**, програма, якою редагувався

файл, може вивести діалогове вікно, подібне тому, яке виводить в таких випадках поширений додаток **Microsoft Word** (рис. 3.21).

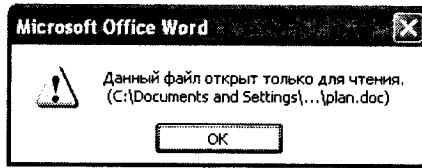


Рис. 3.21

Встановлення атрибута **Скрытый** означає, що без встановлення опції відображення прихованих файлів файла з цим атрибутом під час перегляду вмісту папки видно не буде. Для того, щоб увімкнути режим відображення прихованих файлів, необхідно виконати такі дії:

- зайти в меню **Сервис** вікна папки **Windows**;
- обрати команду **Свойства папки**, після чого з'явиться однойменне вікно;
- у цьому вікні, на вкладці **Вид** (рис. 3.22), у списку **Дополнительные параметры** потрібно активізувати перемикач **Показывать скрытые файлы и папки**;
- натиснути кнопку **OK** або **Применить** (різниця між ними була вказана вище). Після цього приховані файли будуть відображатися у вікнах папок блідими.

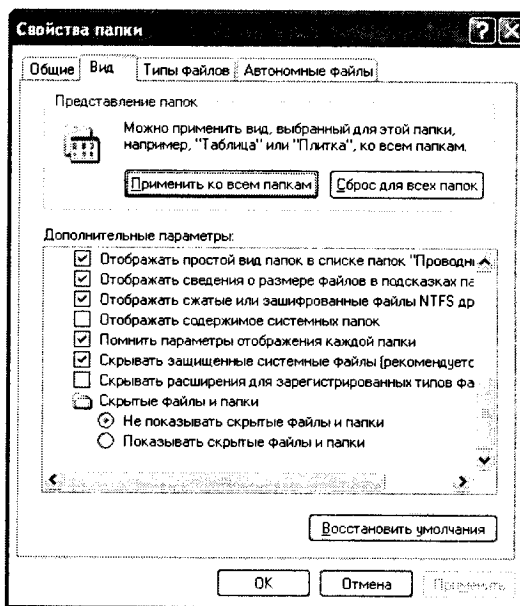


Рис. 3.22

Встановлений атрибут **Архівний** вказує на те, що даний файл може підлягати стисканню з боку певної програми, яка аналізує даний файл на предмет наявності цього атрибута.

3.5.3. Типи файлів Windows XP

Закінчуючи розгляд операцій над файлами, необхідно навести основні файлові формати, з якими користувач може зіткнутись при роботі в ОС **Windows XP**. Поширені формати файлів та деякі додатки, які можуть з ними працювати, наведені в таблиці 3.3.

Таблиця 3.3

Формати файлів Windows XP

Формат	Опис	Додаток
Exe	Виконавчий файл	-
Txt	Текстовий файл	Notepad, Microsoft Word
Dll	Бібліотека динамічного компонування	-
Jpg	Стиснуте з втраченою якістю графічне зображення	Adobe Photoshop, Corel Photo-Paint, ACDSee
Bmp	Графічне зображення	Adobe Photoshop, Corel Photo-Paint, ACDSee
Psd	Графічне зображення	Adobe Photoshop, Corel Photo-Paint
Doc	Текстовий документ	Microsoft Word
Htm	Інтернет-сторінка	Internet Explorer
Dwg	Графічний формат інженерних креслень	AutoCad
Ai	Графічний формат	Adobe Illustrator
Cdr	Графічний формат	CorelDraw
Mp3	Звуковий формат	WinAmp
Avi	Відео формат	Windows Media Player
Rar	Архів	WinRar
Xls	Електронна таблиця Excel	Microsoft Excel
Ppt	Презентація PowerPoint	Microsoft PowerPoint
Pdf	Електронна документація	Acrobat Reader, Acrobat
Lnk	Ярлик	-
Ttf	Файл шрифту	ACDSee
Zip	Архів	WinRar
Djv	Електронна документація	DjVuReader
Sys	Системний файл	-
Mdb	Файл бази даних	Microsoft Access
Rtf	Текстовий формат	Microsoft Word
Wav	Звуковий формат	WinAmp

Встановлення відповідності додатків типам файлів

Іноді виникає необхідність змінити додаток, який би відкривав за замовчуванням файл певного типу. Для цього необхідно:

- у вікні папки зайти в меню **Сервіс** та обрати команду **Свойства папки**;
- у вікні **Свойства папки**, яке з'явиться на екрані, перейти на вкладку **Типы файлов**. Вигляд вікна **Свойства папки** з відкритою вкладкою **Типы файлов** наведений на рис. 3.23;
- для того, щоб встановити відповідність типу файлу іншому додатку, необхідно знайти цей тип у списку **Зарегистрированные типы файлов**, виділити його та натиснути на кнопку **Изменить**;
- у вікні **Выбор программы**, яке з'явиться на екрані, користувач має змогу обрати нову програму для відкриття файлів даного типу та натиснути кнопку **ОК**. Після такої послідовності дій даному типу файлів буде адресований новий додаток.

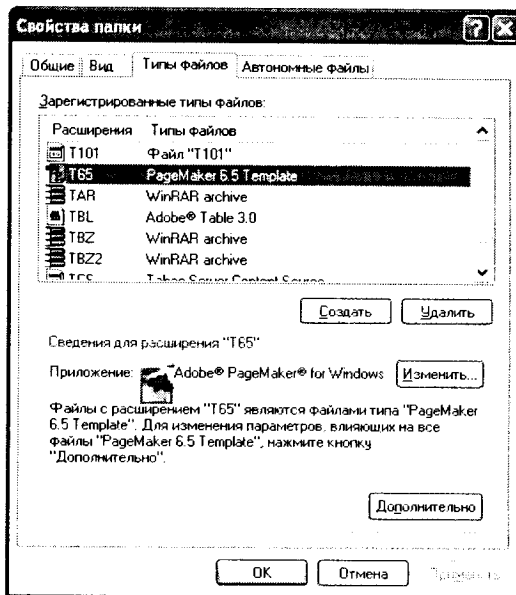






Рис. 3.23

Потрібно бути обережним, щоб не адресувати типу файлів той додаток, який не зможе відкрити файли даного типу.

3.5.4. Навігація по папках Windows XP

Навігація папками зводиться до переходу на вищий рівень ієрархії, вперед або назад та відкритті нижчих за ієрархією папок. Для виконання цих дій **Windows** містить відповідні кнопки на панелі інструментів **Обычные кнопки**:

- кнопка **Назад**  – перехід до папки, яка була відкритою перед цим. Цій кнопці відповідає комбінація клавіш **Alt+←**;
- кнопка **Вперед**  – відміна переходу назад. Цій кнопці відповідає комбінація клавіш **Alt+→**;
- кнопка **Вверх**  – перехід на один рівень вище, тобто з поточної папки ми переходимо до тієї папки, яка містить поточну папку.

Зручний спосіб для навігації по папках надає дерево папок, показати яке у вікні папки можна натиснувши кнопку **Папки**  на панелі інструментів **Обычные кнопки**. Після натискання на цю кнопку в лівій частині вікна папки з'явиться панель інструментів **Папки**, в якій представлене дерево папок (рис. 3.24). В дереві папок ліворуч тих папок, які містять інші папки, стоїть позначка +, натиснувши на яку, користувач розгортає дану папку, і біля неї позначка + змінюється на -. Для того, щоб переглянути повний вміст папки (папки та файли), необхідно просто її виділити у дереві папок (тоді біля неї з'явиться позначка -), а в правій частині вікна з'явиться повний вміст папки.

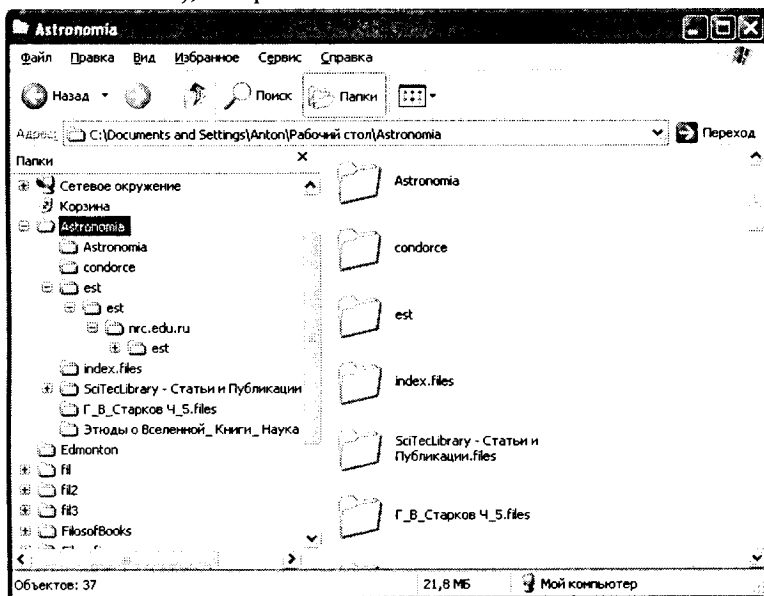


Рис. 3.24

Швидко переходити до папки, яка знаходиться в іншому місці дерева папок, можна також за допомогою **Адресної строки**. Для цього потрібно у текстове поле **Адрес** в **Адресной строке** ввести повний шлях до папки або вибрати потрібний об'єкт зі списку, розкрити який можна, клацнувши лівою кнопкою мишки на кнопці в правій частині рядка **Адрес**. Таким чином особливо легко та швидко переходити до магнітних дисків.

Спеціальні папки Windows XP

Як і в попередніх версіях Windows, у Windows XP є папки, призначені для спеціальних цілей. Після встановлення ОС на системному розділі жорсткого диска будуть створені папки **Windows**, **Program Files**, **Documents and Settings** та деякі інші. Згадані три папки є найважливішими папками, що зазвичай знаходяться на диску C:. Папка **Windows** є папкою, в яку інстальатор ОС копіює всі системні файли, необхідні для системи, та серед інших містить такі папки, як:

- **System32**, яка містить системні файли операційної системи – бібліотеки динамічного компонування, деякі системні програми, необхідні для роботи ОС;
- **System**, яка містить системні файли (бібліотеки динамічного компонування, програми), що необхідні для сумісності Windows XP з попередніми версіями Windows;
- **Fonts**, яка містить шрифти, які встановлені в ОС. Цю папку можна відкрити також з **Панелі управління**;
- **Temp**, яка містить тимчасові файли. Періодично вміст цієї папки можна видаляти. Якщо певний тимчасовий файл потрібний програмі, яка в даний момент виконується в системі, то буде виведене повідомлення про неможливість його видалення.

В папку **Program Files** за замовчуванням встановлюється більшість програм. Саме папки встановлених програм і зберігаються в цій папці. Окрім папок програм користувача, в папці **Program Files** знаходиться папка **Common Files**, яка містить папки, де знаходяться деякі файли програм, які можуть використовуватися іншими додатками.

Папка **Documents and Settings** містить приватні папки користувачів, які наявні в ОС. Коли додається новий користувач, то відразу в папці **Documents and Settings** створюється відповідна папка, яка має таку саму назву, як ім'я нового користувача (це домашня папка цього користувача). В домашній папці користувача знаходяться наступні основні папки:

- **Рабочий стол**, яка містить вміст **Рабочего стола Windows XP**;
- **Мои документи**, яка може використовуватись для збереження робочих файлів даного користувача;
- **Главное меню**, яка містить деякі, індивідуальні для даного користувача, елементи меню **Пуск**;
- **Недавние документы**, яка, виходячи з назви, містить ярлики, що посилаються на файли, з якими нещодавно працював користувач;
- **Local Settings**, яка є прихованою та містить деякі папки, в яких зберігаються індивідуальні тимчасові файли, які створювали програми, запущені в сеансі роботи даного користувача (папка **Local Settings/Temp**), а також які були створені під час роботи в мережі **Internet** (папка **Temporary Internet Files**).

Папку **Temp** в папці **Local Settings**, як і її аналог в папці **Windows**, потрібно періодично чистити (повністю видаляти вміст), але оскільки це робити не достатньо зручно через значну вкладеність папки **Temp**, а також через необхідність робити приховані файли та папки видимими, то у **Windows XP** є можливість змінити місце, куди будуть зберігатися тимчасові файли. Для цього необхідно виконати наступну послідовність дій.

1. Клацнути правою кнопкою мишки на піктограмі **Мой компьютер** та в контекстному меню, яке з'явиться на екрані, обрати команду **Свойства**, після чого на екрані з'явиться діалогове вікно **Свойства системы**.
2. В діалоговому вікні **Свойства системы** необхідно перейти на вкладку **Дополнительно** та натиснути на кнопку **Переменные среды**, після чого на екрані з'явиться однойменне вікно (рис. 3.25).

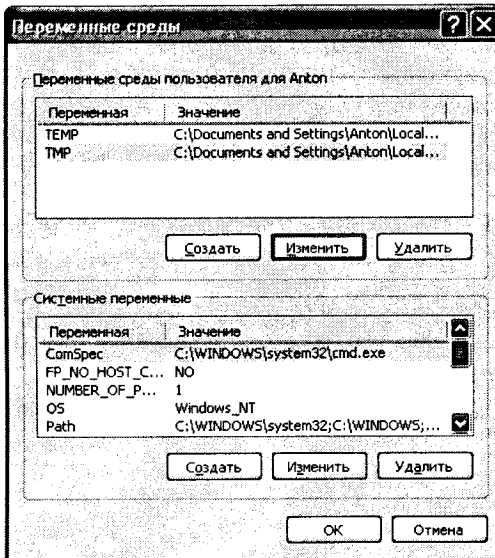


Рис. 3.25

3. В діалоговому вікні **Переменные среды** у списку **Переменные среды пользователя** для (ім'я користувача) необхідно задати нові значення змінних **TEMP** та **TMP**, для чого потрібно виділити змінну у списку та натиснути кнопку **Изменить**, після чого на екрані з'явиться діалогове вікно **Изменение пользовательской переменной**.

У вікні **Изменение пользовательской переменной** в текстовому полі **Значение переменной** необхідно задати нове значення змінної, а саме новий шлях до папки, де будуть зберігатися тимчасові файли. Значенням обох наведених у списку змінних за замовчуванням є:

%USERPROFILE%\Local Settings\Temp

Без зміни даного параметра тимчасові файли будуть зберігатися в папці **Temp**, яка міститься в папці **Local Settings** з домашньої папки користувача. Замість цього шляху можна ввести **C:\Temp** або інший на вибір користувача.


4. Натиснути кнопку **ОК** в поточному діалоговому вікні, а потім кнопку **ОК** в діалоговому вікні **Переменные среды**.

Також можна встановити новий шлях до папки для зберігання системних тимчасових файлів (це папка **Windows\Temp**). Для цього у списку **Системные переменные** діалогового вікна **Переменные среды** слід обрати параметри **TEMP** і **TMP** та натиснути кнопку **Изменить**, після чого у діалоговому вікні задати новий шлях до системної папки для тимчасових файлів.

3.6. Параметри налагодження вигляду об'єктів ОС

Оскільки під час своєї роботи користувач працює з об'єктами різного типу та виконує різні за метою та складністю завдання, то часто виникає необхідність змінити спосіб відображення певного об'єкта так, щоб робота з ним була максимально зручною та естетичною. Сюди можна віднести сортування піктограм у вікні папки, зміну режимів відображення вмісту папки, настройки вигляду папки тощо.

3.6.1. Режими відображення вмісту папки

Для того, щоб змінити режим перегляду файлів у папці, потрібно зайти в меню **Вид** або натиснути на кнопку **Вид**  на панелі **Обычные кнопки** та обрати один з наступних режимів відображення.

1. **Эскизы страниц** (рис. 3.26).

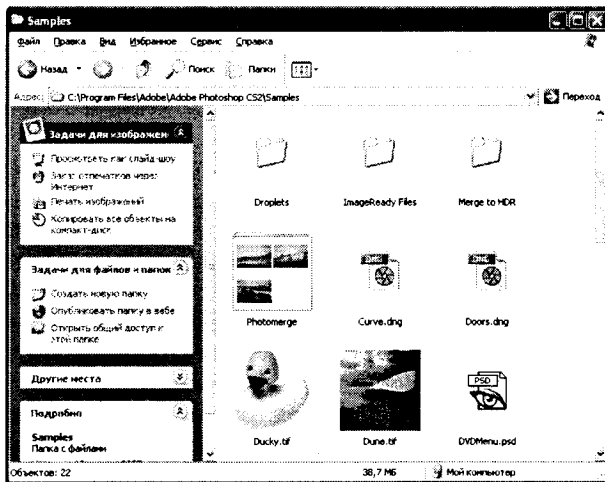






Рис. 3.26

Особливістю цього режиму є можливість перегляду вмісту графічних файлів та першого кадра файлів відео без їх відкриття. Якщо в папці присутні графічні файли, то вміст цих файлів відображається на піктограмі папки, як у випадку з папкою **Photomerge** на рис. 3.26.

2. Діафільм (рис. 3.27).

В цьому режимі ми, окрім перегляду вмісту графічних файлів, як в попередньому режимі, маємо змогу, пересуваючись по графічних зображеннях в папці (кнопки **Предыдущее изображение (стрелка влево)**  та **Следующее изображение (стрелка вправо)** ) , переглядати графічні файли у збільшеному вигляді, а також повертати їх кнопками **Повернуть по часовой стрелке**  та **Повернуть против часовой стрелки** .

Потрібно зазначити, що режим **Діафільм** для папки наявний у списку тільки тоді, коли у якості шаблону для цієї папки встановлений шаблон **Фотоальбом** або **Изображения**. Для того, щоб встановити ці шаблони, необхідно виконати такі дії:

- зайти на вкладку **Настройки** діалогового вікна **Свойства папки** (для відкриття цього вікна необхідно виконати команду **Свойства папки...** з меню **Сервис** вікна папки **Windows**);
- у списку **Использовать в качестве шаблона следующую папку** встановити один з вказаних шаблонів;
- натиснути кнопку **ОК**. Після цього режим **Діафільм** стає доступний.

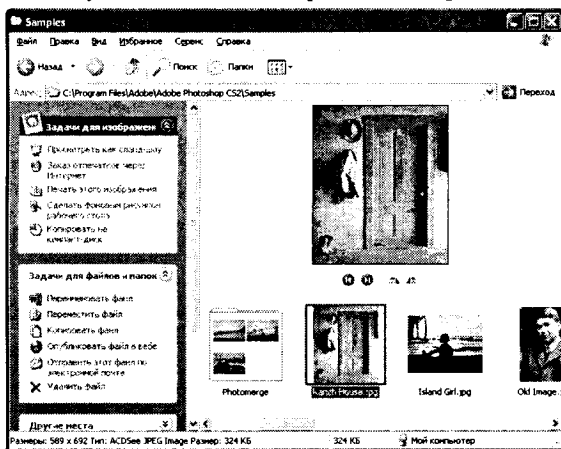


Рис. 3.27

3. Плитка (рис. 3.28,а).

У цьому режимі відображаються піктограми файлів та папок достатньо великого розміру, а під кожним об'єктом подаються короткі дані про тип об'єкта та його розмір. Може бути тільки вертикальна смуга прокрутки.



Рис. 3.28

4. Значки (рис. 3.28,б).

У цьому режимі піктограми меншого розміру, ніж у попередньому режимі, а під об'єктами відсутня інформація про тип об'єкта. Як і в попередньому режимі, наявна тільки вертикальна смуга прокрутки.

5. Список.

У цьому режимі файли та папки відображаються у стовпчик піктограмами невеликих розмірів. Вікно папки, в режимі **Список**, не може мати вертикальної смуги прокрутки. В режимі **Список** не відображається ніяка інформація про файли та папки окрім безпосередньо ім'я файлу та його піктограми.

6. Таблиця (рис. 3.29).

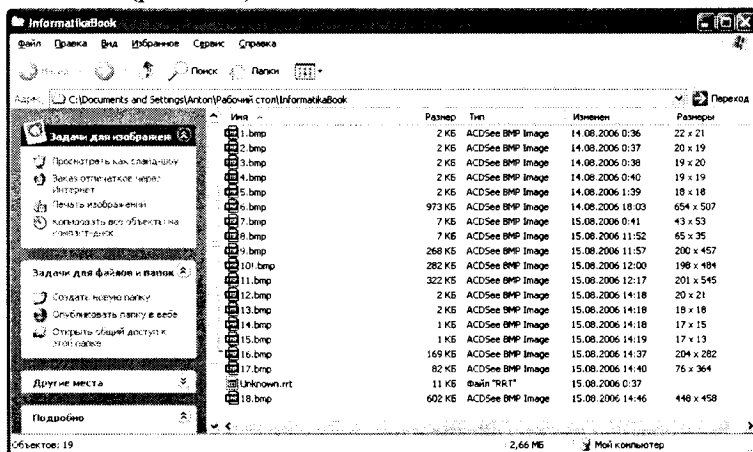


Рис. 3.29

Режим здатний надавати найбільш повну інформацію про об'єкт у поточній папці. Режим **Таблиця** одночасно може інформувати користувача

про ім'я файлу, розмір, дату створення, встановлені атрибути, тип, власника та надавати багато іншої інформації, частина з якої може бути отримана тільки для файлів певного типу (наприклад, якість звуку та тривалість для звукових файлів).

Як бачимо з рис. 3.29, в режимі **Таблиця** у структурі вікна папки з'являється новий елемент – рядок заголовка стовпчиків. Для того, щоб додати новий стовпчик до таблиці, необхідно виконати наступне.

1. Зайти в меню **Вид** та виконати команду **Выбор столбцов в таблице**.
2. На екрані з'явиться вікно **Выбор столбцов в таблице**, в якому можна додати до таблиці стовпчик, встановивши прапорець навпроти його назви.
3. По закінченні встановлення всіх потрібних прапорців, щоб зберегти зміни, потрібно натиснути кнопку **ОК**.

Для того, щоб вирівняти ширину стовпчика за його максимальним елементом, потрібно:

- навести курсор на межу стовпчика в рядку заголовка стовпчика;
- коли курсор набуде вигляду подвійної стрілки, двічі клацнути лівою кнопкою мишки.

3.6.2. Сортування піктограм файлів у вікні відображення вмісту папки

Для того, щоб відсортувати піктограми за певним принципом, необхідно зайти в меню **Вид** → **Упорядочить значки** та клацанням лівої кнопки мишки на назві обрати один з наявних у списку способів сортування піктограм. По-іншому сортування можна провести таким чином:

- клацнути правою кнопкою мишки у вільному місті області відображення вмісту папки;
- зайшовши в підменю **Упорядочить значки** контекстного меню, яке з'явиться на екрані, обрати потрібний спосіб впорядкування.

Кількісний та якісний склад наявних способів сортування залежить від кількості стовпчиків, які відображаються в режимі **Таблиця** – це стосується всіх вікон папок, незалежно від того, який режим відображення вони мають.

3.6.3. Налаштування Рабочего стола та екрану

Рабочий стол Windows XP є спеціальним вікном папки, який завжди розтягнутий на весь екран та має специфічний вигляд та властивості. Клацнувши правою кнопкою мишки на вільному місці **Рабочего стола** та обравши у контекстному меню команду **Свойства**, ми запускаємо діалогове вікно **Свойства** для екрану, за допомогою якого можна задавати як **настройки екрану монітора** (наприклад, екранну розподільну здатність або частоту оновлення), **настройки вигляду самого Рабочего стола** (фоновий рисунок), так і **настройки загального інтерфейсу Windows XP** (тема та стиль оформлення елементів інтерфейсу – кнопок, меню, тексту).

Перша вкладка діалогового вікна **Свойства** для екрану носить назву **Тема** та призначена для попереднього перегляду та задання поточної теми оформлення **Windows XP**. За замовчуванням наявні тільки дві теми – класична, тобто така, в якій елементи інтерфейсу (рядок заголовку, кнопки, **Панель задач**) мають такий самий вигляд, як і в попередніх версіях **Windows**; **Windows XP** – тема, властива тільки **Windows XP**.

Наступна вкладка **Рабочий стол** дозволяє задати рисунок **Рабочего стола**. Його можна обрати зі списку **Фоновые рисунки** або, натиснувши на кнопку **Обзор...**, викликати однойменне діалогове вікно, в якому можна обрати будь-який з наявних на жорсткому диску графічних файлів форматів **bmp** та **jpeg** у якості фонового рисунка. Натиснувши на кнопку **Настройка рабочего стола...**, ми відкриваємо вікно **Элементы рабочего стола**, в якому можна задати, чи будуть такі стандартні елементи інтерфейсу, як піктограми **Мой компьютер** або **Сетевое окружение** відображатися на **Рабочем столе**, а також визначити піктограми цих об'єктів.

Наступна вкладка **Заставка** дозволяє задати заставку **Windows XP**, яка буде запускатися тоді, коли з системою певний час ніхто не працював. Потрібний для цього час можна встановити тут же (лічильник **Интервал**).

На вкладці **Оформление** можна задати кольорову схему інтерфейсу **Windows XP** (список **Цветовая схема**), стиль вікон та кнопок ОС (список **Окна и кнопки**), розмір тексту (список **Размер шрифта**), який зображено в рядках заголовків програм та в іменах файлів у вікнах папок. Натиснувши на кнопку **Дополнительно...**, ми запускаємо діалогове вікно **Дополнительное оформление** (рис. 3.30), в якому ми можемо задати колір та розмір елементів інтерфейсу окремо.

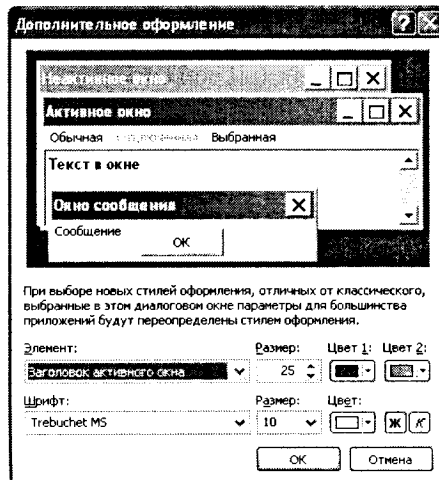


Рис. 3.30

На вкладці **Параметры** ми можемо задати наступні опції.


Екранна розподільна здатність (параметр **Разрешение экрана**) – кількість екранних пікселів по вертикалі та горизонталі. Потрібно пам'ятати, що оскільки при встановленні нової розподільної здатності фактичний розмір піктограм файлів та папок залишається незмінним, то при збільшенні роздільної здатності візуально елементи інтерфейсу будуть зменшуватися і навпаки.


Качество цветопередачи – кількість біт для кодування кольору одного пікселя. Чим опція **Качество цветопередачи** більша, тим більше можливих кольорів може відобразити монітор, але тим більше навантаження йде на оперативну та відеопам'ять.


Уваги заслуговує також опція частоти оновлення екрану, яку можна задати за допомогою списку **Частота обновления экрана** на вкладці **Монитор** діалогового вікна **Свойства монитора**, яке можна викликати, натиснувши на кнопку **Дополнительно...** на вкладці **Параметры** вікна **Свойства** екрану. Підвищення частоти оновлення екрану зменшує його мерехтіння, що є кращим для очей, тому необхідно завжди ставити найбільшу частоту оновлення серед тих, які підтримуються монітором.

3.6.4. Настройки в Панели управления

Центральним пультом ОС, акумулюючим всі основні та доступні для простого користувача настройки **Windows XP**, є **Панель управления**. У **Windows XP** користувач має можливість обрати з-поміж двох виглядів **Панели управления** – класичному, який є звичним для інших версій **Windows**, та за категоріями. Переключатись між ними можна за допомогою посилання в лівій частині вікна **Панели управления**. Послідовно розглянемо деякі з розділів **Панели управления** класичного виду.

Дата и время  – відкриває діалогове вікно **Свойства: Дата и время**, в якому можна встановити поточний час та часовий пояс, в якому знаходиться комп'ютер, а також задати сервер в Інтернеті з яким буде синхронізовуватись системний час на комп'ютері користувача.

Звуки и аудиоустройства  – відкриває діалогове вікно **Свойства: Звуки и аудиоустройства**, в якому можна встановити гучність звуку, задати стандартні звуки ОС (наприклад, звук, який лунає при виникненні критичної помилки), визначити пристрої, які будуть використовуватися для програвання звуку та запису.

Принтеры и факсы  – відкриває діалогове вікно **Принтеры и факсы** (рис. 3.31), в якому можна переглянути статус кожного з принтерів системи. Поряд з піктограмою кожного принтера вказується його

назва, нижче вказана кількість документів в черзі до друку та ще нижче – статус готовності. Чорне коло з білим прапорцем біля піктограми принтера вказує на те, що даний принтер є принтером за замовчуванням, тобто всі документи, які посилаються на друк без зміни поточного принтера, будуть посилатися саме на цей принтер. Для того, щоб змінити принтер за замовчуванням, необхідно клацнути на піктограмі принтера, який потрібно зробити принтером за замовчуванням, правою кнопкою мишки та у контекстному меню обрати команду **Использовать по умолчанию**.

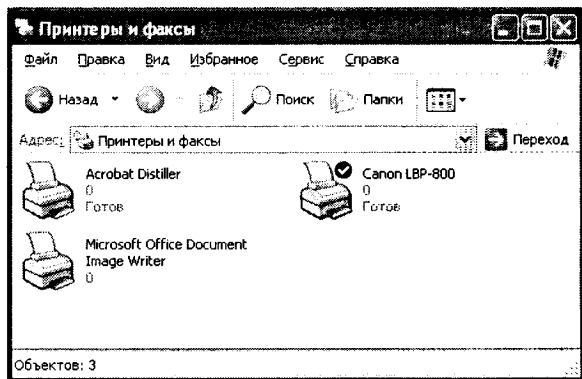



Рис. 3.31

Для встановлення нового принтера потрібно зайти в меню **Файл** та обрати команду **Установить принтер**, після чого запуститься майстер встановлення принтерів, слідуючи вказівкам якого, користувач зможе встановити новий принтер при наявності драйвера принтера – тобто програмного забезпечення, яке забезпечить роботу принтера в даній операційній системі. Для видалення принтера необхідно виділити його у вікні **Принтеры и факсы**, зайти в меню **Файл** та обрати команду **Удалить**. Після того, як користувач підтвердить свої наміри, принтер буде видалений з системи.

Для того, щоб тимчасово друкувати на принтері документи, користувач має змогу тимчасово призупинити друк, для чого необхідно виділити принтер і в меню **Файл** обрати команду **Приостановить печать**. Коли потрібно, обравши в тому ж меню команду **Возобновить печать**, можна продовжити друк на виділеному принтері.

Якщо користувач раптом передумав друкувати вже посланий на друк документ, то він має змогу відмінити друк, виділивши принтер та обравши в меню **Файл** команду **Очистить очередь печати**.

Система  – запускає вікно **Свойства системы**. Це також можна зробити, клацнувши правою кнопкою мишки на піктограмі **Мой компьютер** та в контекстному меню обрати команду **Свойства**.

Діалогове вікно **Свойства системы** має велику кількість вкладок, серед яких вкладка **Общие** має вигляд, як зображено на рис. 3.32. На цій вкладці користувач має змогу тільки отримувати інформацію, а не задавати параметри. Згорі наведена версія встановленої поточної ОС, нижче – ім'я користувача ОС, в самому низу – процесор, його тактова частота та кількість оперативної пам'яті, встановленої на комп'ютері.

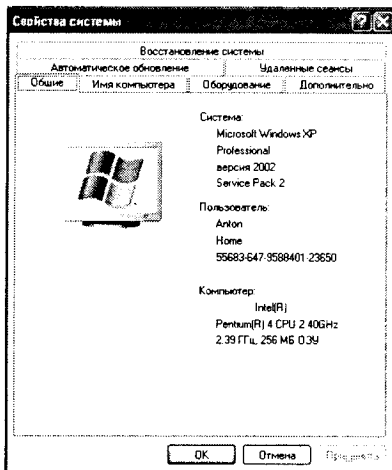


Рис. 3.32

З вкладки **Оборудование** ми отримуємо доступ до можливості запуску **Диспетчера устройств**, для чого необхідно просто натиснути на кнопку **Диспетчер устройств**. На екрані з'явиться вікно **Диспетчер устройств** (рис. 3.33), в якому наведені пристрої, що працюють в комп'ютері.

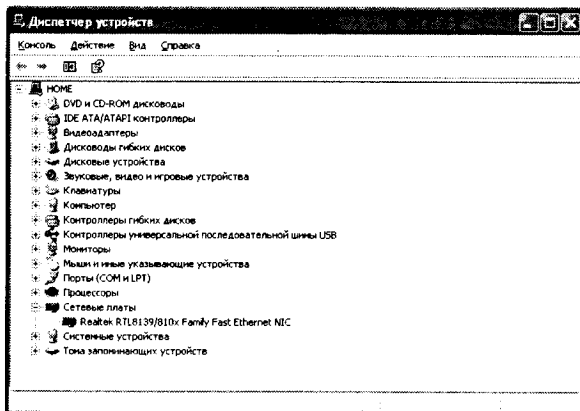


Рис. 3.33


Змінювати способи відображення пристроїв у вікні **Диспетчер устройств** можна за допомогою меню **Вид** цього вікна. Кожний тип пристроїв може бути розгорнутий шляхом клацання на позначку + біля назви типу, після чого у вікні з'являться всі пристрої даного типу, встановлені в комп'ютері.

Якщо пристрій позначений червоним хрестом, то це означає, що він відключений. Для того, щоб його увімкнути, необхідно клацнути на ньому правою кнопкою мишки та у контекстному меню обрати команду **Задійствовать**. Якщо пристрій необхідно, навпаки, тимчасово виключити, то потрібно знов клацнути на нього правою кнопкою мишки та виконати команду **Отключить**, після чого підтвердити свої дії.

Якщо певний пристрій, позначений знаком питання, та підписаний як **Неизвестное устройство**, то це означає, що для даного пристрою не були встановлені драйвери, тобто програмне забезпечення, за допомогою якого пристрій міг би працювати в даній операційній системі. Такий пристрій працювати не буде. Для того, щоб встановити драйвер для цього пристрою, можна скористатись установчим компакт-диском, який поставляється разом з самим пристроєм – тоді вікно **Диспетчер устройств** може не знадобитися. Якщо ж такого компакт-диска немає, а є тільки сам драйвер, який знаходиться на локальному жорсткому диску, то вікно **Диспетчер устройств** дозволяє встановити новий драйвер, для чого необхідно:

- виділити **Неизвестное устройство**;
- в меню **Действие** обрати команду **Обновить драйвер...**;
- запуститься майстер встановлення нового обладнання, слідуочи вказівкам якого, користувач зможе встановити драйвер потрібного пристрою.

Диспетчер устройств також має можливість аналізувати конфігурацію обладнання на предмет нових пристроїв. Для цього необхідно в меню **Действие** обрати команду **Обновить конфигурацию оборудования**. Якщо **Диспетчер устройств** знайде нове обладнання, то автоматично запуститься майстер встановлення нового обладнання.

Установка и удаление программ  – відкриває однойменне вікно, основне призначення якого – все таки видалення програм, для чого необхідно вибрати програму зі списку та натиснути кнопку **Удалить**, після чого запуститься програма, слідуочи вказівкам якої, користувач зможе видалити потрібний продукт.


Для встановлення (інсталяції) програмного устаткування як правило користуються програмою-інсталятором, яка постачається разом з програмним продуктом. Запускається ця програма як правило шляхом запуску вихідного файлу **Setup.exe** або **Install.exe**, після чого користувач у покроковому режимі задає параметри установки та починає копіювання файлів.

Учетные записи пользователей  – відкриває однойменне діалогове вікно, яке використовується для створення профілів користувачів. Існує

два основні типи облікових записів користувачів – адміністратор та обмежений запис. Адміністратор має право установки та видалення програм, внесення змін на рівні всієї системи, на зміну паролів будь-яких користувачів а також на їх додавання, видалення та зміну типу облікових записів. Користувач з обмеженим обліковим записом має право тільки на зміну паролю свого користувача та на зміну рисунка свого облікового запису.

Для того, щоб створити обліковий запис, необхідно:

1. У вікні **Учетные записи пользователей** обрати розділ **Создание учетной записи**.
2. Ввести в текстовому полі у вікні назву свого облікового запису.
3. Натиснути кнопку **Далее**.
4. Обираємо тип облікового запису.
5. Натискаємо кнопку **Создать учетную запись**.

Шрифты . Шрифт описує зовнішній вигляд символів, які виводяться на пристрої (пристроєм може бути як екран монітора, так і принтер). Оскільки кожен символ має певний код в системі **ASCII (American Standard Code for Information Exchange)**, то якщо ОС не знайде шрифт, який би коректно описував зовнішній вигляд символів, то вони будуть відображатися некоректно. В таких випадках, як відсутність потрібного шрифту, у **Windows XP** передбачена можливість встановлення додаткових шрифтів. Для цього потрібно виконати наступні дії.

1. Двічі клацнути на піктограмі папки **Шрифты** в **Панелі управління**, після чого вікно папки з'явиться на екрані (рис. 3.34).
2. Для того, щоб додати новий шрифт, потрібно мати його файл або на жорсткому диску, або на дискеті, або на компакт-диску.
3. Зайти у меню **Файл** та обрати команду **Установить шрифт....**
4. На екрані з'явиться вікно **Добавление шрифтов**, де необхідно вказати папку, в котрій знаходиться файл шрифту, які шрифти з тих, файли яких знаходяться в папці, потрібно встановити, та натиснути кнопку **OK**.

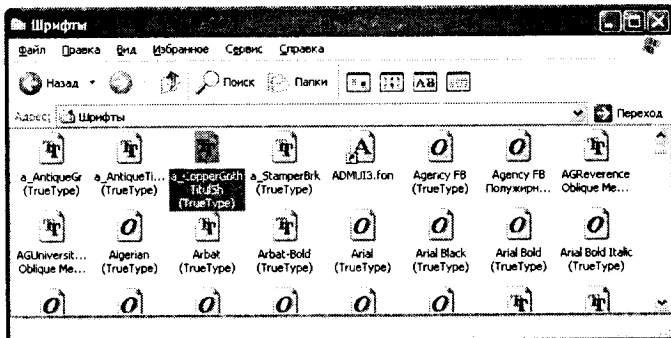



Рис. 3.34

Видаляти шрифти можна так само, як інші файли, а ось в режимах відображення вмісту папки **Шрифты** є відмінності. Серед режимів відображення вмісту папки **Шрифты** є наступні:

- Таблица;
- Список;
- Крупные значки;
- Группировать схожие шрифты.

Переключатись між цими режимами можна також, як і у випадку з іншими папками, через меню **Вид**.

Язык и региональные стандарты  – викликає однойменне діалогове вікно, в якому можна задати як регіональні стандарти (формат чисел, грошова одиниця, формат грошових сум, формат часу та дати), так і додати мову текстового вводу. Для того, щоб додати додаткову мову текстового вводу, необхідно виконати послідовність таких дій.

1. Зайти на вкладку **Языки**.
2. Натиснути на кнопку **Подробнее...**, після чого на екрані з'явиться діалогове вікно **Языки и службы текстового ввода** (рис. 3.35).
3. На вкладці **Параметры** необхідно натиснути на кнопку **Добавить...**
4. На екрані з'явиться вікно **Добавление языка ввода**, де необхідно вибрати потрібну мову та натиснути кнопку **ОК**.
5. Повернувшись до вікна **Языки и службы текстового ввода** та побачивши, що нова розкладка клавіатури з'явилася в списку **Установленные службы**, варто закріпити внесені зміни, натиснувши кнопку **Применить**. Щоб видалити розкладку, необхідно виділити відповідну службу у списку **Установленные службы** та натиснути на кнопку **Удалить**.

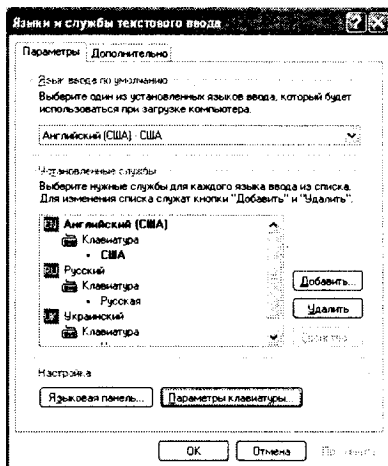


Рис. 3.35

Зміна розкладки клавіатури за замовчуванням

Розкладка клавіатури за замовчуванням – це розкладка, яка є активною відразу після завантаження ОС. В списку **Установленные службы** розкладка за замовчуванням відокремлюється жирним текстом (рис. 3.35). Для того, щоб змінити розкладку за замовчуванням необхідно просто вибрати її зі списку **Язык ввода по умолчанию**. Звісно, що вибір у цьому списку обмежений тільки встановленими текстовими службами.

Вибір комбінації клавіш для зміни мови введення тексту

Багато користувачів встановлюють комбінацію клавіш **Ctrl+Shift** для переключення між мовами введення тексту. Для того, щоб встановити іншу комбінацію клавіш, потрібно виконати наступну послідовність дій:

1. В діалоговому вікні **Языки и службы текстового ввода** натиснути на кнопку **Параметры клавиатуры...**
2. В діалоговому вікні **Дополнительные параметры клавиатуры**, яке з'явиться на екрані, необхідно натиснути кнопку **Смена сочетания клавиш...**
3. В діалоговому вікні **Смена сочетания клавиш** потрібно встановити нову комбінацію клавіш для переключення між мовами набору;
4. Послідовно виходити з відкритих вікон, натискаючи на кнопку **ОК** в кожному вікні.

3.7. Стандартне програмне забезпечення

Стандартне програмне забезпечення є сукупністю програм, які виконують як службові функції (наприклад, дефрагментація), так і функції набору тексту, перегляду графічних файлів (відео) та музичних файлів.

Набір стандартного програмного забезпечення у **Windows XP** порівняно з попередніми версіями практично не змінився. Як і в минулих версіях, у **Windows XP** наявні програми сканування диска та дефрагментації, текстовий редактор **Notepad** (Блокнот), який працює з простим текстом, текстовий редактор **WordPad**, який окрім звичайного текстового формату підтримує також і формат **RTF**, графічний редактор **Paint**, програвач відео та музичних файлів **Windows Media Player**, програми **Калькулятор**, **Проводник**, **Звукозапис** тощо. Як і в минулих версіях, доступ до стандартних програм можна отримати, обравши пункт головного меню **Стандартные** з розділу **Программы**, після чого обрати потрібну програму (рис. 3.36).

Серед нововведень у **Windows XP** можна відмітити скорочення можливостей програми **Сведения о системе**, деякі з яких перейняло на себе діалогове вікно **Управление компьютером** (наприклад, можливість редагування списку служб, які автоматично запускаються при завантаженні ОС), яке можна викликати, клацнувши правою кнопкою мишки на піктограмі **Мой компьютер** та у контекстному меню обравши команду **Управление**.

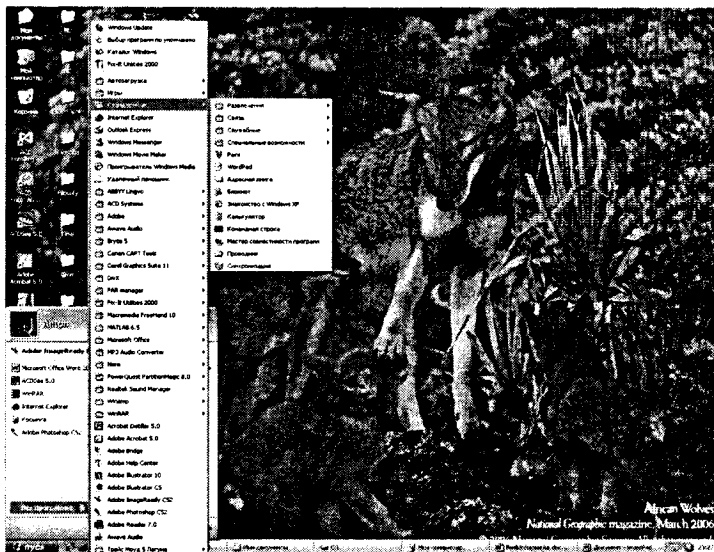


Рис. 3.36

Запитання до самоперевірки

1. Що називається операційною системою?
2. Які основні особливості **Windows XP**?
3. Які елементи інтерфейсу присутні на **Робочем столі**?
4. Що називається піктограмою?
5. Що називається контекстним меню?
6. Які типи вікон існують у **Windows**?
7. Що означають три крапки в кінці назви команди меню?
8. Які операції з вікнами існують та яким чином вони виконуються?
9. Які існують режими відображення **Головного меню**?
10. Що таке файлова система?
11. Які основні файлові системи підтримує **Windows XP**?
12. Які переваги та недоліки файлових систем **FAT, NTFS**?
13. Які основні операції можна виконувати над дисками?
14. Як створюють, копіюють, переміщують та вилучають файли і папки?
15. Які символи не можна вводити до імені файла або папки?
16. Що називається ярликом? Як створити ярлик до файла або папки?
17. Що називається буфером обміну?
18. Яким чином виконується пошук файлів за маскою?
19. Чим відрізняється параметр **Размер** від параметра **На диску** у вікні властивостей файла?
20. Які існують режими відображення вмісту папки?

4. ОПЕРАЦІЙНА СИСТЕМА LINUX

📖 План викладу матеріалу:

1. Можливості та переваги **Linux**.
2. Дистрибутиви **Linux**.
3. Перше знайомство з **Linux**.
4. Робота з файлами та каталогами в **Linux** через командний рядок.
5. Графічна оболонка **KDE**.

↔ Ключові терміни розділу

✓ Рядок підказки	✓ Історія команд
✓ <i>root</i>	✓ Завантажувальний запис
✓ Консоль	✓ Суперблок
✓ Віртуальний термінал	✓ Графічна оболонка KDE

4.1. Можливості та переваги **Linux**

Linux – операційна система, яка належить до сімейства **UNIX**-подібних систем та всі компоненти якої поширюються з ліцензією на вільне копіювання та встановлення для необмеженої кількості користувачів. **Linux** підтримує стандарти відкритих систем (**OSI – Open System Interface**) та протоколи мережі **Internet**, а також сумісна з операційними системами **DOS** та **MS Windows**. ОС **Linux** достатньо широко використовується окремими користувачами на індивідуальному рівні, а також на корпоративному рівні – для обробки даних у сфері фінансів, економіки, медицини, в учбових програмах з програмування та проектування операційних систем, які впроваджені в навчальних закладах. Серед переваг та можливостей ОС **Linux** відзначимо наступні:

- висока швидкодія та низькі вимоги до оперативної пам'яті комп'ютера – мінімально 4 Мб для роботи з командним рядком та 32 Мб при роботі з графічною оболонкою **KDE**;
- відсутність «зависань», які властиві ОС сімейства **Windows**;
- реальна багатозадачність – всі процеси незалежні та ресурси центрального процесора ефективно розподіляються між процесами, що відбуваються. Досягається за рахунок того, що ядро здійснює режим розподілу часу центрального процесора, по черзі виділяючи кожному процесу інтервали часу для виконання (на відміну від режиму багатозадачності **Windows**, де сам процес повинен поступитися в черзі іншим процесам);
- стійкість до впливів вірусів;

- комп'ютер під управлінням ОС **Linux** легко інтегрується до всесвітньої або локальної мережі;
- сторінкова організація пам'яті – системна пам'ять **Linux** організована у вигляді сторінок обсягом 4 Кб;
- збереження оперативної пам'яті на диск, що дозволяє ОС працювати при обмеженій оперативній пам'яті. Для цього вміст деяких сторінок оперативної пам'яті записується в область жорсткого диска, яка виконує роль додаткової оперативної пам'яті;
- дозволяє використовувати прикладні програми інших ОС – **Windows**, **DOS**. Для запуску програм **DOS** та **Windows 3.1/95** для **Linux** розроблені емулятори цих операційних систем;
- сумісне використання програм без використання оперативної пам'яті для кожної з запущених копій;
- використання частини оперативної пам'яті для збереження часто використовуваних даних з жорсткого диска;
- відповідність стандарту **POSIX 1003.1 (Portable Operating System Interface** – інтерфейс мобільної операційної системи);
- підтримка різних форматів файлових систем;
- робота на різних апаратних платформах;
- ядро **Linux** підтримує завантаження потрібних сторінок пам'яті, тобто з диска завантажуються тільки використовувані сегменти програми;
- графічний режим користувача (система **X Window**) та, відповідно, запуск програм, що мають графічний інтерфейс, під **Linux**;
- низька вартість дистрибутиву та вільне копіювання ОС;
- висока надійність **Linux**;
- доступність довідкової інформації по цій ОС;
- низькі системні вимоги, які наведені в таблиці 4.1.

Таблиця 4.1

Вимоги до апаратної частини комп'ютера під управлінням ОС Linux

Задачі, що виконуються в Linux	Вимоги до оперативної пам'яті	Вимоги до обсягу жорсткого диска
Робота в текстовому режимі з командного рядка	4 Мбайт	10 Мбайт
Робота в текстовому режимі через Midnight Commander	4 Мбайт	40 Мбайт
Для запуску графічного інтерфейсу X Window	16 Мбайт	-
Для запуску інтегрованого графічного середовища KDE	32 Мбайт	500 Мбайт
Для запуску кожного окремого великого додатку (графічного редактора, текстового процесора тощо)	34 Мбайт	550-600 Мбайт
Для роботи з офісним пакетом StarOffice	64 Мбайт	800 Мбайт

4.2. Дистрибутиви Linux

Існує велика кількість дистрибутивів Linux – **Debian**, **Red Hat**, **Caldera**, **Slackware** та **S.u.S.E.** Кожний з них має власну програму встановлення та утиліти супроводження, які спрощують адміністрування системи та інсталяцію. Кожний дистрибутив націлений на окрему аудиторію користувачів.

Серед зазначених дистрибутивів ОС для початківця можна порекомендувати дистрибутив **Red Hat**, тому що встановлення операційної системи **Red Hat Linux** є одним з найбільш легких процесів встановлення **Linux**. Наведемо основні відмінності **Red Hat Linux** від інших дистрибутивів ОС:

- формат програмних пакетів. Використовується **Red Hat Package Management (RPM)**, який призначений для компонування складових частин **Linux**. Головна перевага **RPM** полягає в достатньо легкому встановленні програмного забезпечення під **Linux**;
- легкість встановлення ОС – в процесі встановлення користувач має змогу встановити перші параметри системи – обрати потрібні програмні пакети, розбити диск на розділи, налаштувати графічний інтерфейс (шляхом настройки відеоадаптера та монітора), визначити облікові записи користувачів та параметри мережі;
- наявність сценаріїв виконання **UNIX System V** – використовується для організації процесу запуску та зупинки системних служб;
- графічні середовища та оточення **Рабочего стола (GNOME та KDE)** – **GNOME** є графічним інтерфейсом **Linux**, який встановлюється за замовчуванням разом з **Linux**. Наведені графічні середовища підтримують знайому по роботі з **Windows** технологію роботи з об'єктами **drag and drop** (див. розділ присвячений **Windows**), що робить більш легким перехід на **Linux** користувачів, які до цього працювали з ОС **Windows**.

Все таки при виборі дистрибутива потрібно бути уважним – бажання змінити дистрибутив, встановлений на комп'ютері, призведе то повної переінсталяції ОС **Linux**.

4.3. Перше знайомство з Linux

4.3.1. Запуск Linux

Після інсталяції ОС **Linux**, якщо на комп'ютері користувача не встановлено інші ОС, на екрані ненадовго з'явиться напис:

• **LILO boot:**

Якщо на комп'ютері встановлено декілька операційних систем, за необхідності вибору між ними потрібно при виникненні цього напису натиснути клавішу **Tab**, після чого на екрані з'явиться список пунктів,

кожен з яких відповідає різним операційним системам, встановленим на комп'ютері. У версії 21 меню вибору між ОС виникає автоматично. За відсутності дій користувача та відсутності автоматичного завантаження графічної оболонки на екрані з'явиться запрошення на введення імені користувача:

Login:

У відповідь на це запрошення необхідно ввести ім'я користувача або, якщо вхід у систему виконується вперше після встановлення **Linux**, ім'я користувача **root**, який є єдиним користувачем, для якого обов'язково створюється обліковий запис під час встановлення ОС. Потрібно зазначити, що входячи в систему під ім'ям **root**, користувач має необмежений доступ до всіх ресурсів системи, тому діяти в сеансі цього користувача потрібно обережно – проте перший вхід у систему потрібно робити саме в цьому сеансі, тому необхідно ввести ім'я **root** та натиснути клавішу **Enter**. На екрані з'явиться запрошення на введення пароля:

Password:

У відповідь на це запрошення необхідно ввести пароль для того користувача, ім'я якого було введене у відповідь на попереднє запрошення. Отже необхідно ввести пароль, який був заданий для користувача **root** при інсталяції, та натиснути клавішу **Enter**. Після правильного введення пароля користувач може безпосередньо працювати з командним рядком, вводячи туди команди з клавіатури. Якщо ж пароль був введений невірною, то на екрані з'явиться такий напис:

Login incorrect:

Після правильного введення пароля на екрані з'явиться наступний напис:

```
[root@localhost /root]#
```

або

```
/home/Tony#
```

Перший приклад містить ім'я користувача **root** та шлях до поточного каталога **localhost /root**; другий приклад містить тільки шлях до поточного каталога **localhost /root**. Цей рядок носить назву підказки (рядок підказки) та поява його означає, що програма-оболонка **Shell** готова сприймати команди, які вводять з клавіатури користувач.

Загалом, для введення команди потрібно ввести її з клавіатури та натиснути клавішу **Enter**. Потрібно пам'ятати, що при введенні команд потрібно враховувати регістр символів.

4.3.2. Додавання користувача

Відразу після входу в **Shell** потрібно додати нового користувача. Для цього в командному рядку необхідно ввести команду:

```
[root]# useradd anton
```

Тут **root** – поточний користувач (тобто користувач, під ім'ям якого йде робота); **useradd** – назва команди; **anton** – параметр команди, тобто ім'я нового користувача. Результатом виконання даної команди буде поява в ОС нового користувача, що має ім'я **anton** та створення в каталозі **home** домашнього каталога користувача **anton** (*домашній каталог* користувача – це каталог, до якого даний користувач має всі права), проте доки для цього користувача не був заданий пароль, працювати в його сеансі ще не можна. Для додавання паролю для користувача **anton** необхідно ввести рядок

```
[root]# passwd anton
```

після чого на екрані з'явиться рядок

```
New UNIX password:
```

який є запрошенням на введення паролю для користувача **anton**. Необхідно ввести пароль, після чого система запропонує ввести цей пароль знову:

```
Retype new UNIX password:
```

При правильному введенні копії пароля на екрані з'явиться повідомлення:

```
passwd: all authentication tokens updated successfully
```

Після цього користувач вже має змогу знаходитись під сеансом користувача, відмінного від **root**. Вкажемо на відмінність вигляду запрошення в сеансі **root** від інших – в **root** запрошення закінчується символом **#**, а в інших **\$**.

У випадку, коли користувач введе невірну команду в командному рядку, на екрані з'явиться наступний напис:

```
(ім'я команди): command not found
```

Іншими простими командами в **Linux** є:

whoami	Повертає ім'я, під яким користувач ввійшов в систему
who	Повертає імена користувачів, які працюють в системі на даний момент
pwd	Повертає ім'я поточного каталога
ls -l	Повертає список підкаталогів та файлів поточного каталога
cd (назва каталога)	Змінює поточний каталог
ps ax	Повертає перелік процесів, які виконуються на даний момент
man (назва команди)	Виводить довідкову інформацію по команді

4.3.3. Консоль та віртуальні термінали

➤ **ОС Linux** дозволяє підключати до комп'ютера віртуальні термінали (сеанси роботи) та забезпечує можливість роботи з декількома віртуальними терміналами з однієї консолі (*консоль* – монітор та клавіатура, які фізично підключені до комп'ютера). За замовчуванням **Linux (Red Hat)**

відкриває шість віртуальних терміналів, між якими можна переключатися за допомогою комбінацій клавіш **Ctrl+Alt+F1 – Ctrl+Alt+F6** (в графічному режимі) та **Alt+F1 – Alt+F6** в текстовому режимі. Для того, щоб дізнатись, в якому терміналі зараз працює користувач, потрібно ввести команду

tty:

яка виводить назву терміналу в наступному форматі:

/dev/tty2

Для завершення сеансу роботи з системою в одному з терміналів необхідно натиснути комбінацію клавіш **Ctrl+D**. Те саме можна зробити, ввівши або команду **logout** , або команду **exit**.

Якщо необхідно для виконання певних дій швидко перейти до сеансу **root** (без перезавантаження комп'ютера), то це можна зробити за допомогою команди **su**. Після введення цієї команди поточна оболонка запустить ще один свій екземпляр, але вже з правами користувача **root**.

4.3.4. Робота з командним рядком

Робота з командним рядком – це не тільки наявна можливість введення команд, а також переміщення по введених символах, переміщення по словах, історії команд, а також редагування вмісту командного рядка. Нижче наведений перелік команд, які використовуються при роботі з командним рядком оболонки та належать до оболонки **GNU bash 1.14.7(1)**.

Таблиця 4.2

Команди для роботи з командним рядком

Комбінація клавіш	Дії
1	2
→ або Ctrl+F	Переміщення праворуч по командному рядку по вже набраних символах
← або Ctrl+B	Переміщення на один символ ліворуч
Esc+F	Переміщення на одне слово праворуч
Esc+B	Переміщення на одне слово ліворуч
Home або Ctrl+A	Переміщення на початок набраного рядка символів
End або Ctrl+E	Переміщення на початок/кінець рядка символів
Del або Ctrl+D	Видалення символу, на який показує курсор
Backspace	Видалення символу з позиції, яка передує курсору
Ctrl+K	Видалення правої частини рядка, починаючи з символу, на який вказує курсор
Ctrl+U	Видалення лівої частини рядка, включаючи символ, який знаходиться в кінці від курсора

1	2
Enter або Ctrl+M	Запуск команди, яка введена в командному рядку
Ctrl+L	Очищення екрану та переміщення поточної команди в верхню частину екрану
Ctrl+T	Обмін місцями двох символів: символу на який показує курсор та символу ліворуч від курсору, після чого курсор потрібно перевести на один символ праворуч
Esc+T	Обмін місцями двох слів: те, на яке вказує курсор, та те, яке знаходиться ліворуч від нього
Ctrl+K	Вирізання частини рядка від поточної позиції курсора до кінця рядка (потім вирізану частину можна вставити з буфера обміну в інше місце)
Esc+D	Вирізання частини рядка від поточної позиції курсора до кінця поточного слова
Esc+Del	Вирізання частини рядка від поточної позиції курсора до початку поточного слова (слова, на якому знаходиться курсор)
Ctrl+W	Вирізання частини рядка від поточної позиції курсора до попереднього пробілу
Ctrl+Y	Вставка вирізаного тексту з буфера обміну
Esc+C	Зміна символу, на який вказує курсор, на великий
Esc+U	Перетворення символів поточного слова у великі, починаючи з символу, на який вказує курсор
Esc+L	Перетворення символів, починаючи з позиції, на яку вказує курсор, до кінця поточного слова на маленькі.
Shift+PgUp Shift+PgDown	Перегляд сторінок екранного виведення (використовуються тоді, коли результатом виконання команди є великий масив символів інформації)
Ctrl+C	Призупинення виконання команди, яка в даний момент виконується

В Linux є можливість більш зручного введення команд до командного рядка. Тобто користувач може ввести частину команди, а після натискання клавіші **Tab** система виведе перелік команд, які мають введену користувачем частину. Звісно, кількість команд у списку залежить від кількості символів, які для початку ввів користувач. Якщо клавішу **Tab** натиснути тоді, коли в командному рядку немає жодного символу, то на екрані з'явиться попередження:

There are 1217 possibilities. Do you really wish to see them all? (y or n)

Це означає, що система попереджає користувача про те, що виведено буде 1217 команд, що не є прийнятним для користувача.

ОС Linux зберігає *історію раніше введених команд* – за замовчуванням кількість команд, які запам'ятовуються, дорівнює 1000, що є цілком достатнім. Історія команд зберігається в файлі та користувач має змогу обирати певну команду з раніше або пізніше введених. В таблиці 4.3 наведений перелік комбінацій клавіш, які використовуються при роботі з історією команд.

Таблиця 4.3

Комбінації клавіш при роботі з історією команд

Комбінація клавіш	Дії
Ctrl+O	Аналогічне натисканню Enter, після чого відображається чергова команда з історії команд
↑ або Ctrl+P	Перехід до попередньої команди
↓ або Ctrl+n	Перехід до наступної команди
!, N	Виконується n-а команда зі списку історії команд
!, -, N	Виконується n-а команда з кінця списку
!, (рядок символів)	Виконується команда, назва якої починається з рядка символів
PgUp	Перехід до першої команди зі збережених в списку історії команд

4.3.5. Вихід з системи

Для виходу з системи простіше за все можна скористатись командою `exit`, ввівши її в командний рядок.

```
/home/anton# exit
```

4.4. Робота з файлами та каталогами в Linux через командний рядок

Необхідність роботи з файлами, що зберігаються на жорсткому диску, а також наявність файлів та каталогів робить необхідним визначення типів файлових систем, які підтримуються Linux та методів роботи з файлами та каталогами.

4.4.1. Файлові системи, які підтримуються Linux

Нижче наведемо перелік файлових систем, з якими може працювати Linux (таблиця 4.4).

Таблиця 4.4

Типи файлових систем, які підтримуються Linux

Тип	Коментар
extfs	Рання версія файлової системи для Linux, яка тепер не використовується.
ext2fs	Основна файлова система для Linux
minix	Перша файлова система Linux, яка зараз використовується рідко
xiaf	Подібна до ext2fs, проте рідко використовується через меншу кількість можливостей
msdos	Використовується для розділів, відформатованих в MS-DOS та Windows, тобто це FAT
umsdos	Використовується для встановлення ОС Linux в розділі DOS
swap	Розділ для swap-файла ОС Linux
proc	Використовується для звернення до структур даних ядра. Файли цієї файлової системи не займають дисковий простір
ReiserFS	Достатньо нова файлова система, яка використовує так звані збалансовані дерева для збереження всіх об'єктів файлової системи
sysv	Файлова система Unix System V
iso9660	Використовується для встановлення CD-ROM, що відповідає стандарту ISO 9660
vfat	Файлова система FAT-32
smb	Мережева файлова система, яка використовується у Windows, Windows NT та Lan Manager.
ncpfs	Мережева файлова система, яка застосовується в Novell NetWare.
hpfs	Застосовується для розділів OS/2. В Linux забезпечується тільки читання з розділів hpfs
nfs	Використовується для доступу до файлів систем NFS та NFS+ (MacOS)

Як зазначено в таблиці 4.4, основною файловою системою Linux є **ext2fs**. В цій файловій системі (як і в будь-якій іншій) створюється завантажувальна область, яка містить *завантажувальний запис* – фрагмент коду для ініціювання процесу завантаження операційної системи. Інший дисковий простір поділяється на блоки, які можуть мати розмір 1, 2 або 4 Кб та є одиницями дискового простору, що адресується. Блоки в свою чергу об'єднуються в групи блоків, перший елемент яких – *суперблок*, однаковий для всіх груп та використовується для організації доступу до всіх даних на диску. При завантаженні ОС суперблок завантажується до пам'яті, та всі зміни файлової системи спочатку впливають на копію суперблоку та записуються на диск тільки періодично через певний проміжок часу. Це дозволяє значно підвищити продуктивність системи, а також вимагає при вимиканні системи обов'язкового запису вмісту копії суперблоку з оперативної пам'яті на жорсткий диск. Це означає, що

не можна вимикати комп'ютер простим вимкненням живлення, тому що після його наступного завантаження інформація, яка записана в суперблоці, не буде відповідати реальному стану файлової системи.

4.4.2. Каталоги Linux

Каталог **Linux** – це теж саме, що й папка у **Windows**. Як і в **Windows** існують спеціальні папки для збереження файлів певного типу або певної категорії (наприклад, папка **Program Files** – для збереження папок та файлів програм, що встановлені в системі, папка **Windows** для файлів самої системи, папка **Temp** – для збереження тимчасових файлів тощо), так і в **Linux** існує навіть більша кількість каталогів спеціального призначення. Проте в організації каталогів в **Linux** є суттєва відмінність – для всіх логічних розділів жорсткого диска є лише одна структура дерева папок та один кореневий каталог, який позначається символом **/**. Перед тим, як розглядати операції з файлами та папками, необхідно саме визначити призначення деяких каталогів **Linux**.

Після встановлення системи **Linux**, входу до оболонки та введення вже згадуваної нами команди **cd** з параметром **/** (вхід у кореневий каталог) з'являється можливість продивитись перелік об'єктів, що знаходяться в кореновому каталозі. Можливо, у списку користувач може побачити каталоги **bin**, **dev**, **etc**, **home**, **install**, **lib**, **mnt**, **proc**, **root**, **tmp**, **user**, **usr** та **var**, проте перелік може бути іншим та відрізнитись від типу дистрибутиву ОС **Linux**. Коротко розглянемо призначення основних каталогів.

/bin – в даному каталозі знаходяться важливі системні програми, зокрема виконавчі файли програм, що відповідають командам (наприклад, ввівши команду **ls** в командному рядку, ми запускаємо програму **ls**, файл якої знаходиться у каталозі **bin**. Якщо праворуч від імені файла знаходиться зірочка *****, то це означає, що цей файл є виконавчим.

/dev – в даному каталозі знаходяться файли драйверів пристроїв.

/etc – містить системні конфігураційні файли (наприклад, **passwd** – файл паролів).

/sbin – використовується для зберігання системних двійкових файлів, які використовує системний адміністратор.

/home – містить каталоги облікових записів користувачів. Відразу після встановлення системи цей каталог зазвичай порожній через відсутність інших облікових записів, окрім **root**;

/lib – містить образи виконавчих бібліотек (**shared library images**). Ці файли містять код, який можуть використовувати інші програми.

/tmp – каталог тимчасових файлів.

/usr – містить програми та файли конфігурації, які використовує система. Тут в підкаталогах знаходяться найбільш важливі та корисні програми.

Рекомендовано виділяти для цього каталога окремий розділ диска – такий розділ роблять доступним тільки для читання та розташовують в ньому спільні конфігураційні та виконавчі файли.

/usr/bin – містить багато програм, які відсутні в каталозі **/bin**.

/usr/etc – також містить системні програми та конфігураційні файли.

/usr/src – містить вихідні коди для різних програм **Linux**, у тому числі й ядра **Linux**.

/var – містить файли, в яких зберігаються різні змінні дані, які визначають конфігурацію декотрих програм при наступному завантаженні або тимчасову інформацію, яка буде використана пізніше в ході поточного сеансу.

/var/backups – використовується для збереження резервної копії важливих системних файлів.

/var/lock – містить управляючі файли системи, які використовуються для резервування використання тих чи інших ресурсів системи.

/var/tmp – містить тимчасові файли.

/root – домашній каталог користувача **root**.

/boot – містить основні файли, необхідні для завантаження **Linux**.
Файли з цього каталога потрібні тільки при завантаженні системи.

Ми розглянули не всі каталоги, а тільки основні з них. Файли в цих каталогах, як правило, є важливими для правильної роботи системи – видалення їх небажане. Окрім конкретних каталогів, в **Linux** є також загальні типи каталогів. Ми вже розглянули як позначається кореневий каталог, але окрім нього є ще батьківський каталог, який можна визначити як каталог, що містить даний каталог, відносно якого ми визначаємо, є він батьківським чи ні. Наприклад, шлях до каталога **anton** – облікового запису користувача **anton** буде виглядати так:

/home/anton

В цьому прикладі каталог **home** є батьківським для каталога **anton**, тобто містить сам каталог **anton**. Звичайно, що коли мова йде про батьківські каталоги та підкаталоги, виникає питання, яким чином в ОС **Linux** можна проводити навігацію по каталогах, тобто перехід до каталога, відкриття каталога, перехід до кореневого каталога. Для виконання цих дій існують наступні команди.

1. Для зміни поточного каталога:

cd (назва каталога)

Потрібно зазначити, що просто назву каталога можна вказувати, якщо цей каталог знаходиться безпосередньо у поточному каталозі. В протилежному випадку потрібно вказати повний шлях до цього каталога, наприклад:

> **/var# cd /var/spool/news**

Часто виникає потреба перейти відразу в кореневий каталог, для чого виклик команди **cd** приймає вигляд

/var/spool/news# cd.. ,

а ввівши **cd** з такими параметрами:

```
cd./catalog ,
```

користувач підніметься на один рівень вгору, а потім перейде до підкаталога **catalog**.

В **Linux** можна швидко звертатись до домашнього каталога користувача за допомогою символа **~**. Якщо поточним користувачем є, наприклад, **anton**, то наступна команда:

```
/home/anton# more ~docs
```

еквівалентна команді

```
/home/anton# more /home/anton/docs
```

2. Для перегляду вмісту каталога використовується команда

ls повний або відносний шлях до каталога ,

а якщо команду **ls** ввести без параметрів, то на екран буде виведено вміст поточного каталога.

4.4.3. Види файлів **Linux**

В **Linux** під словом «файл» розуміються не тільки звичайні файли (програм, графічних зображень тощо) – файлами в **Linux** є також:

- файли фізичних пристроїв;
- іменовані канали (**named pipes**);
- гнізда (**sockets**);
- символічні посилання (**symlinks**).

Всі пристрої, які підключаються до комп'ютера, з позиції **Linux** є файлами. Фізичні пристрої бувають двох типів: символічними та блочними, між якими існує різниця, яка полягає в різних способах запису та зчитування інформації з пристрою. У першому випадку передача інформації здійснюється потоками байт, а в другому – блоками. Для взаємодії з фізичними пристроями **Linux**, як і **Windows**, використовує драйвери пристроїв, які для взаємодії з іншими частинами ОС створюють комунікаційний інтерфейс у вигляді файла. Більшість таких файлів знаходяться в каталозі **/dev**.

Ще один тип файлів – це іменовані канали, які є аналогом буфера обміну **Windows**, проте на відміну від останнього, де буфер обміну є частиною оперативної пам'яті і тільки в деяких випадках може частково зберігатися на жорсткому диску, в **Linux** іменовані канали зберігаються на диску. Як і буфер обміну у **Windows**, іменовані канали використовуються для обміну інформацією між додатками (процесами) **Linux**. Якщо два процеси, один з яких поміщає інформацію в канал, а інший зчитує, належать до одного батьківського процесу, то канал може бути неіменованим. В інших випадках необхідно створити іменований канал за допомогою команди:

```
mkfifo
```

Іншим типом файлів є гнізда, які є з'єднаннями між процесами, що дозволяють цим процесам взаємодіяти незалежно від інших процесів, які на даний момент мають місце.

Символічні посилання – це фактично ще одне ім'я для вихідного файла. Символічні посилання можуть вказувати на файли, які розташовані на іншому розділі жорсткого диска або на іншому комп'ютері. Якщо вихідний файл був видалений, символічне посилання стає непотрібним, тому що воно посилається на відсутній об'єкт. Для того, щоб створити символічне посилання, необхідно скористатись командою **ln**:

ln -s (ім'я файла чи каталога) (ім'я посилання)

Потрібно сказати, що коли користувач вніс зміни, відкривши файл безпосередньо, він ці зміни побачить також, якщо відкриє даний файл, звернувшись до нього за ім'ям-посиланням.

4.4.4. Операції з файлами

Як завжди, під операціями з файлами та каталогами маються на увазі операції створення, перегляду, копіювання, переміщення, видалення та перейменування. Нижче ми розглянемо, як виконувати ці операції за допомогою команд, які вводяться в командний рядок.

Створення каталогів виконується за допомогою команди **mkdir**:

/home/anton# mkdir rrr

В наведеному прикладі в каталозі **anton**, який знаходиться в каталозі **home**, створюється каталог з назвою **rrr**.

Копіювання файлів виконується за допомогою команди **cp**:

/home/anton/ua# cp termcap newdir

В даному прикладі команда **cp** була використана для копіювання файла **termcap** з поточного каталога в каталог **newdir**.

Переміщення файлів виконується за допомогою команди **mv**:

/home/anton/ua# mv termcap dvgig

У даному прикладі команда **mv** була використана для перейменування файла **termcap** з поточного каталога у файл **dvgig** – звісно, команду **mv** можна використовувати й при звичайному переміщенні файлів.

Для видалення файлів та каталогів використовується команда **rm**:

/home/anton/ua# rm dvgig

Подібною до команди **rm** є також команда **rmdir**, яка видаляє тільки каталоги, але каталоги, які є порожніми. Також потрібно зазначити, що команди **cp**, **mv**, **rm** виконуються, не питаючи підтвердження користувача, тому користуватись ними потрібно досить обережно.

Для перегляду файлів існують команди **more**, яка видає вміст файла «поекранно», та **cat**, яка виводить вміст всього файла відразу. Прикладом виконання команди **more** може бути наступний рядок:

/home/anton/ua# more tm

Якщо використовується команда **more**, то для переходу до наступної частини файлу необхідно натиснути клавішу **Space**.

Таким чином ми розглянули, як можна проводити основні операції з файлами та каталогами за допомогою команд **Linux**. Потрібно зазначити, що для користувача **Windows**, який вирішив перейти на **Linux**, можливо кращим виходом є користування графічними оболонками, які надають більш наочні та зручні способи проведення операцій з об'єктами файлової системи, ніж програма **Shell**. Проте про графічні оболонки мова піде далі.

4.4.5. Символи підстановки

Як і у **Windows** при заданні маски, **Linux** надає можливість звертатись до певної множини файлів за маскою, використовуючи символи підстановки. Існує два символи підстановки: ***** – означає будь-яку кількість будь-яких символів; **?** – означає один будь-який символ. Так, наприклад, команда

```
/home/anton/ua# rm *a
```

видаляє всі файли в поточному каталозі, які закінчуються на **a**, а команда

```
/home/anton/ua# rm ??a
```

видаляє всі файли, які закінчуються на **a** та мають у своєму складі три символи (два знаки **?** та сама літера **a**).

4.5. Графічна оболонка KDE

Linux – це консольна система, робота в якій здійснюється за допомогою введення команд в командному рядку, проте орієнтація на більш широке коло користувачів вимагає не тільки вищезазначених особливостей системи, а й організації графічного інтерфейсу, який є сукупністю зручних та наочних графічних засобів взаємодії користувача з системою без використання консолі. Графічний інтерфейс забезпечує з одного боку дружність робочого інтерфейсу, а з іншого – більш легкий перехід на **Linux** користувачів **Windows**. Неможливість безпосередньої сумісності як інтерфейсу (в **Linux** це спочатку тільки консоль, а в **Windows** – це повноцінний графічний інтерфейс), так і форматів файлів (з консолі **Linux** не можна працювати з такими популярними форматами, як **doc**), стримує популярність ОС **Linux**. Змінити становище на краще допомогло створення графічних оболонок, які забезпечують дружній для користувача інтерфейс, подібний до інтерфейсу **Windows**, а також дозволяють використовувати програмне забезпечення, яке виконує ті ж функції, що й популярне програмне забезпечення у **Windows**.

Було розроблено достатню кількість графічних оболонок для **Linux**, серед яких можна відзначити **Gnome** або **KDE**. Остання – це інтегрована графічна оболонка для **Linux**, яка в теперішній час включає в себе більш ніж 100 графічних додатків та підтримує 40 різноманітних мов (у тому числі й українську). **KDE** дозволяє сумістити зручність роботи в ОС сімейства **Windows** з надійністю ОС **Linux**. Далі піде мова саме про графічну оболонку **KDE 3.2**.

4.5.1. Відразу після запуску

Відразу після повного завантаження графічної оболонки на екрані з'явиться робочий стіл (**desktop**) – елемент, подібний до такого ж елемента середовища **Windows**. Внизу екрана знаходиться панель, за допомогою якої можна переключатися між вікнами програм, які є відкритими, отримувати доступ до панелі швидкого запуску, головного меню та переключатися між робочими столами **Linux**. Загалом, одночасно в графічній оболонці можуть бути активовані до 16 робочих столів, переключатися між якими можна за допомогою кнопок, які знаходяться на панелі внизу екрана (рис. 4.1).



Рис. 4.1

Окрім кнопок переключення робочих столів, на панелі внизу екрана є область, в якій відображаються поточний, встановлений у системі, час, поточна дата та календар, а також останньою праворуч є кнопка, натискання на яку згортає панель з екрана (рис. 4.2).



Рис. 4.2

На панелі відображаються кнопки вікон програм, які в даний момент відкриті в даному робочому столі. В крайній правій частині знаходиться панель швидкого запуску та кнопка з зображенням зірки (рис. 4.3), натискання на яку призводить до відкриття головного меню, про яке мова піде пізніше.



Рис. 4.3

Вся інша частина екрана є безпосередньо робочим столом, на якому розташовуються деякі піктограми. Потрібно пам'ятати, що для кожного користувача **Linux** існує свій індивідуальний робочий стіл. Повний вигляд екрану оболонки **KDE** без відкритих вікон програм поданий на рис. 4.4.

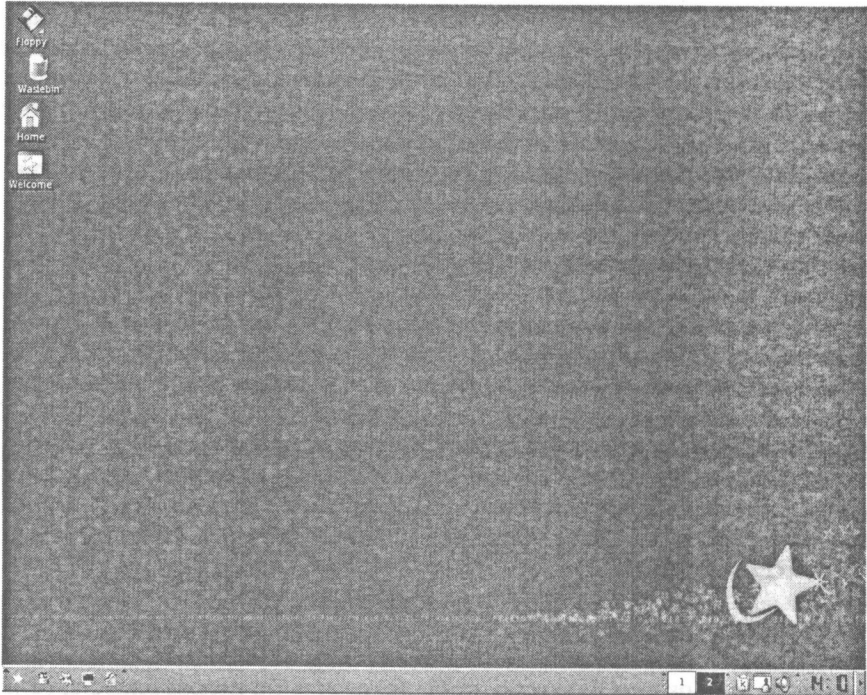


Рис. 4.4

4.5.2. Головне меню

Головне меню **Linux** є близьким аналогом меню **Пуск Windows** та складається з трьох областей (рис. 4.5):

- верхня область – **Most used Applications** – відображає значки програм, які часто або нещодавно запускалися;
- середня область – **All Applications** – відображає значки різноманітних програм, поділених за функціональною приналежністю на наступні групи підменю:
 - **Office** – офісні програми;
 - **Internet** – програми для **Internet**;
 - **Multimedia** – програми для роботи з мультимедійними засобами;
 - **System** – програми для роботи з системою;
 - **More applications** – інші програми;
 - **Home** – домашній каталог поточного користувача.
- нижня область – **Actions** – містить підменю або піктограми команд для виконання певних дій.

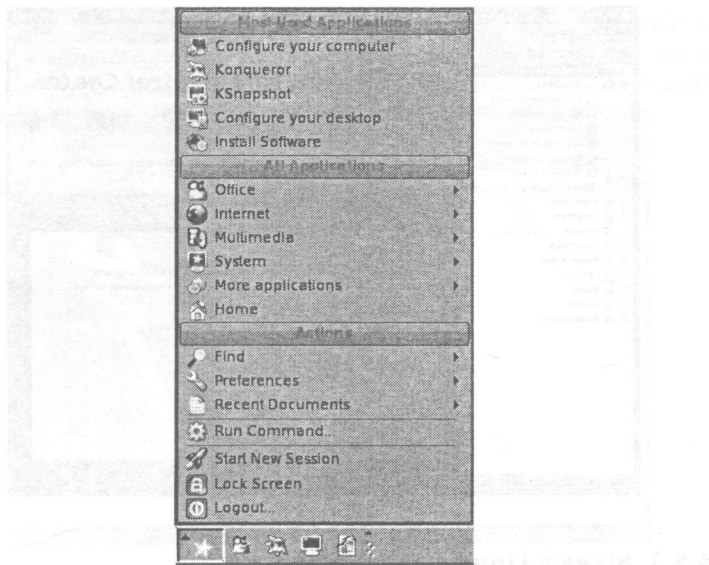


Рис. 4.5

Як і в **Windows**, в головному меню KDE праворуч деяких команд знаходиться чорний трикутник – нагадаємо, що це означає, що після вибору цієї команди з'явиться підменю (рис. 4.6). Також як і у **Windows**, три крапки праворуч назви команди означають те, що безпосередньо команда не виконує дій, а тільки викликає вікно, в якому користувачу пропонується задати параметри команди.

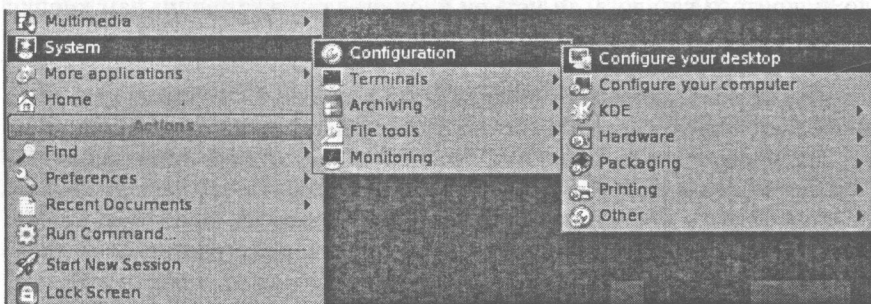


Рис. 4.6

На рис. 4.6 в кінцевому підменю, яке відкрилося, можна обрати команду **Configure your desktop**, після чого відкриється центр управління KDE (**KDE Control Centre**) (рис. 4.7), в якому можна змінити параметри системи.

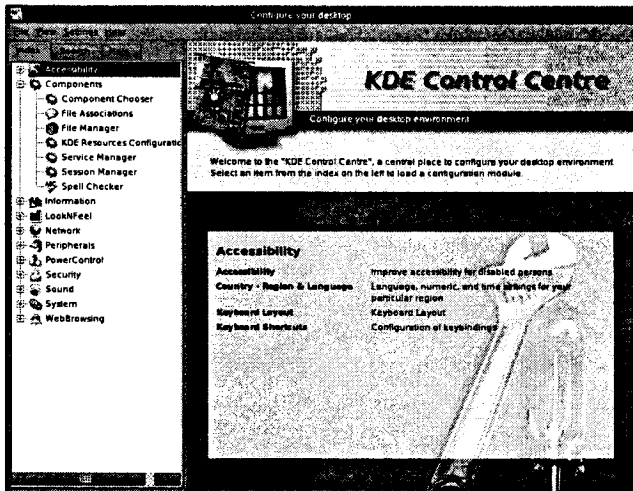


Рис. 4.7

4.5.3. Вікна в Linux

Вікна в **Linux** відіграють таку ж роль, як і в **Windows** – слугують для діалогу користувача з програмним забезпеченням. Оскільки охопити інтерфейс вікон навіть основних **Linux**-програм важко, то обмежимося розглядом структури стандартного вікна каталога **Linux** в оболонці **KDE**.

Стандартне вікно каталога **Linux** має вигляд, як зображено на рис. 4.8, та має наступні основні елементи.

Згори розташовується рядок заголовка, який відображає повний шлях до відкритого каталога, та чотири кнопки – три вже звичні нам кнопки керування розміром вікна: згортання, закриття та розгортання та одну кнопку для отримання допомоги з вікна папки.

Нижче розташовується рядок меню, який, як і у **Windows**, містить меню, які, в свою чергу, містять згруповані певним чином команди.

Під рядком меню присутній **Main Toolbar**, який є **Linux**-аналогом **Windows**-панелі інструментів **Обычные кнопки**. На цій панелі зокрема розташовуються кнопки навігації по дереву каталогів, піктограми кнопок копіювання, вирізання в буфер та вставки з нього, піктограми кнопок збільшення масштабу перегляду та зміни режиму відображення вмісту каталога.

Нижче **Main Toolbar** розташовується рядок адреси, аналогічний **Адресной строке** з **Windows**.

Центральну область вікна каталога займає область відображення вмісту папки, яка, на відміну від **Windows**, має тільки два можливі режими відображення: **Tree view**; **Icon view**.

В нижній частині вікна каталога **Linux** присутній рядок статусу.

Якщо весь вміст відкритого каталога не вміщується в області відображення вмісту каталога, то відображаються горизонтальна та вертикальна смуги прокрутки, користуватись якими потрібно аналогічно **Windows**.

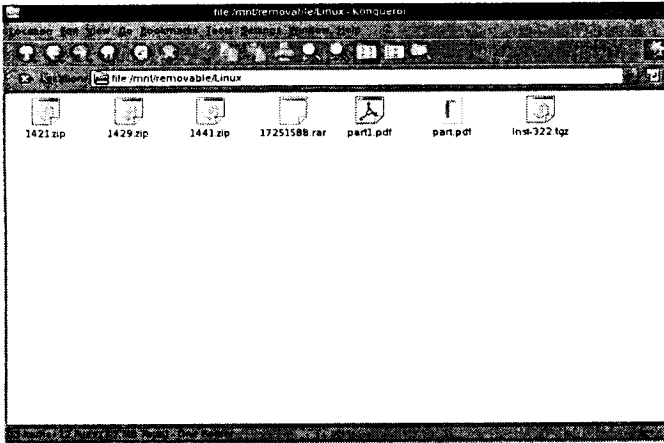


Рис. 4.8

Пересування вікна каталога **Linux** в оболонці **KDE** здійснюється точно так, як це виконується у **Windows** – тобто за методом **drag and drop**. Крім того, при натиснутій клавіші **Alt** можна перетягувати вікно програми не тільки «тримаючи» його за рядок заголовку, а й за будь яку частину вікна. Зміна геометричних розмірів вікна, як горизонтальних так і вертикальних, виконується також аналогічно з **Windows**.

Для впорядкування вікон на робочому столі необхідно клацнути правою кнопкою мишки на вільному місці робочого столу, обрати у контекстному меню команду **Windows** та в підменю, яке з'явиться на екрані, обрати одну з двох команд – **Cascade windows** або **Uncluter windows**.

Зміна режиму відображення вмісту каталога

Як вже зазначалося, в **KDE** існує два режими відображення вмісту каталога – **Tree view** та **Icon view**, переключатися між якими можна за допомогою двох кнопок на панелі **Main Toolbar** (рис. 4.9).



Рис. 4.9

Вигляд вмісту каталога в режимі **Icon view** був зображений на рис. 4.8. В цьому режимі відображаються піктограми файлів разом з їх іменами, мініатюри декількох файлів. Збільшувати розмір піктограм можна за допомогою кнопок зміни масштабу перегляду вмісту каталога на панелі **Main Toolbar** (рис. 4.10).



Рис. 4.10

Вміст каталога в режимі *Tree view* виглядає дещо інакше (рис. 4.11).

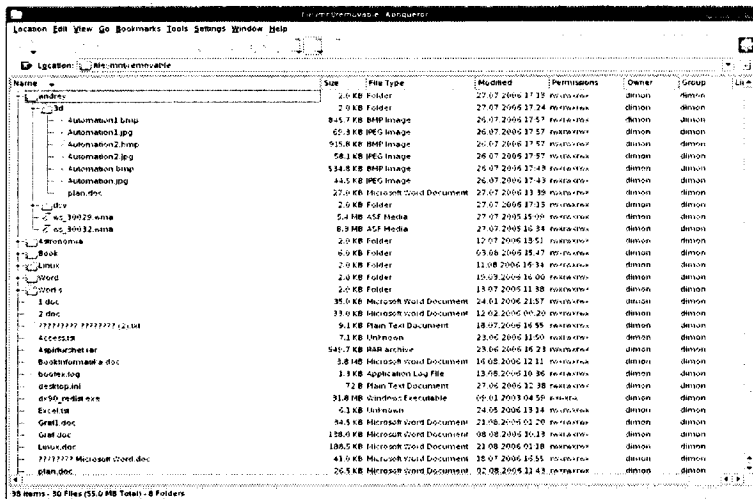


Рис. 4.11

В даному режимі відображаються заголовки стовпчиків, натискаючи на які, можна відсортувати об'єкти у відкритому каталозі за тим параметром файлу, який відображає стовпчик.

Фактично в режимі *Tree view* відображається дерево каталогів, які знаходяться у відкритому каталозі. Біля каталога, який містить підкаталоги та файли, наявна позначка **+**, натиснувши на яку лівою кнопкою мишки, можна розгорнути вміст даного каталога. Якщо біля каталога знаходиться позначка **-**, то це означає, що даний каталог є вже розгорнутим й внизу, трохи з відступом праворуч, відображається його вміст.

4.5.4. Операції з файлами та каталогами в оболонці KDE

На відміну від оболонки *Shell*, яка працює в текстовому режимі, графічна оболонка *KDE* надає зручні засоби проведення операцій з файлами та каталогами, подібні до своїх аналогів в ОС *Windows*. Розглянемо ці операції.

Створення файлів або каталогів

Для створення файлу або каталога в іншому каталозі необхідно клацнути правою кнопкою мишки на вільному місці області відображення вмісту цього каталога та обрати команду **Create New**. Після цього у підменю потрібно обрати ту команду, яка відповідає типу об'єкта, який необхідно створити – **Folder...** (каталог), **File** (файл), **Device** (пристрій) (рис. 4.12).

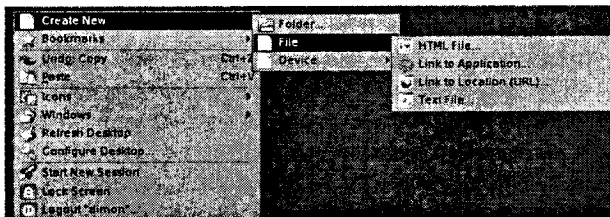


Рис. 4.12

Наступним кроком може бути задання параметрів нового об'єкта, після чого він буде створений.

Якщо ми обираємо в підменю команду **File**, то як видно з рис. 4.12, ми маємо змогу обрати в наступному підменю команду **Link to Application**, яка виведе діалогове вікно, де користувач має змогу задати параметри посилання на потрібний об'єкт. Звісно, що видаляючи посилання у **Linux**, ми не видаляємо сам об'єкт, для якого створене це посилання.

Копіювання та переміщення об'єкта

Копіювання об'єкта може здійснюватися способом, який використовується також і в **Windows** при застосуванні для копіювання буфера обміну. Тобто необхідно виділити об'єкт (виділення виконується подібно до виділення у **Windows**), натиснути комбінацію клавіш **Ctrl+C** та, перейшовши в каталог, куди потрібно скопіювати об'єкт, натиснути комбінацію клавіш **Ctrl+V**, після чого на екрані з'явиться вікно копіювання об'єкта (рис. 4.13), яке певною мірою відрізняється від подібного вікна **Windows**.

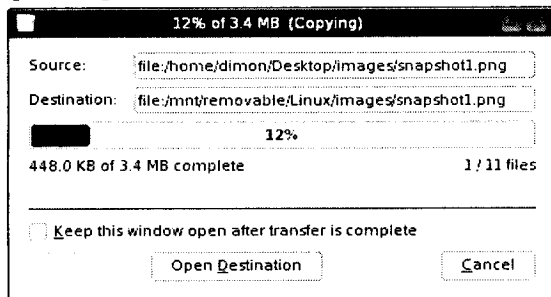


Рис. 4.13

В цьому вікні користувач має право не тільки скасувати дію (кнопка **Cancel**), а й відкрити каталог-адресат, переглянути повний шлях до каталога-джерела та каталога-адресата, а також задати опцію **Keep this window after transfer is complete** для того, щоб після закінчення операції копіювання вікно копіювання не зникло.

Переміщення об'єкта може виконуватися подібно до копіювання, тільки замість комбінації клавіш **Ctrl+C** потрібно використовувати **Ctrl+X**. Діє й проста операція перетягування при натиснутій лівій кнопці мишки.

Перейменування об'єкта

Для перейменування об'єкта можна використовувати наступний простий спосіб – виділити об'єкт та клацнути через деякий час лівою кнопкою мишки на його назві, після чого користувач матиме змогу ввести нове ім'я.

Перегляд властивостей об'єкта

Для перегляду властивостей об'єкта необхідно клацнути на ньому правою кнопкою мишки та у контекстному меню обрати команду **Properties**, після чого на екрані з'явиться одноименне вікно (рис. 4.14).

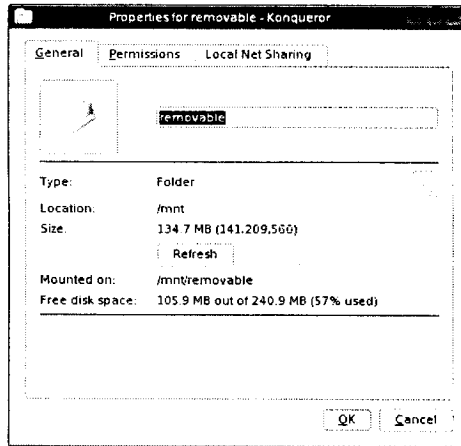


Рис. 4.14

Видалення об'єкта

Для видалення об'єкта (групи об'єктів) необхідно їх виділити та натиснути клавішу **Del**, після чого об'єкт буде видалений до корзини (**Waste-bin**), або натиснути **Shift+Del** для видалення об'єкта, минаючи корзину.

Запитання до самоперевірки

1. Наведіть основні переваги та особливості **Linux**.
2. Яким чином можна вводити команди з клавіатури?
3. Як можна додати нового користувача?
4. Які права має користувач **root**?
5. Наведіть основні команди для роботи з командним рядком.
6. Яку роль виконує клавіша **Tab** у більш зручному введенні команд?
7. Які комбінації клавіш використовуються для роботи з історією команд?
8. Яким чином виконується вихід з системи?
9. Яка файлова система є основною для **Linux**?
10. Як позначається кореневий каталог **Linux**?
11. Наведіть основні команди для роботи з файлами або каталогами.
12. Складіть порівняльну характеристику **Linux** та **Windows XP**.

5. ОСНОВИ ПРОГРАМУВАННЯ МОВОЮ ОБ'ЄКТ PASCAL

📖 План викладу матеріалу

1. Основні конструкції мови.
2. Галуження і цикли.
3. Модульне програмування.
4. Структуровані типи даних.
5. Об'єктно-орієнтоване програмування.
6. Програмування віконних застосувань.

↔ Ключові терміни розділу

✓ Консольні застосування	✓ Алфавіт мови; ідентифікатори
✓ Літерали, константи, змінні	✓ Операції та вирази
✓ Типи даних; приведення типів	✓ Оператор присвоєння
✓ Оператори галуження та вибору	✓ Оператор циклу з параметром
✓ Оператори циклу з умовою	✓ Безумовна передача керування
✓ Процедура; виклик процедури	✓ Функція; виклик функції
✓ Зв'язок параметрів і аргументів	✓ Модуль; структура модуля
✓ Масиви; вектори і матриці	✓ Записи (комбіновані типи)
✓ Множини; використання множин	✓ Робота з рядками символів
✓ Робота з файлами	✓ Властивість інкапсуляції класу
✓ Наслідкування та поліморфізм	✓ Клас; оголошення класу
✓ Конструктори і деструктори	✓ Використання об'єктів
✓ Бібліотека візуальних компонентів	✓ Базові класи VCL
✓ Властивості та події TControl	✓ Події класу TWinControl
✓ Процедури опрацювання подій	✓ Параметр Sender

5.1. Основні конструкції мови

5.1.1. Історична довідка

Мова програмування Pascal була розроблена швейцарським ученим Ніклаусом Віртом у 1968 році і спочатку призначалася для навчальних цілей. Мова є добре детермінованою, тобто все підкоряється певним правилам, винятків з яких не так багато. Відносно невелика кількість базових понять, простий синтаксис і швидкі компілятори досить швидко зробили її популярною у всьому світі.

У лютому 1995 р. Borland International випустила пакет Delphi 1.0 – інтегроване середовище розроблення програмних продуктів на базі мови Object Pascal в операційній системі Windows. З тих пір пакет Delphi

постійно розвивають та удосконалюють. На цей час найбільш розповсюдженими є сьома та восьма версії. Усі навчальні програми і проекти, які описано у підручнику, були протестовані у пакеті Delphi 7. Більшість програм без усяких проблем будуть виконуватися у будь-якій версії Delphi.

5.1.2 Інтегроване середовище Delphi

Інтегроване середовище Delphi (надалі *середовище*) – це програмний продукт, у якому є все необхідне для проектування, запуску і тестування застосувань (програм) під Windows мовою Object Pascal. Основні вікна середовища (рис. 5.1): **головне вікно** складається з *рядка меню*, *палітри компонентів* і *панелі швидкого доступу*; **вікно інспектора об'єктів** складається зі *списку властивостей* (Properties) і *списку подій* (Events); **вікно форми** є робочою поверхнею для розміщення елементів керування; **вікно редактора коду** дає змогу вводити та редагувати тексти програм.

Інспектор об'єктів Меню Панель швидкого доступу Палітра компонентів

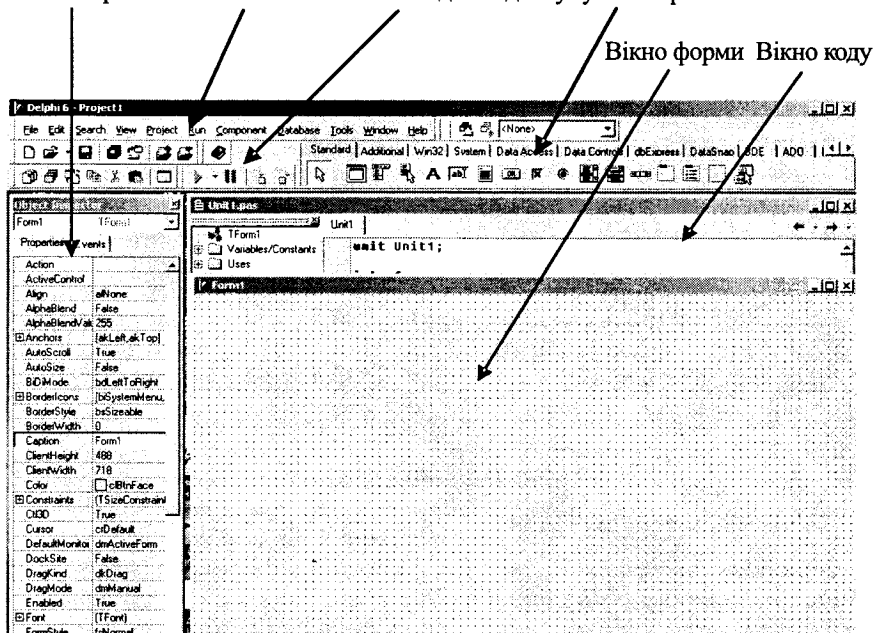


Рис. 5.1. Інтегроване середовище розробки Delphi

Рядок меню надає швидкий і гнучкий інтерфейс до середовища Delphi за допомогою команд меню. *Палітра компонентів* дає можливість вибирати потрібні об'єкти (компоненти) для розміщення їх на формі. Для цього треба перший раз клацнути мишкою на одному з об'єктів і потім другий раз – на формі.

Панель швидкого доступу (ПШД) містить набір кнопок, які надають швидкий доступ до окремих команд меню. На панелі можна розміщувати кнопки команд меню за власними потребами або вилучати маловживані кнопки. Для прикладу доповнимо ПШД кнопкою виклику команди **Copy**:

ПКМ // клацаємо правою кнопкою миші над панеллю ПШД

➤**Customize ...**// виконуємо команду меню **Customize**

Commands // розкриваємо закладку **Commands**

↓**Categories** ↗ **Edit** // вибираємо елемент **Edit** у списку **Categories**

↓**Commands** ↗ **Copy** // кнопку **Copy**, перетягуємо на ПШД

Close // клацаємо на кнопці **Close**.

Увага! Надалі позначки , ➤, , ↓ і ↗ будемо вживати без додаткових пояснень.

Для вилучення кнопки з панелі швидкого доступу необхідно (при відкритому вікні **Customize**) перетягнути її за межі панелі.

Програми, які працюють в операційних системах Windows, прийнято називати додатками (або застосуваннями). В основі таких додатків є **проект** – множина програмних файлів (або **модулів**) та допоміжних файлів (файли ресурсів, файли конфігурації тощо).

Застосування Delphi зазвичай мають *інтерфейс користувача* на базі вікон (*форм*). Delphi дає можливість створювати і простіші *консольні* застосування, у яких для введення/виведення даних використовують *консоль* – об'єднання клавіатури і дисплея. Консольні застосування дуже корисні при початковому вивченні мови.

5.1.3. Консольні застосування

Створити консольне застосування можна різними способами. Найпростіше так: ➤**File** ➤ **New** ➤ **Other...** **New** **Console Application**.

У консольному застосуванні можуть бути розділи: **program** (назва програми); **uses** (підключення модулів); **label** (оголошення позначок); **const** (оголошення констант); **type** (оголошення нових типів); **var** (оголошення змінних); **procedure** (опис власних процедур); **function** (опис власних функцій); **begin** послідовність_операторів **end** (тіло програми; після **end** – крапка).

Обов'язковою частиною є лише *тіло програми*. Заголовок програми є хоч і обов'язковим, але бажаним елементом. Порядок оголошень і описів не регламентується. Для пояснення окремих частин програми використовують *коментарі* двох видів:

{ текст, який може бути на місці будь-якого пропуску }

// текст до кінця рядка, на якому він з'явився

Коментарі, які обмежують з обох боків фігурними дужками, можуть займати будь-яку кількість рядків. Коментар ігнорується під час компіляції, окрім *директив компілятора*: { \$ директива }.

Введення даних у консольному режимі здійснюють з клавіатури, а виведення – на дисплей. Виклик процедури введення даних:

Read / Readln(список_змінних) | Readln.

Змінні у списку_змінних відокремлюють комами; Readln – після введення переводить курсор на новий рядок; Readln (без аргументів) – організація паузи (для продовження – <Enter>).

Виклик процедури виведення даних:

Write / Writeln(список_виведення) | Writeln

У списку_виведення крім назв змінних можна писати рядкові константи (послідовність символів в апострофах) і навіть вирази (виводитимуться їхні значення); Writeln – після виведення переводить курсор на новий рядок; Writeln (без аргументів) – переводить курсор на новий рядок.

Існує *форматоване виведення*, в якому після виразу або змінної через двокрапку вказують кількість позицій для виведення результату (цілих чисел). Якщо результат є раціональним числом, то додатково через двокрапку вказують кількість цифр у дробовій частині числа (одночасно відбувається заокруглення результату).

5.1.4. Алфавіт мови. Ідентифікатори

Алфавіт мови містить великі та малі латинські букви, _ (символ підкреслення; арабські цифри: 0, 1, ..., 9; спеціальні символи:

+ - * / < > = ' @ # \$ ^ : ; , . () [] { } .

Використання символів кирилиці допускається у коментарях, рядках символів, назвах файлів. Речення (*оператори*) мови завершують крапкою з комою. У деяких випадках крапку з комою можна опускати. Зокрема, крапку з комою можна не ставити (а можна й ставити) перед ключовим словом **end**.

Символом-роздільником елементів (*лексем*) мови зазвичай є символ пропуску (код 32), який на екрані не зображується. До роздільників відносять і символи керування (коди від 0 до 31), зокрема: *табуляцію* (код 09); складений символ *переходу до нового рядка* (послідовність кодів 13 – “переведення каретки” і 10 – “переведення рядка”). Між будь-якими двома лексемами можна задавати довільну кількість символів-роздільників.

Ідентифікатор – послідовність букв і цифр довільної довжини (знавущими є перші 64 символи), перший символ мусить бути буквою. Символ підкреслення `_' вважається буквою. Великі і малі букви в ідентифікаторах вважають однаковими символами. Тому, наприклад, ідентифікатори `Suma`, `suma` позначають *один і той самий* елемент програми.

Деякі ідентифікатори мови зарезервовані для службових цілей і їх не можна використовувати з метою найменування констант, змінних і функцій, наприклад: **array**, **const**, **do**, **if** тощо. Такі ідентифікатори називають *ключовими (зарезервованими) словами* і вони входять у алфавіт мови.

У рядку програми можна записати декілька операторів. Однак з погляду простоти читання тексту цим не треба зловживати. Рекомендовано виділяти об'єднані одним змістом оператори в групи, використовуючи для цього відступи і коментарі. Операторні дужки **begin** і **end** виділяють *складений оператор*. Всі оператори, розміщені між операторними дужками, сприймаються синтаксично як один оператор.

5.1.5. Операції та вирази

Операції та вирази задають певну послідовність дій, але не є закінченими реченнями мови. *Прості вирази* містять знак операції та операнди, наприклад $3.14+x$. Операції можна виконувати з одним або двома операндами. Відповідно до цього розрізняють *унарні* та *бінарні* операції. Операндами можуть бути *константи*, *змінні*, *виклики функцій* і *вирази*.

Константами називають елементи програми, значення яких не змінюються в процесі її виконання. Вони бувають або безпосередньо у вигляді значення (*літерала*), або у вигляді ідентифікатора константи, описаного в розділі `Const`. Константами можуть бути числа, логічні константи, символи та рядки символів.

Цілі числа містять лише цифри і знак: -24 , 73 ; *раціональні* – можуть містити також десяткову крапку, показник степеня і символ 'e', який замінює основу 10 в числі: -0.5 , $-1e-5$, $7.2e+15$.

Логічні константи – це значення `False` або `True`.

Символьна константа – це символ ANSI, взятий в апострофи (наприклад: `'f'`). Якщо є необхідність записати сам символ апострофа, то його записують так: `''`. Допускається записувати символ через задання його внутрішнього коду, перед яким стоїть символ `#`, наприклад: `#97` – символ `'a'`; `#90` – символ `'Z'`.

Рядок символів – це послідовність довільних символів ANSI (окрім символу `CR` – повернення каретки), взята в апострофи (наприклад: `'Рядок'`). Якщо у рядку символів треба вказати сам символ апострофа, то він подвоюється, наприклад: `'Ім' 'я'`.

Змінними називаються елементи програми, які можуть міняти своє значення у процесі її виконання. Всі без винятку змінні мають бути описані в розділі програми `var`.

Виклик функції – це вказівка назви функції, за якою у круглих дужках записують список аргументів, як-от: $F(x+c, y)$.

Зауважимо, що для роботи з числовими даними (константи, змінні та вирази) є велика кількість вбудованих (стандартних) функцій: `abs`, `sqrt`, `sqrt`, `sin`, `cos`, `exp`, `ln` та ін.

Символи: `+`, `-` і `*` є знаками *бінарних* арифметичних операцій *додавання*, *віднімання* і *множення* відповідно. Символ `/'` позначає операцію ділення раціональних чисел. Для цілих чисел визначені бінарні

операції *цілочисельного ділення* (**div**) і *знаходження залишку* від цілочисельного ділення (**mod**).

Прикладом *унарної* арифметичної операції є операція *зміни знака числа*, яка позначається символом “мінус”, що стоїть перед одним операндом, як-от: $-x+y$.

Вираз – це послідовність знаків операцій, операндів і круглих дужок, яка задає обчислювальний процес отримання результату певного типу. *Найпростішими* виразами є *константи*, *змінні* та *виклики функцій*.

Порядок обчислення виразу визначають через розміщення круглих дужок, знаків операцій та *пріоритетність* операцій. Операції з найвищим пріоритетом виконують першими. У сумнівних випадках доцільно використовувати круглі дужки.

Окрім арифметичних операцій є ще операції *відношення*, *логічні* (бульовські) операції, *логічні порозрядні* операції, операції з *рядками*, операції з *показжчиками*, операції з *множинами*, операції з *класами*, операція *отримання адреси* (@).

5.1.6. Прості типи даних

У програмі оперують зі *змінними* та *константами*, для яких попередньо визначають тип даних, щоб повідомити компілятору:

- назву (ідентифікатор) змінної чи константи;
- розмір і спосіб виділення пам'яті для зберігання значень;
- допустимі дії, які можна виконувати зі змінною чи константою.

Типи ділять на *стандартні* (вбудовані) і типи, *визначені програмістом*. Стандартні типи ділять на *прості* типи, *рядки символів*, *структури*, *показжчики* (або вказівники) та *процедурні* типи.

Прості типи даних ділять на *порядкові*, *раціональні* і *типи дати-часу*. У свою чергу *порядкові* типи даних ділять на *цілі*, *символьні*, *логічні*, *перелічені* та *обмежені*. До структур відносять *множини*, *масиви*, *записи*, *файли*, *класи* та *інтерфейси*.

Програміст може визначити власний тип безпосередньо при описі змінної у розділі **var**, або, найчастіше, у спеціальному розділі опису типів – **type**. Після ключового слова **type** для кожного типу слідує конструкція, яка має вигляд:

ідентифікатор_типу = визначення_типу;

Наприклад, у програмі можна розмістити оголошення:

```
type TColor=(Blue, Red, White);
```

```
var c1, c2: TColor;
```

Можна використати й еквівалентне оголошення:

```
var c1, c2: (Blue, Red, White);
```

Діапазони можливих значень для цілих типів даних:

```
Byte (0 ÷ 255); Word (0 ÷ 65 535); LongWord (0 ÷ 4 294 967 295);
```

ShortInt $(-128 \div 127)$; **SmallInt** $(-32\,768 \div 2\,767)$; **Cardinal** $(0 \div 4\,294\,967\,295)$; **Integer** $(-2\,147\,483\,648 \div 2\,147\,483\,647)$; **LongInt** $(-2\,147\,483\,648 \div 2\,147\,483\,647)$; **Int64** $(-2^{63} \div 2^{63}-1)$.

Діапазони можливих значень для раціональних типів даних:

Real48 $(\pm 2.9e^{-39} \div \pm 1.7e^{38})$; **Real** $(\pm 5.0e^{-324} \div \pm 1.7e^{308})$; **Single** $(\pm 1.5e^{-45} \div \pm 3.4e^{38})$; **Double** $(\pm 5.0e^{-324} \div \pm 1.7e^{308})$; **Extended** $(\pm 3.6e^{-4932} \div \pm 1.1e^{4392})$; **Comp** $(-2^{63}+1 \div 2^{63}-1)$; **Currency** $(|x| < 922\,337\,203\,685\,477.5807)$.

Раціональні числа зберігаються наближено, отож під час віднімання близьких чисел можлива втрата значущості. Це ж пояснює, чому потрібно уникати порівняння раціональних величин на точну рівність. Вважають, що раціональні числа a і b є рівними, якщо виконується умова $\text{abs}(a-b) < \text{eps}$, де eps – мале додатне число.

Раціональні типи є *впорядкованими*, але не *порядковими*. Тип відносять до *впорядкованих*, якщо для змінних і виразів цього типу визначені операції відношення ($=$, $<$, $>$, $<=$, $>=$). Будь-який *порядковий* тип є *впорядкованим*, але ніколи навпаки.

Кожному значенню *порядкового* типу функція $\text{Ord}(x)$ підбирає відповідне натуральне число – *порядковий номер* значення виразу x у множині значень цього типу. Для цілих типів повертається саме значення, для логічного 0 або 1, для символного – значення від 0 до 255. До будь-яких *порядкових* типів також можна застосовувати функцію $\text{Pred}(x)$, яка повертає попереднє значення, і $\text{Succ}(x)$ – наступне значення.

Змінні логічного типу **Boolean** займають у пам'яті один байт і можуть набувати одне з двох значень **False** (хибне) або **True** (істинне). Результат будь-яких операцій *відношення* має логічний тип і може бути привласнений логічній змінній.

Змінні *символьного* типу **Char** дають змогу працювати з окремими символами. На відміну від самого тексту програми символи, які відповідають малим і великим буквам, розрізняються. Функція Ord видає код відповідного символу (від 0 до 255). Оберненою функцією, яка за кодом видає відповідний символ, є функція Chr .

Перелічений тип (є *порядковим*; нумерація з нуля) задається множиною ідентифікаторів у круглих дужках через кому.

Приклад 5.1.1. Використання перелічених типів:

```
Type Colors = (Red, Green, Blue);
Numbers = (Zero, One, Two, Three, Four, Five);
var z:Colors;    n:Numbers;
begin z:=Red;    write(Ord(z));    {Виведеться 0}
      n:=Four;    write(Ord(n));    {Виведеться 4}
      z:=Succ(z);    {z=Green}
for n:=One to Five do write(Ord(n)); {12345} end.
```

Тип *діапазон* також називають *обмеженим* або *інтервальним* типом. Він є підмножиною свого **базового** типу, яким може виступати будь-який порядковий тип. Тип діапазон задається **межами** своїх значень у базовому типі (через двокрапку) і успадковує всі властивості цього базового типу.

Є дві стандартні функції, які працюють з типом діапазону: `High(x)` – повертає максимальне значення діапазону, до якого належить змінна `x`; `Low(x)` – повертає мінімальне значення.

Приклад 5.1.2. Використання типу діапазон:

```
type Numbers = (Zero, One, Two, Three, Four, Five);
Num = Two .. Four; {діапазон на базі типу Numbers}
Abc = 'A' .. 'z'; {всі англійські букви: діапазон на
базі типу Char}
Digits = 0 .. 9; {цифри}
var n: Num; z, d: Abc; x: SmallInt;
begin
n:= Four; writeln(Ord(n)); {4, як у базовому типі}
n:= Succ(n); {Помилка: наступне значення поза
діапазоном}
read(z, d);
if z = d then write('Однакові букви');
writeln(Low(z), ' .. ', High(z)); { A .. z }
writeln(Low(x), ' .. ', High(x)); { -32768 .. 32767 }
end.
```

5.1.7. Оператор присвоєння. Перетворення типів даних

Алгоритмічні дії над початковими даними і робочими об'єктами мови, необхідні для розв'язання поставленої задачі, описують за допомогою *операторів*. Оператори розділяються крапкою з комою. Найпростіший випадок – це *лінійні* ділянки алгоритмів, у яких оператори виконують послідовно один за одним.

У програмі найчастіше трапляється *оператор присвоєння*, який використовується для задання значення змінним:

назва_змінної:= вираз;

Під час виконання оператора присвоєння:

- обчислюється вираз, що стоїть у правій частині оператора;
- результат обчислення перетворюється до типу змінної;
- результат присвоюється змінній лівої частини оператора.

У виразах та операторах присвоєння можуть міститися елементи (змінні та константи) різних типів. У цьому випадку здійснюється *приведення* (перетворення) одного типу даного в інший.

Неявне приведення типів автоматично здійснює компілятор Delphi за принципом: якщо операція має операнди різних типів, то тип операнда

“молодшого” типу (займає менший обсяг пам’яті) зводиться до типу операнда “старшого” типу (займає більший обсяг пам’яті).

Наприклад, якщо в операції беруть участь коротке ціле і довге ціле, то коротке ціле зводиться до довгого; якщо беруть участь цілий і дійсний операнди, то цілий зводиться до дійсного і т.д. Після зведення типів обидва операнди виявляються одного типу.

У операторі присвоювання відбувається зведення типу результату обчислення виразу правої частини до типу лівого операнда. Якщо тип лівого операнда “молодший”, ніж тип результату, то можливою є втрата точності чи отримання неправильного результату взагалі, а тому такі перетворення спричиняють помилки компіляції. Наступна послідовність операторів призводить до помилки:

```
var i:byte; a:real;
```

```
begin a:=2; i:=a+1; // Помилка приведення типів
```

Як видно з наведеного прикладу, неявне перетворення типів даних не завжди дає бажаний результат. Це можна виправити, застосувавши операцію *явного* перетворення типів даних:

```
ідентифікатор_типу (вираз)
```

Явне приведення типів даних можна здійснювати для будь-яких типів, що мають однаковий розмір. Водночас не допускається взаємне перетворення цілих та дійсних даних. Для перетворення дійсних чисел у цілі використовують функції `Inc` і `Trunc`. А цілі числа перетворюються у дійсні неявно. Так, наприклад, попередній приклад можна модифікувати так:

```
begin a:=2; i:= Trunc(a+1); // Все добре!
```

Явне приведення типів даних можна використовувати як у лівій, так і правій частині оператора присвоювання.

5.2. Галуження і цикли

5.2.1. Складений оператор. Оператори галуження та вибору

Складений оператор – послідовність довільних операторів, взята в операторні дужки (слова `begin` і `end`). Він застосовується там, де за синтаксичними правилами мови може стояти тільки один оператор, а нам треба виконати декілька дій, наприклад:

```
begin a:=b+2; b:=b+1 end
```

Оператор галуження (або *умовний оператор*) забезпечує виконання деякого оператора залежно від виконання певної умови, яку задає логічний вираз:

```
if логічний_вираз then оператор;
```

Якщо значення логічного виразу істинне, виконується оператор, а інакше виконується наступний за `if` оператор. Гілка `else` є розширенням оператора `if` для виконання альтернативної дії:

if логічний_вираз **then** оператор1 **else** оператор2;

Якщо значення логічного_виразу істинне, виконується оператор1, а інакше виконується оператор2. За правилами кожна з гілок може містити або один оператор, або декілька, об'єднаних у складений оператор. Крапку з комою перед **else** вважають помилкою. Рекомендуємо у гілці **then** не використовувати новий оператор **if**, оскільки це може призвести до важко передбачуваних помилкових результатів. Якщо ж все-таки без цього обійтися не можна, то оператор **if** після **then** треба взяти в операторні дужки.

Приклад 5.2.1. Обчислити $z = \max(x, y)$:

if $x > y$ **then** $z := x$ **else** $z := y$;

Приклад 5.2.2. Обчислити $z = \min(x, y, t)$.

if $(x < y)$ **and** $(x < t)$ **then** $z := x$
else if $y < t$ **then** $z := y$ **else** $z := t$;

Приклад 5.2.3. Обчислити $z = \begin{cases} \min(x, y), & \text{якщо } x < 0, \\ \max(x, y), & \text{якщо } x \geq 0. \end{cases}$

if $x < 0$ **then begin if** $x < y$ **then** $z := x$ **else** $z := y$ **end**
else if $x > y$ **then** $z := x$ **else** $z := y$;

Приклад 5.2.4. Обчислити $y = \begin{cases} a + b, & \text{якщо } x < -2; \\ a - b, & \text{якщо } -2 \leq x \leq 3; \\ a \cdot b, & \text{якщо } x > 3. \end{cases}$

if $x < -2$ **then** $y := a + b$
else if $x \leq 3$ **then** $y := a - b$ **else** $y := a \cdot b$;

Зручним засобом вибору одного варіанта з множини можливих варіантів є оператор **case** (оператор *вибору*), який має вигляд:

case вираз **of**
 значення1: оператор1;
 ...
 значенняN: операторN;
else оператор;
end;

Гілка **else** не обов'язкова, і на відміну від оператора **if** перед нею можна ставити крапку з комою. Якщо значення виразу = значенню1, тоді виконується оператор1 і виходимо з **case** і т.д. Якщо значення виразу ні разу не збіглося з заданими значеннями (значення1, ..., значенняN), тоді виконується оператор, що стоїть після **else** (або наступний оператор за **case** – за відсутності **else**). Вираз має бути порядкового типу. Якщо для декількох значень виразу дії збігаються, то ці константи можна перерахувати через кому перед двокрапкою або навіть задати діапазон.

Хоча використання варіанта **else** є не обов'язковим, досвідчені програмісти для пошуку помилок часто включають **else**, навіть коли враховані всі варіанти.

Приклад 5.2.5. Задати текстове значення осin, яка є оцінкою.

```

case osin of
2: s:='не задовільно';
3: s:='задовільно';
4: s:='добре';
5: s:='відмінно';
else s:='Вихід за діапазон';
end;

```

Приклад 5.2.6. Введено ціле число. Якщо це цифра, то визначити, парна вона чи ні, а якщо число, то визначити, чи потрапляє воно у діапазон від 10 до 100. Видати відповідне повідомлення.

```

program chyslo; {$APPTYPE CONSOLE}
var      i:integer;
begin
write('Vvedit cile chyslo:'); readln(i);
case i of
0,2,4,6,8: writeln('Parna cyfra');
1,3,5,7,9: writeln('Neparna cyfra');
10...100: writeln('Chyslo vid 10 do 100');
else writeln('Chyslo - ne cyfra i za diapazonom');
end; readln
end.

```

5.2.2. Оператор циклу з параметром

Для виконання програми часто виникає необхідність багаторазово виконувати одну і ту ж саму послідовність дій (*цикл*). Мова Object Pascal має оператор циклу з параметром (**for**) і два оператори циклу з умовою (**while** і **repeat**).

Оператор **for** організовує виконання деякого оператора певне число разів. Існує дві форми оператора:

```

for параметр: = pz to kz do оператор;
for параметр: = pz downto kz do оператор;

```

Тут параметр циклу (лічильник) – змінна порядкового типу; pz і kz – вирази, що визначають початкове і кінцеве значення лічильника; оператор – один (можливо складений) оператор, який називають *тілом циклу*, що повторюється певну кількість разів.

На першому кроці циклу параметр набуває значення pz. У цей же момент відбувається обчислення kz – значення параметра на останньому кроці циклу. Після кожного виконання тіла циклу, якщо параметр циклу

не дорівнює kz , відбувається зміна параметра на наступне (більше або менше) значення:

```
параметр: = Succ(параметр); //форма do
параметр: = Pred(параметр); //форма downto
```

У випадку $pz > kz$ у першій формі оператора або $pz < kz$ у другій його формі помилки немає, але цикл не буде виконаний ні разу. Після завершення циклу значення параметра рівне kz .

Приклад 5.2.7. Знайти кількість додатних чисел серед 10 введених.

```
program dodatni; {$APPTYPE CONSOLE}
var      i, kn:byte; x:real;
begin kn:=0; // Лічильник додатних чисел
for i:=1 to 10 do
begin write('Vvedit ', i, ' chyslo: '); readln(x);
if x>0 then kn:=kn+1 // Збільшуємо лічильник
end;
writeln('Vvely ',kn,' dodatnyh chysel'); readln
end.
```

Приклад 5.2.8. Протабулювати значення функції $y = \sqrt{x^2 + 1} \cdot \cos x$ на проміжку $[a; b]$ з кроком h .

```
program Tabul;{$APPTYPE CONSOLE}
var a, b, h, x, y: single;
begin
write(' Vvedit a<b i h >0 '); readln( a, b, h);
if a>b then begin y:=a; a:=b; b:=y end;
if h<0 then h:= -h;
writeln(' ----- ');
writeln(' | x          |          y          | ');
writeln(' ----- '); x:=a;
while x<=b do
begin
y:=sqrt(x*x+1)*cos(x);
write(' | ', x:7:2, ' | ');
writeln(y:7:2, ' | ');
writeln(' ----- '); x:=x+h
end; readln
end.
```

5.2.3. Оператори циклу з умовою

Якщо кількість повторень циклу заздалегідь невідома, то використовують цикли з умовою, які є двох типів: **while** і **repeat**.

Цикл типу **while** є *циклом з передумовою*. Він використовується зазвичай у випадках, коли кількість повторень циклу заздалегідь невідома

і можливим є випадок, коли цикл не буде виконаний жодного разу. Цикл типу **while** має вигляд:

while логічний_вираз **do** оператор;

До тих пір, доки значення логічного виразу буде істинним, доти буде виконуватися один оператор (можливо складений). Оскільки значення логічного виразу перевіряється на початку кожної ітерації, то тіло циклу може бути не виконаним взагалі. Отже, у цьому циклі логічний вираз – це умова продовження циклу.

Приклад 5.2.9. Пари додатних раціональних чисел вводять з клавіатури.

Обчислити добуток кожної пари і суму всіх чисел.

```

program cycle_while; {$APPTYPE CONSOLE}
var      x, y, sum:real;      v:char;
  begin sum:=0; v:='Y';
  while (v='Y') or (v='y') do
  begin write('Vvedit chysla x i y >0');
    readln(x,y);
    if (x>0) and (y>0) then
    begin writeln('Dobutok =', x*y:8:3);
      sum:=sum+x+y end;
    write('Nastupni (Y/N)?'); readln(v);
  end;
  writeln('Suma vsih chysel=', sum:8:3); readln
end.

```

Інший варіант циклу з умовою – це цикл **repeat**, який називають *циклом з постумовою*. Він має вигляд:

repeat оператор1;... операторN **until** логічний_вираз;

Оператор **repeat** організує виконання декількох операторів доти, доки не стане істинним логічний вираз (умова виходу з циклу). Тіло циклу обов'язково виконається принаймні один раз.

Приклад 5.2.10. Виконати приклад 5.2.9, використовуючи **repeat**.

```

program cycle_repeat; {$APPTYPE CONSOLE}
var      x, y, sum:real;      v:char;
  begin sum:=0;
  repeat write('Vvedit chysla x, y >0'); readln(x, y);
  if (x>0) and (y>0) then
  begin writeln('Dobutok=', x*y:8:3); sum:=sum+x+y end;
  write('Nastupni (Y/N)?'); readln(v)
  until (v='N') or (v='n');
  writeln('Suma vsih chysel= ', sum:8:3); readln
  end.

```

На місці оператора у циклі може стояти інший цикл (отримуємо *вкладені цикли*). В усіх трьох циклах, у яких на місці оператора є складений

оператор, всередині можна записати оператор **break**. Якщо під час виконання циклу оператор **break** отримує керування, то відбувається негайний вихід з циклу і перехід на наступний оператор, який записаний за цим циклом. Процедура **continue**, яку можна записати тільки у тілі циклу, пропускає записані за нею оператори і переходить до наступної ітерації циклу.

5.2.4. Безумовна передача керування

Оператор безумовної передачі керування має вигляд:

```
goto <позначка>;
```

Він дає змогу передати керування безпосередньо на потрібний оператор програми. Перед цим оператором має бути розміщена *позначка* – ідентифікатор, який відокремлено від оператора двокрапкою. Позначки описують у розділі оголошення позначок:

```
label <список_позначок>;
```

Конкретною позначкою у програмі може бути позначений тільки один оператор. Операторів переходу з однією і тією ж позначкою можна задати будь-яку кількість. Необхідно, щоб розділ оголошення позначок, сама позначка і оператор переходу з її використанням були розміщені у межах одного блоку програми (див. тему *процедури і функції*). Крім того, не можна передавати керування всередину операторів галуження і циклу.

5.3. Модульне програмування

5.3.1. Загальні положення

Паскаль дає можливість виділяти фрагменти програми у допоміжні алгоритми, за допомогою яких можна писати добре структуровані програми, які зазвичай простіші для розуміння та налагодження. У Паскалі допоміжні алгоритми – це *підпрограми*, які можна оформити у вигляді *процедури* або у вигляді *функції*.

У підпрограмах, як і у звичайних програмах, можуть бути описані власні мітки, константи, типи, власні змінні і навіть підпрограми. Кожна підпрограма має власну назву, яку використовують у разі її виклику на виконання (процедура викликається окремим оператором, а виклик функції є частиною виразу).

У програмі *опис підпрограм* зазвичай розміщують між розділами змінних і операторів, хоча Object Pascal допускає довільну послідовність опису констант, змінних, типів, позначок і підпрограм. Але необхідно пам'ятати **базове правило** Object Pascal: будь-яка назва у програмі має бути описана перед тим, як вона з'явиться серед операторів, що виконуються. Кожна підпрограма визначається лише один раз, але може використовуватися багаторазово.

Використання підпрограм є першим кроком, який дає змогу підвищити ступінь абстрактності програми і спрощення її структури. Наступним кроком підвищення абстрактності програм є групування підпрограм і зв'язаних з ними даних в окремі *файли* (модулі), які компілюються незалежно від інших модулів чи програм.

Модуль (unit) у мові Паскаль – це особливим чином оформлена бібліотека підпрограм (процедур і функцій). Модуль не може бути запущений на виконання самостійно, він може тільки брати участь у побудові програми та інших модулів. Модуль – це окрема програмна одиниця, що незалежно компілюється і зберігається.

Отримані у результаті компілювання *об'єктні модулі* об'єднуються в єдину програму (виконавчий файл) за допомогою спеціальної програми – *компонувальника*. Розбиття на модулі зменшує час перекомпілювання і спрощує процес налагодження, приховуючи несуттєві деталі за інтерфейсом модуля і дозволяючи налагоджувати програму частинами (або різними програмістами).

5.3.2. Опис підпрограми. Виклик процедури/функції

Опис підпрограми нагадує опис консольного застосування. Заголовок процедури має такий вигляд:

Procedure *назва* (список_формальних_параметрів);

Процедура може бути і без параметрів, тоді у заголовку вказують лише її назву. За допомогою параметрів передають вихідні дані у процедуру, а також результати роботи процедури назад у головну програму.

Заголовок функції має такий вигляд:

Function *назва* (список_формальних_параметрів):тип_результату;

Функція може бути і без параметрів, тоді у заголовку вказують лише її назву. Типом результату функції може бути будь-який стандартний тип, окрім файлових. Серед операторів тіла функції має бути хоча б один оператор, що присвоює назві функції значення результату.

Список формальних параметрів може складатися з *параметрів-значень*, *параметрів-змінних* (перед ними стоїть слово **var**), й деяких інших категорій формальних параметрів. Заголовок процедури з параметрами-значеннями, наприклад, має вигляд:

Procedure MyProc1(pr1, pr2: type1; pr3: type2);

Заголовок функції з параметрами-змінними, наприклад, має вигляд:

Function MyFunc1(**var** p1, p2: type1; **var** p3: type2): type;

У заголовку підпрограми можуть бути і параметри-змінні й параметри-значення, наприклад:

Procedure MyProc2(**var** p1, p2: type1; p3, p4, p5: type2);

Виклик процедури на виконання:

Назва_процедури(список_фактичних_параметрів);

Виклик функції на виконання знаходиться у деякому виразі і має вигляд: назва_функції(список_фактичних_параметрів). Між формальними і фактичними параметрами має бути відповідність за кількістю і типами. Фактичні параметри іноді ще називають *аргументи*, а формальні параметри – просто *параметри*.

Приклад 5.3.1. Обчислити
$$u = \frac{\max(x, x \cdot y) + \max^3(y, y + x)}{1 + \max^2(x + y^2, xy)}$$
.

Для розуміння різниці між процедурами і функціями – обчислення $\max(a, b)$ оформимо: а) у вигляді процедури; б) у вигляді функції.

а) Розв'язок:

```

program Project1; {$APPTYPE CONSOLE}
uses SysUtils; var x, y, t, u:real;
procedure max(a,b:real; var c:real);
  begin if a>b then c:=a else c:=b end;
begin writeln('Enter x, y'); readln(x, y);
  max(x, x*y, u); max(y, y+x, t); u:=u+sqr(t)*t;
  max(x+y*y, x*y, t); u:=u/(1+t*t);
  writeln('u=', u:7:3); readln;
end.

```

б) Розв'язок:

```

...
var x, y, t, u:real;
function max(a,b:real):real;
  begin if a>b then max:=a else max:=b end;
begin writeln('Enter x,y'); readln(x,y);
  u:=max(x, x*y); t:=max(y, y+x); u:=u+sqr(t)*t;
  t:=max(x+y*y, x*y); u:=u/(1+t*t);
  writeln('u=', u:7:3); readln;
end.

```

У Object Pascal для присвоєння результату можна використовувати стандартну змінну *result*, наприклад:

```

function max(a, b:real):real;
  begin if a>b then result:=a else result:=b end;

```

Результат роботи функції *max* в обох випадках буде однаковий. Назва функції може фігурувати тільки у лівій частині оператора присвоєння і не може входити у вирази. А *result* – звичайна змінна, тому вона може входити у будь-який вираз, наприклад:

```

function fab(a, b:real):real;
  begin result:=a+b; result:=sqr(result)*result end;

```

У Паскалі існує стандартна процедура *exit*, яка забезпечує вихід з підпрограми у будь-якому місці підпрограми, наприклад:

```
function fex(a, b:real):real;
begin result:=a+b; if result<0 then exit;
      result:=sqr(result)*result      end;
```

5.3.3. Механізм зв'язку параметрів і аргументів

Під час виклику підпрограми формальним *параметрам-значенням* виділяють нове місце у пам'яті і туди заносять значення фактичних параметрів. У таких випадках на місці фактичних параметрів можуть бути вирази. Сумісність типів параметрів і аргументів визначається можливістю присвоєння. Після виконання підпрограми пам'ять під формальні параметри звільняється. Зміна формальних параметрів у підпрограмі не позначається на значенні фактичних параметрів. Отже, *параметри-значення* містять копії даних, які їм передаються. Після завершення підпрограми ці копії знищуються. Отож їх використовують лише для передачі у підпрограму вихідних даних.

Під час виклику підпрограми формальні *параметри-змінні* займають те саме місце у пам'яті, що й відповідні їм аргументи (додаткове місце у пам'яті не виділяється і зміни параметра означають і зміну аргументу). Параметри-змінні зазвичай використовують для *передачі результатів* з підпрограми у програму чи іншу підпрограму і для передачі у підпрограму структурованих даних.

5.3.4. Модулі

Модуль – це сукупність програмних елементів (константи, типи, змінні, підпрограми), визначених для використання іншими програмами. Модуль має таку загальну структуру:

```
unit назва_модуля; {заголовок модуля}
interface { видимі програмні елементи }
implementation { приховані програмні елементи }
initialization { оператори ініціалізації елементів }
finalization { оператори завершення модуля }
end.
```

Якщо у модулі немає потреби застосовувати оператори ініціалізації, то ключове слово **initialization** можна не вказувати. Розділ завершення є необов'язковим і використовується лише разом з розділом **initialization**.

Заголовок підпрограми містить всі відомості, необхідні для її виклику: назву, перелік і тип параметрів, тип результату функції. Ця інформація має бути доступна іншим програмам і модулям. Тому заголовок процедур і функцій вміщують в інтерфейсну частину модуля, а текст підпрограми – у частину реалізації (у заголовку може не бути списку формальних параметрів).

Назву модуля беруть із заголовка модуля. Крім того, назва файлу, в якому зберігається модуль, має збігатися з назвою модуля. Для підключення модуля до програми необхідно вказати його назву в розділі опису модулів (розділ **uses**), наприклад:

```
uses CRT, Graph, myUnit;
```

У випадку, коли назви змінних в інтерфейсній частині модуля і у програмі, що використовує цей модуль, збігаються, звернення буде відбуватися до змінної, описаної в програмі. У такому випадку для звернення до змінної, описаної в модулі, необхідно застосувати складну назву, до якої входить назва модуля і назва змінної, розділені крапкою.

Приклад 5.3.2. Побудувати модуль, який включає опис комплексного типу даних Complex (числа вигляду $m + n \cdot i$, де m і n – дійсні числа, а i – уявна одиниця) і функції/процедури операцій над комплексними числами: додавання, віднімання, множення, ділення; присвоєння початкових значень через введення даних з клавіатури; виведення на екран комплексного числа. Перевірити роботу модуля на прикладі обчислення виразу $d = \frac{a+b-(2+3 \cdot i) \cdot c}{a+2 \cdot b}$, де a , b і c – комплексні числа.

Увага! Для моделювання комплексних чисел використовуватимемо такий структурований тип даних, як *записи* (див. 5.4.3).

```
program Project_532; {$APPTYPE CONSOLE}
uses SysUtils, Cmplx in 'Cmplx.pas';
var a,b,c,d,e: Complex;
begin
  writeln('Enter coplex numbers:');
  readc(a); readc(b); readc(c); e.re:=2; e.im:=3;
  d:=AddC(a,b); d:=DivC(SubC(d,MulC(e, c)),
  AddC(d,b)); writec(d); readln;
```

end.

Файл модуля:

```
unit Cmplx; // Збережено у файлі Cmplx.pas
interface
Type Complex=record re,im:real end;
Function AddC(x,y:Complex):Complex;
Function SubC(x,y:Complex):Complex;
Function MulC(x,y:Complex):Complex;
Function DivC(x,y:Complex):Complex;
Procedure ReadC(var x:Complex);
Procedure WriteC(x:Complex);
implementation
Function AddC(x,y:Complex):Complex;
  begin AddC.re:=x.re+y.re; AddC.im:=x.im+y.im; end;
```

```

Function SubC(x,y:Complex):Complex;
  begin SubC.re:=x.re-y.re; SubC.im:=x.im-y.im; end;
Function MulC(x,y:Complex):Complex;
  begin MulC.re:=x.re*y.re-x.im*y.im;
    MulC.im:=x.re*y.im+x.im*y.re;
  end;
Function DivC(x,y:Complex):Complex;
  var z:real;
  begin z:=sqr(y.re)+sqr(y.im);
    if z=0 then begin writeln('Divide by 0');
      readln; halt end;
    DivC.re:=(x.re*y.re+x.im*y.im)/z;
    DivC.im:=(x.re*y.im-x.im*y.re)/z;
  end;
Procedure ReadC(var x:Complex);
  begin write('Enter Re: '); read(x.re);
    write('Enter Im: '); read(x.im); end;
Procedure WriteC(x:Complex);
  begin write('Result=', x.re:8:2);
    if x.im>0 then write('+');
    write(x.im:8:2,' I'); readln; end;
end.

```

5.4. Структуровані типи даних

5.4.1. Масиви

Масив – це впорядкована структура *однотипних* даних, яка зберігає ці дані у пам'яті комп'ютера послідовно. Доступ до елемента масиву здійснюється через його індекси. *Тип масиву* описують таким чином:

```

назва_типу_масиву = array [діапазони_індексів]
of тип_елемента_масиву;

```

Як тип_елемента_масиву можна використати будь-який тип Паскаля, окрім файлового. Діапазони_індексів – це один діапазон або декілька діапазонів, які перераховані через кому.

Кількість діапазонів в описі типу масиву (кількість *індексів*) визначає *розмірність* масиву, яка може бути довільною (на практиці – не більше трьох). Одновимірні масиви часто називають *векторами*, а двовимірні – *матрицями* (перший індекс задає номер рядка, а другий – номер стовпця).

Приклад 5.4.1. Опис типу вектора і матриці:

```

type vector = array [0..5] of integer;
matrix = array [1..4, 1..5] of real;

```

Після опису типу масиву у розділі **var** можна описати змінні цього типу (масиви), наприклад:

```
var v1, v2:vector; m1, m2, m3: matrix;
```

Тип масиву можна вказувати і безпосередньо під час опису змінних-масивів (у розділі **var**), наприклад:

```
var w1: array [1..8] of real;
```

Оголошення масиву можна поєднувати з заданням його елементам початкових значень. Список значень з кожного виміру беруть у круглі дужки, а самі значення і списки розділяють комами.

Приклад 5.4.2. Оголошення масиву з присвоєнням значень:

```
type matr = array [1..3, 1..4] of byte;
var a1: array [1..6] of integer=(1, 2, 3, 4, 5, 6);
a2: matr = ((11, 12, 13, 14),
(21, 22, 23, 24),
(31, 32, 33, 34));
```

5.4.2. Одновимірні масиви (вектори)

Одновимірні масиви найчастіше використовують під час розв'язання задач типу: "Дано натуральне n і числа a_1, a_2, \dots, a_n . Знайти ...". У Паскалі кількість елементів масиву, їхню впорядкованість і тип задають явно до початку виконання програми. Тому, якщо границі масиву точно невідомі, то їх вибирають "із запасом" (це стосується переважно верхньої межі діапазону індекса). Наприклад, після опису

```
a: array [1.. 100] of Real;
```

уведене n має належати діапазону 1..100. Зазвичай верхню межу діапазону індексу описують у вигляді константи у розділі **const**.

Приклад 5.4.3. У масиві a кожний елемент дорівнює 0 чи 1. Скласти фрагмент програми заміни усіх нулів одиницями і навпаки.

Досить одного оператора присвоєння у тілі циклу:

```
for i:= 1 to n do a[i]:= 1-a[i];
```

Приклад 5.4.5. У масиві кожен елемент дорівнює 3, 4 чи 8. Скласти фрагмент програми перестановки елементів масиву так, щоб спочатку розмішувалися всі 3, потім усі 4 і, нарешті, 8. Додаткового масиву не заводити.

Можна не переставляти елементи масиву, а підрахувати кількості 3, 4 і заповнити масив заново у потрібний спосіб:

```
k3:=0; k4:=0;
for i:= 1 to n do
  if a[i] = 3
  then k3:=k3+1 else if a[i] = 4 then k4:=k4+1;
for i:= 1 to n do a[i]:=8;
for i:= 1 to k3 do a[i]:=3;
for i:= k3+1 to k3+k4 do a[i]:=4;
```

Приклад 5.4.6. Дано натуральне число n і послідовність дійсних чисел a_1, a_2, \dots, a_n . Знайти перше від початку послідовності найменше число a_i ($i=1..n$) і останнє за порядком у послідовності найбільше число a_j ($j=1..n$) та поміняти їх місцями.

Достатньо знайти *порядкові номери* першого від початку найменшого числа і останнього за порядком найбільшого числа.

```

program arr_546; {$APPTYPE CONSOLE}
const nmax = 30;
var a: array [1..nmax] of real;
    x: real; i, n, kmin, kmax: byte;
begin
  repeat write('Kilkist chysel vid 1 do ', nmax, ': '); read(n);
  until (n>=1) and (n<=nmax);
  writeln(' Chysla: ');
  for i:=1 to n do read(a[i]); kmin:=1; kmax:=1;
  for i:=2 to n do
    begin if a(kmin)>a[i] then begin kmin:=i end;
      if a(kmax)<=a[i] then begin kmax:=i end
    end;
  x:= a[kmin]; a[kmin]:= a[kmax]; a[kmax]:=x;
  for i:=1 to n do write(a[i]:6:2, ' '); readln
end.

```

Приклад 5.4.7. Написати фрагменти програм, які реалізують основні операції над матрицями a і b типу **array**[1..m, 1..n]: а) додавання двох матриць; б) множення матриці на скаляр; в) множення двох матриць; г) транспонування квадратної матриці.

а) Матриці в операції додавання мають однакові розміри:

```

for i:=1 to m do
  for j:=1 to n do c[i,j]:=a[i,j]+b[i,j];

```

б) Множення матриці на скаляр виконують поелементно:

```

for i:=1 to m do
  for j:=1 to n do c[i,j]:=a[i,j]*p;

```

в) Матриці мають такі розміри: $m \frac{n}{A} \times n \frac{k}{B} = m \frac{k}{C}$. Тоді будь-який еле-

мент c_j ($i=1..m; j=1..k$) знаходять за формулою $c_{ij} = \sum_{l=1}^n a_{il} \cdot b_{lj}$:

```

for i:=1 to m do
  for j:=1 to k do
    begin s:=0; for l:=1 to n do s:=s+a[i,l]*b[l,j];
      c[i,j]:=s
    end;

```

г) Під час транспонування головна діагональ матриці залишається без змін, а обмін здійснюємо *тільки* для елементів, які розміщені над цією

діагоналлю (елементи під діагоналлю поміняються автоматично внаслідок симетрії):

```
for i:=1 to n-1 do
  for j:=i+1 to n do
    begin s:=d[i,j]; d[i,j]:=d[j,i]; d[j,i]:=s end;
```

5.4.3. Записи (комбіновані типи)

Запис (або комбінований тип) – це структура даних, яка може зберігати інформацію різних типів, об'єднаних під однією назвою. Компоненти запису називають *полями*. Опис *типу запису*:

```
type назва_типу =
record
список_назв_полів_1: тип_1;
...
список_назв_полів N: тип_N;
end;
```

Типи полів запису можуть бути будь-якими. У свою чергу, тип запису може використовуватися для створення нових масивів і записів. Приклад оголошення типу запису:

```
type men = record fio, adress: string; year: byte; end;
```

Після опису типу можна визначити відповідні змінні, наприклад:

```
var a, c: men;
```

Можна оголошувати запис і безпосередньо під час опису змінних, не вводячи типу, наприклад:

```
var a, c: record fio, adress: string; year: byte; end;
```

Для звернення до полів запису застосовують *складну назву*, яка містить назву змінної-запису, крапку та назву поля, наприклад:

```
a.fio:= 'Жук П.І.'; a.year:= 1981; a.adress:= 'Свободи, 40/10';
```

Для зручності звернення до полів записів може використовуватися *оператор приєднання*: **with** змінна **do** оператор;

який дає змогу не вказувати перед кожним полем назву змінної типу запису, наприклад:

```
with a do begin fio:= 'Жук П.І.'; year:= 1981;
adress:= 'пр. Свободи, 40/10'; end;
```

5.4.5. Множини

У Паскалі допускаються тільки скінченні множини, що складаються не більш ніж з 256 різних елементів. Всі елементи мають бути значеннями одного порядкового типу, який називають *базовим типом* цієї множини, який задають діапазоном або переліком з урахуванням того, що елементи мають лежати у діапазоні від 0 до 255. Зі стандартних типів як базові підходять типи **Byte** і **Char**.

Елементи множини беруть у квадратні дужки (*порожню* множину зображують конструкцією []). Приклади:

Математичний запис:

Запис на Паскалі:

\emptyset	[]
{2, 3, 5, 7, 17, 19}	[2, 3, 5, 7, 17, 19]
{0, 1, 2, 3, 4}	[0, 1, 2, 3, 4] або [0..4]

У квадратних дужках можуть знаходитися допустимі вирази зі змінними, наприклад, [i, j] – множина, що містить поточні значення змінних i, j типу Byte. Як і в математиці, вважають, що порядок перерахування елементів ролі не грає i, що кожний елемент враховується тільки один раз, тому [true, false] еквівалентно [false, true], а [1, 4, 2] еквівалентно [1, 2, 4]. Записи типу [5..5], [9..0] відповідають одноелементній множині [5] і порожній множині.

Після завдання базового типу сукупність значень відповідного множинного типу визначається автоматично. У неї входять всі можливі множини, що є довільними комбінаціями значень базового типу. Як відомо, всяка n-елементна множина має 2^n підмножин. Для опису множинного типу використовують конструкцію вигляду: **set of** базовий_тип

Наприклад, множинний тип **set of** 1..3 має базовим типом проміжок цілих чисел [1; 3] і його можливими значеннями є: [1], [2], [3], [1, 2], [1, 3], [2, 3], [1, 2, 3] і [].

Набір операцій над множинами у Паскалі відображає відповідний набір операцій математики:

$A = B$ ($A <> B$) – множини A і B збігаються (не збігаються);

$A \leq B$ – усі елементи множини A належать множині B;

$x \text{ in } A$ – значення x входить у A; $A + B$ – об'єднання A і B;

$A * B$ – перетин A і B; $A - B$ – різниця A і B.

Приклад 5.4.8. Підрахувати кількість цифр та знаків '+' і '-' у рядку символів.

```

program Project548; {$APPTYPE CONSOLE}
var s:string; i,k:byte;
begin write('Enter s'); readln(s);
    k:=0; for i:=1 to length(s) do
        if s[i] in ['0'..'9','+', '-'] then inc(k);
    write('k=', k); readln
end.

```

Приклад 5.4.9. Підрахувати кількість різних цифр у десятковому записі натурального числа n.

```

program Project549; {$APPTYPE CONSOLE}
var n:longword; k:byte; d:0..9; sd: set of 0..9;
begin write('Enter n'); readln(n); sd:=[];

```

```

repeat d:=n mod 10; sd:= sd+[d]; n:=n div 10 until n=0;
k:=0; for d:=0 to 9 do if d in sd then inc(k);
write('k=', k); readln
end.

```

Приклад 5.4.10. Дано текст з малих латинських букв, який закінчується крапкою. Надрукувати перші входження букв у текст, зберігаючи їхній взаємний порядок.

```

program Project5410; {$APPTYPE CONSOLE}
var c: char; st: set of 'a'..'z'; s:string;
begin st:=[]; s:= '';
write('Enter c'); readln(c);
while c<>'.' do
begin
if not(c in st) then begin st:=st+[c]; s:=s+c; end
write('Enter c'); readln(c)
end;
write('s=', s); readln
end.

```

5.4.6. Рядки символів

Тип **String** (*рядок символів* або просто *рядок*) у Паскалі використовується для опрацювання текстів. З символами рядка можна працювати як з елементами масиву символів, але на відміну від масивів, рядки можна вводити цілком, порівнювати один з одним і зчіплювати операцією “+”.

Приклад 5.4.11. Використання операції зчеплення.

```

var s1, s2, x, y:string; z:char;
begin
x:='object'; y:='pascal';
s1:=x+' '+y; // s1='object pascal'
s2:=''; // порожній рядок
for z:='a' to 'z' do s2:=s2+z; // s='abcd...xyz'
writeln(s1, ' ', s2); readln
end.

```

Порівнюють рядки посимвольно до першого незбігу їхніх кодів. Якщо один з рядків закінчився до першого незбігу, то він вважається меншим. Порожній рядок є меншим за будь-який непорожній рядок, наприклад:

```

'abcd'>'abcd' { 'd'>'D' }; 'abcd'>'abc' { 'd'>' ' };
'abc'<'axxc' { 'b'<'x' }; 'abcd'='abcd'

```

Наявна низка стандартних функцій і процедур роботи з рядками:

- **Length(s: string): integer** – видає поточну довжину рядка *s*;
- **Copy(s: string; Index, Count: integer): string** – повертає фрагмент рядка *s*, який починається у позиції *Index* і має довжину *Count*;

- `Pos(s, s1: string): byte` – шукає перше входження підрядка `s` у рядку `s1` і повертає номер першого символу з `s` у рядку `s1` або 0, якщо такого входження немає;

- `Delete(var s: string; Index, Count: integer)` – видаляє з рядка `s` фрагмент, який починається у позиції `Index` і має довжину `Count`;

- `Insert(s: string; var s1: string; Index: integer)` – вставляє у рядок `s1` підрядок `s`, починаючи зі заданої позиції `Index`.

Можливі два способи опрацювання рядків символів:

- опрацювання рядка за допомогою стандартних функцій і процедур роботи з рядками;

- опрацювання рядка як масиву символів: можна звертатися до окремого символу рядка через його індекс, наприклад: `s[i]` – `i`-ий символ рядка `s`.

Приклад 5.4.12. Підрахувати кількість повторення у рядку символу 'x'.

```
var s:string; i,k:byte;
```

```
begin write('Enter s'); readln(s); k:=0;
```

```
  for i:=1 to length(s) do if s[i]='x' then inc(k);
```

```
  write('k=',k); readln
```

```
end.
```

Приклад 5.4.13. Замінити у рядку кожен символ '!' на символ '.'.

```
var s:string; i:byte;
```

```
begin write('Enter s'); readln(s);
```

```
  for i:=1 to length(s) do if s[i]='!' then s[i]='.';
```

```
  write(s); readln
```

```
end.
```

Приклад 5.4.14. Замінити у рядку кожен символ '.' на '...'.
 var s:string; i:byte;

```
begin write('Enter s'); readln(s); i:=1;
```

```
  while i<=length(s) do
```

```
    if Copy(s,i,1)='.'
```

```
      then begin Insert('..',s,i+1); i:=i+3 end
```

```
      else inc(i);
```

```
  write(s); readln
```

```
end.
```

Приклад 5.4.15. Вияснити, чи трапляються у рядку символи ', -'.

```
var s:string; i:byte;
```

```
begin write('Enter s'); readln(s); i:=pos(',-',s);
```

```
  if i=0 then write('No') else write('Yes'); readln
```

```
end.
```

Приклад 5.4.16. Відомо, що у рядку є хоча б один символ ',' (кома). Знайти номер першого і останнього входження ',' у рядок.

```
var s:string; i,k,m:byte;
```

```
begin write('Enter s: '); readln(s);
```

```

k:=pos(',',s); i:=length(s); m:=0;
while (i>=1) and (m=0) do
  begin if copy(s,i,1)=',' then m:=i; i:=i-1 end;
  write('l vh=',k,' Ost.vh=', m);readln
end.

```

Приклад 5.4.17. Порахувати у рядку усі букви 'b' і видалити ті з них, перед якими стоїть буква 'a'.

```

var s:string; c:char; i,k:byte;
begin write('Enter s: '); readln(s);
c:=chr(0); { позначка видалення} s:=' '+s; k:=0;
for i:=2 to length(s) do
  if s[i]='b' then
    begin inc(k); if s[i-1]='a' then s[i]:=c end;
  while pos(c,s)<>0 do delete(s,pos(c,s),1);
  write(s); write('k=', k); readln
end.

```

5.4.7. Файли

Файловий тип даних, або *файл*, визначає впорядковану сукупність довільного числа однотипних *компонент* (або *елементів*), які зберігаються на зовнішніх пристроях запам'ятовування. Кількість компонент файлу може бути довільною. Файли на зовнішніх пристроях часто називають *фізичними файлами*.

Під час роботи з файлами виконують операції введення-виведення. Операція *введення* (*читання* – Read) означає перепис даних зі зовнішнього пристрою (з вхідного файлу) в основну пам'ять комп'ютера, операція *виведення* (*запису* – Write) – це пересилання даних з основної пам'яті на зовнішній пристрій (у вихідний файл).

Для роботи з файлами у програмі необхідно визначити файлову змінну. Паскаль підтримує три файлових типи: *текстові* файли, *компонентні* файли, *безтипові* файли. Описують файлові змінні текстового типу за допомогою слова Text, наприклад:

```
var tStory: Text;
```

Опис компонентних файлів має вигляд: **var fComp: File of T;**

де T – тип компоненти файлу. Приклади опису файлів компонентного типу: **var fl: File of Real; f2: File of Char;**

Безтипові файли описують словом File: **var f: File;**

Текстові файли – це послідовність рядків, а рядки – послідовність символів. Рядки мають змінну довжину, кожний рядок завершується ознакою кінця рядка – EOLN (*end of line*), який зображає символ #13 (переведення каретки – CR), за яким, можливо, розміщений символ #10 (переведення рядка – LF). Закінчується файл символом #26 (закінчення файлу – EOF).

Компонентний (або типізований) файл – це файл з оголошеним типом його компонент. Компонентні файли складаються з машинних зображень значень змінних; вони зберігають дані у тому ж вигляді, що і пам'ять комп'ютера.

Безтипові файли дають можливість записувати на диск довільні ділянки пам'яті комп'ютера і прочитувати їх з диска у пам'ять за допомогою процедури `BlokRead` і `BlockWrite`.

Файлові змінні, що їх описано в програмі, називають *логічними файлами*. Всі основні процедури і функції, що забезпечують введення/виведення, працюють тільки з логічними файлами. Фізичний файл має бути пов'язаний з логічним до виконання процедур відкриття файлів. Паскаль уводить низку процедур і функцій, застосовних для будь-яких типів файлів:

- процедура `Assign` (`var f; FName: String`) зв'яже логічний файл `f` з фізичним файлом, назва якого задана у `FileName`;
- процедура `Reset` (`var f`) відкриває логічний файл `f` для наступного читання даних;
- процедура `Rewrite` (`var f`) створює і відкриває логічний файл `f` для запису даних;
- процедура `Close` (`var f`) закриває логічний файл;
- логічна функція `EOF` (`var f`): `Boolean` повертає `TRUE`, коли при читанні досягнуто кінця файлу;
- `ReadLn` (`T`) – пропускає рядок файлу `T` до початку наступного;
- `WriteLn` (`T`) – завершує рядок файлу `T`, в який проводиться запис, ознакою кінця рядка і переходить до початку наступного.
- `Read` (`T, X1, X2, ... XK`) – читання значень текстового файлу `T` у змінні `X1, X2, ... XK` з одночасним перетворенням їхнього текстового зображення у машинне;
- `Write` (`T, X1, X2, ... XK`) – запис значень змінних `X1, X2, ... XK` у текстовий файл `T` з одночасним перетворенням їхнього машинного зображення текстове.

Паскаль уводить додаткові процедури і функції, застосовні *тільки* до текстових файлів:

- процедура `Append` (`var f: Text`) відкриває вже наявний фізичний файл для дозаписування даних у кінець файлу;
- процедура `Flush` (`var f: Text`) записує дані з буфера у файл незалежно від ступеня його заповнення;
- функція `SeekEOLn` (`var f: Text`): `Boolean` повертає `True`, якщо до кінця рядка залишилися тільки пропуски;
- функція `SeekEOF` (`var f: Text`): `Boolean` повертає `True`, якщо до кінця файлу залишилися рядки, заповнені пропусками.

Приклад 5.4.18. Нехай на диску є текстовий файл `ID.DAT`, який містить числові значення дійсного типу по два числа в кожному рядку –

значення аргументу і функції відповідно. Кількість пар чисел не більша за 20. Скласти програму, яка читає файл, значення аргументу і функції записує в одновимірні масиви, підраховує їхню кількість, виводить на екран дисплея і записує у файл компонентного типу RD.DAT.

```

program Project1; {$APPTYPE CONSOLE}
uses SysUtils;
var rArg, rF: Array[1..20] of Real; inf: Text;
    outf: File of Real; n, l: Integer;
begin Assign(inf, 'ID.DAT'); Assign(outf, 'RD.DAT');
    Reset(inf); Rewrite(outf); n:=0;
    while not EOF(inf) do
        begin n:=n+1; ReadLn(inf, rArg[n], rF[n]) end;
    for l:=1 to n do
        begin WriteLn(l:2, rArg[l]:8:2, rF[l]:8:2);
            write(outf, rArg[l], rF[l]) end;
    readln; close(outf)
end.

```

5.5. Об'єктно-орієнтоване програмування

5.5.1. Основні концепції

Розвитком ідеї модульного програмування є моделювання об'єктів предметної області за допомогою *класів*. Об'єкт, який необхідно моделювати, уявляється людині цілісно, хоча він складається з частин та інших об'єктів. Цілісне уявлення об'єкта є базовим принципом об'єктно-орієнтованого програмування (*ООП*, скор.).

Основним поняттям ООП є *клас* – визначена програмна структура (абстрактний тип), яка має три важливі властивості:

- *інкапсуляції* (“вміст в оболонці”) – об'єднання і локалізація у рамках класу даних і підпрограм їхнього опрацювання;
- *наслідування* – можливість породжувати своїх *нащадків* – класи, які успадковують властивості батьківського класу, а також містять нові дані та підпрограми чи перевизначають по-новому деякі батьківські підпрограми;
- *поліформізму* – можливість для класів-родичів по-різному здійснювати однотипні (і навіть однаково названі) дії, наприклад: дія “намалювати на екрані” має по-різному реалізовуватися для класів-родичів: “точка”, “пряма” і “прямокутник”.

5.5.2. Оголошення класу

Клас – це деякий абстрактний тип, який створюється на основі наявних типів. Окремий клас включає такі *елементи* (або члени), як *поля*,

методи, властивості і події. Поле класу – це змінна довільного типу, що служать для зберігання інформації про об'єкт.

Методи – це процедури і функції, призначені для опрацювання полів. Заголовок методу розміщується в описі класу, а сам код методу знаходиться в розділі реалізації модуля.

Властивості займають проміжне положення між полями і методами. Кожній властивості відповідає поле, що містить її значення, і два методи (читання/запис) доступу до цього поля.

Подія – спеціальна властивість, яка може повідомити про виникнення певної події (натискання клавіші, клацання кнопкою миші тощо). Якщо у програмі є *процедура опрацювання* події об'єкта, то при її виникненні керування передається відповідній процедурі.

Клас має таку *структуру*:

```
type НазваКласу = class (НазваКласуПредка)
public // Відкриті поля, методи властивості і події
private // Закриті поля, методи властивості і події
protected // Захищені поля, методи властивості і події
published // Опубліковані поля та властивості
end;
```

Клас **TObject** є базовим класом для всіх інших класів Object Pascal. Об'єктів цього класу не існує. Нові класи за домовленістю успадковують усі методи класу **TObject**. Якщо батьківським класом є **TObject**, то його назву після слова **class** можна не вказувати. Тоді перший рядок опису класу буде виглядати так:

```
type НазваКласу = class
```

НазваКласу може бути довільним ідентифікатором. Прийнято назву класу починати буквою “**T**” (від слова *Type*).

Розділ **public** містить відкриті члени класу, які видимі у будь-якому місці програми, де доступний сам клас. Специфікатор **public** зазвичай використовують для задання відкритого доступу методам класу, які організують зв'язок об'єкта цього класу із зовнішнім світом.

Розділ **private** містить *закриті* члени класу, які доступні тільки для методів свого класу і підпрограм, які знаходяться у тому ж модулі, що й цей клас.

Розділ **protected** містить *захищені* члени класу, які доступні всередині модуля, в якому вони знаходяться. Члени класу з цього розділу доступні також і для породжених класів за межами названого модуля. Статус захищений (**protected**) використовується в класах при застосуванні механізму успадкування класів.

Розділ **published** складається з опублікованих елементів, область видимості яких така ж сама, як і відкритих елементів. Головним призначенням розділу **published** є забезпечення доступу до властивостей

об'єктів під час конструювання проєктів. У Інспекторові об'єктів видно тільки опубліковані властивості.

Кількість і порядок розміщення розділів **private**, **protected**, **public**, **published** є довільними.

Оскільки класи найчастіше вводять для використання у багатьох програмах, то їх, як правило, описують у розділі **interface** деякого модуля. Реалізацію методів розміщують у розділі **implementation** цього модуля.

Методи, призначені для створення або видалення об'єктів, називаються *конструкторами* і *деструкторами*; в їхніх заголовках стоять ключові слова **constructor** чи **destructor**. Як назви конструкторів/деструкторів у базовому класі TObject і багатьох інших класах використовують назви Create і Destroy. Конструктори чи деструктори, введені програмістом, *доповнюють* конструктор/деструктор базового класу TObject операторами ініціалізації полів і/або властивостей об'єкта під час його створення. Якщо для створення об'єкта певного класу достатньо тільки стандартних дій, то в цьому класі конструктор можна не задавати.

Приклад 5.5.1. Визначити клас TCircle для опрацювання даних про коло. *Поля* – параметри кола: координати центра кола (x, y) і радіус r. *Методи* – зміна параметрів кола, отримання інформації про параметри кола, обчислення довжини кола і площі круга.

```

unit DefTCircle;
interface
  type TCircle=class
  private x,y,r:Real;
  public
    constructor create(x1,y1,r1:Real);
    function getX:Real; // отримання абсциси центра кола
    function getY:Real; // отримання ординати центра кола
    function getR:Real; // отримання радіуса
    function length:Real; // обчислення довжини кола
    function square:Real; // обчислення площі круга
    procedure setX(x1:Real); // задання абсциси центра
    procedure setY(y1:Real); // задання ординати центра
    procedure setR(r1:Real); // задання радіуса
  end;
implementation
  constructor TCircle.Create;
  begin x:=x1; y:=y1; r:=r1 end;
  function TCircle.getX;   begin getX:=x end;
  function TCircle.getY;   begin getY:=y end;
  function TCircle.getR;   begin getR:=r end;

```

```

function TCircle.length; begin length:=2*Pi*r end;
function TCircle.square; begin square:=Pi*r*r end;
procedure TCircle.setX; begin x:=x1 end;
procedure TCircle.setY; begin y:=y1 end;
procedure TCircle.setR; begin r:=r1 end;
end.

```

5.5.3. Створення і використання об'єктів

Оголошення класу ще не задає об'єкта, а тільки визначає тип того об'єкта, який буде створений у програмі пізніше. Отже, після оголошення класу TCircle можна визначити один або декілька об'єктів, як екземпляри цього класу. Для описаного вище класу TCircle, визначення об'єктів може виглядати так:

```
var c1, c2, c3: TCircle;
```

Тут визначені об'єкти c1, c2 і c3 типу (класу) TCircle.

Перш ніж звертатися до елементів об'єкта, йому треба *створити* за допомогою *конструктора* і, при потребі, задати у ньому початкові значення полів об'єкта. Виклик конструктора може виконуватися двома способами: за допомогою задання *назви класу* або *назви конкретного об'єкта*. При створенні нового об'єкта виклик конструктора реалізується тільки через *назву класу*, наприклад:

```
c1:=TCircle.Create;
```

Якщо конструктор Create доповнювався у відповідному класі, то під час виконання він може ініціалізувати поля і/або властивості об'єкта за допомогою значень, що передаються як аргументи конструктора, наприклад:

```
c2:=TCircle.Create(x2, y2, r2);
```

Конструктор класу під час створення об'єкта виконує такі дії:

- реалізує стандартні дії зі створення об'єкта (викликається конструктор **Create** класу **TObject**);
- виконує дії, які задані програмістом.

Якщо конструктор викликається як метод уже наявного об'єкта (за допомогою *назви цього об'єкта*), то виконуються тільки дії конструктора, які задані програмістом. У такий спосіб, як правило, викликають конструкторів-нащадків. Для доступу до відкритих полів, властивостей чи методів об'єкта використовують складені назви: об'єкт.поле/властивість або об'єкт.метод[аргументи]

Приклад 5.5.2. Використати клас TCircle для програмування задач:

1. На площині задати 15 кіл.
2. Надрукувати інформацію про кола, довжина яких перевищує задане значення L.
3. Кола, площею меншою за 2,75, пересунути на початок координат.

```

program Project2;
{$APPTYPE CONSOLE}
uses SysUtils, DefTCircle in 'DefTCircle.pas';
const NC=15;
var circles: array[1..NC] of TCircle;
    x,y,r:Real; i:byte; C,L,S:Real;
begin // Реалізація задачі 1
    writeln('Enter centr and radius');
    for i:=1 to NC do
        begin read(x,y,r); circles[i]:=TCircle.Create(x,y,r) end;
    // Реалізація задачі 2
    writeln('Enter limit of C'); read(L);
    for i:=1 to NC do
        begin C:= circles[i].length;
            if C>L then
                begin r:=circles[i].getR; x:=circles[i].getX;
                    y:=circles[i].getY; writeln('1: Centr: ',x:6:1,
                        ' ',y:6:1,' R=',r:6:1) end
                end;
    // Реалізація задачі 3
    for i:=1 to NC do
        begin S:=circles[i].square;
            if S<2.75 then
                begin r:=circles[i].getR;
                    circles[i].setX(0); circles[i].setY(0);
                    x:=circles[i].getX; y:=circles[i].getY;
                    writeln('2: Centr: ', x:6:1,' ',
                        y:6:1,' R=',r:6:1) end
                end;
    end;
end.

```

5.5.4. Наслідування та поліформізм

Властивість *наслідування* означає, що будь-який клас може бути породжений іншим класом, як-от:

```
TNewClass=class(TOldClass) ...
```

Клас-нащадок автоматично успадковує всі поля, методи і властивості класу-предка і може доповнюватися новими. Але разом з тим видалити або перевизначити поля класу-предка у класі-нащадкові неможливо.

Властивість *поліформізму* дає можливість використовувати однакові назви для методів, які входять у різні класи. При звертанні до таких методів буде виконуватися той з них, що відповідає класові об'єкта. Ця властивість забезпечується через *перекриття* у класові-нащадкові відповідного методу батьківського класу.

Метод, оголошений в класі, можна викликати різними способами, що залежать від *типу методу*, який визначає механізм його перекриття у класах-нащадках. За домовленістю методи класу є *статичними* і викликаються як звичайні підпрограми. Якщо у класові-нащадкові є статичний метод з назвою, що збігається з назвою деякого статичного методу класу-предка, то дія методу класу-предка відмінюється для класу-нащадка повністю. Заміщення статичних батьківських методів методами нащадків відбувається на етапі компілювання програми.

Динамічні та віртуальні методи відрізняються від статичних тим, що заміщення батьківських методів методами нащадків відбувається під час виконання програми. Для оголошення динамічного/віртуального методу в батьківському класі використовують слово **dynamic/virtual**, а у класі-нащадкові – слово **override**. Різниця між віртуальними та динамічними методами полягає в особливостях реалізації їхніх викликів.

Якщо при створенні об'єктів класу-нащадка треба виконати додаткові дії, які відсутні у предкові, то у конструкторі нащадка треба попередньо викликати конструктор предка, а потім вже здійснювати додаткові дії. Викликати будь-який *перекритий метод* предка, у т.ч. конструктор/деструктор, можна за допомогою слова **inherited** (успадкований).

Приклад 5.5.3. Для класу TCircle побудувати клас-нащадок TRing для опрацювання даних про кільце (два концентричні кола). Доповнити його новим полем Q – для зберігання радіуса другого кола. Перекрити метод обчислення площі круга (для площі кільця). Використати класи TCircle і TRing для програмування задач:

1. На площині задати масив з восьми об'єктів (перші три – класу TCircle, а решта – класу TRing).

2. Надрукувати інформацію про площі об'єктів.

```

unit DefTRing;
uses DefTCircle in 'DefTCircle.pas';
  // У DefTCircle метод square має такий вигляд:
  // function square:Real; virtual;
interface TRing=class(TCircle) private Q:Real;
  public constructor create(x1,y1,r1,q1:Real);
  function square:Real; override; end;
implementation
  constructor TRing.create;
  begin inherited Create(x1,y1,r1);Q:=q1 end;
  function TRing.square; begin square:=Pi*Abs(r*r-Q*Q) end
end.

```

→ Програма тестування:

```

program Test;
{$APPTYPE CONSOLE}

```

```

uses SysUtils, DefTRing in 'DefTRing.pas';
var list: array[1..8] of TCircle;
    x,y,r,q:Real; i:byte; S:Real;
begin // реалізація задачі 1
  for i:=1 to 8 do
    begin if i<4
      then begin writeln('Enter centr and radius');
        read(x,y,r); list[i]:=TCircle.Create(x,y,r) end
      else begin writeln('Enter centr and 2 radiuses');
        read(x,y,r,q); list[i]:=TRing.Create(x,y,r,q) end
      end; // реалізація задачі 2
    for i:=1 to 8 do
      begin S:=List[i].Square; writeln('Square=',S) end;
    readln
  end.

```

5.6. Програмування віконних застосунків

5.6.1. Бібліотека візуальних компонентів (VCL)

Основою створення проектів у Delphi є компоненти та об'єкти, класи яких розміщені у *бібліотеці візуальних компонентів* (Visual Component Library, VCL). *Компоненти* – екземпляри класів, які є нащадками класу TComponent. Екземпляри інших класів називають *об'єктами* (наприклад, об'єкти класу TFont). Об'єкти виконують допоміжні функції й не можуть розміщуватися на формі. Деякі компоненти є видимими тільки на етапі проектування інтерфейсу користувача (наприклад, компонент Timer). Коли компонент розміщується на формі, Delphi генерує для нього код класу на основі відповідного класу VCL.

Важливі нащадки класу TComponent:

- TControl – базовий клас усіх візуальних компонентів, що мають властивості та методи визначення розмірів компонента, його розміщення на формі, вирівнювання, видимості тощо. Нащадки цього класу мають спільну назву – *елементи керування* (ЕК). У цьому класі визначені *події миші*.
- TwinControl – базовий клас усіх віконних компонентів, що мають властивості та методи визначення вигляду вікна, фокусування, процесів перетягування тощо. У цьому класі визначені *події клавіатури*.
- TGraphicControl – базовий клас *невіконних* компонентів (графічні компоненти, кнопки тощо). Визначена властивість Canvas (полотно) – основа зображень.

У класі TControl розрізняють *батьківський* і *дочірній* елемент керування (ЕК). Батьківським може бути тільки *віконний* ЕК (нащадок

TWinControl). Зазвичай батьківським ЕК є *форма* або *компонент групування* (наприклад, *панель* – клас TPanel).

Клієнтська область компонента – це доступна користувачеві частина компонента. Саме у клієнтській частині користувач вводить значення (рядки символів) чи малює зображення. У клієнтську область не входить *обрамлення* компонента (заголовок, меню, рамка, статусний рядок тощо).

5.6.2. Основні властивості та події класу TControl

Для відображення елементів керування (ЕК) на екрані використовують властивості *позиціонування*:

- Left / Top – задає горизонтальну/ вертикальну координату лівого верхнього кута ЕК (у пікселях) щодо компонента-батька;
- Width – задає ширину ЕК (у пікселях);
- Height – задає висоту ЕК (у пікселях).

Наведемо інші важливі властивості класу TControl:

- Align – визначає *спосіб вирівнювання* компонента у батьківському компоненті;
- Anchors – визначає *прив'язку* компонента до меж батьківського під час зміни розмірів останнього;
- AutoSize – визначає, чи будуть розміри компонента автоматично пристосовуватися до розмірів його вмісту;
- Caption – визначає рядок тексту, який ідентифікує компонент для користувача (напис на позначці, кнопки тощо);
- ClientHeight – визначає висоту клієнтської області (у пікселях);
- ClientWidth – визначає ширину клієнтської області (у пікселях);
- Color – визначає колір фону елемента керування;
- Enabled – визначає, чи реагує компонент на події, зв'язані з мишею, клавіатурою і таймером;
- Font – визначає характеристики шрифту;
- Hint – містить текст, що відображається у вікні підказки;
- ShowHint – дає змогу або забороняє показувати вікно підказки;
- Text – містить текст, що асоціюється з цим ЕК;
- Visible – визначає, чи відобразиться ЕК на екрані.

Операційною системою Windows керують події, які виникають унаслідок маніпуляцій з мишею, натискання клавiш на клавіатурі, події таймера тощо. Кожний компонент спроектовано так, щоб він реагував на відповідні події. Так, компонент Button (кнопка) буде реагувати на кліцання мишею.

Реакцію користувача на виникнення події програмують за допомогою *процедур опрацювання події* (event handlers). Список подій, на які реагує компонент, наведено на закладці Events вікна інспектора об'єктів. У програмі немає потреби опрацьовувати усі події, на які реагує компонент.

Якщо подія не опрацьована, то відповідне повідомлення або відкидається без реагування, або виконуються дії за домовленістю.

У класі TControl описані події, які виникають при користуванні мишею, зокрема: OnClick (клацання лівою кнопкою миші); OnDblClick (подвійне клацання лівою кнопкою миші); OnMouseDown/MouseUp (натискання/відпускання кнопки миші, коли курсор миші знаходиться над ЕК); OnMouseMove (переміщення миші, коли її курсор знаходиться над ЕК).

Кожна процедура опрацювання події буде мати принаймні один параметр Sender, який представляє у процедурі компонент, який посилає повідомлення цій процедурі. У випадку процедури опрацювання події для конкретного компонента параметр Sender, у принципі, є зайвим (однак *опускати його не можна*). Взагалі, Sender дає змогу мати процедуру опрацювання однієї події одночасно для декількох компонентів (приклад 5.6.2).

5.6.3. Побудова проекту віконного застосування

Процес побудови будь-якого *проекту* віконного застосування у Delphi складається з таких основних етапів:

1. Проектування інтерфейсу користувача (етап *візуального програмування*).
2. Програмування процедур опрацювання подій та інших необхідних підпрограм (етап *кодування*).
3. Налаштування і запуск проекту на виконання.

Для створення найпростіших віконних проектів будемо використовувати компоненти введення/виведення даних (*позначку і текстові редактори*) і компонент ініціалізації дій (*командну кнопку*).

Текстовий підпис або *позначка* є компонентом класу TLabel, який використовують для відображення незмінного тексту. Властивість Caption містить текст, який зазвичай є поясненням (розгорнутою назвою) іншого компонента. Часто позначку використовують і для відображення результатів роботи програми.

Компонент класу TEdit (однорядковий текстовий редактор чи поле введення) задає вікно введення і редагування одиночного рядка тексту, який є значенням властивості Text.

Компонент класу TMemo є вікном редагування багаторядкових текстів. Зміст області редагування визначає масив рядків, який є значенням властивості Lines.

Компонент класу TButton є командною кнопкою, яка ініціалізує якусь дію. Напис на кнопці задається властивістю Caption. Основна подія – OnClick, що виникає при клацанні мишею на кнопці.

При розміщенні на формі компонент отримує стандартну назву, як-от: Label1, Label2, ..., Edit1, Edit2, ..., Button1 і т.п. Ця назва

зберігається у властивості Name (у великих проєктах назви компонентів зазвичай змінюють на осмисленіші, які відповідають їхньому призначенню).

Приклад 5.6.1. Побудуємо проєкт віконного застосування, в якому будемо з'ясовувати, чи уведений рядок є *паліндромом* (читається однаково як зліва направо, так і справа наліво). На формі розміщено дві позначки, дві командні кнопки і поле введення. Після натиснення кнопки з написом “Аналіз” перевірятиметься рядок, введений користувачем у поле введення. Кнопка “Вихід” закриває форму.

Перший етап – *проектування інтерфейсу користувача.*

Оскільки *проєкт* – це множина різних спеціальних файлів, то передусім необхідно створити каталог для зберігання цих файлів. Виконавши команди **File > New Application**, отримаємо нове порожнє вікно форми (Form1). Далі послідовно виконаємо такі дії:

- змінимо за допомогою інспектора об'єктів стандартний заголовок форми “Form1” (значення Caption) на заголовок “Паліндром”;
- на формі розміщуємо позначку Label1 і змінимо значення її властивості Caption на “Введіть рядок.”;
- на формі розміщуємо позначку Label2 і змінимо значення її властивості Caption на “Результат.”;
- на формі розміщуємо поле введення Edit1;
- на формі розміщуємо командну кнопку Button1 і змінимо значення її властивості Caption на “Аналіз”;
- на формі розміщуємо командну кнопку Button2 і змінимо значення її властивості Caption на “Вихід”;
- за допомогою мишки встановимо розміри форми та компонентів на ній приблизно такими, як на рис. 5.2.

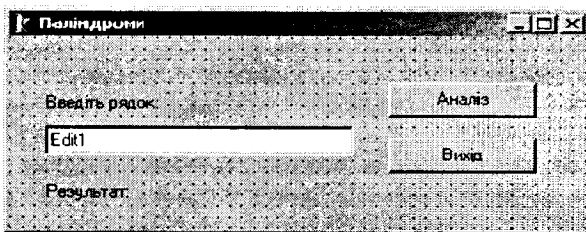


Рис. 5.2

Далі варто зберегти результати нашої роботи. Виконавши команди **File > Save Projects As...** отримаємо послідовно два діалогові вікна: **Save Unit1 As** (збережемо файл форми *Unit1.pas*) і **Save Project1 As** (збережемо файл проєкту *Project1.dpr*). На цьому перший етап побудови застосування завершено.

Як бачимо, проєкт простий, а слів для опису етапу проектування інтерфейсу сказано немало. Описуючи наступні проєкти, користуватимемося

скороченою формою запису, опускаючи стандартні кроки відкриття нової форми та збереження результатів роботи. Скорочено етап проектування інтерфейсу нашого першого проекту віконного застосування можна описати так:

```
Form1: Label1; Label2; Edit1 // розміщуємо компоненти
      Button1; Button2
Form1.Caption := 'Паліндроми'
Label1.Caption:= 'Введіть рядок: \'
Label2.Caption:= 'Результат: \'
Button1.Caption:= 'Аналіз'; Button2.Caption:= 'Вихід'
```

Другий етап – програмування методів опрацювання подій.

Для обох кнопок необхідно запрограмувати процедуру опрацювання події `OnClick`. Клацнувши два рази на кнопці `Button1` перейдемо на сторінку редактора коду, на якій побачимо заготовку:

```
procedure TForm1.Button1Click(Sender: TObject);
begin end;
Запрограмуємо процедуру Button1Click:
procedure TForm1.Button1Click(Sender: TObject);
var s,s1:string;i,n:byte;
begin s1:='рядок \'; Label2.Caption:='Результат: \';
      s:=Edit1.Text;
      while pos(' ',s)<>0 do delete(s,pos(' ',s),1);
      i:=1; n:=length(s);
      while (i<=n div 2) and (s[i]=s[n-i+1]) do inc(i);
      if i<=n div 2 then s1:=s1+'- не паліндром'
      else s1:=s1+'- паліндром';
      Label2.Caption:=label2.Caption+s1
end;
```

Процедура опрацювання події `OnClick` для кнопки `Button2`:

```
procedure TForm1.Button2Click(Sender: TObject);
begin close end;
```

Третій етап – налагодження і запуск проекту на виконання.

Натиснемо на кнопку **Run**. Після декількох секунд, які витрачаються на компілювання та компонування проекту, отримаємо вікно застосування. Увівши декілька різних рядків (паліндромів і не паліндромів) впевнюємося, що завдання виконане.

5.6.4. Основні події класу `TWinControl`

Клас `TWinControl` є нащадком класу `TControl` і використовується як базовий для створення віконних ЕК, які під час виконання програми можуть отримувати фокус уведення і реагувати на події, які виникають при використанні клавіатури.

Характерними представниками елементів `TWinControl` є однорядковий редактор `Edit`, багаторядковий редактор `Memo`, список `ListBox`, кнопка `Button` і т.д.

У класі `TControl` описані події, які виникають при користуванні клавіатурою: `OnKeyDown/OnKeyUp` (момент початку натискання/відпускання клавіші); `OnKeyPress` (натискання символічної клавіші). Усі події викликаються для *активного* ЕК (має фокус уведення) при натисканні клавіші.

Якщо ЕК одержує *фокус*, то це певним чином відображається на екрані – текстове поле отримує мерехтливий маркер уведення, а кнопка виділяється пунктиром. З передачею фокуса пов'язані події `OnEnter` і `OnExit`. Якщо перейти від одного ЕК до іншого, то для попереднього ЕК викликається подія `OnExit`, а для нового – `OnEnter`. Передача фокуса програмно здійснюється методом `SetFocus`.

Подія `OnKeyPress` виникає, коли натиснута *символьна* клавіша. Після цього повертається *ANSI-код* натиснутої клавіші. Можуть перехоплюватися тільки такі спеціальні клавіші, як `[Enter]`, `[Esc]` і `[Backspace]`. Параметр `Key` є параметром-змінною, тобто його можна змінювати у процедурі опрацювання події. Це використовують для *фільтрування символів*, що вводяться користувачем, – якщо символ неприпустимий, то, встановивши значення `Key` рівним нулю, запобігають його передачі для дальшого опрацювання.

Події `OnKeyDown/OnKeyUp` викликаються навіть при натисканні спеціальних клавіш керування, наприклад, функціональних. Параметр `Key` і містить *клавіатурний код* натиснутої клавіші, а параметр `Shift` інформує про стан клавіш `[Shift]`, `[Ctrl]` і `[Alt]`.

Якщо властивість форми `KeyPreview` дорівнює `True`, то подія, пов'язана з клавіатурою, передається спочатку формі, а потім поточному ЕК (форма може попередньо аналізувати повідомлення). Цей факт широко використовують при розробці проектів.

Приклад 5.6.2. Знаходження дійсних коренів квадратного рівняння. У програмі будемо аналізувати коди клавіш, які натискатиме користувач при введенні числових даних для коефіцієнтів рівняння. Символи, які не сумісні з *представленням чисел*, будемо блокувати. У проекті властивість форми `KeyPreview` дорівнює `True`, тому події, пов'язані з клавіатурою, передаються формі.

Етап проектування інтерфейсу проекту:

```
Form1: Label1; Label2; Label3; Label4
  Edit1; Edit2; Edit3; Button1; Button2; Button3
Form1.Caption:= ' Квадратні рівняння '
Form1.KeyPreview:= True // аналіз на рівні форми
Button1.Caption:= 'Корені'
```

```

Button1.Caption:= 'Нові значення'
Button3.Caption = 'Вийти'
Edit1.Text:= ''; Edit2.Text:= ''; Edit3.Text:= ''
Label2.Caption:= 'Коефіцієнт А (<>0) ' // до Edit1
Label3.Caption:= 'Коефіцієнт В' // до Edit2
Label4.Caption:= 'Коефіцієнт С' // до Edit3
Label1.Caption:= '' // результат у Label1. Caption
Label1.WordWrap:= True; Label1.AutoSize:= True
Етап програмування процедур опрацювання подій:
procedure TForm1.EditKeyPress(Sender: TObject; var Key: Char);
var buf:string[20];
begin if Sender = Edit1 then buf:=Edit1.Text
      else if Sender = Edit2 then buf:=Edit2.Text
      else buf:=Edit3.Text;
case key of
  '0'.. '9', chr (8): ; // допустимі символи
  // Символ '\,' має бути один!
  '\,': if pos('\,',buf)<>0 then key:=chr(0);
  // Символ '-' має бути першим!
  '-': if length(buf)<>0 then key:=chr(0);
  // Завершення введення. Перехід на наступний елемент
  chr(13): if Sender = Edit1 then Edit2.SetFocus
          else if Sender = Edit2 then Edit3.SetFocus
          else button1.SetFocus;
  else key:=Chr(0) // блокування недопустимих символів
end end;
procedure TForm1.Button2Click(Sender: TObject);
begin Edit1.text:=''; Edit2.text:=''; Edit3.text:='';
      Label1.caption:=''; Edit1.SetFocus
end;
procedure TForm1.Button3Click(Sender: TObject);
begin Form1.Close end;
procedure TForm1.Button1Click(Sender: TObject);
var a,b,c:real; // Коефіцієнти рівняння
      d:real; {Дискримінант} x1,x2:real; // Корені рівняння
begin
  if (Edit1.Text='')or(Edit2.Text='')or(Edit3.Text='')
    then begin ShowMessage('Задайте коефіц.!!'); exit end;
  a:=StrToFloat(Edit1.text);
  b:=StrToFloat(Edit2.text);
  c:=StrToFloat(Edit3.text);
  if a=0 then begin Label1.caption:= 'Коефіцієнт a=0!»; end

```

```

else
begin { Обчислення коренів } d:=Sqr(b)-4*a*c;
  if d<0 then
    begin Labell.caption:=' Дійсних коренів нема!' end
  else
    begin x1:=(b+Sqrt(d))/(2*a); x2:=(-b+Sqrt(d))/(2*a);
      Labell.caption:=' Дійсні корені: '
      + #13 + 'x1=' + FloatToStrF(x1,ffGeneral,7,4)
      + #13 + 'x2=' + FloatToStrF(x2,ffGeneral,7,4);
    end end end;

```

Процедура TForm1.EditKeyPress має *універсальне* значення і може використовуватися у всіх випадках уведення чисел.

5.6.5. Компоненти відображення списків рядків

Компоненти класів TListBox і TComboBox відображають списки рядків і дають змогу користувачеві вибрати певне значення з деякої множини значень. Ці компоненти відрізняються один від одного передусім тим, що TListBox тільки *відображає* дані, а TComboBox дає можливість також і *редагувати* дані. Крім того, TListBox відображає список у розкритому вигляді і автоматично додає у список вертикальну смугу прокручування, якщо всі рядки не вміщуються у вікні компонента; TComboBox дає змогу відображати список як у розгорненому вигляді, так і у вигляді випадаючого списку.

Основною властивістю компонентів є властивість Items. Заповнити список (властивість Items) під час проектування можна, якщо натиснути кнопку з трьома крапками біля цієї властивості у вікні інспектора об'єктів. Після цього відкривається вікно редагування списку. Кожний записаний у вікні рядок буде відповідати окремому рядку списку. Доповнювати список рядками під час виконання програми можна за допомогою методу Add.

Індекс вибраного рядка у списку рядків зберігає властивість ItemIndex, яка доступна тільки під час виконання програми. Якщо жоден рядок не вибраний, то ItemIndex = -1.

Приклад 5.6.3. Проект переведення фунтів у кілограми (рис. 5.3). У різних країнах коефіцієнт переведення фунтів у кілограми є різним, а тому у проекті буде створено список країн. Залежно від

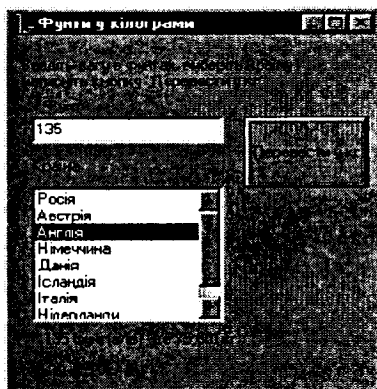


Рис. 5.3

вибору країни у списку програма вибиратиме відповідний коефіцієнт перерахунку і перераховуватиме фунти у кілограми.

Етап проектування інтерфейсу проекту:

Form1: Button1; Label1; ... Label4; Edit1; ListBox1

Label1.Caption:= 'Введіть вагу в фунтах, виберіть країну і натисніть кнопку «Перевести у кг»'

Label2.Caption:= 'Фунти' Label3.Caption:= 'Країна'

Label4.Caption:= '' ; Edit1.Text:= ''

Button1.Caption:= 'Перевести у кг'

ListBox1.Items:= '' //Заповнення списку у програмі

Етап програмування процедур опрацювання подій:

procedure TForm1.FormCreate(Sender: TObject);

begin ListBox1.Items.Add('Росія');

ListBox1.Items.Add('Австрія');

ListBox1.Items.Add('Англія');

ListBox1.Items.Add('Німеччина');

ListBox1.Items.Add('Данія');

ListBox1.Items.Add('Ісландія');

ListBox1.Items.Add('Італія');

ListBox1.Items.Add('Нідерланди');

ListBox1.ItemIndex:=0;

end;

procedure TForm1.Button1Click(Sender: TObject);

var funt:real; { вага у фунтах }

kg:real; { вага у кілограмах }

k:real; { коефіцієнт перерахунку }

begin

case ListBox1.ItemIndex **of**

0: k:=0.4059; { Росія }

1: k:=0.453592; { Англія }

2: k:=0.56001; { Австрія }

3..5,7:k:=0.5; { решта країн }

6: k:=0.31762; { Італія }

end; Label4.Caption:='';

try funt:=StrToFloat(Edit1.Text); kg:=k*funt;

label4.caption:=Edit1.Text + ' фунт(a/ів) - це '

+ FloatToStrF(kg, ffFixed, 6, 3) + 'кг.';

except on EConvertError **do** // помилка перетворення

begin ShowMessage('Задайте число.');

form1.Edit1.SetFocus

end

end end;

Компонент класу TListBox має властивість MultiSelect, яка дає змогу користувачеві здійснити множинний вибір у списку. У разі множинного вибору (MultiSelect = true) значення ItemIndex відповідає тому елементу списку, який знаходиться у фокусі. При множинному виборі перевірити, чи вибрано даний елемент, можна за допомогою властивості Selected.

Компонент ListBox не допускає появи горизонтальної смуги прокручування. Тому процедурі FormCreate необхідно посилати повідомлення Windows з вимогою дорисувати до списку горизонтальну смугу прокручування (виконання методу Perform).

Приклад 5.6.4. Проект вибору страви з меню (рис. 5.4). Меню сформуємо на етапі проектування інтерфейсу. Передачу вибраних страв у компонент Memo1 реалізуємо через множинний вибір.

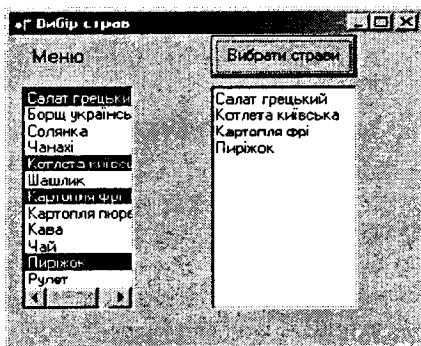


Рис. 5.4

Етап проектування інтерфейсу:

```
Form1: Button1; Label1; Label2; Memo1; ListBox1
Label1.Caption:='Меню'; Label2.Caption:='Замовлено'
Memo1.Lines <input type="checkbox"> (...):=>>; Memo1.ScrollBars:=ssBoth
ListBox1.Items <input type="checkbox"> (...):=><Салат грецький; Борщ український; ... Рулет'
ListBox1.MultiSelect:=true; Form1.Caption:='Вибір страв'
Button1.Caption:= 'Вибрати страви'
```

Етап програмування процедур опрацювання подій:

```
procedure TForm1.FormCreate(Sender: TObject);
var maxWidth, w, i:integer; text:String;
begin
  maxWidth:= 0;
  with ListBox1 do
    begin // Знаходимо максимальну довжину рядка
      for i:=0 to Items.Count-1 do
        begin text:=Items.Strings[i];
          w:=Canvas.TextWidth(text);
          if w> maxWidth then maxWidth:=w;
        end; perform(LB_SETHORIZONTALEXTENT, maxWidth, 0);
      end; Memo1.ReadOnly:=true;
    end;
end;
procedure TForm1.Button1Click(Sender: TObject);
var i:integer; text:String;
```

```
begin if Memol.ReadOnly then Memol.ReadOnly:=false;
  Memol.Lines.Clear;
  with ListBox1 do
    for i:=0 to Items.Count-1 do
      if Selected[i] then
        begin text:=Items.Strings[i];
          Memol.Lines.Add(text) end;
    Memol.ReadOnly:=true;
  end;
```

Запитання для самоперевірки

1. Що розуміють під типом даних?
2. Як зображують окремі символи та рядки символів у мові Паскаль?
3. Що таке вираз? Вкажіть правила обчислення виразу.
4. Запишіть і охарактеризуйте оператор циклу **for**.
5. Запишіть і охарактеризуйте оператор циклу **while ... do**.
6. Запишіть і охарактеризуйте оператор циклу **repeat ... until**.
7. Як порівнюються рядки символів?
8. Що таке масив? Як описують тип масиву?
9. Що таке запис? Як описують тип запису?
10. Для чого використовують оператор приєднання?
11. Що таке множина? Як описують множину? Операції над множинами.
12. Як описують процедуру? Як описують функцію?
13. Що таке оператор процедури? Як реалізовується виклик функції?
14. Що таке параметри-значення? Що таке параметри-змінні?
15. Що таке модуль? Яка структура модуля у мові Паскаль?
16. Перерахуйте й охарактеризуйте типи файлів.
17. Дайте означення класу. Інкапсуляція, наслідування, поліморфізм.
18. Для чого використовуються поля класу? Що таке методи класу?
19. Дайте означення конструктора і деструктора.
20. Коротко охарактеризуйте етап візуального програмування.
21. Коротко охарактеризуйте етап кодування.
22. Для чого у проекті використовують позначки?
23. Для чого у проекті використовують поля введення?
24. Для чого у проекті використовують командні кнопки?

6. ТЕКСТОВИЙ ПРОЦЕСОР Microsoft Word

📄 План викладу матеріалу

1. Інтерфейс програми Word.
2. Робота з документами.
3. Введення і редагування тексту.
4. Форматування тексту.
5. Пошук та заміна фрагментів тексту.
6. Використання мовних функцій Word.
7. Таблиці.
8. Вставлення та форматування ілюстрацій.
9. Математичні формули.
10. Основи верстання. Вставлення тексту.
11. Формування зовнішнього вигляду документа.
12. Робота зі стилями.
13. Шаблони.
14. Робота з ілюстраціями та ілюстративними засобами.
15. Виноски.
16. Зміст документа.
17. Макрокоманди.
18. Друкування документа.

🔑 Ключові терміни розділу

✓ <i>Текстові процесори</i>	✓ <i>Верстання</i>
✓ <i>Документи Word</i>	✓ <i>Розділи документа</i>
✓ <i>Введення тексту</i>	✓ <i>Колонтитули</i>
✓ <i>Форматування тексту</i>	✓ <i>Стилі</i>
✓ <i>Списки</i>	✓ <i>Шаблони Word</i>
✓ <i>Ілюстрації</i>	✓ <i>Зміст документа</i>
✓ <i>Таблиці</i>	✓ <i>Друкування документа</i>

Microsoft Word давно відомий як потужний текстовий процесор, що надає великі можливості користувачам різного рівня та кваліфікації. Зручний інтерфейс і можливість його налаштування під конкретного користувача, розгалужена система довідок та підказок, відкритість для модифікацій і доповнень з боку самого користувача вигідно вирізняють цю програму серед інших. Крім звичайних засобів редагування тексту, Microsoft Word має цілу низку властивостей, які дають змогу віднести його до видавничих систем середнього класу.

6.1. Інтерфейс програми Word

Загальний вигляд вікна програми Microsoft Word 2003 показано на рис. 6.1.

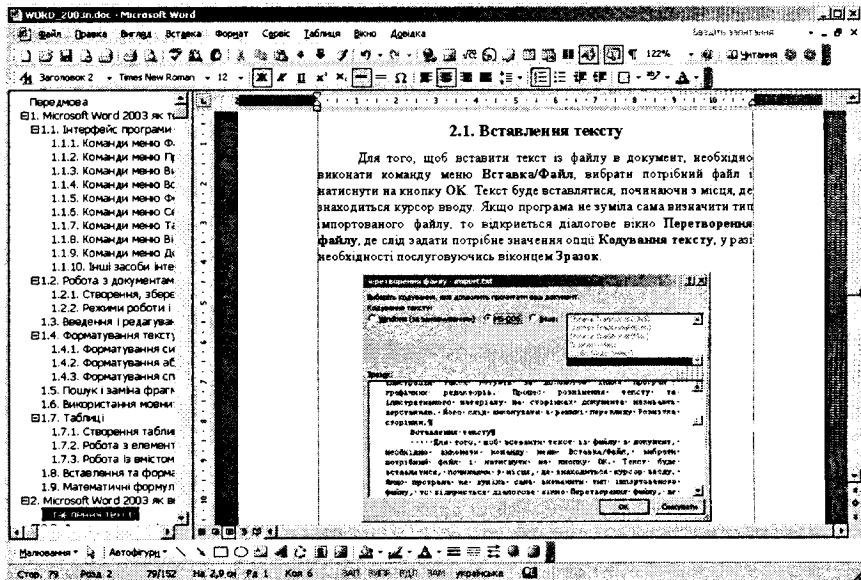


Рис. 6.1

Доступ до всіх функцій програми Word 2003 можна отримати через її головне меню. Воно містить дев'ять позицій: **Файл**, **Правка**, **Вигляд**, **Вставка**, **Формат**, **Сервіс**, **Таблиця**, **Вікно** і **Довідка**. Наявність після назви команди трьох крапок означає, що після її вибору буде відкрите діалогове вікно для конкретизації завдання.

Крім головного меню, можна також використовувати локальні (контекстні). Вони відкриваються після натискання на праву клавішу миші, а їхній вигляд залежить від того, на якому фрагменті документа перебував у цей момент курсор: на текстовому абзаці, ілюстрації чи іншому об'єкті. Локальні меню містять переважно пов'язані з цим фрагментом команди головного меню.

Ще один спосіб виконання окремих команд – натискання спеціальних комбінацій клавіш, які пов'язані з цими командами (їх можна побачити у головному меню праворуч від назв команд).

Багато команд можна виконати, натискаючи на піктограми, що містяться на так званих панелях інструментів. Для їх візуалізації використовують команду **Вигляд/Панелі інструментів**. Вона відкриває підменю зі

списком найвживаніших панелей, де треба ввімкнути або вимкнути перемикач ліворуч від потрібного імені. Щоб побачити повний список, слід виконати підкоманду **Настройка** і в однойменному діалоговому вікні вибрати вкладку **Панелі інструментів**. Тут також можна візуалізувати або приховати потрібні панелі, після чого натиснути на кнопку **Закрити**.

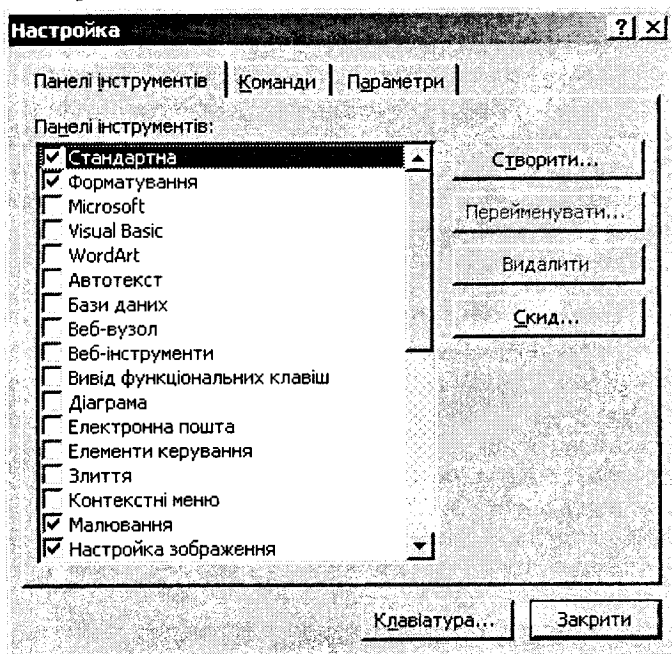


Рис. 6.2

Ще один засіб комунікації з програмою – так звана область завдань. Це спеціальне вікно, що відкривається праворуч від робочої області і містить поля вводу, кнопки, піктограми і команди у вигляді гіперпосилань. Для його візуалізації чи приховання використовують команду **Вигляд/Область завдань**, конкретну ж область можна вибрати, розкривши список у заголовку вікна. Деякі команди меню (наприклад, **Файл/Створити**, **Формат/Стилі та форматування** та інші) відразу відкривають потрібну область завдань.

6.2. Робота з документами

6.2.1. Створення, збереження та відкривання документа

Для створення документа у Word 2003 треба виконати команду **Файл/Створити**, яка відкриває область завдань **Створити документ**.

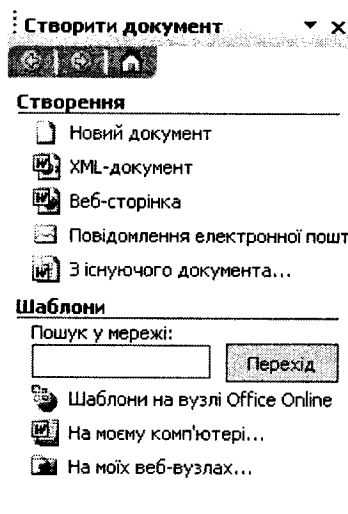


Рис. 6.3

Виконання певної команди ділянки **Створення** призведе до створення нового документа відповідного типу, причому команда **Новий документ** використовує для цього стандартний шаблон **Normal.dot**, а команда **З існуючого документа** дає змогу створити непоіменовану копію вибраного у діалоговому вікні файлу. Команда **На моєму комп'ютері** ділянки **Шаблони** відкриває однойменне діалогове вікно для вибору шаблону, на якому базуватиметься новий документ.

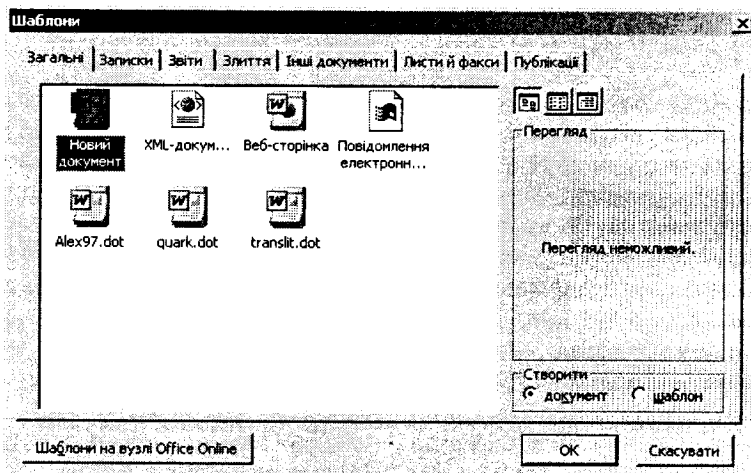


Рис. 6.4

У вікні треба:

- вибрати потрібну вкладку з шаблонами, а в ній – сам шаблон, вигляд якого з'явиться (якщо це можливо) у віконці **Перегляд**;
- надати опції **Створити** значення документ;
- натиснути на кнопку **ОК** (натискання кнопки **Скасувати**, як і в інших діалогових вікнах, відміняє виконання операції).

Щоб швидко створити документ на базі стандартного шаблону **Normal.dot** (**Новий документ**), можна також скористатися піктограмою **Створити файл** на панелі інструментів **Стандартна**. Кнопка **Шаблони** на вузлі **Office Online** дає змогу скористатися шаблонами, розміщеними на сайті Microsoft Office Online.

Для збереження активного документа використовують команду **Файл/Зберегти**. Якщо документ непоіменований (тобто його ще не зберігали), то активізується діалогове вікно **Збереження документа**, де потрібно задати:

- диск і каталог (поле **Папка**);
- назву файлу (поле **Ім'я файлу**);
- формат, у якому зберігатиметься файл (поле **Тип файлу**).

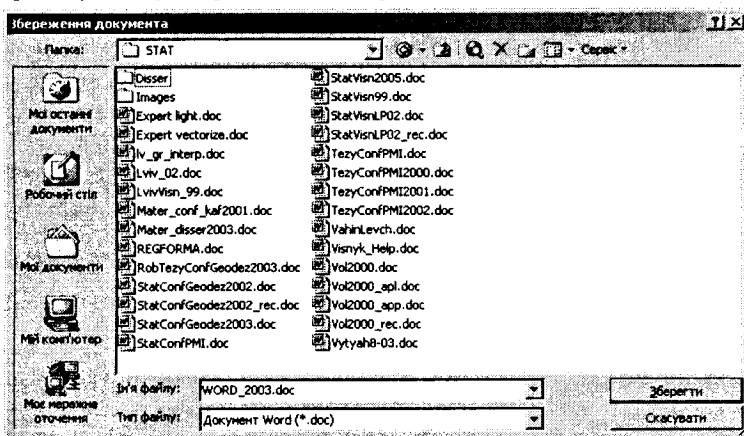


Рис. 6.5

Натискання на кнопку **Зберегти** ініціює процес збереження. Наступний запис документа цієї ж командою (або з використанням піктограми **Зберегти** панелі **Стандартна**) відбуватиметься без діалогу, а якщо потрібно зберегти документ під іншою назвою, то треба скористатися командою **Файл/Зберегти як**. Крім того, Word 2003 надає користувачам можливість збереження в одному файлі кілька версій документа, до кожної з яких можна записати окремий коментар, – для цього слід виконати команду **Файл/Версії**, натиснути на кнопку **Зберегти**, ввести примітку і натиснути на кнопку **ОК**.

Щоб відкрити вже збережений документ, використовують команду **Файл/Відкрити** або однойменну піктограму. У діалоговому вікні **Відкриття документа**, яке з'явиться після цього, слід вибрати файл і натиснути на кнопку **Відкрити** (або, розгорнувши список підкоманд на кнопці, вибрати одну з них: **Відкрити для читання** (без можливості внесення змін), **Відкрити як копію** (створити непоіменовану копію документа) тощо). Список **Подання**, призначений для зміни способу відображення вмісту активного каталогу, містить, окрім стандартних, ще й значення **Перегляд**, яке дає змогу попередньо переглядати вибраний файл, та **Властивості** для перегляду властивостей документа. Слід також зазначити, що **Word 2003** – багатівіконний текстовий процесор, тобто одночасно можна відкривати або створювати багато документів, перемикання між якими відбувається вибором потрібної позиції у меню **Вікно**.

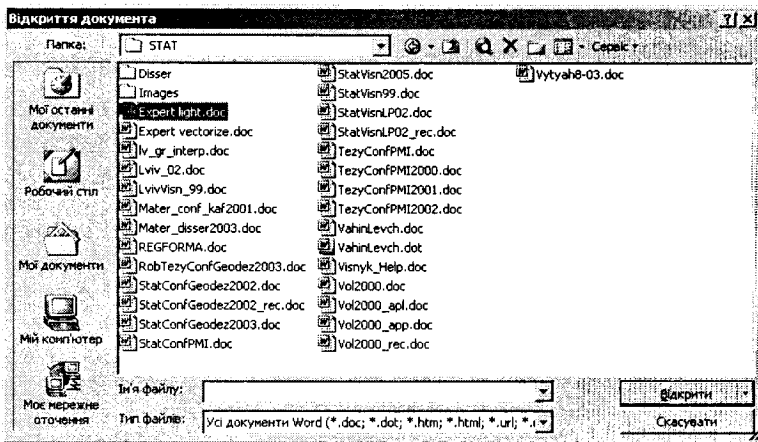


Рис. 6.6

Тут також можна організувати пошук потрібних файлів, якщо у списку **Сервіс** вибрати команду **Знайти**. Внаслідок цього відкриється діалогове вікно **Пошук файлів** із вкладками **Звичайний** та **Розширений**. Зокрема, для простого пошуку документа, який містить деякий текст, потрібно задати:

- у полі **Текст** – шуканий фрагмент тексту;
 - у полі **Область пошуку** – зону, де слід шукати файл;
 - у полі **Типи файлів результатів** – типи шуканих файлів,
- і натиснути на кнопку **Знайти**. Один із знайдених файлів можна відкрити, вибравши його у списку **Результати** і натиснувши на кнопку **ОК**.

Для задання складніших умов пошуку використовують вкладку **Розширений**. Цілковим аналогічним способом файли можна шукати за допомогою області завдань **Звичайний пошук файлів**, яку відкривають командою **Файл/Пошук файлів**.

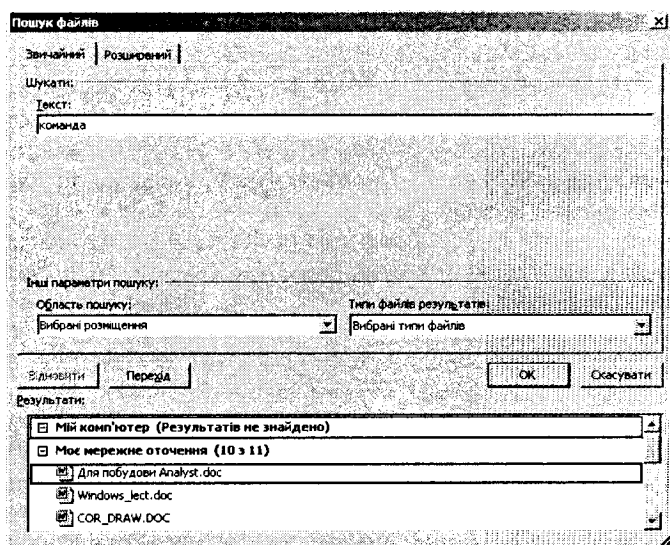


Рис. 6.7

Якщо відкритий документ містить у собі кілька версій, то для роботи з ними використовують діалогове вікно **Версії документа**, яке активізується командою **Файл/Версії**. Вибрану у списку **Найявні версії** версію документа можна відкрити кнопкою **Відкрити** або вилучити кнопкою **Видалити**, а також переглянути повний текст коментаря, натиснувши на кнопку **Примітки**. Кнопка **Зберегти** призначена для збереження поточного стану документа як нової версії, а ввімкнення опції **Автоматично зберігати версію під час закриття** дає змогу автоматично зберігати версію під час кожного закриття документа.

□ *Кілька (до дев'яти) останніх документів, з якими працювали у попередніх сеансах роботи з Word 2003, можна відкрити простішим способом. Для цього потрібно виконати команду **Сервіс/Параметри**, яка активізує діалогове вікно **Параметри**, на вкладці **Загальні** увімкнути опцію **пам'ятати список із ... файлів** і задати кількість документів, які необхідно пам'ятати. Тепер для відкриття одного з таких файлів достатньо просто вибрати його зі списку, що міститься в нижній частині меню **Файл**.*

□ *Під час роботи з документом його бажано періодично зберігати. Особливо зручний для цього режим автоматичного зберігання. Для його увімкнення слід виконати команду **Сервіс/Параметри**, вибрати вкладку **Збереження** і увімкнути опцію **автозбереження кожні:**, задавши у полі **хвилин проміжок часу**, через який ініціюватиметься процес автоматичного зберігання – записування файлу на диск у спеціальному форматі.*

Якщо у процесі роботи станеться аварія, то під час повторного запуску Word 2003 на екрані з'явиться список збережених таким чином файлів, і їх можна відновити. Однак, завершивши роботу з документом, необхідно зберегти його командою **Файл/Зберегти** або відповідною піктограмою. Для більшої надійності можна також увімкнути опцію **завжди створювати резервну копію**, яка дає змогу зберігати попередню версію документа у файлі з розширенням **.wbk**.

□ Якщо під час сеансу роботи з Word 2003 одночасно відкрито кілька документів, то для послідовного переміщення між ними можна скористатися комбінацією клавіш **Ctrl+F6**.

6.2.2. Режими роботи і масштаб відображення

Під час роботи з документами Word можна використовувати один із п'яти режимів перегляду. Режимом **Звичайний**, у якому відображається спрощений варіант документа, особливо зручно користуватись під час введення та редагування тексту. Режим макета **Розмітка сторінки**, який виводить на екран сторінки документа, призначений для формування друкованої публікації, режим **Структура** дає змогу працювати зі структурою довгих та складених документів, **Режим читання** оптимальний для читання документа, а режим **Веб-документ** використовують для роботи з документами у форматі HTML. Кожен із перелічених режимів задають відповідною командою меню **Вигляд** або піктограмою, що міститься ліворуч від горизонтальної смуги прокручення. Крім того, відобразити структуру документа у спеціальній ділянці вікна ліворуч від тексту можна за допомогою команди **Вигляд/Схема документа** або однойменної піктограми панелі **Стандартна**, однак робити це доцільно у випадку, коли документ структурований, тобто його заголовки оформлені стандартними стилями **Заголовок 1-9** чи аналогічними до них (мова про це йтиме далі). Команда **Вигляд/Ескізи** відображає там же спрощені моделі сторінок документа.

Для виконання різних операцій з документом потрібний різний масштаб відображення сторінок на екрані монітора. Щоб його змінити, можна використати команду **Вигляд/Масштаб**, яка відкриває діалогове вікно **Масштаб**. У ньому в рамці **Масштаб** вибирають одне із стандартних значень, серед яких є нечислові:

- **за шириною сторінки** – масштаб такий, що видно сторінку на всю ширину;
- **за шириною тексту** – видно сторінку на ширину тексту;
- **ціла сторінка** – видно цілу сторінку;
- **декілька сторінок** – видно декілька сторінок, кількість і розміщення яких задають натисканням на кнопку нижче від опції.

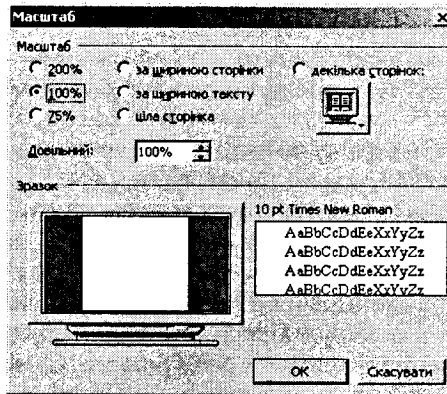


Рис. 6.8

Потрібне значення масштабу можна задати також вручну в полі **Довільний**. Однак значно простіше вибрати масштаб відображення у спеціальному полі **Масштаб** на панелі інструментів **Стандартна**, вводячи його з клавіатури або вибираючи зі стандартних значень, до списку яких додане значення **Дві сторінки**, але відсутня можливість вибору більше двох сторінок.

Для переміщення у потрібну ділянку документа використовують вертикальну та горизонтальну смуги прокручення. Натискаючи на піктограми із зображенням одинарних чорних стрілок вгору та вниз вертикальної смуги, можна змістити документ на один рядок вниз або вгору, натискання ж на смугу вище або нижче прямокутного індикатора призводить до зміщення на величину робочої області документа. Якщо ж потрібно переміститись на конкретну сторінку, то слід просто захопити індикатор і тягнути його вгору або вниз, при цьому праворуч у невеличкому віконці висвічується номер сторінки, на яку відбуватиметься перехід. Аналогічним чином документ зміщують і по горизонталі.

За допомогою вертикальної смуги прокручення можна рухатись по документу не лише на певну відстань, а й до об'єктів заданого типу або фрагментів тексту. Для визначення способу переміщення потрібно натиснути на піктограму із зображенням кульки на вертикальній смугі прокручення і вибрати тип об'єктів: **Поля**, **Кінцеві виноски**, **Виноски**, **Примітки**, **Розділи**, **Сторінки**, **Виправлення**, **Заголовки**, **Малюнки**, **Таблиці**. Тепер, щоб перейти до попереднього або наступного об'єкта вибраного типу, достатньо натиснути на піктограму із зображенням подвійної синьої стрілки, спрямованої відповідно вгору або вниз.

□ Якщо раптом виявилось, що горизонтальної чи вертикальної смуги прокручення немає на екрані, то їх можна візуалізувати увімкненням опції **Показувати/гориз. смугу прокрутки та верт. смугу прокрутки** на вкладці **Вигляд** діалогового вікна **Параметри**. Там же задають виведення

рядка статусу (опція **рядок стану**), з якого користувач може одержати багато корисної інформації, зокрема, скільки сторінок є у документі, номер видимої сторінки та місце перебування курсора вводу.

6.3. Введення і редагування тексту

Текст вводять у документ із того місця, де розташований курсор, в одному з двох режимів: вставлення (стандартний), коли попередній текст зберігається, зсуваючись праворуч, або заміни, у якому введені літери замінюють собою наявні. Для перемикання між режимами використовують клавішу **Insert** клавіатури. Перед уведенням тексту бажано вибрати основні його параметри (гарнітуру, розмір шрифту, накреслення та спосіб вирівнювання) у відповідних полях панелі **Форматування**.

Щоб вставити в документ символи, яких немає на клавіатурі, або спеціальні символи, використовують команду **Вставка/Символ**. Після її виконання відкривається вікно **Символ**, де слід вибрати потрібну вкладку – **Символи** або **Спеціальні знаки**, а на ній – шуканий символ (поле **Шрифт** вкладки **Символи** дає змогу змінити гарнітуру, а поле **Набір** – показати певний набір символів для вибраного шрифту), і натиснути на кнопку **Вставити**. Під час роботи з вікном **Символ** можна переходити у текст документа, переносити курсор на нову позицію і продовжувати процес вставлення. Кнопка **Сполучення клавіш** призначена для пов'язування з вибраним символом певної комбінації клавіш, а кнопка **Автозаміна** дає змогу використати для його введення функцію автоматичної заміни (див. зауваження в кінці цього пункту). Після вставлення всіх потрібних символів вікно закривають кнопкою **Закрити**.

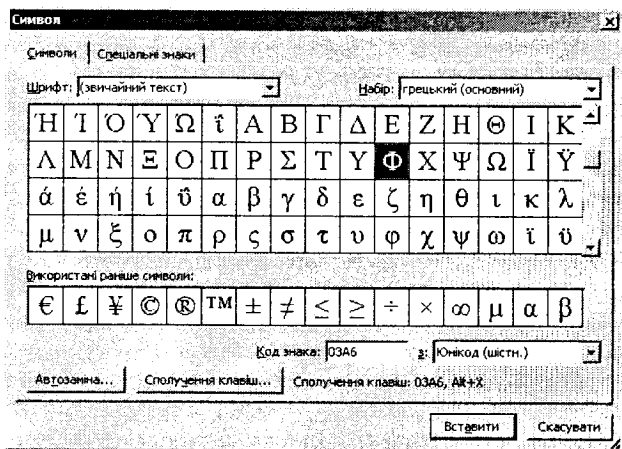


Рис. 6.9

Основною структурною одиницею тексту є абзац, закінчення якого позначають спеціальним символом – ¶ (**Знак абзацу**). Його вводять натисканням на клавішу **Enter**, що автоматично переносить курсор вводу на новий абзац. Якщо потрібно продовжити текст абзацу з нового рядка, використовують комбінацію клавіш **Shift+Enter**, яка в кінці попереднього рядка залишає символ ↵ (**Розрив рядка**). Обидва згаданих символи належать до категорії недрукованих, куди входять також:

- символи табуляції → (**Знак табуляції**);
- пробіли (**Пробіл**);
- умовні переноси ~ (М'який перенос), які стають видимими лише тоді, коли опиняються на правому краю тексту;
- прихований текст (**Прихований текст**);
- маркери кінця сторінки (**Розрив сторінки**), колонки (**Розрив колонки**) та розділу (**Розрив розділу**).

Показати або сховати недруковані символи у тексті можна за допомогою піктограми **Недруковані знаки** панелі **Стандартна**.

Більшість операцій у Word 2003 виконують над виділеними фрагментами тексту. Для виділення фрагмента за допомогою миші слід помістити курсор у потрібне місце, натиснути на ліву клавішу і, не відпускаючи її, перемістити курсор на кінець фрагмента. Зазначимо, що у Word 2003, на відміну від ранішніх версій, можна виділити кілька окремих фрагментів, утримуючи клавішу **Ctrl**. Якщо ж виділяти за допомогою клавіатури, то треба натиснути на клавішу **Shift** і, не відпускаючи її, розширити межі фрагмента клавішами переміщення курсора.

Виділені фрагменти програма інвертує (відображає білим кольором на чорному тлі). Щоб виділити вертикальні блоки, використовують мишу з утриманням клавіші Alt або навігаційні клавіші після натискання на комбінацію Ctrl+Shift+F8.

Виділені фрагменти можна вирізати або копіювати в буфер командами **Правка/Вирізати** та **Правка/Копіювати**, щоб потім мати змогу вставити їх у довільне місце тексту командою **Правка/Вставити** (ці ж функції виконують однойменні піктограми панелі **Стандартна**). Варто зазначити, що таким чином можна обмінюватися фрагментами тексту не лише з іншими відкритими документами Word (переходячи до них через меню **Вікно**), а й з усіма іншими програмами, які містять команди роботи з буфером. Для вилучення виділеного фрагмента використовують команду **Правка/Очистити/Вміст** або клавішу **Delete**.

У текст активного документа можна також вставляти вміст інших документів. Для цього треба виконати команду меню **Вставка/Файл**, вибрати необхідний файл і натиснути на кнопку **ОК**. Текст буде вставлено, починаючи з місця розташування курсора вводу.

□ Щоб у файлі достатньо великих розмірів потрапити туди, де попереднього разу було закінчене його редагування, слід натиснути на комбінацію клавіш **Shift+F5**.

□ Якщо час від часу потрібно використовувати літери російського алфавіту **ы, э та ъ**, не обов'язково встановлювати для цього російську розкладку клавіатури – достатньо на вкладці **Символи** вікна **Символ** пов'язати з ними поєднання клавіш. Однак, щоб ці символи Word сприймав саме як літери, треба у полі **Шрифт** вибрати значення **звичайний текст**. Це ж саме стосується відсутніх на українській клавіатурі апострофа ' (стандартна комбінація **Ctrl+'**, а потім сам ') та літери **г (Ctrl+Alt+U)**, якщо стандартні комбінації не спрацьовують.

□ Швидко змінити регістр літер виділеного фрагмента або поточного слова можна за допомогою комбінації клавіш **Shift+F3**. Більші можливості для цього надає команда **Формат/Регістр**, однак для неї, зрозуміло, потрібно більше часу.

□ Для того, щоб два слова завжди містилися в одному рядку, між ними слід вводити не звичайний пробіл, а спеціальний символ **Нерозривний пробіл**, якому відповідає комбінація клавіш **Ctrl+Shift+Space**.

□ Виділені фрагменти тексту можна копіювати або переміщувати не тільки через буфер, а й за допомогою миші: для цього потрібно лише захопити виділений фрагмент і перетягнути його (якщо натиснута клавіша **Ctrl** – його копію) на нове місце. Однак, щоб така операція стала можливою, необхідно ввімкнути опцію **Використосувати перетягнення тексту** на вкладці **Правка** діалогового вікна **Параметри**.

□ Подвійне натискання мишею на будь-якому слові виділяє його, а потрійне – цілий поточний абзац. Натискання мишею з утриманням клавіші **Ctrl** виділяє поточне речення. Щоб виділити весь текст, використовують команду **Правка/Виділити все** або комбінацію **Ctrl+A**. Поєднання **Ctrl** з клавішами **Delete** або **Backspace** дає змогу вилучати цілі слова.

□ Якщо доводиться часто вставляти в документ однакові, але достатньо великі або незручні для набору фрагменти тексту (які можуть містити ілюстрації, формули, спеціальні символи тощо), доцільно скористатися функцією **Автотекст**: виділити потрібний фрагмент, виконати команду **Вставка/Автотекст/Автотекст**, у полі **Ім'я елемента** задати ім'я фрагмента як елемента автотексту (бажано коротке) і натиснути на кнопку **Додати**. Тепер для вставлення цього фрагмента у текст документа достатньо знову виконати названу команду, вибрати у списку необхідний елемент автотексту і натиснути на кнопку **Вставити**. Створити новий елемент автотексту можна іще швидше, використавши після виділення фрагмента тексту комбінацію клавіш **Alt+F3**, а для швидкого вставлення слід набрати ім'я елемента (як окремого слова) і натиснути на клавішу **F3**.

□ Якщо сталося щось на перший погляд непоправне, наприклад, після виділення великого фрагмента тексту несподівано спрацювала клавіша **Delete**, не слід впадати у відчай – для таких випадків у *Word* передбачені команда **Правка/Скасувати** та однойменна піктограма на панелі **Стандартна**. За їхньою допомогою можна відмінити практично всі зміни, зроблені у документі (якщо у комп'ютера достатньо пам'яті): для цього треба виконати команду або натиснути на зображення стрілки у піктограмі – і так стільки разів, скільки потрібно. Якщо необхідно відмінити відразу кілька операцій, то слід розкрити їхній список і вибрати останню з потрібних. А у випадку, коли й відміна виявилась помилковою, достатньо у цілком аналогічний спосіб скористатись командою **Правка/Повторити** або однойменною піктограмою, які повторюють виконання відмінених операцій.

□ Припустимо, що вкрай необхідно ввести у текст позначення (R). Однак швидше за все воно дивовижним чином перетвориться на символ ®. Справа в тому, що *Word 2003* виконав функцію автоматичної заміни тексту під час введення, яка називається **Автозаміна**. Виходів із ситуації принаймні три: на вкладці **Автозаміна** однойменного вікна, яке відкривається командою **Сервіс/Автозаміна/Параметри автозаміни**, вимкнути опцію **Замінювати під час вводу**; там же вилучити непридатний елемент автозаміни, вибравши його у списку і натиснувши на кнопку **Видалити**; відмінити останню виконану операцію (а нею буде саме автозаміна) командою **Правка/Скасувати**, однойменною піктограмою панелі **Стандартна** або комбінацією **Ctrl+Z**. Саму ж функцію автозаміни потрібно використовувати з вигодою для себе: можна задавати власні елементи для заміни, вводячи у поле **замінити**, наприклад, скорочення найчастіше вживаних слів, а в поле **на** – їхнє повне написання, і натискаючи на кнопку **Додати**, щоб внести елемент у список. Якщо ж деякі подібні прикращі продовжують траплятись, слід це переглянути вкладку **Автоформат при вводі** цього ж діалогового вікна.

6.4. Форматування тексту

Вислів «форматування тексту» означає пов'язування з фрагментами документа різних атрибутів, які визначають зовнішній вигляд тексту. Всі операції форматування виконуються для виділених фрагментів або тексту, що вводиться далі з клавіатури, за допомогою команд меню **Формат** або піктограм панелі інструментів **Форматування**. Для символів параметрами форматування є шрифт (гарнітура), розмір шрифту, накреслення, підкреслення, колір, міжсимвольна відстань, розміщення символу в рядку, кернінг та спеціальні ефекти, а для абзаців – відступи, міжрядковий відстані, спосіб вирівнювання, умови розриву рядків, точки табуляції, рамки та тіні.

6.4.1. Форматування символів

Для задання параметрів форматування символів використовують команду **Шрифт**, яка відкриває однойменне діалогове вікно з трьома вкладками (рис. 6.10).

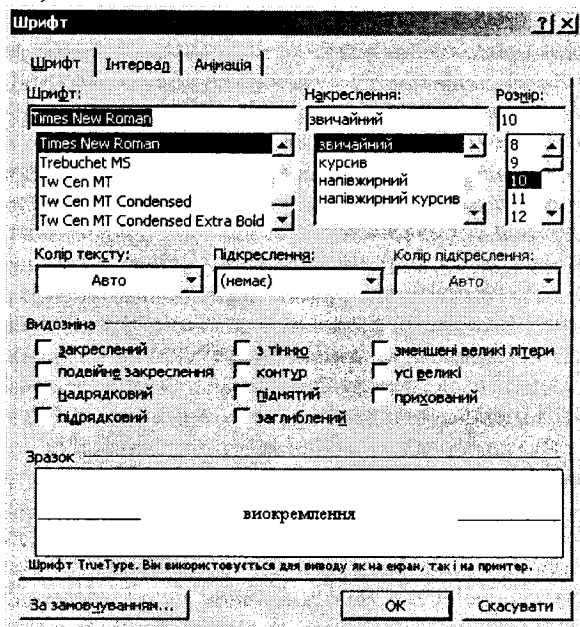


Рис. 6.10

На вкладці **Шрифт** визначають:

- гарнітуру (поле **Шрифт**);
- спосіб накреслення (**Накреслення**) зі стандартних значень **звичайний**, **курсив**, **напівжирний** та **напівжирний курсив**;
- розмір шрифту (**Розмір**), який задають у спеціальних друкарських одиницях – пунктах (10 пт = 3,5 мм);
- стиль підкреслення (**Підкреслення**) зі значень: **немає** – без підкреслення, **Лише слова** – підкреслення однією лінією, але тільки слів, а також іншими запропонованими лініями;
- кольори символів (**Колір тексту**) та підкреслення (**Колір підкреслення**);
- спеціальні ефекти (**Видозміна**): **закреслений** – закреслений текст, **подвійне закреслення** – закреслений подвійною лінією, **надрядковий**, **підрядковий** – текст як верхній та нижній індекси, **з тінню**, **контур** – текст з тінню або контуром, **піднятий**, **заглиблений** – ефект піднятого або втопленого тексту, **зменшені великі літери** – символи тексту

будуть переведені у капітель, усі великі – символи тексту будуть переведені у верхній регістр, **прихований** – текст буде прихованим.

Основні параметри форматування символів (гарнітуру, розмір та тип накреслення) можна задати за допомогою панелі інструментів **Форматування**, причому ті з них, які відповідають уже застосованим до виділеного фрагмента тексту параметрам форматування, буде виділено.

На вкладці **Інтервал** задають:

- у полі **Масштаб** – ширину символів у відсотках до стандартної;
- у полі **Інтервал** – відстань між символами (трекінг) зі значень: **Звичайний** – стандартна, **Ущільнений** – зменшена і **Розріджений** – збільшена, причому величину збільшення чи зменшення (в пунктах) задають у полі **на праворуч**;
- у полі **Зсув** – позицію тексту відносно базової лінії зі значень: **Немає** – звичайна, **Вгору** і **Вниз** – вище або нижче від базової лінії, причому величину зміщення (в пунктах) задають у полі **на праворуч**.

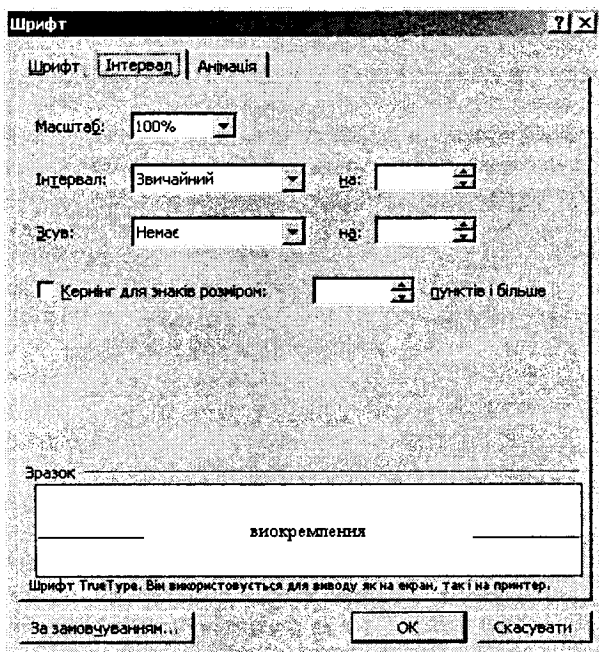


Рис. 6.11

Для використання кернінгу (операції зменшення відстані між спеціальними парами символів – У і А, W і А тощо) необхідно увімкнути опцію **Кернінг для знаків розміром:**, а в полі **пунктів і більше** задати розмір, починаючи з якого буде виконуватися ця операція.

І, нарешті, на вкладці **Анімація** можна застосувати до виділеного фрагмента уподобаний анімаційний ефект зі списку **Вигляд**, який буде видно лише на екрані монітора. Слід також зазначити, що на кожній вкладці є вікно **Зразок**, де відображаються результати застосованих до тексту параметрів, а кнопка **За замовчуванням** робить ці параметри стандартними, тобто їх використовуватимуть за замовчуванням в усіх нових документах, що базуються на шаблоні активного документа.

□ Під час уведення тексту можна швидко змінювати параметри форматування символів, використовуючи такі комбінації клавіш: **Ctrl+B** – увімкнення/вимкнення напівжирного накреслення, **Ctrl+I** – курсиву, **Ctrl+U** – підкреслення, **Ctrl+=** – переведення у нижній регістр і назад, **Ctrl+Shift+=** – переведення у верхній регістр і назад.

6.4.2. Форматування абзаців

Якщо у роботі з Word дотримуватися правил «хорошого тону», то ніколи не слід користуватися клавішею **Tab**, щоб задавати відступ для першого рядка абзацу, і вставляти порожні рядки для відокремлення абзаців один від одного (за винятком хіба що титульних сторінок) – для цього є свої засоби, описані далі.

Команда **Абзац** відкриває однойменне діалогове вікно для форматування поточного (у якому розташований курсор вводу) або виділених абзаців.

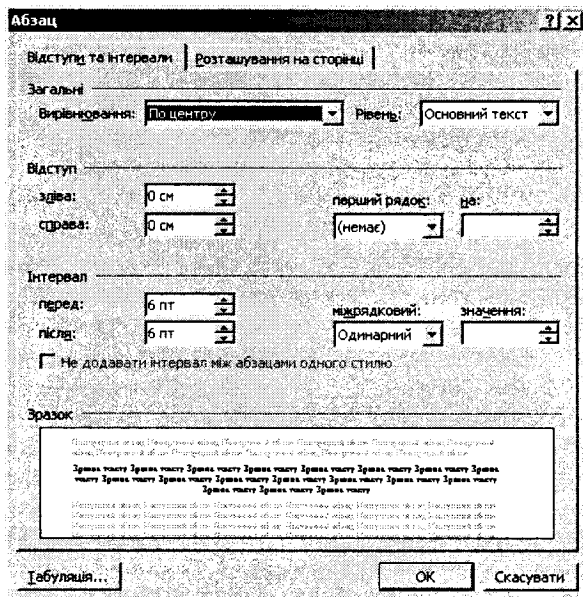


Рис. 6.12

На вкладці **Відступи та інтервали** (рис. 6.12) задають:

у полі **Вирівнювання** – спосіб вирівнювання абзацу:

- **По центру** – в центрі сторінки;
- **За лівим краєм** – рівний лівий край тексту;
- **За правим краєм** – рівний правий край тексту;
- **За шириною** – рівні обидва краї;

у полі **Рівень** – рівень абзацу в структурі документа:

- **Основний текст** – абзац розглядається як звичайний текст;
- **Рівень 1-9** – абзац розглядається як заголовок рівня 1-9;

у групі **Відступ**:

- відстань від абзацу до лівого (**зліва**) та правого (**справа**) полів сторінки документа;
- відступи всередині абзацу (**перший рядок**) зі значень: **немає** – без відступів, **Відступ** – задано відступ для першого рядка абзацу, **Нависаючий** – задано відступ для всіх рядків абзацу, крім першого; величину відступу зазначають у полі **на** праворуч;

у групі **Інтервал**:

- відступи (в пунктах) перед першим (**перед**) та після останнього (**після**) рядка абзацу;
- міжрядковий інтервал (**міжрядковий**) зі значень: **Одинарний**, **Полуторний**, **Подвійний** – один, півтора та два інтервали, **Мінімум** – одинарний, але не менший (в пунктах) від заданого у полі **значення**, **Точно** – точно заданий у полі **значення**, **Множник** – у частинах (від одинарного), заданих у полі **значення**.

Зазначимо, що спосіб вирівнювання абзаців та міжрядковий інтервал можна швидко змінити, використовуючи відповідні піктограми на панелі інструментів **Форматування**.

Вкладка **Розташування на сторінці** (рис. 6.13) містить такі опції:

- **заборона нависаючих рядків**, яка забороняє появу висячих рядків (тобто одиноких рядків абзацу, що містяться в кінці або на початку сторінки);
- **не розривати абзац**, яка дає змогу зберігати повністю весь абзац на одній сторінці;
- **не відривати від наступного**, яка зберігає абзац на одній сторінці з наступним;
- **з нової сторінки**, яка поміщає абзац на початок нової сторінки;
- **заборонити нумерацію рядків**, яка виключає рядки абзацу з послідовності пронумерованих рядків (якщо задано режим їх нумерації на вкладці **Макет** діалогового вікна **Параметри сторінки**, що відкривається за допомогою команди **Файл/Параметри сторінки**);
- **заборонити автоматичний перенос слів**, яка відміння автоматичний перенос слів в абзаці.

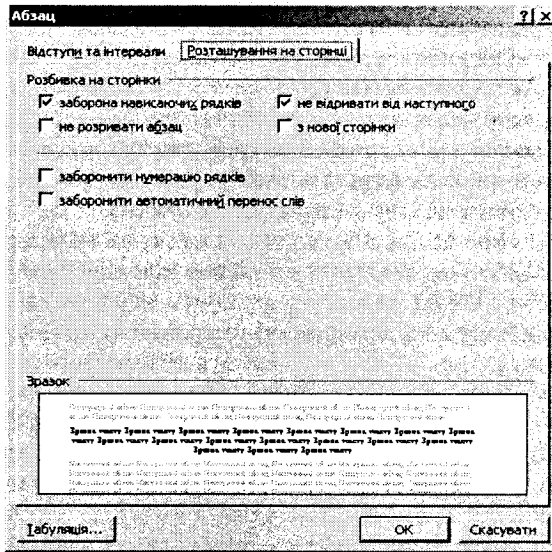


Рис. 6.13

Кнопка **Табуляція** призначена для задання позицій точок табуляції. У вікні, що відкривається після її натискання (рис. 6.14), треба:

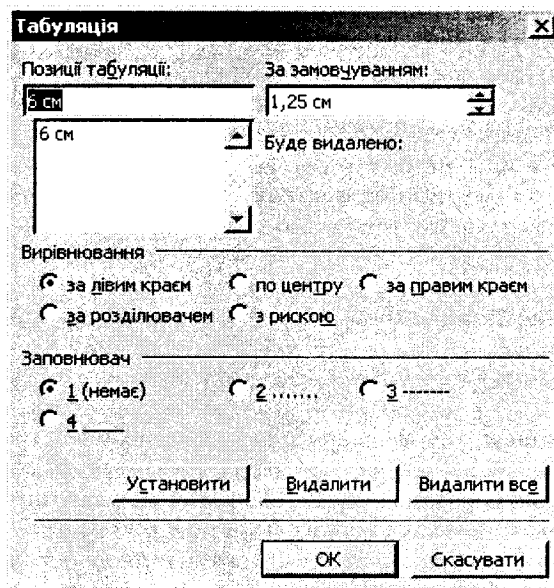


Рис. 6.14

- у полі **Позиції табуляції** задати положення точки;
- вибрати потрібне значення опції **Вирівнювання**, яка означає спосіб вирівнювання тексту, введеного після знака табуляції (**за лівим краєм**, **по центру**, **за правим краєм** – відповідно ліворуч, до центру та праворуч від точки табуляції, **за розділювачем** – відносно десяткової крапки, **з ризкою** – у позицію табуляції вставляється вертикальна лінія);
- задати значення опції **Заповнювач**, яка визначає заповнювач для порожнього місця;
- натиснути на кнопку **Установити**, після чого точка буде додана у список.

Кнопки **Видалити** та **Видалити все** призначені для вилучення однієї або всіх точок табуляції зі списку, а натискання на кнопку **ОК** закриває діалогове вікно **Табуляція**. Тепер, якщо ввести де-небудь в абзаці символ табуляції клавішею **Tab**, то курсор відразу з'явиться у найближчій справа точці табуляції і текст, який вводитимуть далі, вирівнюватиметься відповідно до типу табуляції.

Точки табуляції можна також задати, виконавши команду **Формат/Табуляція**.

Для зміни вигляду облямування і затінення абзацу використовують команду **Формат/Межі й заливка**, яка відкриває однойменне діалогове вікно із вкладками **Межа**, **Сторінка** і **Заливка**. На першій з них задають параметри облямування поточного або виділених абзаців:

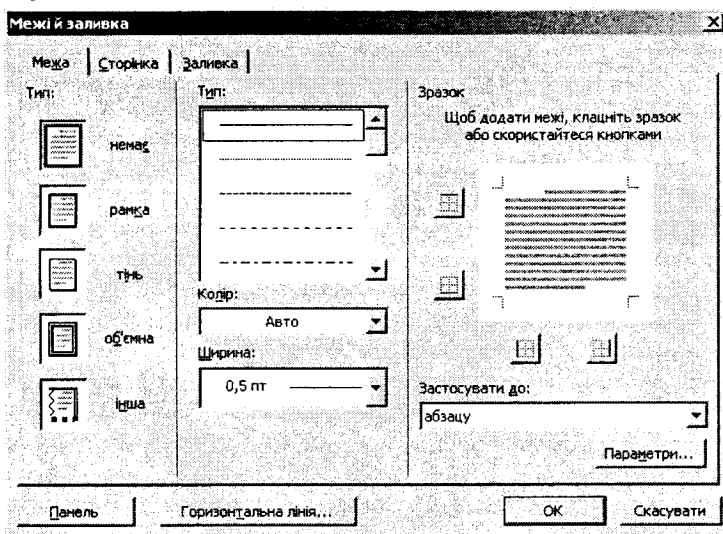


Рис. 6.15

- опція **Тип** дає змогу вибрати один із заготовлених типів облямування: **немає** – без облямування, **рамка** – абзаци буде обведено прямокутною

рамкою, **тїнь** – прямокутна рамка матиме тїнь, **об'ємна** – рамка буде об'ємною, **їнша** – вигляд рамки визначає користувач, причому кожна сторона може мати свої атрибути відображення;

- зі списку **Тип** вибирають стиль рамки або сторони;
- у полі **Колір** задають колір рамки (або вибраної у вікні **Зразок** сторони, якщо використовують тип рамки **їнша**);
- у полі **Ширина** – товщину рамки;
- у полі **Застосувати до** – до чого слід застосовувати задані параметри: **абзацу** – до всього виділеного фрагмента тексту, **тексту** – до кожного виділеного рядка.

Кнопка **Параметри** призначена для зміни відстаней від тексту до кожної з чотирьох сторін рамки. Зовнішній вигляд облямування з урахуванням усіх заданих параметрів відображається у вікні **Зразок**.

Для швидкого задання способу облямування абзаців тексту можна скористатися спеціальною піктограмою **Межі** на панелі інструментів **Форматування**.

У такий самий спосіб на вкладці **Сторінка** можна задати вигляд облямування для сторінок документа. Тут є лише одне додаткове поле **Малюнок**, яке дає змогу скористатись великим набором графічних рамок для всієї сторінки.

Спосіб затінення абзацу визначають на вкладці **Заливка**:

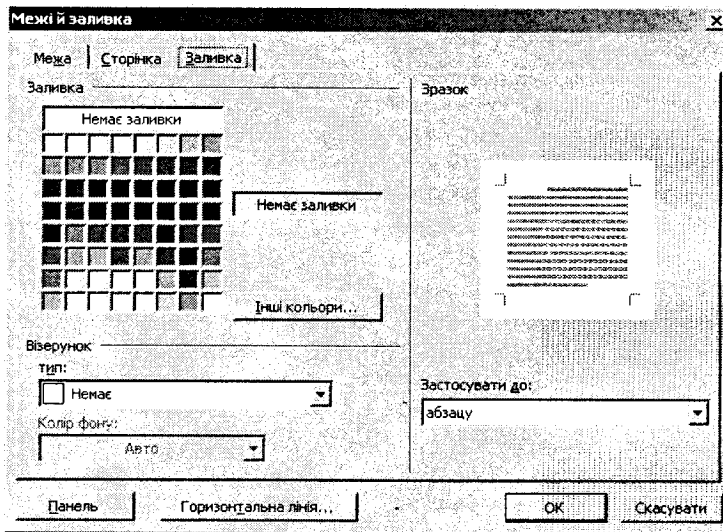


Рис. 6.16

- у рамці **Заливка** вибирають колір шаблону заповнення (значення **Немає заливки** дає змогу відмінити затінення);

- у полі **тип** задають сам шаблон;
- у полі **колір фону** – фоновий колір заповнення;
- у полі **Застосувати до** – область застосування параметрів заповнення.

Для виконання всіх названих операцій зручно користуватись піктограмами панелі інструментів **Таблиці та межі**, яка візуалізується, окрім звичайних засобів, кнопкою **Панель діалогового вікна Межі й заливка**. Кнопка **Горизонтальна лінія** вставляє у місце розташування курсора вводу вибрану в діалоговому вікні ілюстрацію у формі лінії.

Зручним та наочним інструментом для форматування абзаців є лінійка форматування, яка візуалізується командою **Вигляд/Лінійка**.

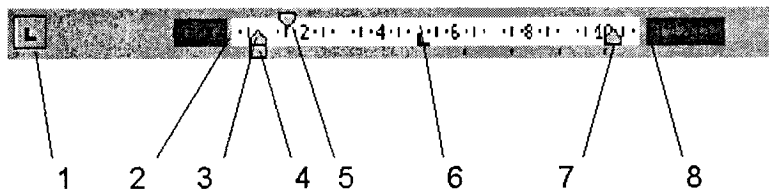


Рис. 6.17. 1 – піктограма типу точки табуляції; 2 – індикатор лівого поля; 3 – індикатор виступу всіх рядків абзацу, крім першого; 4 – індикатор лівого відступу абзацу; 5 – індикатор відступу першого рядка абзацу; 6 – точка табуляції; 7 – індикатор правого відступу абзацу; 8 – індикатор правого поля.

На ній містяться індикатори, переміщенням яких можна змінити відповідні параметри поточного або виділених абзаців. Натискання на піктограму ліворуч від лінійки форматування змінює тип точок табуляції, які потім легко вставити мишею у потрібному місці на лінійці. Встановлені точки можна переміщувати по лінійці або вилучати, вивівши їх за межі лінійки. Натискаючи мишею на якому-небудь індикаторі з утриманням клавіші **Alt**, можна отримати числову інформацію стосовно параметрів форматування поточного абзацу. Переміщення по лінійці звичайно прив'язані до позначок, відстань між якими становить 2,5 мм, однак утримання клавіші **Alt** дає змогу уникнути цього обмеження.

□ **Найкращий спосіб створити у чистому бланку довгий підкреслений рядок, куди потім зручно набирати текст, такий: встановити у місці, де закінчується рядок, точку табуляції, потім на початку рядка набрати необхідний стандартний текст, увімкнути режим підкреслення і ввести пробіл та символ табуляції. Тепер увесь рядок від пробілу до кінця виглядатиме підкресленим, як і все те, що буде введене між пробілом та символом табуляції.**

★ □ **Піктограма *Копіювати формат* дає змогу перенести параметри форматування від одного фрагмента тексту до іншого: слід виділити фрагмент-зразок, натиснути на згадану піктограму і виділити фрагмент-**

одержувач. Для копіювання параметрів форматування лише абзацу потрібно просто натиснути на абзаці-одержувачі. Подвійне натискання на піктограму дає змогу її зафіксувати, щоб виконати описану операцію кілька разів поспіль, після чого слід просто відтиснути її назад.

□ Введення слів, вставлення вмісту буфера, зміну параметрів форматування та ще багато-багато операцій можна повторити в іншому місці документа, натиснувши на клавішу F4.

6.4.3. Форматування списків

Якщо у тексті є послідовність абзаців, пронумерованих або позначених якимось символом-маркером, то їх можна оформити як список. У Word 2003 є два типи списків – з нумерацією та позначками, можливі також комбіновані списки з кількома рівнями (до дев'яти). Якщо до списку з позначками додати новий елемент, то він автоматично позначається відповідним символом, а у випадку додання або вилучення елемента з нумерованого списку його номери автоматично коригуються.

Для створення списку необхідно виділити потрібні абзаци або поставити курсор вводу у новий абзац і виконати команду **Формат/Список**, яка відкриє діалогове вікно **Список** із вкладками: **Маркірований**, **Нумерований**, **Багаторівневий** та **Список стилів**.

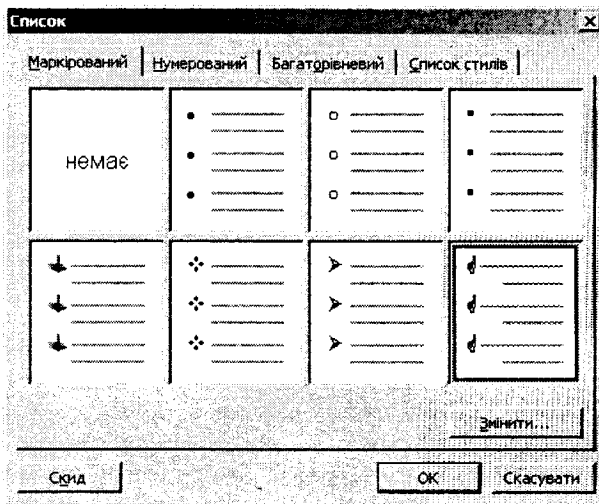


Рис. 6.18

Для створення списку з позначками вибирають вкладку **Маркірований**. Із семи пропонуваніх варіантів оформлення списку необхідно вибрати один, натиснувши на ньому мишею, а в разі потреби його модифікувати –

натиснути на кнопку **Змінити**, яка відкриє діалогове вікно **Зміна маркірованого списку**. У ньому задають:

- символ позначки списку (ділянка **Знак маркера**); якщо потрібного серед пропонувананих немає, то його можна вибрати в описаному раніше вікні **Символ**, яке відкривається за допомогою кнопки **Знак**, змінити параметри шрифту у вікні **Шрифт**, що активізується одноіменною кнопкою, а також задати графічний маркер кнопкою **Малюнок**;
- відстань від лівого краю тексту до маркера (поле **відступ рамки Розташування маркера**);
- відступ для тексту першого рядка абзацу (не враховуючи маркера), реалізований за допомогою табуляції (поле **табуляція після рамки Розташування тексту**) та виступ для решти рядків (поле **відступ праворуч у тій же рамці**).

Варто зазначити, що відступ для абзаців списку можна також задавати у діалогових вікнах **Абзац** і **Табуляція**. Зручна для цього й лінійка форматкування.

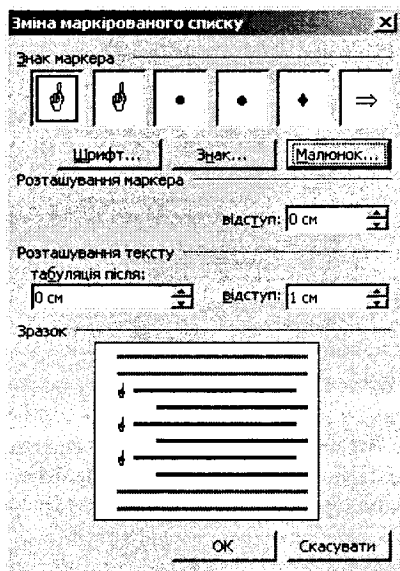


Рис. 6.19

Для заданих параметрів на списку відображається у ділянці **Зразок**. Після натискання на кнопку **ОК** поточний або попередньо виділені абзаци будуть оформлені у вигляді списку. Щоб закінчити список і повернутися до введення звичайного тексту, слід двічі ввести кінець абзацу клавішею **Enter**, після чого зайвий порожній абзац бажано видалити.

Для вилучення виділених абзаців зі списку потрібно у діалоговому вікні **Список** вибрати варіант **немає**. Кнопка **Скид** дає змогу повернутись до стандартних параметрів вибраного варіанту.

Нумерований список створюють за допомогою вкладки **Нумерований** вікна **Список** подібно до списку з позначками, однак тут присутня додаткова опція **Нумерація**, значення якої **почати заново** визначає створення списку з новою нумерацією, а значення **продовжити** дає змогу продовжити нумерацію попереднього списку. Змінити стандартні параметри списку можна у діалоговому вікні **Зміна нумерованого списку**, яке відкривається після натискання на кнопку **Змінити**.

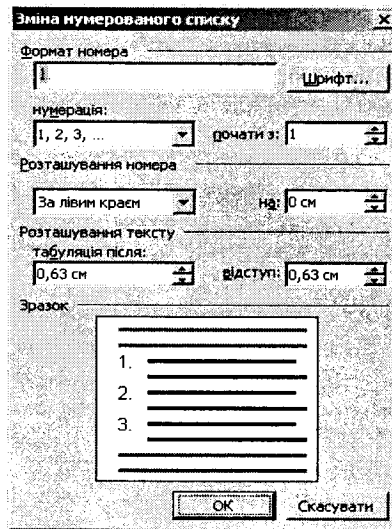


Рис. 6.20

У рамці **Формат номера** :

- у полі зверху задають текст перед номером (якщо курсор помістити у початок рядка) та після нього (крапку, дужку тощо); кнопка **Шрифт** дає змогу встановити параметри шрифту для номера в однойменному діалоговому вікні;
- у полі **нумерація** вибирають стиль нумерації (арабськими чи латинськими цифрами, літерами латинського алфавіту або словами), а в полі **почати з** задають початкове число або літеру для нумерації.

У лівому полі рамки **Розташування номера** вибирають спосіб вирівнювання номера відносно позиції, заданої у полі **на**, а у рамці **Розташування тексту** – відступи для тексту. Відстані у списках можна також регулювати у діалогових вікнах **Абзац** і **Табуляція** та за допомогою лінійки форматування цілком аналогічно до списків з позначками.

Для швидкого оформлення абзаців у список без діалогу використовують піктограми **Нумерований список за замовчуванням** та **Маркірований список за замовчуванням** панелі інструментів **Форматування**, у цьому випадку будуть застосовані стандартні параметри форматування списків. Якщо поточний або виділені абзаци є елементами списку, то залежно від його типу відповідні піктограми виглядатимуть виділеними. Щоб вилучити ці абзаци зі списку, достатньо ще раз натиснути на піктограму.

Щоб створити багаторівневий список, необхідно перейти на вкладку **Багаторівневий** вікна **Список**. Як і в попередніх випадках, тут можна вибрати один із наперед заготовлених варіантів, а також модифікувати його у діалоговому вікні **Зміна багаторівневого списку**, що відкривається після натискання на кнопку **Змінити**.

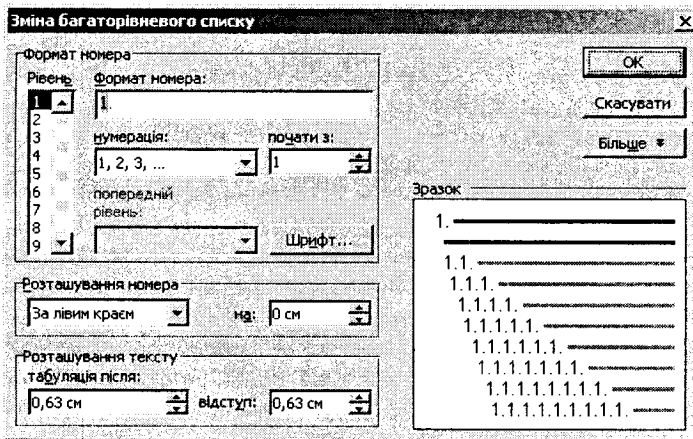


Рис. 6.21

Параметри кожного ієрархічного рівня задають окремо: для цього спочатку необхідно вибрати його у рамці **Рівень**, потім визначити тип у полі **нумерація**, задаючи потрібний стиль нумерації або символ маркера, і далі визначити необхідні параметри форматування, як це було описано вище. Крім того, у полі **попередній рівень** можна вибрати інший, відмінний від стандартного, попередній рівень, – той, від якого братиметься перша складова нумерації.

Кнопка **Більше** відкриває нижню область діалогового вікна, де задають додаткові параметри багаторівневого списку:

- поле **Зв'язати рівень зі стилем** дає змогу задати автоматичну нумерацію заголовків, пов'язавши вибраний рівень з одним із стилів, абзаци якого будуть далі пронумеровані;
- у полі **Знак після номера** можна змінити символ табуляції, що стандартно вставляється між номером (чи маркером) і текстом, на пробіл або зовсім вилучити його;

- поле **Ім'я списку** поля **ListNum** використовується для формування списку за допомогою засобу **ListNum**;
- опція **Замінити римські цифри арабськими** дає змогу замінити в складеній нумерації римські цифри на арабські;
- опція **Нумерувати знову** задає режим, коли після зміни номера вибраного праворуч рівня нумерація поточного починається спочатку;
- поле **Застосувати** визначає межі дії заданих параметрів.

Закрити додаткову область діалогового вікна можна тією ж кнопкою, яка змінює свою назву на **Менше**.

І, нарешті, на вкладці **Список стилів** діалогового вікна **Список** можна використати один із заготовлених стилів (наборів параметрів форматування) для оформлення багаторівневих списків, а також за допомогою кнопок **Додати**, **Змінити** та **Видалити** створювати власні стилі, модифікувати та вилучати їх.

Для зміни рівня поточного або виділених елементів багаторівневого списку використовують піктограми **Зменшити відступ** та **Збільшити відступ** панелі **Форматування**, а також однойменні команди локального меню. Для звичайного тексту вони просто зменшують або збільшують відступ зліва.

□ *Якщо розпочати абзац з «1. », «1) » або латинських A, A, i, I з крапкою або дужкою і пробілом, то, найімовірніше, після введення символу кінця абзацу автоматично утвориться нумерований список. Введення ж на початку абзацу символів *, - або > з пробілом спричинить виникнення списку з позначками. Якщо такий сервіс не потрібен, то слід виконати команду **Сервіс/Параметри автозаміни** і на вкладці **Автоформат при вводі в групі Застосовувати** під час вводу вимкнути опції **стилі нумерованих списків** і **стилі маркірованих списків**. Більш трудомісткий, але універсальний спосіб такий: щоразу після небажаної автозаміни натискати на **Ctrl+Z**, **Alt+Backspace** або на піктограму **Скасувати** панелі **Стандартна**.*

6.5. Пошук і заміна фрагментів тексту

Під час роботи з документом можна організувати пошук фрагментів тексту за допомогою команди **Правка/Знайти**. Вона відкриває діалогове вікно **Знайти й замінити** з активною вкладкою **Знайти**, де у полі **Знайти** слід увести текст, який треба знайти, і натиснути на кнопку **Знайти далі**. Якщо заданий фрагмент знайдено, то він буде виділений і показаний у робочому вікні документа (і кнопкою **Знайти далі** можна шукати його наступне місцезнаходження у тексті), інакше програма видасть повідомлення про відсутність фрагмента або його наступного входження в документ. Вмикання опції **Виділити всі елементи, знайдені в:** дає змогу відшукати та виділити всі задані фрагменти у тексті документа або колоніти-тулах, у цьому випадку кнопка **Знайти далі** змінює назву на **Знайти все**.

Діалогове вікно **Знайти й замінити** дозволяє без свого закривання захопити у текст документа і повернутися назад для продовження діалогу.

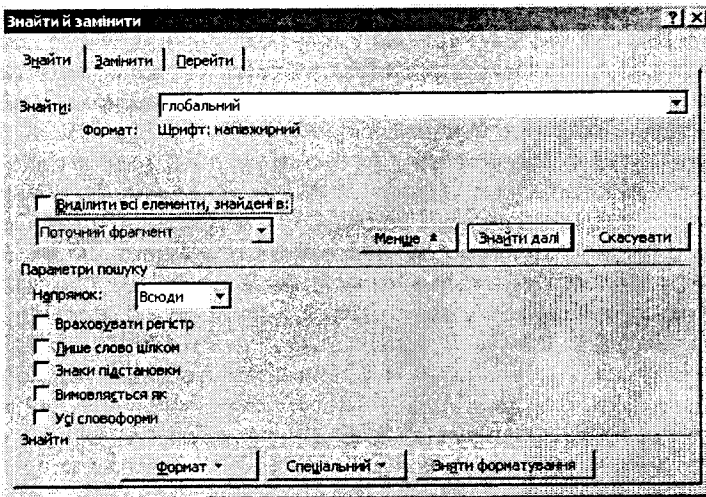


Рис. 6.22

Шуканий фрагмент тексту може містити спеціальні символи (табуляції, кінця абзацу, сторінки, розділу тощо), для введення яких треба натиснути на кнопку **Спеціальний**, або символи маски. Щоб мати змогу скористатись останнім засобом, слід натиснути на кнопку **Більше**, яка відкриває нижню ділянку діалогового вікна для задання додаткових параметрів пошуку (після чого кнопка змінює назву на **Менше**, і нею можна повернути вікно до попереднього вигляду), і ввімкнути опцію **Знаки підстановки**. Найважливіші символи маски означають таке:

*	довільна кількість будь-яких символів;
?	один будь-який символ (?іч задає пошук фрагментів ніч, річ, піч тощо);
[]	будь-який із символів, записаних у дужках ([мк]аска задає пошук фрагментів <i>маска</i> і <i>каска</i>);
[-]	будь-який із символів заданого діапазону ([в-д]ата задає пошук фрагментів <i>вата</i> , <i>гата</i> і <i>дата</i>);
[!]	один символ, відмінний від записаних після ! ([!мк]аска задає пошук фрагментів <i>ласка</i> , <i>паска</i> тощо, але не <i>маска</i> і <i>каска</i>);
[!-]	один символ, що не потрапляє у заданий діапазон ([!в-д]ата задає пошук фрагментів <i>фата</i> , <i>хата</i> тощо, але не <i>вата</i> , <i>гата</i> і <i>дата</i>);
<	початок слова (<від задає пошук слів <i>відчай</i> , <i>відпустка</i> тощо, але не <i>привід</i> , <i>розвідка</i>);
>	кінець слова (>ка задає пошук слів <i>сітка</i> , <i>відпустка</i> тощо, але не <i>кара</i> , <i>скарб</i>).

Для пошуку ж тих символів, які використовують як спеціальні для маски, перед ними слід поставити знак \ (наприклад, запис * означає, що треба відшукати символ *).

Опція **Враховувати регістр** дає змогу розрізняти малі та великі літери, **Лише слово цілком** – шукати тільки повне слово, а **Вимовляється як** та **Усі словоформи** працюють лише для англійської мови. У полі **Напрямок** задають напрям пошуку:

- **Всюди** – у всьому документі;
- **Назад** – вище від курсора вводу;
- **Вперед** – нижче від курсора вводу.

Кнопка **Формат** відкриває спеціальне меню, за допомогою команд якого (**Шрифт**, **Абзац**, **Табуляція**, **Стиль** тощо) можна включити в умови пошуку параметри форматування фрагмента, задавши їх у відповідних діалогових вікнах. Якщо при цьому не вводити у поле **Знайти** жодного тексту, то шукатимуться всі фрагменти із заданими параметрами форматування. Кнопка **Зняти форматування** вилучає форматування з умов пошуку.

Якщо в напрямках пошуку **Назад** або **Вперед** досягнуто початку (кінця) тексту, то буде видано відповідне повідомлення і запит на продовження пошуку в необстеженій частині тексту, після чого треба натиснути на кнопку **Так**, щоб продовжити пошук, або **Ні**, щоб завершити операцію.

Для заміни фрагментів тексту використовують вкладку **Замінити** діалогового вікна **Знайти й замінити**, яку можна активізувати безпосередньо у вікні (якщо воно вже відкрите) або за допомогою команди **Правка/Замінити** і яка, крім усіх названих вище полів, кнопок та опцій, містить:

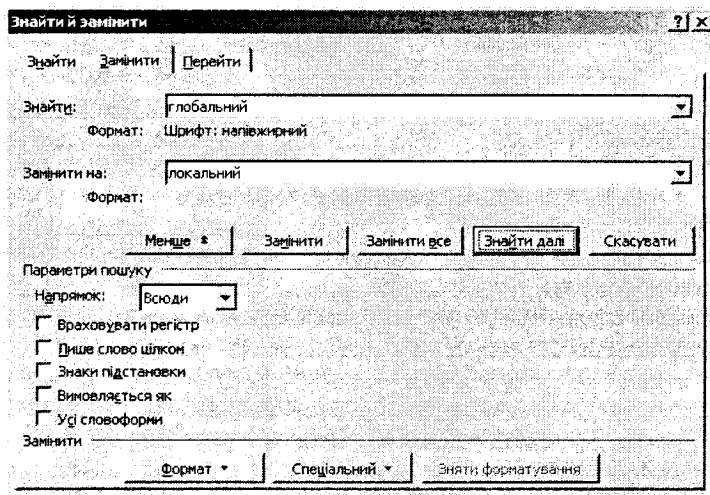


Рис. 6.23

- поле **Замінити на**, де слід увести фрагмент тексту для заміни шуканого;
- кнопку **Замінити** для заміни наступного знайденого фрагмента;
- кнопку **Замінити все** для заміни всіх фрагментів, що відповідають заданим умовам.

Третю вкладку діалогового вікна **Знайти й замінити** – **Перейти** – використовують для швидкого переходу на потрібний фрагмент тексту. Для цього у списку **Об'єкт переходу** вибирають тип об'єкта, до якого слід перейти (**Сторінка**, **Розділ**, **Примітка**, **Малюнок** тощо), а у полі праворуч – номер об'єкта або їх кількість, на яку потрібно зміститись (зі знаком «+» – вперед, зі знаком «-» – назад). Перехід здійснюється після натискання на кнопку **Перейти**, якщо номер або зміщення задано, або кнопку **Далі** чи **Назад**, якщо не задано.

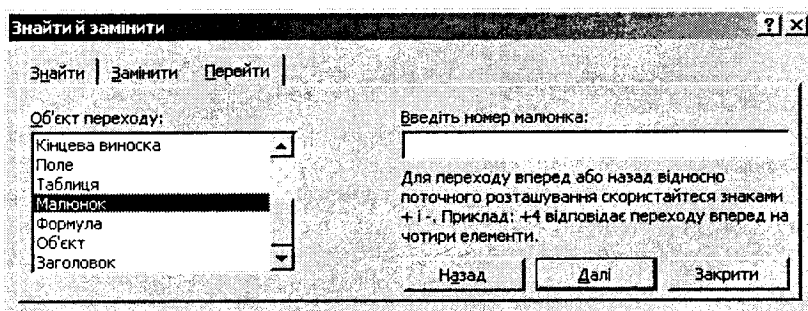


Рис. 6.24

□ Швидко відшукати наступний фрагмент тексту, заданий у діалоговому вікні **Знайти й замінити**, можна за допомогою комбінації клавіш **Shift+F4**.

6.6. Використання мовних функцій Word

Для того, щоб можна було скористатися такими функціями Word, як підтримка правопису, використання словника синонімів, автоматичне вставлення переносів тощо, потрібно насамперед надати фрагментам тексту атрибут «мова». Якщо цього не передбачено за замовчуванням (наприклад, у параметрах використовуваних стилів, див. п. 6.12), то слід виділити потрібні фрагменти, виконати команду **Сервіс/Мова/Вибрати мову** і задати відповідну мову. Опція **Визначати мову** автоматично дає змогу присвоювати введеному тексту мову, відповідну до вибраної клавіатури, а кнопка **За замовчуванням** – зробити вибрану мову стандартною для нових документів, базованих на шаблоні активного документа.

Тепер для пошуку та виправлення помилок у тексті документа треба виконати команду **Сервіс/Правопис**, яка відкриває відповідне діалогове вікно. У ньому:

- у полі **Немає у словнику** буде показано речення з помилковим (відсутнім у словнику) словом, виділеним червоним кольором;
- у списку **Варіанти** – пропозиції для заміни виділеного слова;
- у полі **Мова словника** – робочий словник (відповідний до заданої мови фрагмента), який можна змінити.
- опція **Граматика** дає змогу одночасно перевіряти відповідність речення граматичним правилам.

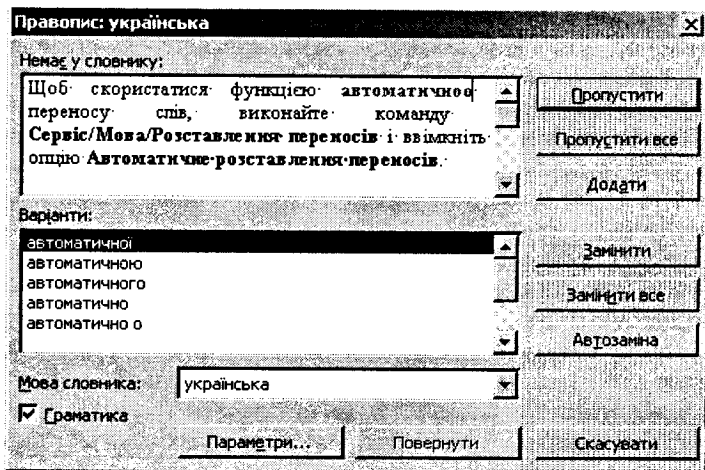


Рис. 6.25

Натискання на кнопки у правій та нижній ділянках діалогового вікна означає таке:

- **Пропустити** – пропущення поточного входження виділеного слова;
- **Пропустити все** – пропущення усіх входжень виділеного слова;
- **Додати** – додання виділеного слова у словник користувача;
- **Замінити** – заміна виділеного слова на вибране у списку **Варіанти**;
- **Замінити все** – заміна усіх входжень виділеного слова на вибране у списку **Варіанти**;
- **Автозаміна** – внесення пари «виділене слово – вибраний варіант» у список автозаміни;
- **Повернути** – повернення до попередньої помилки;
- **Параметри** – відкривання вкладки **Правопис** діалогового вікна **Параметри** для зміни параметрів перевірки правопису.

Слід зазначити, що вмикання на вкладці **Правопис** опції **автоматично перевіряти орфографію** за вимкненої не виділяти слова з помилками дає змогу бачити в тексті ймовірні помилки як слова, підкреслені червоною хвилястою лінією, з можливістю їх виправлення за допомогою контекстного меню, що відкривається правою кнопкою миші.

Мовні засоби Word 2003 дають також можливість скористатися функцією автоматичного перенесення слів. Для цього треба виконати команду **Сервіс/Мова/Розставлення переносів** і увімкнути опцію **Автоматичне розставлення переносів**.

□ *Мовний атрибут виділених фрагментів або тексту, що вводиться з поточної позиції курсора вводу, відображається у рядку статусу ближче до правого краю.*

□ *Увімкнута опція **автоматична зміна клавіатури** вкладки **Правопис** діалогового вікна **Параметри** задає режим, у якому поміщення курсора у фрагмент із заданою мовою автоматично змінює клавіатуру на відповідну до цієї мови. Якщо це заважає вставленню у текст іншомовних слів, опцію слід вимкнути.*

6.7. Таблиці

У Word 2003 засоби роботи з таблицями помітно посилені порівняно з ранішими версіями (зокрема з Word'97). Наприклад, у клітинку таблиці можна вставляти іншу таблицю, задавати режим автодобору, змінювати розміри всієї таблиці за допомогою миші тощо. Однак користуватися деякими з них слід обережно: скажімо, таблиця в таблиці може інколи стати джерелом незрозумілих ефектів.

6.7.1. Створення таблиць

Для створення таблиці стандартним способом використовують діалогове вікно **Вставка таблиці**, яке активізується командою **Таблиця/Вставити/Таблицю**. У ньому потрібно:

- у полях **Кількість стовпців** та **Кількість рядків** задати кількість стовпців та рядків;
- для опції **Автодобір ширини стовпців** вибрати одне із значень: **постійна** визначає сталу та однакову для всіх ширину стовпців, задану у полі ліворуч (**Авто** надасть усій таблиці ширину сторінки); **за вмістом** надає таблиці мінімальної ширини, однак розміри стовпців змінюватимуться залежно від уведеного тексту; **за шириною вікна** надає таблиці розміри сторінки, однак ширина стовпців може змінюватися залежно від вмісту;
- за бажанням у вікні **Автоформат таблиці**, що відкривається за допомогою кнопки **Автоформат**, вибрати стиль таблиці (початківцям рекомендовано використовувати стандартний стиль **Сітка таблиці**);
- увімкнути опцію **За замовчуванням для нових таблиць**, якщо потрібно зробити вибрані параметри стандартними для вставлення таблиць.

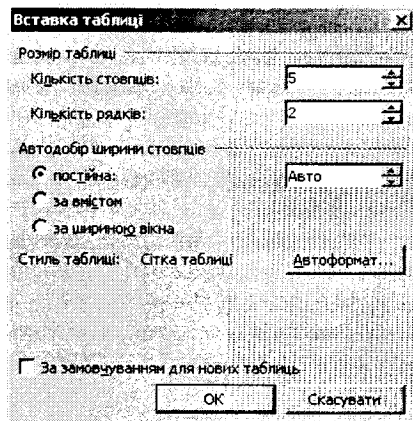


Рис. 6.26

Швидко створити таблицю можна за допомогою піктограми **Вставити таблицю** панелі **Стандартна**: для цього потрібно натиснути на неї і, не відпускаючи лівої кнопки миші, вибрати необхідну кількість рядків та стовпців. Якщо ж перед цим було виділено фрагмент тексту, то така дія (як і виконання команди **Таблиця/Вставити/Таблицю**) призведе до автоматичного поміщення фрагмента в таблицю, розміри якої визначаються кількістю абзаців (рядки) та символів табуляції або інших характерних символів (стовпці).

Окрім цих стандартних засобів, Word надає користувачам спеціальну можливість намалювати таблицю за допомогою інструмента **Створити таблицю**. Для цього потрібно виконати команду **Таблиця/Намалювати таблицю** або натиснути на піктограму **Панель меж** панелі **Стандартна** (в обох випадках відкриється панель інструментів **Таблиці та межі**) й інструментом у формі олівця намалювати прямокутник, відповідний до розмірів таблиці. Далі для додання нових рядків достатньо провести лінію від лівого краю таблиці до правого, а для додання нових стовпців – від верхнього краю до нижнього. Зайві лінії можна витерти інструментом **Гумка**.

Змінити зовнішній вигляд розмежувальних ліній виділених клітинок таблиці можна за допомогою звичайних засобів облямування, які надаються командою **Формат/Межі й заливка**, панеллю інструментів **Таблиці та межі** або (у найпростішому випадку, коли не потрібно змінювати товщину та колір ліній) спеціальною піктограмою **Межі** на панелі **Форматування**. Якщо ж деякі або всі розмежувальні лінії сховані, то для показу структури таблиці слід скористатись командою **Таблиця/Відобразити сітку**, яка відобразить сірого кольору сітку, що не виводиться на друк.

Командою **Таблиця/Розділити таблицю** таблицю можна розділити на дві окремі, починаючи з місця, де розташований курсор вводу, а дві таблиці – об'єднати, вилучивши всі абзаци між ними.

Для переміщення по таблиці, крім миші, можна використовувати такі команди клавіатури:

Tab	перехід до наступної клітинки (після останньої – вставлення нового рядка);
Shift+Tab	перехід до попередньої клітинки;
↓	перехід до наступного рядка;
↑	перехід до попереднього рядка;
Alt+Home	перехід до першої клітинки в рядку;
Alt+End	перехід до останньої клітинки в рядку;
Alt+PageUp	перехід до першої клітинки в стовпці;
Alt+PageDown	перехід до останньої клітинки в стовпці.

□ *Змінити режим автодобору ширини стовпців можна за допомогою відповідних підкоманд команди **Таблиця/Автодобір**. Початківцям рекомендовано використовувати постійну ширину (підкоманда **Фіксована ширина стовпця**), яку можна легко змінити самому (див. далі).*

□ *Інструментом **Створити таблицю** панелі **Таблиці та межі** можна намалювати діагональну лінію від одного кута клітинки до протилежного. Однак слід діяти обережно: якщо лінію «недотягнути», результатом буде вставлення у клітинку нової «одноклітинкової» таблиці.*

6.7.2. Робота з елементами таблиці

Для виділення клітинок таблиці слід натиснути ліву клавішу миші і, не відпускаючи її, перемістити курсор у потрібне місце. Виділені клітинки можна об'єднувати в одну командою **Таблиця/Об'єднати клітинки**. Команда **Таблиця/Розділити клітинки** дає змогу розбити виділені або поточну клітинки: у полі **Кількість стовпців** вікна **Розбивка клітинок** задають кількість стовпців для розділення клітинок, у полі **Кількість рядків** – кількість рядків, а опція **Об'єднати перед розбивкою** призначена для задання поділу всіх виділених клітинок як одного цілого (інакше визначеним у вікні способом ділитиметься кожна клітинка).

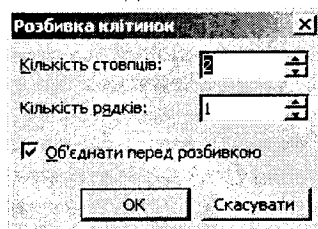


Рис. 6.27

Для виділення цілих рядків або стовпців таблиці, окрім проходження мишею по всіх потрібних клітинках, можна скористатися й іншими способами, наприклад, командами **Таблиця/Виділити рядок** або **Таблиця/**

Виділити стовпець, які виділяють рядки чи стовпці, що містять виділені клітинки. Команда **Таблиця/Виділити таблицю** виділяє всю таблицю.

Якщо у таблиці виділено рядки, стовпці або клітинки, то командами **Таблиця/Вставити/Рядки вище** та **Таблиця/Вставити/Рядки нижче** можна вставити нові рядки вище або нижче від виділених, причому їх вставляється стільки, скільки було виділено перед виконанням команди. Стовпці вставляють аналогічно командами **Таблиця/Вставити/Стовпці зліва** та **Таблиця/Вставити/Стовпці справа**.

Окремі клітинки вставляють у таблицю командою **Таблиця/Вставити/Клітинки** після виділення потрібної їх кількості. Ця команда відкриває вікно **Додавання клітинок**, де треба зазначити спосіб виконання операції:

- **зі зсувом праворуч** – виділені клітинки зсуваються вправо;
- **зі зсувом донизу** – виділені клітинки зсуваються вниз;
- **вставити цілий рядок** – буде вставлено рядки перед першою виділеною клітинкою;
- **вставити цілий стовпець** – буде вставлено стовпці перед першою виділеною клітинкою.

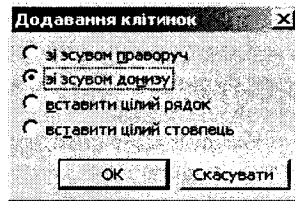


Рис. 6.28

Виділені рядки або стовпці можна вилучити з документа командами **Таблиця/Видалити/Рядки** і **Таблиця/Видалити/Стовпці** відповідно. За допомогою команди **Таблиця/Видалити/Таблицю** вилучають усю таблицю, причому її навіть не треба виділяти. Для вилучення виділених клітинок призначена команда **Таблиця/Видалити/Клітинки**, яка відкриває вікно, де потрібно задати спосіб вилучення аналогічно до вставлення.

Для зміни розмірів клітинок таблиці є кілька способів. По-перше, їх можна змінити безпосередньо в таблиці за допомогою миші, наводячи курсор на розмежувальні лінії і переміщуючи їх на нові місця (причому в разі попереднього виділення клітинок, на які наводять курсор, зміна ширини стосуватиметься тільки їх; слід також мати на увазі, що за замовчуванням висота рядка під час введення в клітинку тексту автоматично стає такою, щоб показати його повністю). По-друге, розміри клітинок наочно задають за допомогою лінійок форматкування (тому висоту рядка не можна змінити у режимі перегляду **Звичайний**, бо вертикальна лінійка там не відображається). Третій спосіб полягає у використанні команди **Таблиця/Властивості таблиці**, яка відкриває однойменне вікно з вкладками **Таблиця**, **Рядок**, **Стовпець** і **Клітинка**.

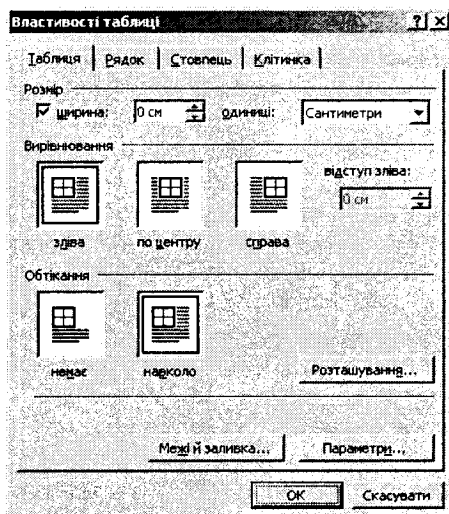


Рис. 6.29

На вкладці **Таблиця** можна задати параметри всієї таблиці:

- у рамці **Розмір** явно змінити ширину (увімкнувши однойменну опцію і ввівши потрібне значення);
- у рамці **Вирівнювання** вибрати спосіб вирівнювання всієї таблиці відносно сторінки, причому для значення **зліва** можна задати відступ у полі праворуч;
- вибрати потрібне значення опції **Обтікання** для обтікання таблиці текстом, причому значення **навколо** дає змогу змінити стандартні параметри обтікання за допомогою кнопки **Розташування**.

Кнопка **Межі й заливка** призначена для задання вигляду обмежувальних ліній та способу затінення таблиці, а кнопка **Параметри** – для зміни деяких стандартних параметрів таблиці (полів клітинок, інтервалу між клітинками, параметрів автодобору).

На вкладці **Рядок**, увімкнувши опцію **висота**, можна явно задати висоту поточного або попередньо виділених рядків (інакше вона буде підбрана автоматично відповідно до розміру тексту), причому за режиму **Мінімум** висота не може бути меншою за задане значення. Опція **дозволити перенос рядків на наступну сторінку** дає змогу розділяти вміст клітинки у випадку його продовження на іншу сторінку, а опцією **повторювати як заголовок на кожній сторінці** виділені рядки визначають як заголовок таблиці, який буде автоматично повторюватися на сторінках-продовженнях (для цього також використовують команду **Таблиця/Заголовки**). Кнопки **Попередній рядок** та **Наступний рядок** призначені для переходу відповідно до попереднього та наступного рядків без повернення у текст документа.

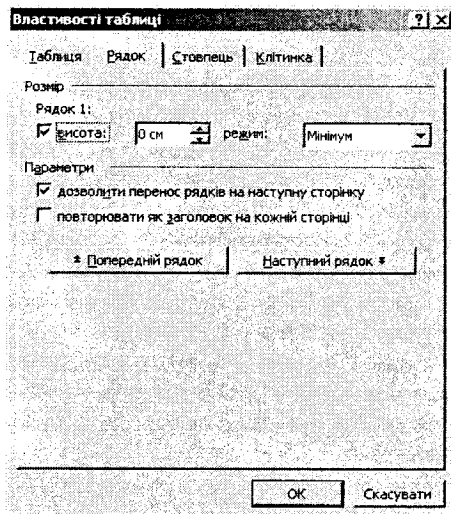


Рис. 6.30

На вкладці **Стовпець** задають параметри стовпців подібно до того, як це роблять на вкладці **Рядок** для рядків. На вкладці **Клітинка** можна змінити ширину поточної або попередньо виділених клітинок, вибрати спосіб вирівнювання їхнього вмісту по вертикалі (опція **Вертикальне вирівнювання**), а також змінити деякі параметри (зокрема, поля у клітинках) за допомогою кнопки **Параметри**.

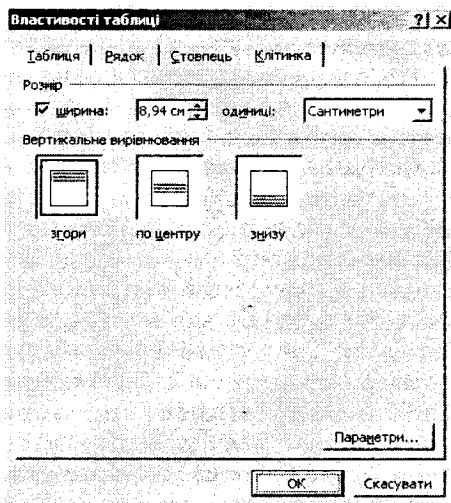


Рис. 6.31

Крім цього, за допомогою команди **Таблиця/Автодобір/Вирівняти висоту рядків** виділеним клітинкам (рядкам) можна надати однакової висоти, а команда **Таблиця/Автодобір/Вирівняти ширину стовпців** встановлює однакову ширину для виділених клітинок або стовпців.

□ Для швидкого виділення рядків або стовпців найпростіше відвести курсор ліворуч або вгору від таблиці і, коли він набуде вигляду стрілки, вибрати потрібні рядки чи стовпці. Цілу таблицю можна виділити за допомогою спеціальної піктограми, яка з'являється після наведення курсора на таблицю в її верхньому лівому куті у режимах перегляду **Розмітка сторінки та Веб-документ**.

□ Щоб швидко додати до таблиці новий рядок, достатньо помістити курсор в останню клітинку і натиснути на клавішу **Tab**.

6.7.3. Робота із вмістом таблиці

Текст усередині таблиці вводять, редагують та форматують за допомогою тих самих засобів, що й звичайний текст. Для форматування елементів таблиці можна скористатися, окрім описаних раніше, ще й автоматизованими засобами, які надаються користувачу після виконання команди **Таблиця/Автоформат таблиці**. У вікні, що відкриється (рис. 6.32), слід вибрати один із наявних стилів форматування (з можливістю його зміни або створення нового) і, досягнувши потрібного результату, який відображається у віконці **Зразок**, натиснути на кнопку **ОК**.

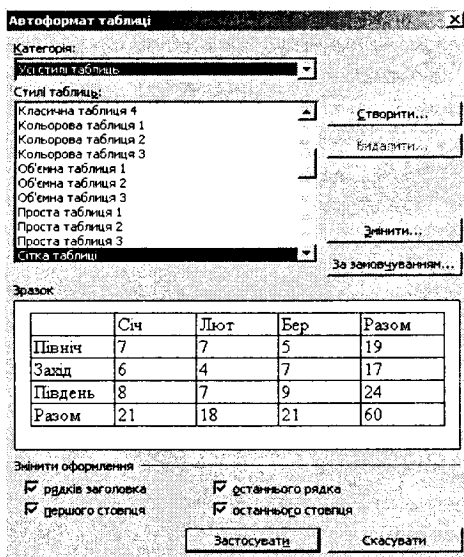


Рис. 6.32

Щоб визначити заголовок таблиці для його автоматичного повторення на сторінках-продовженнях, окрім вкладки **Рядок** діалогового вікна **Властивості таблиці**, використовують команду **Таблиця/Заголовки**, перед виконанням якої слід виділити потрібні рядки. Цією ж командою заданий заголовок відмінюють. Для вирівнювання вмісту виділених клітинок по вертикалі, крім вкладки **Клітинка** діалогового вікна **Властивості таблиці**, використовують підкоманди пункту **Вирівнювання в клітинці** локального меню, а також спеціальну піктограму панелі інструментів **Таблиці та межі**. За допомогою команди **Напрямок тексту** меню **Формат** або локального меню виділених клітинок можна змінити орієнтацію тексту у цих клітинках.

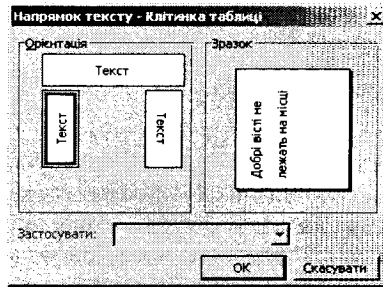


Рис. 6.33

Для автоматичної нумерації виділених клітинок використовують команду **Формат/Список** або піктограму **Нумерація** панелі **Форматування**. Клітинки нумеруються зліва направо і зверху вниз.

Рядки таблиці (як і, взагалі кажучи, просто абзаци тексту) можна сортувати командою **Таблиця/Сортування**. Для цього у діалоговому вікні **Сортування** (рис. 6.34) слід задати:

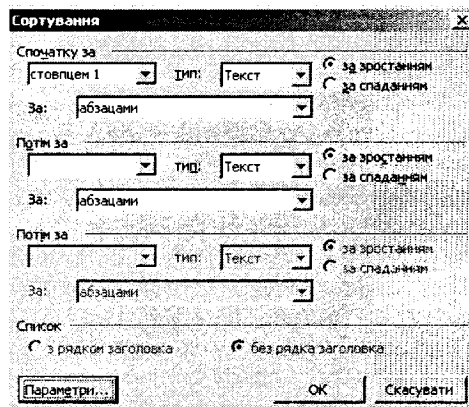


Рис. 6.34

- стовпці, за якими буде відбуватися сортування (спершу у полі **Спочатку за**, а потім послідовно у двох полях **Потім за**);
- тип сортованої інформації (поле **тип**);
- напрямок сортування (опції **за зростанням** або **за спаданням**);
- спосіб сортування (список **За**: за абзацами чи наявними полями, розділювач яких задано у діалоговому вікні **Параметри сортування**, що відкривається кнопкою **Параметри**).

Щоб вилучити з процесу сортування заголовки таблиці, потрібно задати значення **без рядка заголовка** опції **Список**. Кнопка **Параметри** призначена для задання більш детальних параметрів сортування.

□ *Виділені абзаци звичайного тексту, що містять символи-розділювачі певних груп слів, можна перетворити в таблицю командою **Таблиця/Перетворити/Текст на таблицю**, а виділені рядки таблиці – у текст командою **Таблиця/Перетворити/Таблицю на текст**.*

□ *Щоб вставити в клітинку символ табуляції, треба натиснути на комбінацію клавіш **Ctrl+Tab**.*

□ *Для форматування елементів таблиці зручно користуватись панеллю **Таблиці та межі**, яку можна відкрити однойменною піктограмою панелі інструментів **Стандартна** або командою **Таблиця/Намалювати таблицю**.*

□ *Word 2003 надає можливість автоматичного підсумовування числових значень у стовпцях або рядках: для цього слід поставити курсор у нижню клітинку стовпця (або крайню праву рядка) і натиснути на піктограму **Автосума** панелі **Таблиці та межі**. У двозначних ситуаціях перевага надається підсумовуванню стовпців.*

6.8. Вставлення та форматування ілюстрацій

Ілюстрації у документі можна розміщувати двома способами: як незалежні зображення і як елементи тексту. У першому випадку ілюстрацію можна переміщувати в будь-яке місце сторінки, створювати ефект обтікання текстом, накладати на текст та інші ілюстрації тощо. В іншому разі вона буде розміщена в документі подібно до символу тексту, у процесі редагування документа не змінюватиме свого місця розташування щодо сусідніх символів, і до неї можна застосовувати, наприклад, параметри форматування абзацу. Користувачам-початківцям рекомендовано використовувати другий спосіб, і в цьому пункті йтиметься саме про нього. Щоб запровадити такий режим вставлення ілюстрацій, потрібно на вкладці **Правка** діалогового вікна **Параметри** (яке відкривається командою **Сервіс/Параметри**) надати опції **вставляти малюнки як значення у тексті**. Роботу із незалежними зображеннями буде розглянуто під час вивчення можливостей програми Word 2003 як видавничої системи.

Отже, щоб вставити у текст документа ілюстрації з файлів, підготовлених зовнішніми графічними редакторами (власні ілюстративні засоби Word 2003 також буде описано далі), потрібно виконати команду меню **Вставка/Малюнок/З файлу**, яка відкріє діалогове вікно **Додавання малюнка**. У ньому слід вибрати потрібний файл і натиснути на кнопку **Вставити**.

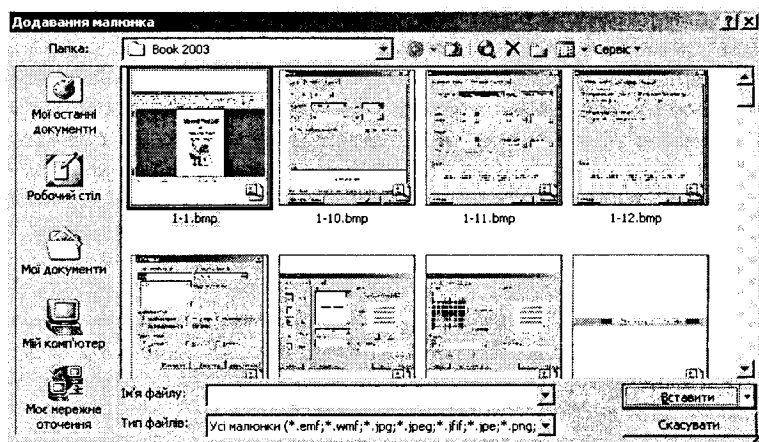


Рис. 6.35

Ілюстрацію можна вставити також із буфера, попередньо помістивши її туди у робочому вікні відповідного графічного редактора, командами **Правка/Вставити** або **Правка/Спеціальна вставка**. Ілюстрація буде поміщена у те місце документа, де розташований курсор вводу.

Вставлену ілюстрацію можна переміщувати, копіювати та вилучати з документа, а також змінювати її розміри. Для виконання будь-якої з цих операцій зображення треба попередньо виділити, натиснувши на ньому мишею. Щоб перемістити ілюстрацію, достатньо просто захопити її і транспортувати у потрібне місце, якщо ж одночасно тримати натиснутою клавішу **Ctrl**, то переміщуватиметься копія. Виділену ілюстрацію можна скопіювати або вирізати в буфер, а також просто вилучити відповідними командами меню **Правка**. Для зміни розмірів об'єкта (зі зміною масштабу відображення) необхідно переміщувати індикатори, що містяться на сторонах позиційної рамки, причому переміщення кутових індикаторів з клавішею **Shift** дає змогу зберегти пропорції. А ось для того, щоб обрізати ілюстрацію або додати до неї вільний простір (тобто змінити її розміри без зміни масштабу відображення), потрібно використовувати спеціальний інструмент **Обтинання** панелі **Налаштування зображення**, яка має автоматично відкриватись після виділення ілюстрації (інакше в її локальному меню слід виконати команду **Відобразити панель налаштування зображення**).

Точніше задати параметри ілюстрації можна у діалоговому вікні **Формат малюнка**, яке активізується командою **Формат/Малюнок**. У ньому на вкладці **Розмір**:

- у полях **висота** і **ширина** групи **Розмір і поворот** можна задати новий розмір ілюстрації, а в полях групи **Масштаб** – масштаб відображення для висоти і ширини, причому відповідні поля цієї та попередньої груп взаємопов'язані;
- опція **зберегти пропорції** призначена для збереження початкових пропорцій малюнка (і тоді перемішувати кутові індикатори за допомогою миші можна й без утримування клавіші **Shift**);
- опція **відносно вихідного розміру** дає змогу задавати масштаб відображення ілюстрації відносно її початкового розміру (у разі вимкнення він визначатиметься відносно попереднього розміру);
- у рамці **Вихідний розмір** виводиться початковий розмір ілюстрації, а кнопка **Скид** дає змогу його відновити.

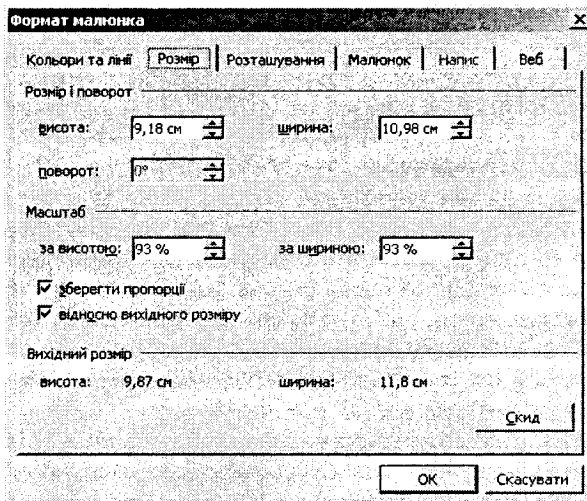


Рис. 6.36

На вкладці **Малюнок**:

- у полях групи **Обтинання** задають розміри полів, що відтинаються або додаються (якщо значення від'ємні) до відповідних сторін ілюстрації (зліва, справа, зверху, знизу);
- у полях групи **Зображення** визначають кольорову гаму відображення ілюстрації (поле **колір** із значеннями: **Авто** – без змін, **Відтінки сірого** – напівтонова (у градаціях сірого), **Чорно-біле** – тільки чорний та білий кольори, **Підкладка** – бліді кольори), яскравість і контрастність (відповідні поля та індикатори);

- кнопка **Скид** дає змогу відновити вихідні параметри, а кнопка **Стиснути** – зменшити розмір пам'яті, необхідної для зберігання ілюстрації, за рахунок зниження якості зображення.

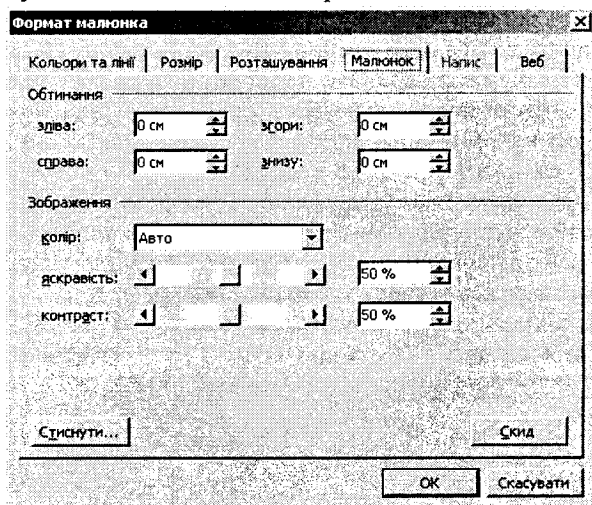


Рис. 6.37

Якщо ілюстрація має векторний формат, то на вкладці **Кольори та лінії** можна змінити кольори її фону та ліній. Вкладку **Розташування** буде розглянуто під час вивчення програми Word 2003 як видавничої системи.

Слід також зазначити, що практично все сказане раніше стосується будь-яких об'єктів (електронних таблиць Excel, малюнків Corel Draw, математичних формул тощо), вставлених у документ через буфер або за допомогою команди **Вставка/Об'єкт**.

Багато цікавих ілюстрацій є у спеціальній графічній бібліотеці **Clip-Art Gallery**. Щоб вставити звідти малюнок у документ, необхідно виконати команду **Вставка/Малюнок/Картинки**, вибрати серед запропонованих малюнків той, що Вам подобається, і вибрати з локального списку команд **Вставити**.

□ *Якщо велика ілюстрація вставлена як елемент тексту, то варто тримати її в окремому абзаці, інакше решта тексту абзацу «тягнутиметься» за лівим чи правим нижніми краями позиційної рамки. Щоб поліпшити сприймання ілюстрації, у її абзаці слід забрати відступ для першого рядка, задати вирівнювання до центру і дати кілька пунктів інтервалу перед та після абзацу. Сказане не стосується невеличких графічних об'єктів (наприклад, формул), які й повинні бути вставлені всередину текстового абзацу.*

□ У багатьох випадках вставлені ілюстрації потрібно підписувати та нумерувати – заняття не вельми цікаве і не завжди просте, особливо якщо після цього доводиться перенумерувати те, що було вставлено раніше. Однак цей процес можна значно полегшити, якщо виділити ілюстрацію, виконати команду **Вставка/Посилання/Назва**, вибрати у полі підпис потрібний підпис (якщо його там немає, то можна створити, натиснувши на кнопку **Створити**), проконтролювати правильність підпису та нумерації у полі **Назва** (для кожного підпису ведеться окрема нумерація) і вставити їх у документ кнопкою **ОК**. У цьому разі вставлені раніше ілюстрації будуть перенумеровані, якщо виділити фрагмент тексту, що містить їх, і натиснути **F9** для оновлення полів.

□ Якщо ілюстрації – елементи тексту не переміщуються за допомогою миші, то слід скористатися даною раніше для тексту порадою: виконати команду **Сервіс/Параметри** і на вкладці **Правка** діалогового вікна **Параметри** увімкнути опцію **використовувати перетягнення тексту**.

□ Коли трапляється таке, що ілюстрація вставлена, а в документі замість неї з'явилась порожня рамка, то, можливо, на вкладці **Вигляд** діалогового вікна **Параметри** увімкнено опцію **пусті рамки малюнків**. Потрібно її вимкнути і ще раз поглянути на ілюстрацію.

6.9. Математичні формули

Однією із суттєвих переваг текстового процесора Word для фахівців у галузі природничих наук є наявність у ньому вбудованого редактора формул, який дає змогу легко, швидко і наочно оформити математичний вираз довільної складності.

Щоб вставити в документ математичну формулу, необхідно помістити курсор у потрібне місце, виконати команду **Вставка/Об'єкт**, перейти на вкладку **Створення** і вибрати у списку **Тип об'єкта** позицію **Microsoft Equation 3.0**, не забувши при цьому вимкнути опцію **У вигляді значка**. Однак значно швидше та простіше зробити це можна за допомогою піктограми **Редактор формул**. Відкриється спеціальне вікно для введення та редагування формул і панель **Формула** для набирання символів, яких немає на клавіатурі (її вимикають або вмикають командою **Вид/Панель інструментов**). Панель містить піктограми, натискання на які викликає появу відповідного набору символів або шаблонів. Для вставлення у формулу потрібного елемента слід просто натиснути на нього.

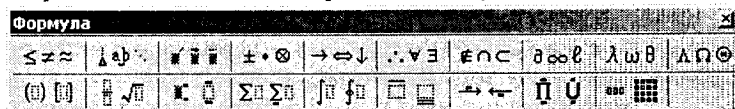


Рис. 6.38

Різні елементи формул формують по-різному, причому відповідні параметри будуть застосовані автоматично залежно від вставляваних символів чи шаблонів. Для вибору ж стилю (набору параметрів форматування) вручну використовують команди меню **Стиль**:

- **Математический** – стиль, що автоматично застосовується до змінних та функцій (стандартний);
- **Текст** – стиль для тексту;
- **Функция** – для функцій;
- **Переменная** – для змінних;
- **Греческий** – для символів грецького алфавіту;
- **Матрица-Вектор** – для елементів векторів та матриць.

Для модифікації параметрів цих стилів застосовують команду **Стиль/Определить**, яка активізує спеціальне діалогове вікно. У ньому для кожного стилю задають шрифт (поле **Шрифт**) та накреслення (опції **Полужирный** та **Наклонный** (опції **Полужирный** та **Курсив**). Якщо потрібно ввести у формулу символи з відмінними від визначених вище параметрами форматування, то використовують команду **Стиль/Другой**.

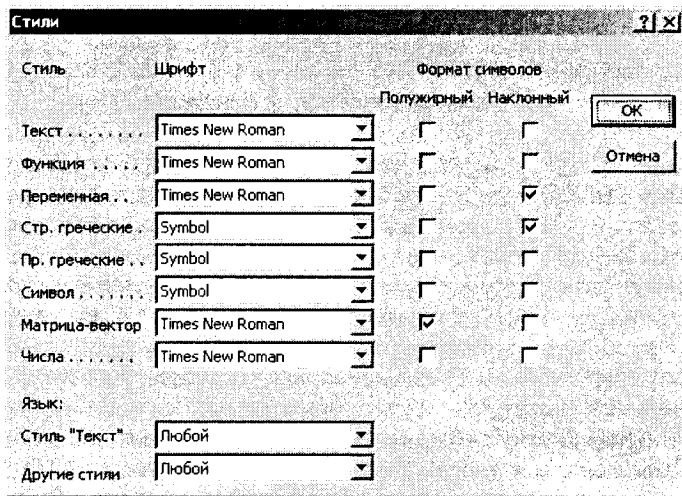


Рис. 6.39

Розмір елементів формул визначають у вікні **Размеры**, що відкривається командою **Размер/Определить**:

- **Обычный** – звичайні символи;
- **Крупный индекс** – символи у верхньому та нижньому індексах;
- **Мелкий индекс** – символи в індексах індексів;
- **Крупный символ** – спеціальні символи: сума, інтеграл тощо;
- **Мелкий символ** – спеціальні символи в індексах.

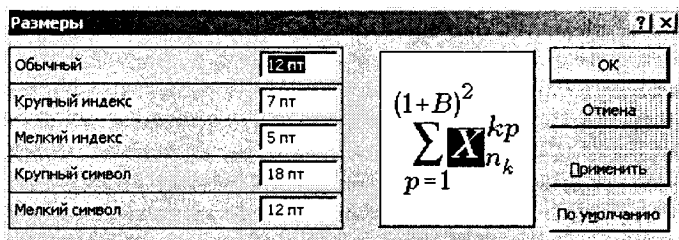


Рис. 6.40

Щоб використати інший розмір, потрібно виконати команду **Размер/Другой**.

Відстані між елементами формул задають у діалоговому вікні **Интервал**, яке активізується командою **Формат/Интервал**. У лівій частині вікна міститься список відстаней між різними елементами формули, а зображення праворуч пояснює їхній зміст. Результат зроблених змін можна побачити у формулі, не закриваючи вікна, якщо натиснути на кнопку **Застосувати**. Кнопка **По умолчанию** призначена для відновлення стандартних значень.

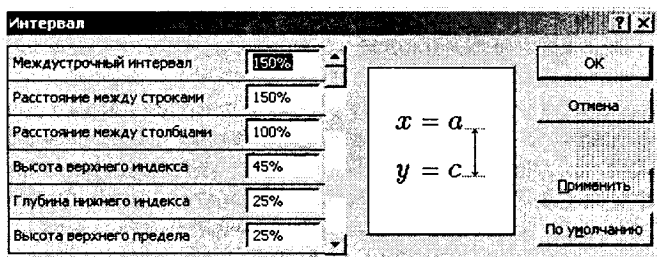


Рис. 6.41

Для завершення редагування формули потрібно клацнути мишею за її межами. Щоб редагувати повторно, слід виконати команду меню **Правка/Об'єкт Формула/Змінити**). Для редагування формули в окремому вікні використовують команду **Правка/Об'єкт Формула/Відкрити**); у цьому випадку можна змінювати масштаб відображення формули командами меню **Вид**.

□ Подвійне натискання мишею на потрібній формулі дає змогу швидко перейти в режим редагування, а клавіша **Esc** – швидко вийти з нього.

□ Розміри для чотирьох нижніх полів діалогового вікна **Размеры** можна задавати у відсотках до значення, вказаного в полі **Обычный**. Це дає змогу в разі потреби змінювати розмір лише у верхньому полі. Стандартні ж значення для нижніх полів у порядку їх розміщення згоризонту становлять відповідно 58%, 42%, 150%, 100%.

□ *Незрозумілі пусті місця всередині абзацу цілком можуть виявитись порожніми формулами, якщо, наприклад, натиснути на піктограму **Редактор формул**, а набирати формулу передумати. Це може призвести до проблем з вирівнюванням окремих рядків, тому слід виділяти такі «пробіли» і вилучати. Хоча краще відразу не допускати їх...*

□ *Якщо формула раптом стала «відриватися» від свого місця в абзаці, то це означає, що вона якимось чином перетворилася з елемента тексту на незалежне зображення. У такому випадку слід її повернути у текст: виділити, виконати команду **Формат/Об'єкт** і на вкладці **Розташування** вибрати значення у **тексті** опції **Обтікання**.*

6.10. Основи верстання. Вставлення тексту

Під час роботи з програмою Word як видавничою системою матимемо на увазі, що частина текстової інформації може бути вже набрана за допомогою інших текстових редакторів (зокрема, й самого Word) і міститись у файлах на запам'ятовувальних пристроях. Переважну частину ілюстрацій також готують за допомогою інших програм – графічних редакторів. Процес розміщення тексту та ілюстративного матеріалу на сторінках документа називають верстанням. Його слід виконувати в режимі перегляду **Розмітка сторінки**.

Для того, щоб вставити текст із файлу в документ, необхідно виконати команду меню **Вставка/Файл**, вибрати потрібний файл і натиснути на кнопку **ОК**. Текст буде вставлятися, починаючи з місця, де розташований курсор вводу. Якщо програма не зуміла сама визначити тип імпортованого файлу, то відкриється діалогове вікно **Перетворення файлу** (рис. 6.42), де слід задати потрібне значення опції **Кодування тексту**, у разі необхідності послуговуючись віконцем **Зразок**.

Ще одним поширеним способом вставлення тексту є обмін фрагментами через буфер за допомогою команд меню **Правка Word 2003** та відповідних команд редагування інших програм.

□ *Якщо Word 2003 неправильно визначає в автоматичному режимі формат імпортованого файлу, слід виконати команду **Сервіс/Параметри** і на вкладці **Загальні** діалогового вікна **Параметри** увімкнути опцію **підтвердити перетворення при відкритті**. Це дасть змогу визначати тип файлу самостійно у діалоговому вікні **Перетворення файлу**.*

□ *Тексти, що зберігаються у форматі ASCII, мають один суттєвий недолік: кожен їхній рядок закінчується символом, який Word розглядає як кінець абзацу. Найпростіше у такому випадку застосувати до вставленого тексту автоформатування (піктограма **Формат/Автоформат**), однак це не завжди допомагає. Тоді можна дати таку пораду: командою **Правка/Замінити** замінити у всьому тексті символи «^p» на «^l»*

(тобто маркер абзацу з пробілом – на кінець рядка з пробілом), потім усі маркери абзаців «^p» – на пробіли « », і, нарешті, «^l» – на «^p». Однак, щоб ці маніпуляції мали успіх, перший рядок кожного абзацу DOS'івського тексту повинен починатися принаймні з одного пробілу, а решта рядків – зі значущого символу.

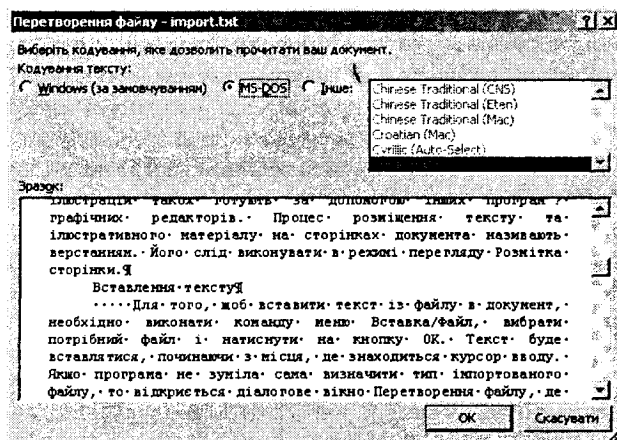


Рис. 6.42

□ Якщо, крім того, у DOS-тексті було застосовано переноси слів, потрібно додатково повилучати «-^p» (замінивши їх на порожні місця), причому зробити це слід перед усіма іншими замінами.

□ Позбутись у тексті зайвих пробілів, які використовують у DOS-текстах для вирівнювання за шириною (а також люблять вживати недосвідчені користувачі Word), не так уже й складно: слід замінити два пробіли на один доти, аж поки не з'явиться повідомлення, що замінити більше нічого. Після цього корисно позамінювати всі «^p» (маркери абзацу з пробілом) на «^p» (без пробілу), а також символи «^p^t» на «^p», оскільки, на жаль, багато хто задає відступ для першого ряду абзацу за допомогою пробілів або табуляції.

6.11. Формування зовнішнього вигляду документа

6.11.1. Оформлення сторінок

Текст у Word 2003 автоматично буде розбито на сторінки, якщо увімкнута опція **фонова розбивка на сторінки** на вкладці **Загальні** діалогового вікна **Параметри**, причому в режимі перегляду **Розмітка сторінки** це робиться завжди. Почати введення тексту з нової сторінки можна за допомогою команди **Вставка/Розрив** і вибору опції **Почати/нову сторінку** або натисканням на комбінацію клавіш **Ctrl+Enter**.

Параметри сторінок задають за допомогою команди **Файл/Параметри сторінки**, яка відкриває однойменне діалогове вікно з трьома вкладками: **Поля**, **Папір** і **Макет**.

На вкладці **Поля** задають:

- у рамці **Поля** – розміри полів: **Верхнє**, **Нижнє**, **Ліве**, **Праве**, – відповідно зліва, справа, зверху та знизу, **Корінець** – поле корінця, **Розташування корінця** – його місце (згори або зліва), доступне лише для режиму **Звичайний** (поле декілька сторінок рамки **Сторінки**);
- опцію **Орієнтація**: **книжкова** для поздовжньої орієнтації або **альбомна** для поперечної;
- у полі **декілька сторінок** рамки **Сторінки** – параметри для верстання декількох сторінок: **Звичайний** – одна сторінка з однібічним верстанням, **Дзеркальні поля** – одна сторінка з двобічним верстанням (тоді замість лівого і правого полів задають **Усередині** – внутрішнє та **Зовні** – зовнішнє), **2 сторінки на аркуші** та **Брошура**.

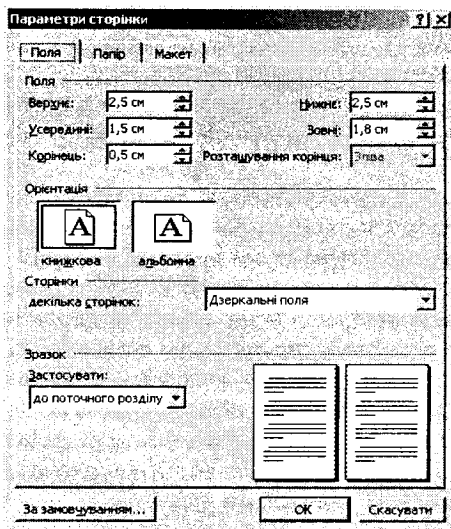


Рис. 6.43

Більшість полів можна задати й за допомогою лінійок форматування, які відображаються на екрані після виконання команди **Вигляд/Лінійка** для горизонтальної лінійки та вмикання опції **вертикальна лінійка** (режим **розмітки**) вкладки **Вигляд** діалогового вікна **Параметри** для вертикальної. Межі смуги набору тексту можна відобразити, увімкнувши опцію **межі тексту** на тій же вкладці вікна **Параметри**.

Розміри сторінок документа визначають на вкладці **Папір** вікна **Параметри сторінки**, де у полі **Розмір паперу** треба вибрати формат або

задати його вручну у полях **Ширина** та **Висота**. У рамці **Подача паперу** задають спосіб подачі паперу у принтер для першої та решти сторінок.

На вкладці **Макет** визначають деякі інші параметри сторінок документа. Поле **Почати розділ** рамки **Розділ** задає місце початку нового розділу (про розділи йтиметься у наступному пункті), а опція **Заборонити кінцеві виноски** запобігає друку кінцевих виносок у поточному розділі (вони будуть розташовані перед виносками наступного розділу).

В рамці **Розрізняти колонтитули** задають параметри для колонтитулів:

- опція **парних і непарних сторінок** дає змогу задавати різні колонтитули для парних і непарних сторінок;
- опція **першої сторінки** – окремий колонтитул для першої.
- в полях групи **Від краю** задають відстані: **до верхнього колонтитула** – від верхнього краю сторінки до верхнього краю верхнього колонтитула; **до нижнього колонтитула** – від нижнього краю сторінки до нижнього краю нижнього колонтитула.

У полі **Вертикальне вирівнювання** рамки **Сторінка** задають спосіб вертикального вирівнювання тексту: **За верхнім краєм** – до верхнього краю, **По центру** – до центру, **За висотою** – рівномірно між верхнім та нижнім краями, **За нижнім краєм** – до нижнього краю.

Кнопка **Нумерація рядків** дає змогу задавати режим нумерації рядків документа, а кнопка **Межі** – облямування сторінки за допомогою діалогового вікна **Межі й заливка**.

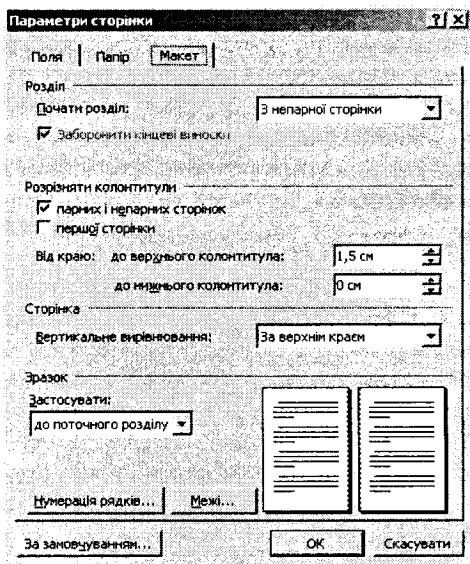


Рис. 6.44

Спільним для всіх вкладок діалогового вікна **Параметри сторінки** є рамка **Зразок**, де у полі **Застосувати** визначають межі дії заданих параметрів (до всього документа, до виділеного тексту, до кінця документа, до виділених розділів, до поточного розділу), а ділянка праворуч відображає результати внесених змін. Кнопка **За замовчуванням** дає змогу зберегти ці параметри як стандартні для всіх нових документів, базованих на поточному шаблоні.

Будь-які параметри сторінок можна змінювати під час верстання. Саме верстання слід виконувати в режимі **Розмітка сторінки**, який вмикається однойменною командою меню **Вигляд** або відповідною піктограмою, що міститься ліворуч від горизонтальної смуги прокручення.

□ Для того, щоб легко активізувати діалогове вікно **Параметри сторінки**, достатньо двічі натиснути мишею трохи збоку від горизонтальної лінійки форматування, яка, зрозуміло, повинна бути увімкненою.

□ Опція **поля між сторінками (режим розмітки)** вкладки **Вигляд** діалогового вікна **Параметри** керує візуалізацією/прихованням верхнього і нижнього полів сторінки. Наочно це можна зробити, натискаючи мишею на проміжку між сторінками в режимі перегляду **Розмітка сторінки**.

□ Якщо для задання розмірів полів документа **Word 2003** пропонує якісь незрозумілі одиниці (наприклад, " або пк), потрібно вибрати у полі **Одиниці вимірювання** вкладки **Загальні** діалогового вікна **Параметри** **сантиметри**.

6.11.2. Розділи

Якщо документ повинен складатись зі сторінок з різними параметрами, то його слід розбити на окремі розділи. Це роблять за допомогою команди меню **Вставка/Розрив**, яка активізує діалогове вікно **Розрив**.

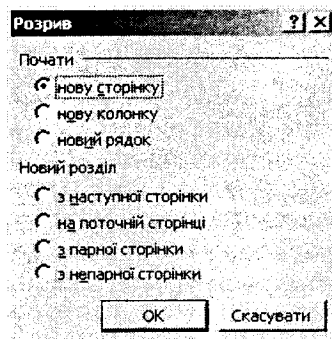


Рис. 6.45

Для вставлення нового розділу необхідно вибрати одне зі стандартних значень опції **Новий розділ**, яка вказує на місце початку розділу:

- з **наступної сторінки** – з наступної сторінки;
- на **поточній сторінці** – далі на тій же сторінці;
- з **парної сторінки** – з найближчої парної сторінки;
- з **непарної сторінки** – з найближчої непарної.

Текст, що міститься за курсором вводу, буде перенесено у новий розділ. Кінець попереднього розділу позначається спеціальним рядком **Кінець розділу**, який належить до недрукованих символів. Тепер для окремих розділів можна встановлювати різні параметри сторінок в описаному раніше вікні, вибираючи у полі **Застосувати значення до виділених розділів** або до **поточного розділу**.

□ *Місце початку розділу можна змінити, вибравши нове значення поля **Почати розділ** вкладки **Макет** вікна **Параметри сторінки**. Окрім уже наведених, можна задавати значення **З нової колонки** – розділ буде починатися з наступної колонки.*

□ *Якщо змінити параметри сторінки для попередньо виділеного фрагмента (вибравши у полі **Застосувати значення до виділеного тексту**), то цей фрагмент буде автоматично перенесений у новий розділ.*

□ *Нагадаємо також, що швидко розпочати нову сторінку (вставити розрив сторінки) можна комбінацією клавіш **Ctrl+Enter**, а новий рядок – комбінацією **Shift+Enter**. Ще один спосіб помістити абзац на нову сторінку – виконати для нього команду **Формат/Абзац** і на вкладці **Розташування на сторінці** увімкнути опцію з **нової сторінки**.*

6.11.3. Колонки

У документах Word 2003 можна використовувати колонки як однакової, так і різної ширини, причому їх допустима кількість обмежена лише розмірами сторінки. Текст послідовно заповнює колонки зліва направо.

Для швидкого створення колонок однакової ширини слід, якщо це необхідно, виділити фрагмент тексту, натиснути на піктограму **Колонки** панелі інструментів **Стандартна 1**, не відпускаючи лівої клавіші миші, вибрати потрібну їх кількість. Для точного задання параметрів колонок використовують діалогове вікно **Колонки**, яке активізується командою **Формат/Колонки** і де у рамці **Тип** можна вибрати один із п'яти готових варіантів. Якщо жоден з них не підходить, можна створити колонки самому, задавши:

- у полі **Кількість колонок** – їхню кількість;
- у полі **ширина** – ширину колонок;
- у полі **проміжок** – відстані між ними.

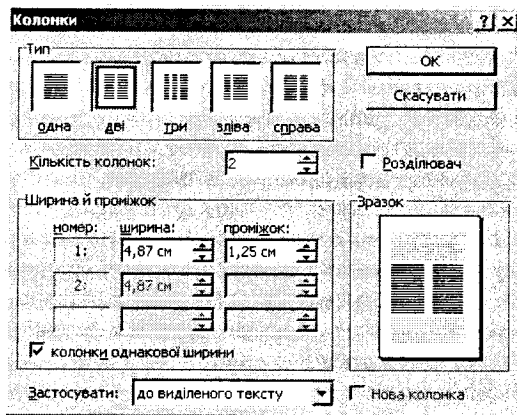


Рис. 6.46

У випадку, якщо ввімкнута опція **колонки однакової ширини**, всі колонки будуть однаковими. Опція **Розділювач** керує виведенням вертикальної лінії між колонками, а в полі **Застосувати** визначають межі дії заданих параметрів. Вмикання опції **Нова колонка** дає змогу перейти до нової колонки. Під час роботи з текстом зробити це можна за допомогою команди **Вставка/Розрив**, у цьому випадку в діалоговому вікні **Розрив** потрібно вибрати значення **нова колонка**.

Якщо виконана команда стосувалась виділеного фрагмента тексту, то він оформляється в окремий розділ, інакше колонки буде створено для всього тексту.

□ *Швидко розпочати нову колонку (вставити розрив колонки) можна комбінацією клавіш **Ctrl+Shift+Enter**.*

□ *Розміри колонок зручно змінювати за допомогою горизонтальної лінійки форматування. Незайве ще раз нагадати, що під час верстання документа обидві лінійки бажано тримати ввімкнутими.*

□ *Якщо фрагмент тексту оформлено кількома колонками, а на екрані й далі видно одну, тільки вузьку, це означає, що потрібно змінити режим перегляду документа зі **Звичайного** на **Розмітку сторінки** – тільки там можна побачити справжній вигляд сторінки.*

6.11.4. Нумерація сторінок і колонтитули

Для нумерування сторінок використовують команду меню **Вставка/Номери сторінок**, яка активізує однойменне вікно. Значення у полі **Розташування** визначає місцезнаходження номера сторінки:

- **Внизу сторінки** – у нижньому колонтитулі;
- **Вгорі сторінки** – у верхньому.

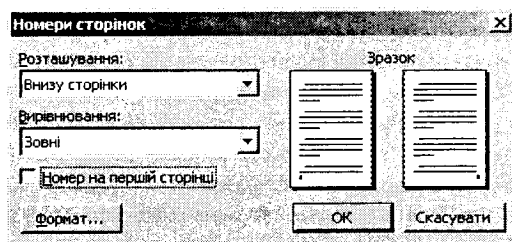


Рис. 6.47

Значення поля **Вирівнювання** задає положення номерів щодо бокових полів сторінки: **Зліва**, **Від центру**, **Справа**, **Усередині**, **Зовні**. Щоб вивести номер на першій сторінці, необхідно ввімкнути опцію **Номер на першій сторінці**. Якщо потрібно змінити стандартний формат нумерації, то слід скористатися кнопкою **Формат**.

Колонтитули – це окремі ділянки документа, куди не можна потрапити простим перенесенням курсора миші. Редагують їх за допомогою команди **Вигляд/Колонтитули**. У цьому випадку відкривається панель інструментів **Колонтитули**, а курсор вводу переміщується в колонтитул активної сторінки.

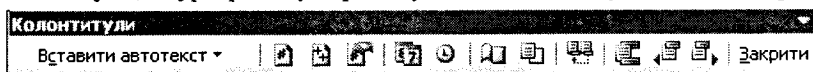


Рис. 6.48

У колонтитул можна вводити і там редагувати текст, ілюстрації, таблиці, а також вставляти номер сторінки, кількість сторінок, дату і час за допомогою піктограм **Вставити поле номера сторінки**, **Вставити кількість сторінок**, **Вставити поле дати** і **Вставити поле часу**. Потрібно пам'ятати, що кожен розділ має свій набір колонтитулів, переміщення між якими відбувається за допомогою піктограм **Перехід до попереднього** і **Перехід до наступного**, а кількість (до трьох: можуть бути окремі колонтитули для першої, парних та непарних сторінок розділу) визначають на вкладці **Макет вікна** **Параметри сторінки** (див. п. 6.11.1). Для перемикання між верхнім та нижнім використовують піктограму **Верхній/нижній колонтитул**, а натиснута піктограма **Як у попередньому** дає змогу використати колонтитул попереднього розділу. Вийти з режиму редагування колонтитула можна, натиснувши на кнопку **Закрити** панелі.

❑ *Якщо потрібно створити складний колонтитул з текстом та графікою, то не варто вставляти номери сторінок командою **Вставка/Номери сторінок** – це може спричинити певні незручності під час узгодження елементів колонтитула, оскільки номер вставляється не безпосередньо у текст, а у текстову рамку. У такому випадку краще використовувати піктограму **Вставити поле номера сторінки** панелі інструментів **Колонтитули**, а для вирівнювання номера застосовувати табуляцію.*

□ У разі наявності непорожнього колоннитула можна швидко перейти до його редагування, двічі натиснувши на ньому мишею, і так само швидко вийти, двічі натиснувши на тексті документа.

□ Чи може у документі після 25-ї сторінки йти 3-тя? Так! Якщо є потреба змінити природний хід нумерації сторінок, необхідно у бажаному місці оформити новий розділ, кнопкою **Формат** діалогового вікна **Номери сторінок** відкрити вікно **Формат номера сторінки** і вибрати значення **почати з опції Нумерація сторінок**, задавши у полі праворуч початковий номер.

□ Якщо після тривалого оформлення колоннитулів кількох розділів несподівано виявилось, що з них усіх залишився тільки останній, залишається лише змиритися та усвідомити, що це наслідок дії натиснутої піктограми **Як у попередньому панелі інструментів Колоннитули**. Щоб такого не трапилось наступного разу, слід для кожного окремого колоннитула відтискати названу піктограму – тоді колоннитули різних розділів будуть незалежними один від одного.

6.12. Робота зі стилями

У більшості документів можна виділити стандартні складові: заголовки, основний текст, підписи тощо, які мають свої особливі параметри форматування. Стиль – це набір параметрів форматування, пов'язаний із конкретним іменем і повністю описує зовнішній вигляд абзацу (для стилів абзацу), сукупності символів (для стилів знака) або таблиці (для стилів таблиці). Набір стилів документа користувач може модифікувати: додавати і вилучати, змінювати, копіювати в шаблони документів.

6.12.1. Вбудовані стилі документа

До таблиці стилів кожного документа належить ціла низка вбудованих стилів. Це, зокрема:

- **Звичайний** – стандартний стиль для основного тексту;
- **Верхній колоннитул**, **Нижній колоннитул** – стилі верхнього та нижнього колоннитулів;
- **Маркірований список**, **Нумерований список** – стилі відповідних списків;
- **Заголовок 1 ... Заголовок 9** – стилі для заголовків відповідного рівня;
- **Зміст 1 ... Зміст 9** – стилі для оформлення абзаців змісту документа;
- **Сітка таблиці** – стандартний стиль для таблиць;
- **Гіперпосилання** – стиль для гіперпосилань.

Назви стилів символів мають позначку **a**, назви стилів абзаців – позначку ¶, а таблиць – спеціальну ґратку з чотирьох клітинок. Деякі вбудовані стилі не можна вилучити, але їх усі можна модифікувати.

Використати стилі для оформлення документа можна такими способами:

- пов'язати абзац зі стилем під час введення;
- застосувати стиль до виділеного фрагмента;
- скористатись автоформатуванням.

Під час набору тексту у Word абзацам за замовчанням присвоєно стиль **Звичайний**. Щоб пов'язати абзаци або символи з іншим стилем, необхідно їх виділити, виконати команду **Формат/Стилі та форматування** і в однойменній області завдань, яка відкривається, у списку **Виберіть форматування для застосування** вибрати потрібний стиль. Вигляд фрагмента відразу зміниться відповідно до параметрів застосованого стилю, а його ім'я відображається у верхньому полі ділянки **Форматування виділеного тексту**. Переглянути параметри форматування виділеного фрагмента можна в області завдань **Показати форматування**, яку відкриває однойменна команда меню **Формат**.

У згаданому списку **Виберіть форматування для застосування** можуть бути відображені не всі наявні у документі стилі – це залежить від значення поля **Показати**:

- **Наявні стилі** – відображаються вбудовані стилі, застосовані в документі, або створені чи модифіковані користувачем;
- **Використовувані** – відображаються стилі, застосовані в документі;
- **Усі** – відображаються всі стилі документа.

Вигляд списку залежить також від опції **вести облік форматування** вкладки **Правка** діалогового вікна **Параметри**. Якщо вона увімкнута, то у списку відображаються не лише самі стилі, а й їхні видозмінені варіанти (наприклад, стиль **Звичайний** + напівжирне накреслення, яке не задане в самому стилі), а у полі **Показати** з'являється додаткове значення **Доступне** – відображаються всі доступні стилі та їхні варіанти. Стає доступною також кнопка **Виділити все**, яка для виділеного фрагмента дає змогу знайти та виділити всі інші з тим самим форматуванням. Після вимкнення опції **вести облік форматування** у списку відображатимуться лише самі стилі, а у полі **Показати** з'явиться додаткове значення **Спеціальне**, після вибору якого відкривається діалогове вікно **Настройки формату**, де можна додатково вказати, які стилі слід відображати у списку, а які – приховати.

Стиль можна застосувати швидше за допомогою спеціального поля **Стиль**, яке міститься ліворуч на панелі інструментів **Форматування**. Там же можна побачити й назву стилю поточного абзацу. Для повної ж індикації стилів усіх абзаців тексту використовують режим перегляду **Звичайний**, при цьому ширину поля, що відводиться для імені стилю, задають на вкладці **Вигляд** діалогового вікна **Параметри** у полі **ширина смуги стилів**.

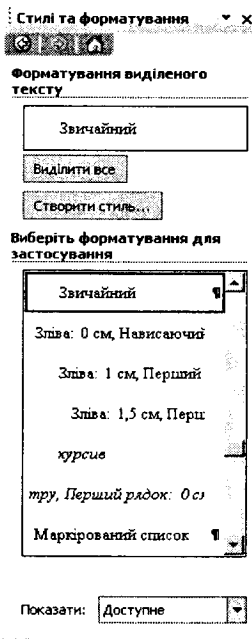


Рис. 6.49

Операцію пов'язування абзаців зі стилями можна виконати й в автоматичному режимі за допомогою команди **Формат/Автоформат**. Вона відкриває діалогове вікно **Автоформат**, де слід вибрати потрібну опцію (**Відразу весь документ** – автоформатування виконується повністю автоматично, **З переглядом кожної зміни** – результати автоформатування можна контролювати) і натиснути на кнопку **ОК**. Після цього програма сама проаналізує абзаци тексту і оформить їх відповідними стилями. Однак у цьому випадку вигляд документа може виявитися далеким від бажаного.

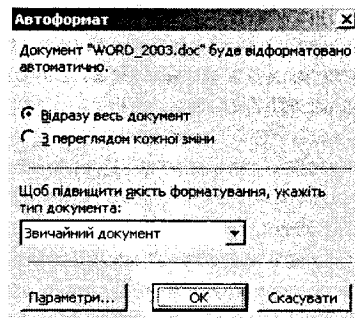


Рис. 6.50

□ У списку поля **Стиль** панелі інструментів **Форматування** для зручності виводяться лише ті стилі, які вже застосовували, створювали чи модифікували. Тому, якщо там немає потрібного стилю, слід повернутися в документ, натиснути на клавішу **Shift** і знову відкрити список.

□ Швидко відкрити область завдань **Стилі та форматування** можна за допомогою піктограми **Панель форматування** панелі інструментів **Форматування**.

□ Для швидкого присвоєння поточному або виділеному абзацу стилю **Звичайний** потрібно натиснути на комбінацію клавіш **Ctrl+Shift+N**.

□ Абзаци, до яких застосовано якийсь стиль, дуже легко переформатити іншим стилем за допомогою команди **Правка/Замінити**. Помістивши курсор вводу в поле **Знайти**, потрібно натиснути на кнопку **Формат**, виконати команду **Стиль** і вибрати той стиль, який треба замінити. Аналогічно для поля **Замінити** на вказують стиль, на який слід замінити попередній, після чого натискають на кнопку **Замінити** або **Замінити все**. Зазначимо, що жодного тексту у заданих полях не повинно бути. Подібним чином, взагалі кажучи, можна змінювати будь-які параметри форматування.

6.12.2. Створення та модифікація стилів

Щоб створити новий стиль, необхідно натиснути на кнопку **Створити стиль** області завдань **Стилі та форматування** і у вікні **Створення стилю** задати:

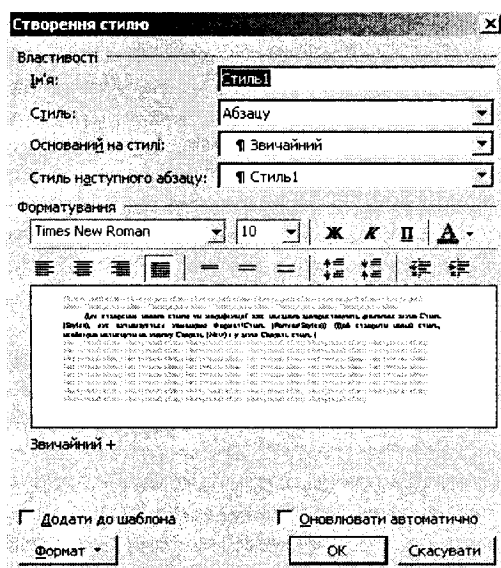


Рис. 6.51

- у полі **Ім'я** – ім'я стилю;
- у полі **Стиль** – тип (**Абзацу** для абзацу, **Знака** для символу, **Таблиці** для таблиці);
- у полі **Оснований на стилі** – ім'я базового стилю, з якого буде взято початкові параметри форматування;
- у полі **Стиль наступного абзацу** – стиль, який буде застосовуватись до наступного абзацу під час введення тексту (тільки для стилів абзацу).

Основні параметри форматування можна змінити просто в діалоговому вікні **Створення стилю** у рамці **Форматування**. Кнопка **Формат** призначена для детальнішого задання параметрів стилю і активізує меню, що містить команди **Шрифт**, **Абзац**, **Табуляція**, **Межа**, **Нумерація**, **Властивості таблиці** тощо. Діалогові вікна, що активізуються після виконання кожної з цих команд (окрім **Сполучення клавіш**, яка відкриває вікно **Настройка клавіатури**) нічим не відрізняються від вікон, відкритих відповідними командами меню **Формат** і **Таблиця**.

Словесний опис заданих параметрів виводиться під демонстраційною ділянкою вікна **Створення стилю**, де подається зовнішній вигляд фрагмента, оформленого цим стилем. Якщо увімкнути опцію **Оновлювати автоматично**, то зміна параметрів форматування фрагмента автоматично змінить параметри відповідного стилю і негайно відобразиться у тексті. Опцію **Додати до шаблону** використовують для додання стилю до шаблону документа. Натискання на кнопку **ОК** додає стиль до набору стилів документа.

Слід також зауважити, що застосування стилю зовсім не означає, що оформлені ним абзаци, символи чи таблиці не можна додатково формувати за допомогою команд меню **Формат**, проте це вже буде саме додаткове форматування.

Подібним чином стилі й модифікують. Для цього в області завдань **Стилі та форматування** слід навести курсор на потрібний стиль, натиснути на піктограму розкриття списку, яка з'явиться праворуч (або просто натиснути на праву клавішу миші), і виконати команду **Змінити**. Відкриється діалогове вікно **Зміна стилю**, аналогічне до вже описаного вікна **Створення стилю**. Після завершення модифікації всі абзаци цього стилю набудуть вигляду, що відповідає зробленим змінам.

Команда **Видалити** локального меню стилю призначена для його вилучення, а команда **Оновити відповідно до виділеного фрагмента** дає змогу надати стилю параметри виділеного фрагмента (або просто поточного абзацу).

□ *Якщо заголовки у документі оформлені відповідними стилями, то, як було сказано раніше, можна користуватись дуже зручним режимом **Схема документа**. У ньому, наприклад, щоб перейти до будь-якого пункту чи розділу документа, достатньо просто натиснути мишею на назві відповідного заголовка. Правда, слід зазначити, що присутність абзацу-*

заголовка відповідного рівня у схемі визначається не застосованим до нього стилем, а значенням **Рівень 1**, ..., **Рівень 9** атрибуту **Рівень**, який задають на вкладці **Відступи та інтервали** діалогового вікна **Абзац**, – просто у стилях заголовків це передбачено заздалегідь.

□ Щоб швидко зняти додаткові параметри форматування абзаців з виділених фрагментів тексту (де можуть бути використані й різні стилі), слід натиснути на комбінацію клавіш **Ctrl+Q**. Для зняття ж додаткових параметрів форматування символів використовують комбінацію **Ctrl+Shift+Z**.

□ Оновити параметри стилю відповідно до змін у фрагменті або поточному абзаці можна також, вибравши стиль із тим самим іменем у полі **Стиль** панелі інструментів **Форматування**. При цьому відкривається діалогове вікно, яке дає змогу не лише виконати зазначену процедуру, а й повернути зміненому фрагменту вигляд, визначений параметрами стилю.

□ Є дуже ефектний засіб оформлення колоннитулів: вставлення для кожної сторінки назв поточного розділу або пункту, якщо вони оформлені окремими стилями. Для цього слід поставити курсор у потрібне місце колоннитула, виконати команду **Вставка/Поле** і в діалоговому вікні, що відкриється, у полі **Категорії** вибрати значення **Зв'язки й посилання**, у списку **Поля** – ім'я **StyleRef**, у списку **Ім'я стилю** – стиль, яким оформлений потрібний заголовок, у рамці **Параметри поля** увімкнути потрібні опції і натиснути на кнопку **ОК**. Цим самим у колоннитул буде вставлено обчислюване поле, яке для кожної сторінки заповнюватиметься назвою поточного розділу або пункту.

□ Як вилучити зі схеми документа абзаци, що не є заголовками? Треба для них у полі **Рівень** вкладки **Відступи та інтервали** діалогового вікна **Абзац** задати значення **Основний текст**. А що робити, якщо вигляд самої схеми виявиться неприсмним чи й узагалі нечитабельним, і його потрібно змінити? Вихід надзвичайно простий, але нетривіальний: треба модифікувати стиль **Схема документа**.

6.13. Шаблони

6.13.1. Шаблон документа

Часто доводиться працювати з документами, які мають однакове оформлення і навіть деякі елементи тексту. Цю стандартну інформацію можна помістити у так званий шаблон. Кожен документ базується на якомусь шаблоні, де зберігаються набір стилів, макрокоманди, список елементів **АвтоТекст**, налаштування клавіатури, панелей інструментів та меню, а також стандартний текст. Під час створення документа у нього копіюються текст і стилі із базового шаблону.

Крім того, кожному документу доступні елементи так званих глобальних (загальних) шаблонів. Стандартно є один глобальний шаблон – **Normal.dot (Новий документ)**, однак глобальним можна зробити будь-який шаблон. Для цього треба скористатись командою **Сервіс/Шаблони та надбудови**, яка відкриє однойменне діалогове вікно.

Щоб додати шаблон до списку глобальних, треба у рамці **Загальні шаблони та надбудови** вкладки **Шаблони** натиснути на кнопку **Додати** і вибрати потрібний файл шаблону. Однак для того, щоб скористатись ним як глобальним, треба ще увімкнути опцію-перемикач ліворуч від його імені у списку. Кнопку **Видалити** використовують для вилучення шаблону зі списку.

В рамці **Шаблон документа** можна змінити базовий шаблон активного документа, натиснувши на кнопку **Приєднати** і вибравши інший шаблон. Опція **Автоматично оновлювати стилі** призначена для модифікації тих стилів документа, які містяться в новому шаблоні з тими самими іменами, відповідно до нових параметрів, причому її необхідно вмикати окремо для кожного документа.

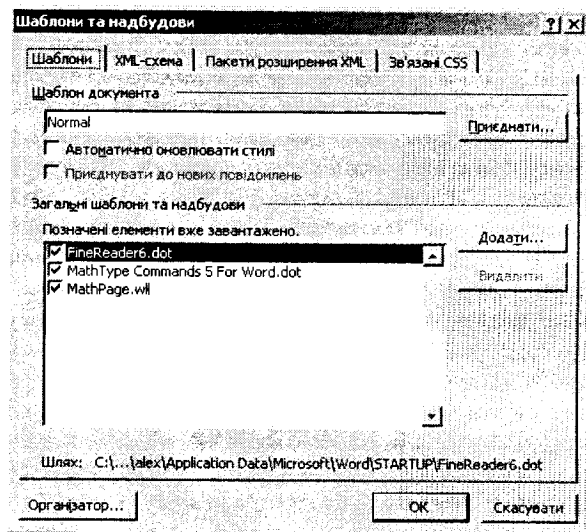


Рис. 6.52

Кнопка **Організатор** призначена для організації обміну елементами між різними шаблонами та документами (вона міститься ще у діалоговому вікні **Макрос**). Після натискання на неї відкривається однойменне діалогове вікно із вкладками **Стилі**, **Автотекст**, **Панелі** та **Макроси** для доступу до потрібних елементів обміну. Користуючись кнопками **Закрити файл** (які потім перетворюються на **Відкрити файл**), у лівій та правій

частинах вікна можна відкривати потрібні шаблони чи документи і за допомогою кнопок **Копіювати**, **Видалити** та **Перейменувати** відповідно копіювати, вилучати або перейменовувати вибрані елементи.

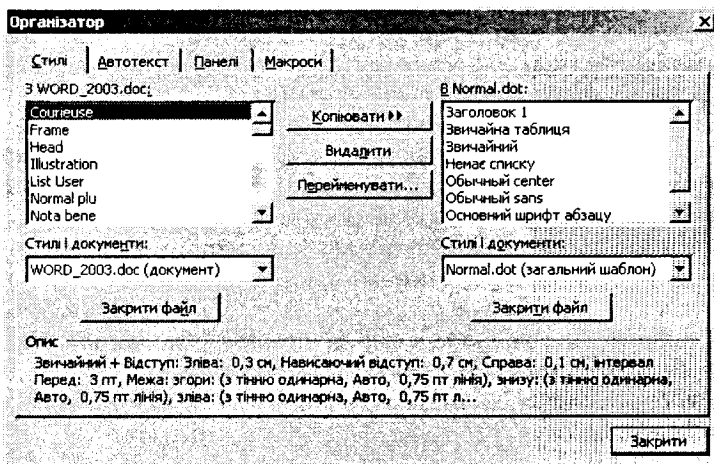


Рис. 6.53

□ Зазвичай із програмою Word 2003 поставляють велику кількість різноманітних готових шаблонів, які можна використати для створення власних документів. Частина цих шаблонів починається зі слова **Майстер** і розрахована на формування документів у процесі діалогу. Необхідно лише відповідати на запитання «майстра», – і потрібний документ буде готовий.

6.13.2. Створення і модифікація шаблонів

Для створення нового шаблону треба виконати команду меню **Файл/Створити** і в області завдань **Створити документ**, яка відкриється, вибрати команду **На носому комп'ютері**. Це активізує діалогове вікно **Шаблони**, де потрібно вибрати шаблон, на якому базуватиметься створюваний, задати для опції **Створити** значення **Шаблон** і натиснути на кнопку **ОК**.

Іншим способом є створення шаблону на базі наявного документа:

- відкрити документ і внести необхідні зміни;
 - виконати команду **Файл/Зберегти як**;
 - у полі **Тип файлу** задати значення **Шаблон документа**;
 - записати з потрібним іменем.
- Зберігаючи шаблон, потрібно мати на увазі, що шаблони повинні бути розміщені не де-небудь, а у спеціально визначеному місці. Тому спочатку потрібно на вкладці **Розташування** діалогового вікна **Параметри** визначити, де розташовано інші шаблони (шлях до них описаний у рядку

шаблони користувача, а якщо він надто довгий, то його можна підглядіти, виділивши цей рядок і натиснувши на кнопку **Змінити**), і записати шаблон саме туди.

Редагують шаблони так само, як і звичайні документи (вводять, вставляють і обробляють текст, модифікують набір стилів), а також налаштовують параметри меню, панелей інструментів та клавіатури, записують чи створюють макрокоманди, після чого шаблон зберігають.

Так само виконують і модифікацію вже створених шаблонів. Крім того, під час роботи з документами може відбуватися й неявна модифікація базового або глобального шаблонів, якщо користувач працював з діалоговими вікнами, що містять функції роботи з шаблонами (наприклад, під час створення елементів автотексту або комбінацій клавіш, після натискання на кнопку **За замовчуванням** тощо).

□ *Якщо не завжди потрібно, щоб внесені до загального шаблону **Normal** зміни були збережені, слід на вкладці **Збереження діалогового вікна Параметри** увімкнути опцію **запит на збереження шаблону Normal.dot**, яка змусить програму перед завершенням сеансу роботи запитати на це дозвіл.*

6.14. Робота з ілюстраціями та ілюстративними засобами

6.14.1. Ілюстрації як незалежні зображення

Про вставлення ілюстрацій як елементів тексту та їх форматування вже йшлося раніше. Такий спосіб має свої переваги, однак якщо Word 2003 використовують для верстання складних публікацій, то варто застосувати гнучкіший варіант вставлення – як незалежного зображення. Для цього у списку **вставляти малюнки як вкладки** Правка діалогового вікна **Параметри** слід вибрати значення обтікання **навколо рамки**. Можна вважати, що текст і незалежне зображення перебувають у різних шарах. Проте кожна така ілюстрація прикріплюється до певного (переважно найближчого) абзацу тексту, який позначено спеціальним символом – якорем (якщо увімкнута опція **прив'язка об'єктів рамки Параметри режиму розмітки й веб-документа** на вкладці **Вигляд** діалогового вікна **Параметри**).

Нагадаємо, що для вставлення у документ ілюстрації з файлу потрібно виконати команду меню **Вставка/Малюнок/З файлу**, у діалоговому вікні **Додавання малюнка** вибрати потрібний файл і натиснути на кнопку **Вставити**. Заданий (у діалоговому вікні **Параметри**) режим обтікання текстом можна змінити на вкладці **Розташування** діалогового вікна **Формат малюнка** у рамці **Обтікання**: значення у **тексті** робить ілюстрацію елементом тексту, а **за текстом** і **перед текстом** відмінюють обтікання. У рамці **Горизонтальне вирівнювання** задають розташування

ілюстрації по горизонталі. Відстань до тексту з різних боків від малюнка задають на вкладці **Обтікання текстом** діалогового вікна **Додаткова розмітка**, яке відкривається після натискання на кнопку **Додатково**.

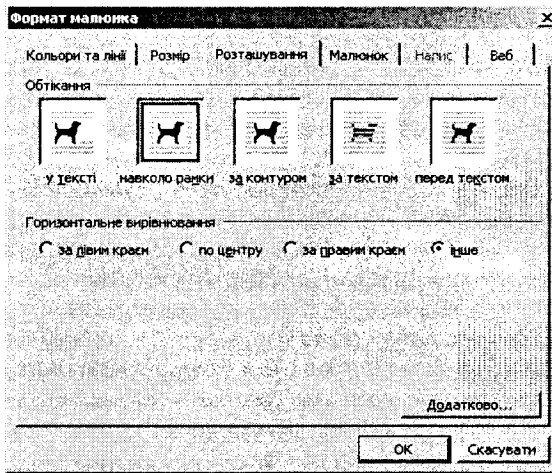


Рис. 6.54

Ілюстрацію – незалежне зображення можна переміщувати в будь-яке місце документа, просто захопивши мишею. Для її точного позиціювання використовують вкладку **Розташування малюнка** діалогового вікна **Додаткова розмітка**, де:

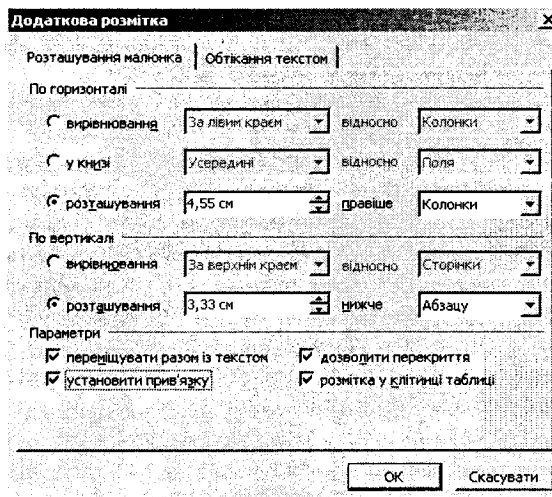


Рис. 6.55

- у рамках **По горизонталі** та **По вертикалі** задають відповідні відстані від ілюстрації до вибраного елемента публікації;
- вмикання опції **переміщувати разом з текстом** задає режим, у якому зображення переміщуватиметься разом з абзацом, до якого воно прикріплене;
- опція **установити прив'язку** забороняє змінювати місце прикріплення ілюстрації;
- опція **дозволити перекривання** дозволяє ілюстраціям «наповзати» одні на одну;
- опція **розмітка у клітинці таблиці** у разі переміщення незалежного зображення у таблицю визначає його як вміст клітинки.

Слід також нагадати, що для зміни розмірів ілюстрації необхідно переміщувати індикатори, що містяться на сторонах позиційної рамки, причому переміщення кутових індикаторів дає змогу зберегти пропорції. А щоб обрізати ілюстрацію або додати до неї вільний простір, потрібно використовувати спеціальний інструмент **Обтинання панелі Налаштування зображення**.

Кілька незалежних зображень можна виділити разом (з утриманням клавіші **Shift**, або зобразивши рамку спеціальним інструментом **Вибір об'єктів** панелі інструментів **Малювання**) і виконувати над ними групові операції, доступні через піктограму **Малювання** панелі інструментів **Малювання**: групувати і розгрупувати (команди **Групувати** і **Розгрупувати**), вирівнювати та рівномірно розподіляти (відповідні підкоманди команди **Вирівняти/розподілити**, причому вмикання режиму **Відносно сторінки** дає змогу робити це відносно сторінки) тощо. Захопивши незалежне зображення за зелений індикатор вгорі, його можна повертати навколо центра, а точно це зроблять за допомогою поля **поворот** вкладки **Розмір** діалогового вікна **Формат малюнка**.

□ *Ілюстрації зазвичай займають набагато більше місця у документі, аніж текст, і тому під час верстання щедро ілюстрованої публікації може настати момент, коли розмір файлу перевищить певну межу і працювати з ним стане дуже важко, а то й зовсім неможливо. У таких випадках ілюстрації слід не безпосередньо вставляти у документ, а давати посилання на них. Для цього у діалоговому вікні **Додавання малюнка** потрібно не просто натиснути на кнопку **Вставити**, а, відкривши пов'язаний з нею список команд, вибрати **Зв'язати з файлом**. Вигляд ілюстрації від цього зовсім не зміниться, але розмір документа значно зменшиться, оскільки у ньому міститиметься лише текст. Правда, переносити файл такого документа в інше місце слід лише разом з файлами ілюстрацій. Для керування ілюстраціями-посиланнями та зміни їхнього статусу використовують команду **Правка/Зв'язки**.*

□ *Рамку для обтинання ілюстрації текстом можна зробити фігурною: для цього слід відкрити палітру **Налаштування зображення**, увімкнути*

пiктограмою **Меню “Обтiкання текстом”** спецiальний iнструмент **Змiнити контур обтiкання** i додати вузли до рамки, захоплюючи її сторони i вiдтягуючи вбiк! При цьому значення опцiї **Обтiкання** змiниться на **за контуром**. Повернутись до нормального обтiкання навколо рамки можна, повернувши опцiї значення **навколо рамки**.

□ Щоб iлюстрацiя-пiдкладка не виглядала надто яскравою та насиченою, треба її видiлити, вiдкрити палiтру **Настройка зображення**, натиснути на пiктограму **Меню “Зображення”** i вибрати значення **Пiдкладка**.

□ Для перемiщення незалежного зображення мишею лише у вертикальному або горизонтальному напрямку потрiбно утримувати натиснутою клавішу **Shift**. Для цього ж можна використовувати клавіші-стрiлки клавіатури, попередньо видiливши iлюстрацiю.

□ Якщо стовiдсотково вставленi як незалежнi зображення iлюстрацiї несподiвано зникли, то можливо було просто задано режим перегляду документа **Звичайний**. Тому насамперед потрiбно спробувати змiнити його на **Розмiтку сторiнки** – тiльки там можна побачити незалежнi зображення.

□ Якщо пiд час перемiщення незалежного зображення мишею воно весь час рухається «стрибками» i «проскакує» потрiбне мiсце, то це просто означає, що запроваджено режим прив'язання об'єктiв до сiтки з надто великими розмiрами клiтинок. У такому разi слiд натиснути на пiктограму **Малювання панелi Малювання**, виконати команду **Сiтка** локального меню i задати у полях **По горизонталi та По вертикалi** меншi величини для кроку змiщення або просто вимкнути опцiю **Прив'язати до сiтки**.

6.14.2. Текстові рамки

Текстові рамки у Word 2003 попри свою невдалу назву (**Напис**) надають користувачу зручнi додатковi можливостi для роботи з текстом: розмiщувати текст в будь-якому мiсцi сторiнки, у тому числi на полях, створювати ефект обтiкання тексту текстом, змiнювати напрямок тексту в рамцi, зв'язувати його з iншими рамками з утворенням єдиного текстового потоку тощо. Подiбнi засоби властивi спецiалiзованим видавничим системам, як наприклад, PageMaker, InDesign чи QuarkXPress.

Щоб створити порожню текстову рамку, потрiбно виконати команду **Вставка/Напис** (або натиснути на пiктограму **Напис** панелi **Малювання**) i курсором, який набуде форми хрестика, задати розмiри рамки. Пiсля вiдпускання лiвої клавіші миші курсор вводу перемiститься всередину рамки, де можна вводити, редагувати i формувати текст так само, як i на сторiнцi документа. Якщо ж попередньо видiлити фрагмент тексту, то згадана команда (а також команда **Формат/Додати Напис**) помiстить його у текстову рамку. Роль текстової рамки може вiдiгравати також виноска, для створення якої слiд натиснути на пiктограму **Автофiгури**,

виконати команду **Виноски** локального меню, вибрати потрібний тип фігури і задати її розміри на сторінці.

Якщо на вкладці **Загальні** діалогового вікна **Параметри** увімкнено опцію **автоматично створювати полотно при вставці автофігур**, після вмикання інструмента формування текстової рамки (чи довільної автофігури) буде створено так зване полотно, яке дає змогу об'єднувати всі розміщені на ньому фігури для виконання деяких групових операцій (переміщення, копіювання, вилучення тощо). Коли така можливість не потрібна, опцію слід вимкнути.

Текстові рамки поводять себе подібно до незалежних зображень: їх можна переміщувати, змінювати розміри, обтїкати текстом, групувати, поміщати перед та позад тексту тощо. Для задання детальних параметрів використовують діалогове вікно **Формат напису**, яке активізується командою **Формат/Напис**. У ньому, крім вже описаних вкладок, стає доступною ще вкладка **Напис**, де:

- у полях групи **Внутрішні поля** задають відстані між текстом і краями рамки;
- опція **переносити текст в автофігурі по словах** дає змогу використовувати рамки довільної ширини (у разі її вимикання ширина рамки стане рівною ширині тексту на сторінці);
- опція **припасовувати розмір автофігури до тексту** автоматично збільшує висоту текстової рамки залежно від вмісту;
- кнопку **Формат виноски** використовують для визначення параметрів, властивих лише виноскам;
- кнопка **Перетворити на рамку** дає змогу перетворити текстову рамку на рамку старого формату (фрейм).

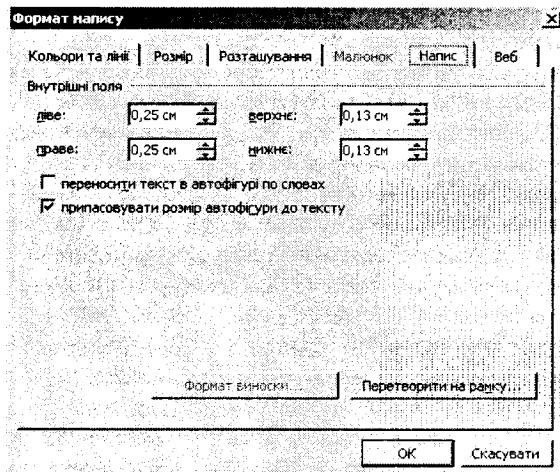


Рис. 6.56

На вкладці **Кольори та лінії** можна задати або змінити кольори заповнення та облямування текстової рамки. Як і в клітинках таблиці, у текстових рамках можна змінювати напрямок тексту: для цього слід виділити відповідну рамку і виконати команду **Формат/Напрямок тексту**, де у діалоговому вікні вибрати потрібний варіант, або скористатись одноіменною піктограмою панелі **Напис**.

Виділену текстову рамку можна зв'язати з іншою, порожньою, рамкою за допомогою піктограми **Створити зв'язок із наступним написом** панелі **Напис** або однойменної команди локального меню, вказавши курсором у вигляді кухля на потрібну рамку. Тепер під час редагування тексту, зміни розмірів чи навіть вилучення текстових рамок текст «перетікатиме» між пов'язаними рамками так, щоб повністю заповнювати перші у цьому «ланцюжку». Для розірвання зв'язку слід використати піктограму **Розірвати зв'язок із наступним написом**.

□ *Швидко вилучити полотно можна, якщо перед рисуванням рамки натиснути на клавішу **Esc**.*

□ *Для швидкого відкриття вікна **Формат напису** досить двічі натиснути мишею де-небудь на штрихованій облямівці рамки.*

□ *Текстова рамка – незамінний засіб для створення підписів до незалежних зображень. Тільки для уникнення казусів слід групувати об'єкт і підпис до нього.*

□ *Щоб використовувати текстові рамки для надписів на малюнках, треба зняти у них заповнення (**Немає заливки**) та облямування (**Немає ліній**) у відповідних полях вкладки **Кольори та лінії** вікна **Формат напису** (або через аналогічні піктограми панелі інструментів **Малювання**).*

□ *Щоб трохи пожвавити вигляд виділеної текстової рамки, можна натиснути на піктограму **Малювання** панелі **Малювання**, виконати команду **Змінити автофігуру** локального меню і вибрати що-небудь оригінальніше.*

□ *Як вилучити всю текстову рамку, якщо клавіша **Delete** вилучає символи тексту всередині рамки? Слід натиснути мишею не там, де розміщено текст, а на штрихованій облямівці рамки.*

□ *Якщо піктограма зв'язування текстових рамок натиснута, а зв'язувати не потрібно або немає з чим, слід натиснути на клавішу **Esc**, інакше можна одержати від **Word** серію повідомлень про помилку.*

6.14.3. Ілюстративні засоби програми Word

Для створення ілюстративних об'єктів на кшталт ліній, прямокутників, овалів, багатокутників та різноманітного вигляду заготовок (автофігур) Word 2003 надає спеціальні засоби, якими найзручніше користуватись через піктограми панелі інструментів **Малювання** (один із засобів її вмикання – натиснути на однойменну піктограму панелі **Стандартна**).

Хоча у Word 2003, на відміну від попередніх версій програми, є і стандартний спосіб доступу до цих інструментів – через команду меню **Вставка/Малюнок/Автофігури**. Слід також зауважити, що створювати автофігури можна лише у режимах **Розмітка сторінки** та **Веб-документ**.

Щоб намалювати фігуру одного з чотирьох найпростіших типів, треба натиснути на відповідну піктограму – **Лінія**, **Стрілка**, **Прямокутник**, **Овал**, помістити хрестоподібний курсор у потрібне місце і, натиснувши на ліву клавішу миші, перемістити його так, щоб фігура набула бажаних розмірів (для формування квадрата, кола або лінії під кутами, кратними 15 градусам, слід утримувати натиснутою клавішу **Shift**; це ж саме застосовують для пропорційної зміни розмірів будь-яких фігур). Після завершення рисування фігури відповідний інструмент вимикається, і програма переходить у звичайний режим роботи. Щоб використати інструмент кілька разів поспіль, слід увімкнути його подвійним натисканням.

Для створення фігури довільної форми потрібно натиснути на піктограму **Автофігури**, виконати команду **Лінії**, вибрати інструмент **Крива**, **Полілінія** або **Мальована крива** і нарисувати фігуру (у перших двох випадках цей процес завершується подвійним натисканням мишею або натисканням на клавішу **Esc**). Незамкнуту криву можна замкнути командою **Замкнути криву** її локального меню, однак незалежно від того, зроблено це чи ні, така фігура може мати внутрішнє заповнення. Щоб редагувати створену фігуру, слід виконати команду **Почати зміну вузлів** її локального меню (або меню, що відкривається після натискання на піктограму **Малювання**) і переміщувати сторони або вузли, причому локальні меню сторін або вузлів відкривають додаткові можливості для їх редагування.

Однак дуже часто можна обійтися використанням вже готових фігур, інструменти для створення яких можна вибрати через меню, що відкривається після натискання на піктограму **Автофігури**. Їх форму не можна редагувати, однак можна змінити на іншу фігуру (для цього, як було вже сказано, треба натиснути на піктограму **Малювання** панелі інструментів **Малювання** і виконати команду локального меню **Змінити автофігуру**). Автофігури можна використовувати у ролі текстових рамок, виконавши команду **Додати текст** їхнього локального меню.

Детальні параметри фігур задають у діалоговому вікні **Формат автофігури**, аналогічному до вікон форматування усіх інших об'єктів. Основні параметри (колір і стиль заповнення та ліній, тип стрілок, кут повороту) можна швидше та наочніше змінити за допомогою відповідних піктограм панелі **Малювання**. Крім цього, за допомогою її засобів до фігур можна додати тінь (піктограма **Стиль тіні**), а також надати їм об'ємного вигляду (піктограма **Об'єм**). Параметри фігури можна зробити стандартними для всіх новостворюваних, виконавши команду **За замовчуванням для автофігур** її локального меню або меню, пов'язаного з піктограмою **Малювання**.

□ Швидко активізувати вікно для форматування фігури можна подвійним натисканням на ній мишею.

□ Якщо потрібно виділити одночасно багато фігур, слід користуватись інструментом, який вмикається піктограмою **Вибір об'єктів**, – ним достатньо просто зобразити рамку потрібних розмірів.

□ Створивши за допомогою автофігур і текстових рамок складну діаграму, не варто передчасно радіти: вже перша спроба змінити її розміри викличе розчарування: самі фігури змінилися, а текст – ні! Щоб уникнути такої ситуації, треба вирізати всю групу фігур у буфер, виконати команду **Правка/Спеціальна вставка** і вибрати у списку форматів вставлення значення, наприклад, **Малюнок (метафайл Windows)**. Таким чином об'єднання фігур перетвориться на звичайну ілюстрацію.

6.14.4. Текстові спецефекти

Для створення ефектів фігурного тексту у Word є спеціальний засіб, який називається WordArt. Щоб вставити в документ об'єкт WordArt, потрібно виконати команду **Вставка/Малюнок/Об'єкт WordArt** або натиснути на піктограму **Додати об'єкт WordArt** панелі **Малювання**. Після цього відкриється діалогове вікно **Колекція WordArt**, де потрібно вибрати стиль оформлення об'єкта, потім – вікно **Зміна тексту WordArt**, у якому слід увести текст і натиснути на кнопку **ОК**. Текстовий об'єкт буде вставлено в документ.

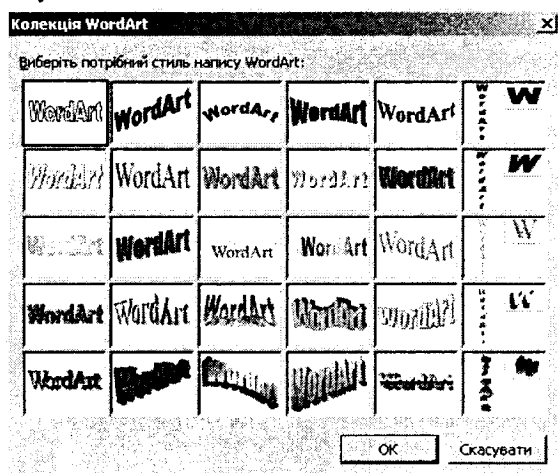


Рис. 6.57

Для редагування об'єкта WordArt використовують піктограми спеціальної панелі **WordArt**. Зокрема, піктограма **Змінити текст** дає змогу змінити текст об'єкта, піктограма **Колекція WordArt** – його стиль, а

піктограма **Меню “Текст – Фігура”** – застосований ефект. Крім цього, форму текстового об'єкта можна модифікувати шляхом переміщення індикатора у вигляді жовтого ромбика. Піктограми **Тінь** та **Об'єм** панелі **Малювання** використовують для зміни параметрів, пов'язаних із тінню та об'ємним виглядом об'єктів WordArt.

□ Для того, щоб швидко змінити текст об'єкта WordArt, потрібно двічі натиснути на ньому мишею.

6.15. Виноски

Слово «**виноска**» у Word 2003 вживають також у двох значеннях: як **текстову рамку** з лінією прив'язання до об'єкта і як **примітку** до тексту, розміщену внизу сторінки або в кінці документа. В цьому пункті вживатимемо згадане слово в його другому значенні.

Отож, для вставлення виноски як примітки до слова чи фрагмента документа використовують команду меню **Вставка/Посилання/Виноска**, яка відкриває діалогове вікно **Виноски**.

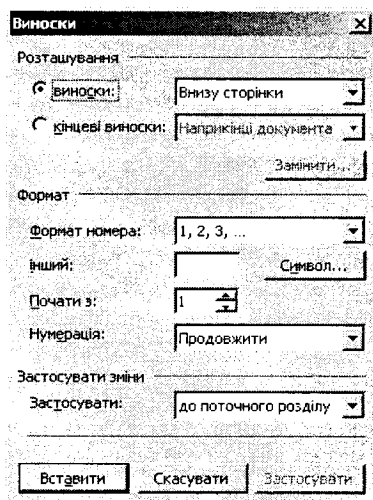


Рис. 6.58

У вікні треба задати потрібні значення опцій та полів: у **рамці Розташування**

- опція **виноски** – текст виноски розміщуватиметься на тій же сторінці в її нижній частині (значення поля ліворуч **Внизу сторінки**) або відразу після тексту, якщо сторінка не заповнена (значення **Внизу тексту**);
- опція **кінцеві виноски** – текст буде розміщено в кінці розділу або документа залежно від значення у полі ліворуч;

у рамці **Формат**

- список **Формат номера** – використовують вибрану послідовну автоматичну нумерацію виносок;
- поле **інший** – використовують заданий символ (його можна ввести з клавіатури або вставити за допомогою кнопки **Символ**);
- поле **Почати з** – задають номер першої виноски;
- поле **Нумерація** задає спосіб автоматичної нумерації виносок: **продовжити** – неперервно в усьому документі; **в кожному розділі** – окремо для кожного розділу; **на кожній сторінці** – окремо для кожної сторінки.

Поле **Застосувати** визначає область застосування заданих параметрів виноска. Кнопка **Вставити** дає змогу вставити виноску, а кнопка **Застосувати** – змінити параметри наявних виноска із врахуванням зони застосування.

Після закриття вікна **Виноски** на місці, де перебував курсор вводу, з'явиться символ виноска, а сам курсор переміститься туди, де буде розташовано її текст. Після вводу тексту виноска у режимі перегляду **Звичайний** слід натиснути на кнопку **Закрити**, а в режимі **Розмітка сторінки** – просто перемістити курсор у потрібне місце. Програма сама стежить за тим, щоб символи і тексти виноска завжди містились на одній сторінці, а в разі використання автоматичної нумерації виноска після додання чи вилучення автоматично перенумерує їх.

Редагувати текст виноска у режимі **Розмітка сторінки** можна безпосередньо на сторінках документа, однак найкраще робити це у режимі **Звичайний** після виконання команди меню **Вигляд/Виноски**, коли одночасно видно тексти усіх виноска. Для вилучення виноска з документа достатньо вилучити її символ з основного тексту, однак це стає можливим тільки після виділення символу.

□ *Швидко вставити виноску можна, натиснувши на комбінацію клавіш **Ctrl+Alt+F**.*

□ *Для швидкого переходу до редагування виноска потрібно двічі натиснути мишею на її символі.*

□ *За допомогою команди **Вставка/Посилання/Перехресне посилання** можна вставляти різноманітні посилання, зокрема, на елементи нумерованих списків (наприклад, літературні джерела). У разі перенумерування списку відповідні посилання можна підправити, виділивши фрагмент тексту, що містить їх, і натиснувши **F9** для оновлення полів.*

6.16. Зміст документа

Зміст документа – це список назв розділів та пунктів, розташованих у тому ж порядку, що й у документі, з можливістю зазначення номерів сторінок. Як заголовки у тексті документа слід використовувати абзаци, оформлені стилями **Заголовок 1 ... Заголовок 9**, причому в самому змісті їм відповідатимуть стилі **Зміст 1 ... Зміст 9**.

Для створення змісту використовують вкладку **Зміст** діалогового вікна **Зміст і покажчики**, яке активізується командою **Вставка/Посилання/Зміст і покажчики**.

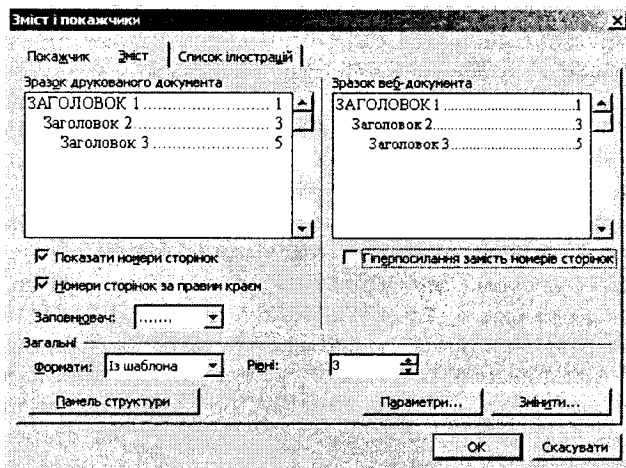


Рис. 6.59

У цьому вікні:

- варіант форматування змісту вибирають у списку **Формати**, спостерігаючи результат у ділянці **Зразок друкованого документа**;
- опція **Показати номери сторінок** дає змогу включати у зміст номери сторінок, а опція **Номери сторінок за правим краєм** – вирівнювати їх до правого краю;
- у списку **Заповнювач** вибирають символ, яким буде заповнено простір між назвою заголовка і номером сторінки;
- кількість використовуваних рівнів заголовків задають у полі **Рівні**;
- кнопка **Параметри** дає змогу змінити спосіб формування змісту;
- кнопку **Змінити** використовують для модифікації стилів змісту;
- кнопка **Панель структури** відкриває панель інструментів **Структура**.

Після натискання на кнопку **ОК** зміст буде вставлено в документ у місці розташування курсора вводу. Для оновлення змісту потрібно знову виконати команду **Вставка/Зміст і покажчики**. Перед самим оновленням з'явиться діалогове вікно із запитом на заміну наявного змісту.

□ Для швидкого оновлення змісту достатньо помістити у нього курсор і натиснути на клавішу **F9**. У діалоговому вікні, що відкривається, можна вибрати один із двох варіантів: оновити весь зміст (значення опції **оновити цілком**) чи тільки номери сторінок (**оновити лише номери сторінок**). Останнє актуальне у тому разі, якщо деякі абзаци зі змісту були вилючені, і повертати їх назад немає потреби.

□ У Word 2003 номери сторінок вставляються у зміст як гіперпосилання. Тому натискання на них мишею (можливо, з одночасним утримуванням клавіші **Ctrl**) дає змогу відразу переміститись на відповідну сторінку.

□ Замість потрібного змісту можна отримати на екрані якийсь незрозумілий і непривабливий рядок, якщо на вкладці **Вигляд** діалогового вікна **Параметри** буде ввімкнута опція **Показувати/коди полів**. Поки ще не освоєні методи роботи з обчислюваними полями, краще тримати її вимкненою.

□ Якщо заголовки оформлено у вигляді списків, то символ табуляції, що автоматично вставляється між номером або позначкою і текстом, може стати джерелом малоприємних ефектів у змісті. Тому бажано оформити такі списки як багаторівневі (навіть коли потрібен лише один рівень) і у полі **Знак після номера** діалогового вікна **Зміна багаторівневого списку** замість символу табуляції вибрати пробіл.

6.17. Макрокоманди

Зазначимо, що досі йшлося лише про стандартні команди програми. Однак кожен, хто постійно працює з Word, рано чи пізно помічає, що йому бракує деяких дуже потрібних функцій – найчастіше просто набору послідовно виконуваних стандартних операцій. У таких випадках чудовим засобом оптимізації роботи з програмою можуть стати макрокоманди, які відтворюють виконання послідовності команд програми Word.

Для створення простих макрокоманд методом їх записування потрібно:

- виконати команду **Сервіс/Макрос/Почати запис**, яка відкриває діалогове вікно **Запис макросу**;

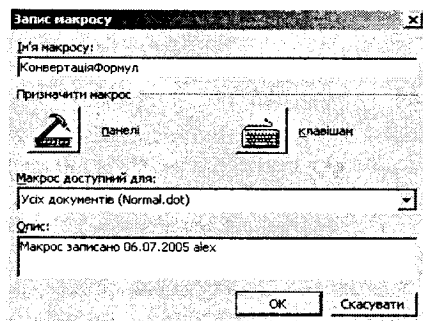


Рис. 6.60

- у полі **Ім'я макросу** цього вікна задати назву макрокоманди;
- у полі **Макрос доступний для** задати місце, де зберігатиметься макрос (шаблон або поточний документ);
- у полі **Опис** увести необов'язковий коментар;

- натиснути на одну з піктограм рамки **Призначити макрос** для пов'язування макрокоманди з командою панелі інструментів (піктограма **Панелі**) чи комбінацією клавіш (піктограма **клавішам**) або на кнопку **ОК**, якщо пов'язування не потрібне.

Після закриття вікна з'являється невеличка панель з піктограмами **Зупинити запис** та **Пауза**, яка свідчить про початок запису макрокоманди. Цей процес полягає у записуванні команд, виконаних користувачем, у спеціальному файлі за допомогою мови **Visual Basic**, і завершується командою **Сервіс/Макрос/Зупинити запис** або натисканням на однойменну піктограму. Запуск макрокоманди просто повторює виконання всіх записаних у ній команд.

Для роботи з макрокомандами використовують діалогове вікно **Макрос**, яке активізується командою **Сервіс/Макрос/Макроси**. Воно дає змогу:

- у полі **Макроси** з задати джерело, звідки братимуться макрокоманди;
- у полі **Ім'я** задати назву макросу (або вибрати його у списку нижче);
- кнопкою **Виконати** запустити вибрану макрокоманду на виконання;
- кнопкою **Налагодження** запустити макрокоманду в режимі відлагоджування;
- кнопкою **Змінити** відкрити текст макросу для редагування;
- кнопкою **Створити** створити нову макрокоманду, попередньо задавши її ім'я у полі **Ім'я**;
- кнопкою **Видалити** вилучити вибрану макрокоманду;
- кнопкою **Організатор** активізувати однойменне вікно для обміну наборами макросів між різними документами та шаблонами.

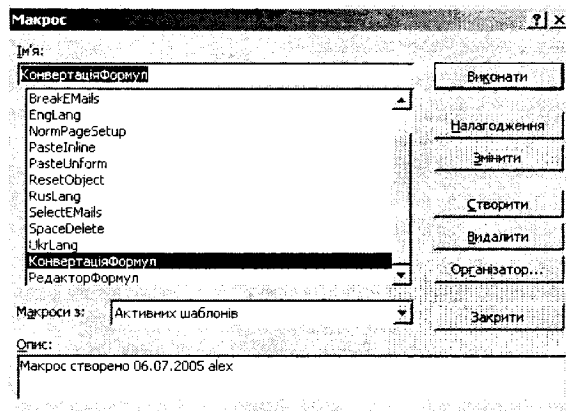


Рис. 6.61

Якщо ж макрос пов'язаний з піктограмою панелі інструментів або комбінацією клавіш, то його можна виконати цілком аналогічно до будь-якої команди Word.

6.18. Друкування документа

Перш ніж роздрукувати підготовлений документ, його можна переглянути на екрані у вигляді, максимально наближеному до друкованого. Це роблять за допомогою команди меню **Файл/Попередній перегляд** або однойменної піктограми панелі **Стандартна**. Вийти з режиму перегляду можна натисканням на кнопку **Закрити** або клавішу **Esc**.

Для надрукування однієї копії всього документа можна використати піктограму **Друк** панелі **Стандартна**. Якщо ж документ потрібно друкувати з іншими параметрами, то виконують команду **Файл/Друк**, яка активізує однойменне діалогове вікно.

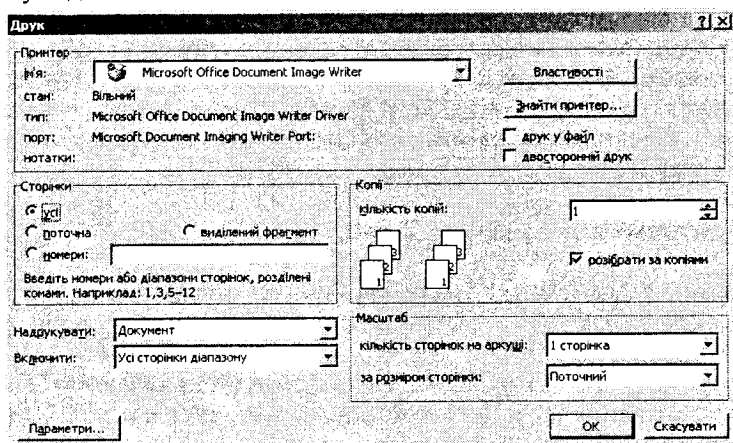


Рис. 6.62

У цьому вікні:

- у полі **Ім'я** рамки **Принтер** вибирають принтер (із наявних у системі), на якому виконуватиметься друкування, причому кнопка **Властивості** дає змогу переглянути та змінити його параметри, а кнопка **Знайти принтер** відкриває діалогове вікно для пошуку інших мережевих принтерів;
- після вмикання опції **друк у файл** образ друкованого документа буде записано у заданий користувачем файл, який потім можна роздрукувати системною командою **copy <ім'я файлу> rpn** або використати для створення pdf-файлу, а опція **двосторонній друк** задає режим друкування документа на обох сторонах аркуша, видаючи запити на вставлення паперу іншим боком;
- у рамці **Сторінки** визначають набір сторінок, які потрібно надрукувати: **усі** – всі сторінки, **поточна** – сторінка, на якій розташовано курсор вводу, **виділений фрагмент** – лише виділена частина, **номери** – набір сторінок або діапазонів, уведений у полі праворуч через кому;

- у полі **Надрукувати** задають значення **Документ**;
- поле **Включити** визначає, які сторінки із заданого у рамці **Сторінки** набору потрібно друкувати: **Усі сторінки діапазону**, **Непарні сторінки** чи **Парні сторінки**;
- у полі **Кількість копій** рамки **Копії** задають потрібну кількість копій документа, а опція **розібрати за копіями** вмикає режим послідовного друкування повних копій документа (інакше по чергово друкуватимуться всі копії окремих сторінок);
- у полі **кількість сторінок на аркуші** рамки **Масштаб** вибирають кількість сторінок, які можна роздрукувати на одному аркуші паперу, а у полі **за розміром сторінки** – формат цього аркуша;
- кнопка **Параметри** відкриває вкладку **Друк** діалогового вікна **Параметри** для зміни параметрів друкування документа.

Командою **Файл/Друк** можна роздрукувати й іншу інформацію, пов'язану з документом (відомості, примітки, список стилів з описом параметрів форматування, елементи автотексту, комбінації клавіш), вибравши у списку **Надрукувати** відповідне значення.

Запитання для самоперевірки

1. Перелічіть усі можливі способи виконання команд Word.
2. Як зберегти документ у потрібному місці з потрібним іменем і потім його відкрити?
3. Як увести в текст документа символи, відсутні на клавіатурі? Як виділити потрібний фрагмент тексту і скопіювати його в інше місце, перенести чи вилучити?
4. Поясніть різницю між форматуванням символів та абзаців.
5. Як здійснити пошук та заміну фрагментів тексту? Чи можна автоматично замінювати параметри форматування? Що таке маска пошуку?
6. Скільки є різних способів створення таблиць і як їх використати? Що таке заголовок таблиці? Як відсортувати рядки таблиці?
7. Назвіть способи вставлення ілюстрацій і різницю між ними. Що таке обтікання ілюстрації текстом і як його застосовувати? Які власні ілюстративні засоби має Word?
8. Що таке верстання? Як вставити у документ готовий текст?
9. Що таке розділи і навіщо вони потрібні? Що таке колоннитули?
10. Дайте означення стилю. Як створити, модифікувати та застосувати стиль?
11. Що таке шаблон документа і що у ньому міститься?
12. Які є типи виносок? Як їх вставити і вилучити з тексту?
13. Що потрібно для формування змісту документа і як це зробити?
14. Наведіть приклад ситуації, коли доцільно скористатися макрокмандою.
15. Як надрукувати кілька окремих сторінок документа?

7. ТАБЛИЧНИЙ ПРОЦЕСОР Microsoft Excel

📖 План викладу матеріалу

1. Основні відомості.
 2. Початок роботи з Microsoft Excel.
 3. Побудова таблиць.
 4. Бази даних в Excel.
 5. Імпорт даних з текстових файлів.
 6. Ділова графіка.
 7. Друкування аркуша.
 8. Макрокоманди.
- Додатки.

🔑 Ключові терміни розділу

✓ <i>Табличний процесор</i>	✓ <i>Формат клітинки</i>
✓ <i>Книга</i>	✓ <i>База даних</i>
✓ <i>Аркуш</i>	✓ <i>Форма даних</i>
✓ <i>Клітинка</i>	✓ <i>Фільтр</i>
✓ <i>Таблиця</i>	✓ <i>Зведена таблиця</i>
✓ <i>Адреса клітинки</i>	✓ <i>Імпорт даних</i>
✓ <i>Формула</i>	✓ <i>Діаграма</i>
✓ <i>Функція</i>	✓ <i>Макрокоманда</i>

7.1. Основні відомості

Бухгалтер, економіст, інженер чи приватний підприємець повсякденно працює з інформацією, яка наведена у вигляді таблиць даних як текстового, так і числового типу. Для табличних розрахунків характерні порівняно прості формули і великі обсяги даних. Якщо вихідних даних небагато і розрахунок за табличною формою є разовим, то можна використати мікрокалькулятор. Однак якщо подібні розрахунки є постійними чи супроводжуються великими обсягами даних, то працівник, який їх виконує, стає неуважним, втомлюється, і, як наслідок, робить помилки. Аналіз великої кількості даних, що пов'язані між собою складними залежностями, потребує надзвичайної ретельності та чималих затрат часу. Швидке створення, заповнення, оформлення та друкування таблиць у зручному вигляді, відшукування сум чи середніх значень чисел, розміщених у стовпцях або рядках таблиці, та складніший аналіз даних – це ті завдання, які найліпше вирішувати за допомогою табличного процесора.

Табличні процесори (інші терміни – електронні таблиці або табличні редактори) – незмінний атрибут програмного забезпечення персонального комп'ютера. Електронна таблиця є універсальним засобом для автоматизації розрахунків над табличними даними. Її створюють у пам'яті комп'ютера, потім її можна переглядати, змінювати, записувати на магнітний чи оптичний диск для зберігання, друкувати на принтері. На екрані дисплея електронна таблиця подається у вигляді матриці, що складається зі стовпців та рядків, на перетині яких є клітинки. Стовпці і рядки мають ідентифікатори, тому кожному клітинку можна визначити єдиним способом. У клітинках розміщують числа, тексти і математичні формули.

За допомогою табличних процесорів можна не тільки автоматизувати розрахунки, а й ефективно проаналізувати їхні можливі варіанти. Змінюючи значення одних даних, можна спостерігати за змінами інших, що залежать від них. Такі розрахунки виконуються швидко і без помилок, надаючи користувачу за лічені хвилини велику кількість варіантів розв'язання задачі. Усе це дає підстави вважати електронну таблицю обов'язковим елементом інженерної, управлінської й наукової діяльності.

Серед найвідоміших табличних процесорів треба назвати програми Excel, Lotus 1-2-3, Quattro Pro. Однак велика популярність пакету офісних програм Microsoft Office де-факто визначає стандартом табличного процесора програму Excel. Тож далі розглянуто українську версію програми Microsoft Office Excel 2003.

Можливості Excel набагато більші, ніж розуміють під терміном табличний процесор. Опрацювання тексту, статистичний аналіз та прогнозування, ділова графіка, керування базами даних – програма настільки потужна, що у багатьох випадках перевершує спеціалізовані програми – текстові редактори чи системи керування базами даних.

Поряд із такими простими завданнями, як підготовка числових, текстових та змішаних таблиць, оформлення найрізноманітніших бланків, наведення результатів у формі ділової графіки, за допомогою Excel вирішують набагато складніші проблеми. Наприклад, використовуючи цю програму, на підприємстві можна розраховувати податки і заробітну плату, вести облік кадрів і витрат, планувати виробництво та керувати збутом. А потужні математичні та інженерні функції Excel дають змогу розв'язувати багато задач у галузі природничих та технічних наук.

7.2. Початок роботи з Microsoft Excel

7.2.1. Файли даних

Файли даних, з якими працює програма Excel (вони мають розширення .xls), називають *книгами*. Кожна книга складається з електронних

таблиць – *аркуші*. Нова книга під час відкриття містить за замовчуванням три аркуші (цю кількість можна змінити у полі **Кількість аркушів** у **новій книзі** сторінки **Загальні** діалогового вікна **Параметри**, яке активізують командою **Сервіс/Параметри**. Книга може містити до 255 аркушів. Крім аркушів-таблиць, книги можуть містити аркуші діаграм. Кожен аркуш-таблиця може мати до 256 стовпців і 256^2 (65536) рядків. Можна відкривати одночасно декілька книг, кожна з яких відкривається у своєму вікні.

Для створення нової книги потрібно виконати команду меню **Файл/Створити**, яка відкриває в області завдань діалогове вікно **Створити книгу** (рис. 7.1).

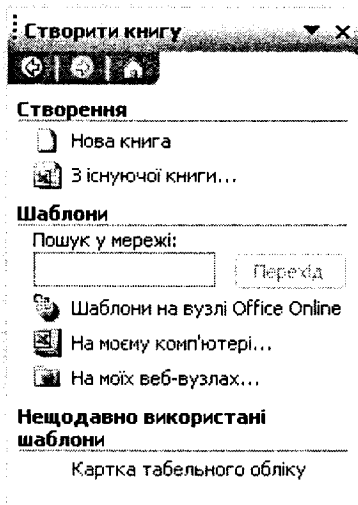


Рис. 7.1. Створення нової книги

У цьому вікні треба вибрати спосіб створення книги та шаблон, на якому базуватиметься книга. Для швидкого створення книги на базі стандартного шаблону можна скористатися піктограмою **Створити** панелі інструментів **Стандартна**.

Для збереження книги використовують команду **Файл/Зберегти**. Якщо книгу зберігають уперше, то відкриється діалогове вікно **Збереження документа** (рис. 7.2), де потрібно вказати:

- назву файлу у полі **Ім'я файлу**;
- диск і каталог у полі **Папка**;
- формат, у якому зберігатиметься файл у полі **Тип файлу**.

Натискання на кнопку **Зберегти** ініціює процес збереження. Кнопкою **Скасувати**, як і в усіх інших діалогових вікнах, відмінюють виконання операції.

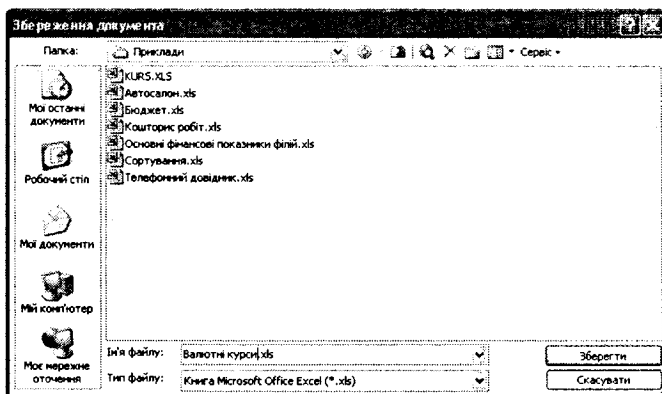


Рис. 7.2. Збереження новоствореної книги

За допомогою меню **Сервіс/Параметри...** відкривають діалогове вікно **Параметри**, у якому на закладці **Безпека** Excel дає змогу встановити захист книги двох типів:

- відкриття книги тільки за паролем, заданим у полі **Пароль для відкриття**,
- внесення змін у книгу тільки за паролем, заданим у полі **Пароль дозволу запису**,

а також задати рекомендований (але не обов'язковий) режим "лише для читання" (опція **рекомендувати доступ лише для читання**) (рис. 7.3). Наступний запис книги командою **Файл/Зберегти** (або з використанням піктограми **Зберегти** панелі **Стандартна**) проходить без діалогу. У разі потреби зберегти книгу під іншим іменем використовують команду **Файл/Зберегти як...**

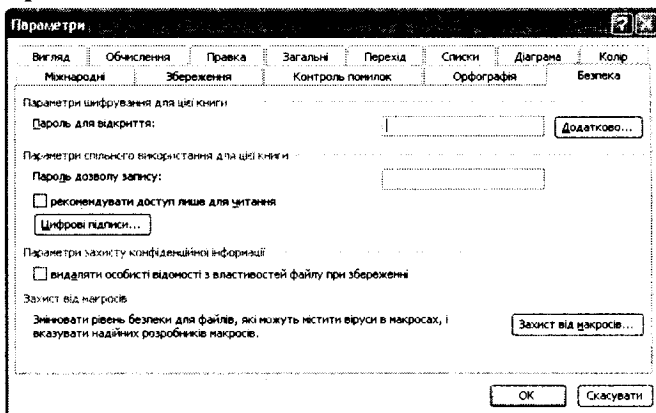


Рис. 7.3. Задання параметрів збереження

Для відкриття наявної книги використовують команду **Файл/Відкрити** або однойменну піктограму панелі **Стандартна**. У діалоговому вікні **Відкриття документа**, що відкривається (рис. 7.4), треба вибрати файл і натиснути на кнопку **Відкрити**.

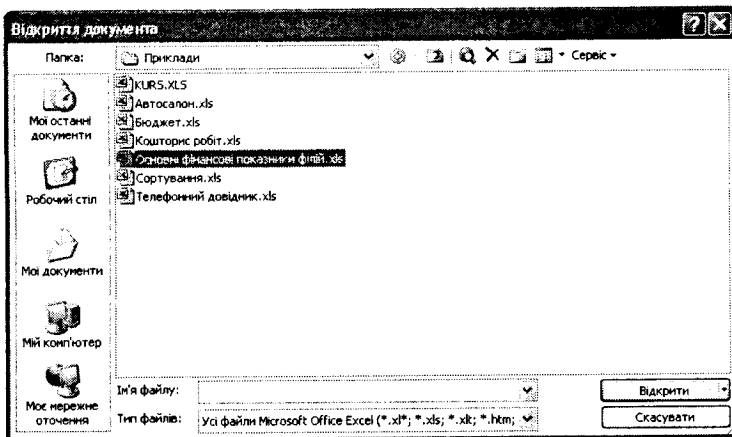


Рис. 7.4. Відкриття збереженої раніше книги

Інформація про місцезнаходження та перелік книг, з якими потрібна одночасна робота, може бути збережена командою **Файл/Зберегти робочу область** у вигляді файлу з розширенням **.xlw**. Під час його відкриття відбувається пошук та завантаження усіх вказаних у ньому книг.

Швидко відкрити одну з книг, з якими працювали в останніх сеансах роботи з Excel, можна шляхом вибору потрібного імені у переліку кількох файлів, що міститься в нижній частині меню **Файл**. Кількість файлів, які потрібно запам'ятати, задають у полі праворуч від опції **Пам'ятати список файлів**, до сторінки **Загальні** діалогового вікна **Параметри**.

7.2.2. Робоче вікно програми

Приклад робочого вікна Excel наведено на рис. 7.5. Основними елементами робочого вікна Excel є:

- Рядок заголовка;
- Рядок меню;
- Панелі інструментів;
- Рядок формул;
- Смуги прокрутки;
- Рядки і стовпці аркуша;
- Закладки з іменами аркушів;
- Рядок стану.

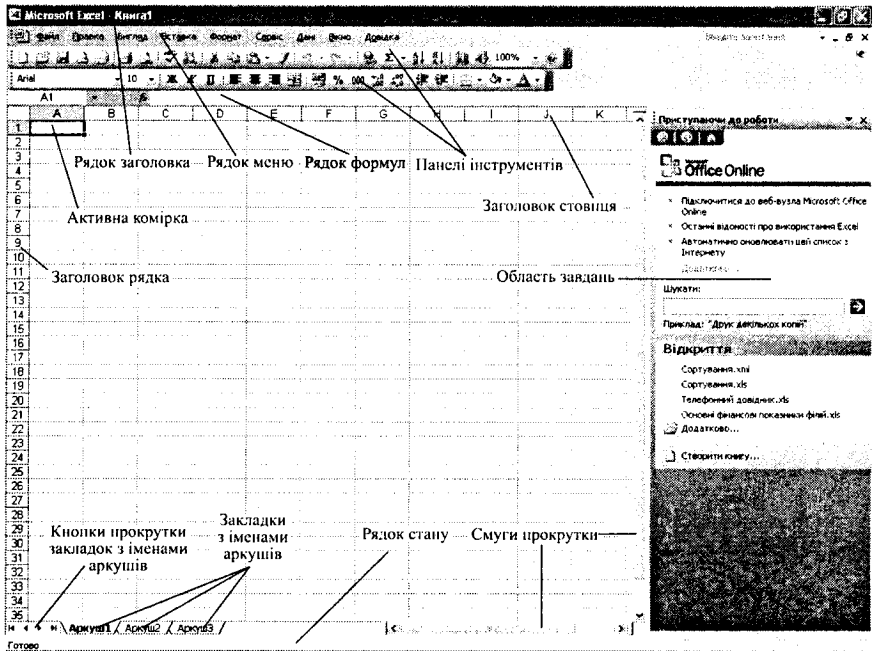


Рис. 7.5. Робоче вікно Excel

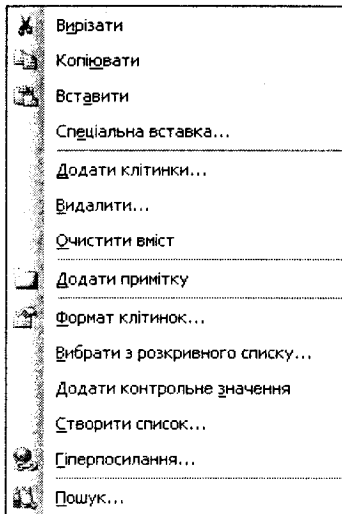


Рис. 7.6. Контекстне меню

Доступ до всіх функцій програми можна отримати через головне меню. Воно містить дев'ять пунктів: **Файл**, **Правка**, **Вигляд**, **Вставка**, **Формат**, **Сервіс**, **Таблиця**, **Вікно** і **Довідка**. Крім цього, у головному меню можуть бути присутніми ще кілька команд, пов'язаних із вбудованими макрокладами, якщо ті були підключені за допомогою команди **Сервіс/Набудови**. Виконання більшості основних команд можна ініціювати, окрім вибору в головному меню, ще двома способами: натисканням на відповідну піктограму однієї з панелей інструментів або комбінацією "гарячих" клавіш. Можна використовувати також команди контекстного меню, яке відкривається після натискання на праву кнопку мишки і містить переважно найуживаніші команди головного меню (рис. 7.6).

Без використання мишки у головне меню можна потрапити, якщо натиснути на клавішу **Alt**. Для вибору потрібної позиції меню треба користуватися клавішами стрілок (і натисканням на **Enter** для виконання команди) або уведенням підкресленої у назві букви.

Нижче від меню, за бажанням користувача, можуть бути розміщені панелі інструментів (звично відкритими є панелі **Стандартна** і **Форматування**). Для візуалізації чи приховання панелей використовують команду **Сервіс/Настройка...** (або **Вигляд/Панелі інструментів/Настройка**), яка відкриває вікно (рис. 7.7), де на сторінці **Панелі інструментів** виводиться список усіх наявних панелей. Найуживаніші можна вибрати безпосередньо з меню, яке відкривають командою **Вигляд/Панелі інструментів** або натисканням на праву кнопку мишки над однією з уже відкритих панелей. Зазначимо, що в Excel рядок меню вважається панеллю з іменем **Рядок меню аркуша**.

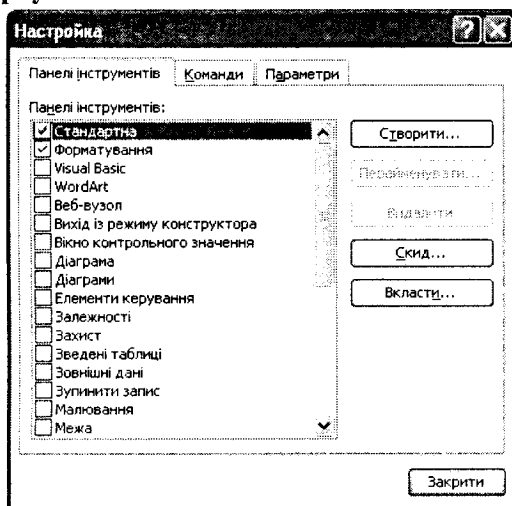


Рис. 7.7. Візуалізація панелей інструментів

Захоплення (за вільне від піктограм місце, наприклад, за розділювачі окремих груп піктограм) і переміщення всередину робочого вікна або подвійне натискання мишкою на вільному місці переводить панель у плаваючу форму, коли можна змінювати її розміри та місцезнаходження, а зворотна операція повертає панель до фіксованого стану.

Нижче від панелей інструментів розташовано рядок формул. У його лівій частині відображається адреса активної клітинки, у правій – уся інформація, яку вводять в активну клітинку або яка вже міститься в ній. Візуалізацію рядка формул можна увімкнути або вимкнути командою **Вигляд/Рядок формул**.

Усередині робочого вікна Excel книги подаються окремими вікнами (вікнами таблиць), які можна максимізувати, мінімізувати (згорнути до вигляду піктограми) або впорядкувати (розмістити одночасно декілька). Вікно таблиці містить смуги прокручування та рядок вибору потрібного аркуша (таблиці) з книги. Для перегляду великих таблиць вікно можна розділити на дві або чотири частини командою **Вікно/Розділити** (або захопленням і "втягуванням" мишкою спеціальних роздільників праворуч від горизонтальної та зверху від вертикальної смуг прокручування). Інший спосіб роботи з великими таблицями – це відкрити нове вікно командою **Вікно/Нове** та впорядкувати копії однієї й тієї ж книги у вікні Excel за допомогою команди **Вікно/Розташувати...** одним з таких способів: поруч, зверху донизу, зліва направо, каскадом (рис. 7.8), або будь-яким довільним чином за допомогою мишки.

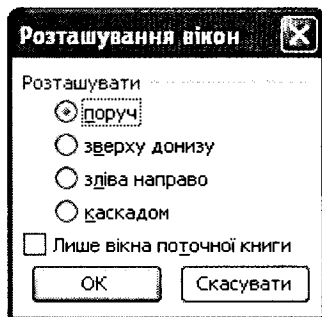


Рис. 7.8. Задання способів розміщення вікон

7.2.3. Управління курсором

Усі дані, з якими працює табличний процесор, а це, зазвичай, числа, текст, дати, формули, зберігаються в клітинках таблиці (клітинка є перетином рядка та стовпця). Після відкриття наявної або створення нової таблиці активною (біжучою) завжди стає ліва верхня клітинка, про що свідчить чорна рамка навколо неї. Іншу клітинку можна активізувати мишкою або за допомогою клавіш керування курсором. Крім цього, для переміщення по таблиці можна використовувати такі навігаційні клавіші та їхні комбінації:

Tab – перехід на одну клітинку праворуч;

Shift+Tab – перехід на одну клітинку ліворуч;

Home – перехід у перший стовпець;

Ctrl+Home – перехід до початку таблиці;

Ctrl+End – перехід до останньої клітинки (яку використовують);

PgUp – одна сторінка вгору;

PgDn – одна сторінка донизу;

Alt+PgUp – одна сторінка ліворуч;

Alt+PgDn – одна сторінка праворуч;

Ctrl+PgUp – перехід до сусіднього (ліворуч) аркуша;

Ctrl+PgDn – перехід до сусіднього (праворуч) аркуша.

Натискання на клавішу **Enter** дає змогу перейти в одну із суміжних клітинок, якщо увімкнута опція **Перехід до іншої клітинки після вводу** сторінки **Правка** діалогового вікна **Параметри**, причому напрям такого переходу визначається у полі у **напрямку**.

Ще одним засобом активізації певної клітинки є занесення її адреси у ліве поле рядка формул. Зауважимо, що для цього можна використати довільний допустимий спосіб записування адрес.

Після уведення в клітинку тексту з клавіатури або натискання на клавішу **F2** з'являється курсор уведення у вигляді I-подібного штриха. Його можна переміщувати вздовж тексту в межах активної клітинки за допомогою згаданих навігаційних клавіш.

Більшість команд Excel виконуються для активної клітинки або виділеної ділянки, яка є групою активних клітинок. Виокремлюють їх шляхом переміщення курсора мишки з одночасним утриманням натиснутою її лівої клавіші. Вміст виділених клітинок (за винятком початкової) інвертується (зображається білим кольором на чорному тлі). Для виокремлення цілих стовпців та рядків треба натиснути мишкою на їхніх заголовках, а якщо натиснути в позиції перетину заголовків стовпців та рядків, то виокремиться увесь аркуш. Утримуючи клавішу **Ctrl**, можна одночасно виокремити декілька несуміжних ділянок. Виокремлення можна також виконувати, утримуючи натиснутою клавішу **Shift** (або після натискання на **F8**), за допомогою клавіш-стрілок та інших навігаційних команд клавіатури.

7.3. Побудова таблиць

7.3.1. Способи адресації даних

Кожна клітинка таблиці має свою адресу. В Excel передбачено два способи адресації клітинок, які умовно позначають **A1** та **R1C1**.

Перший спосіб полягає в тому, що кожен рядок таблиці має свій порядковий номер, а кожному стовпцю відповідає буква (дві букви) англійського алфавіту: **A, B, ..., Z, AA, AB, ..., AZ, BA, ..., BZ, IA, ..., IV**, наприклад:

- **A1** – адреса клітинки, розташованої на перетині стовпця **A** та рядка **1**;
- **C16** – адреса клітинки, розташованої на перетині стовпця **C** та рядка **16**.

Спосіб адресації **R1C1** передбачає використання для ідентифікації як рядків, так і стовпців їхніх порядкових номерів, записаних після буквених позначень: **R** – рядок (англ. – row), **C** – стовпець (англ. – column).

У цьому випадку поряд з адресою, наприклад **R16C3** (шістнадцятий рядок, третій стовпець), у формулах можна використовувати адреси щодо активної клітинки, наприклад:

- **RC(-1)** – адреса клітинки, що розташована ліворуч від активної;
- **R(-1)C(2)** – адреса клітинки, що розташована ліворуч та через одну вниз від активної.

Спосіб адресації клітинок задають на сторінці **Загальні** вікна **Параметри** шляхом вмикання або вимикання опції **Стиль посилань RIC1**.

Використаний спосіб адресації є суттєвим під час копіювання клітинок, що містять формули з адресами клітинок. За замовчуванням Excel вважає всі адреси типу **A1**, **C16** відносними (визначається їхній зсув за рядками та стовпцями стосовно активної клітинки). Тому копіювання клітинки **E3** з формулою **=B3+C3+D3** в клітинку **E4** автоматично спричинить зміну формули на **=B4+C4+D4**. Така автоматична зміна у більшості випадків виправдана і прискорює створення чи редагування електронної таблиці. Для задання абсолютної адреси (така адреса не змінюватиметься внаслідок копіювання) клітинки використовують запис **\$A\$1** замість **A1**.

Отже, знак долара **\$** є символом абсолютної адресації. Якщо маємо запис **\$A1**, то це означає, що внаслідок копіювання (перенесення) буде змінюватись адреса тільки рядка клітинки, а якщо позначено **\$A1** – тільки стовпця.

Абсолютну адресу для обох способів адресації зручно задавати, призначаючи клітинці (або діапазону клітинок) ім'я, яке надалі буде використовуватись у формулах. Застосування імен спрощує побудову та розуміння формул. Швидким способом присвоєння імені є його уведення у поле адреси активної клітинки, що розміщене ліворуч від рядка формул (рис. 7.9).

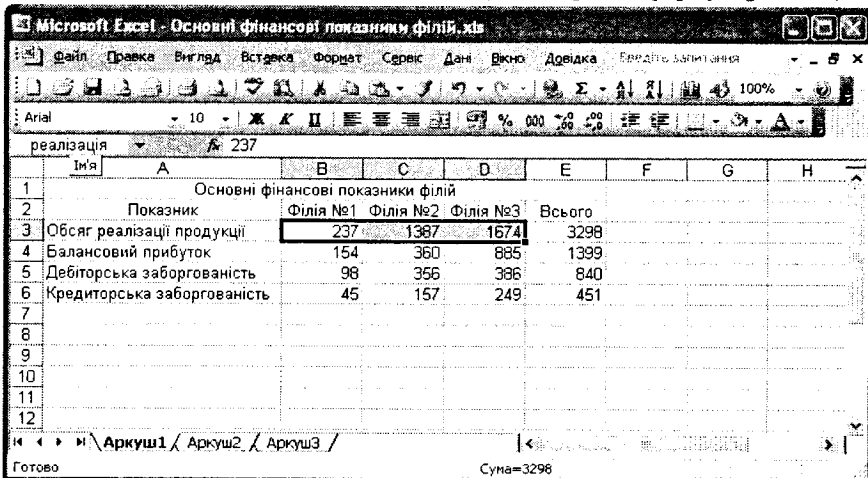


Рис. 7.9. Присвоєння імен діапазнам

Присвоювати імена можна також за допомогою команди **Вставка/Ім'я/Надати...** У діалоговому вікні **Надання імені** записують нове ім'я (у разі редагування вибирають наявне зі списку), а в полі **Формула** – діапазон клітинок разом з іменем аркуша. Адреси клітинок можна вказати як абсолютні (за замовчуванням), так і відносні (у показаному на рис. 7.10 прикладі ім'я **Балансовий_прибуток** присвоюється діапазону клітинок **B4:D4**). Натискання на кнопку **Додати** завершує процедуру присвоювання імені.

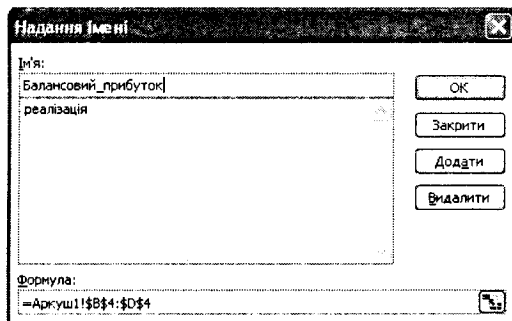


Рис. 7.10. Діалогове вікно **Надання імені**

Створити ім'я можна також, використавши вміст однієї із суміжних з діапазоном клітинок. Для цього треба виокремити діапазон разом із цією клітинкою і виконати команду **Вставка/Ім'я/Надати...**

Визначені в аркуші імена можна вставляти у формули, використовуючи команду **Вставка/Ім'я/Вставити...**, або, якщо формули записані в клітинки таблиці раніше, замінити адреси іменованих діапазонів на їхні імена у всій таблиці або в попередньо виділеній її частині командою **Вставка/Ім'я/Застосувати...**

Інформація в робочому аркуші може містити зв'язки з іншими аркушами тієї ж книги. Щоб використати у формулі адресу клітинки з іншого робочого аркуша, застосовують конструкцію **Назва аркуша!Адреса клітинки**, а із аркуша іншої книги – **[Назва книги. xls]Назва аркуша!Адреса клітинки**.

Наприклад: **[Книга1.xls]Персонал!R2C7**.

7.3.2. Уведення, редагування та опрацювання даних

Для введення даних курсор треба помістити у потрібну клітинку таблиці і набрати цифри або текст. Інформація, яку вводять, одночасно відображається у правому полі рядка формул, куди безпосередньо можна вводити дані. Після введення першого ж знака ліворуч від цього поля з'являються три піктограми. Піктограма, на якій зображений червоний

символ **X**, є аналогом кнопки **Скасувати**, ✓ із зеленою галочкою, аналог клавіші **Enter**, а з символом f_x – піктограма виклику майстра для компонування формул шляхом діалогу. Закінчують введення натисканням на клавішу **Enter** або піктограму **ОК** чи на одну з клавіш керування курсором. Виправляють введені дані у рядку формул або безпосередньо в клітинці після натискання на клавішу **F2** чи подвійного натискання мишкою (для редагування даних у клітинці треба увімкнути опцію **Правка безпосередньо у клітинці** сторінки **Правка** діалогового вікна **Параметри**).

Уведені в клітинку число або дата за замовчуванням вирівнюються до правого краю, а текст до лівого. Для того, щоб увести число як текст потрібно взяти його в лапки. Довгий текст стандартно відображається поперек сусідніх клітинок, якщо вони не містять даних, однак його можна розмістити у кількох рядках, виконавши команду **Формат/Клітинок** і на сторінці **Вирівнювання** увімкнути опцію **переносити по словах** (на цій же сторінці, до речі, можна змінити й стандартний спосіб вирівнювання, про що йтиметься у наступному пункті).

Формули в Excel'і завжди починаються зі знака рівності (=) і можуть містити числові та текстові величини (константи), знаки арифметичних операцій, операції порівняння, операції з текстом, дужки, адреси клітинок та імена, а також вбудовані формули, які називаються функціями, (наприклад, для підсумовування чисел у клітинках з A1 по A5 – **SUM(A1:A5)**). Якщо елементом формули (аргументом функції) є адреса клітинки або її ім'я, то в обчисленнях Excel використовує результат, одержаний в адресованій клітинці. Всі текстові величини у формулах треба брати в лапки, використання пробілів не дозволене.

Після того, як користувач уведе формулу в яку-небудь клітинку, там звичайно відразу з'являється результат обчислень за цією формулою. Щоб побачити власне формулу, треба активізувати відповідну клітинку і переглянути вміст правого поля рядка формул (натискання на функціональну клавішу **F2** відображає формулу в клітинці).

Розглянемо використання функцій у формулах детальніше. Найбільш поширеною в табличних розрахунках є функція суми. Для швидкого задавання цієї функції у формулі призначена піктограма **Автосума** панелі інструментів **Стандартна**. Після її натискання Excel намагається "відгадати", вміст яких саме клітинок слід підсумувати (рис. 7.11).

Змінити діапазон підсумовування можна мишкою (після чого треба натиснути на клавішу **Enter**) або задаванням потрібних адрес з клавіатури. Діапазони позначають адресами першої та останньої клітинок, розділеними двокрапкою (:). Можна задавати одразу кілька діапазонів, відокремлюючи їх один від одного крапкою з комою (;) (щоб задати кілька діапазонів за допомогою мишки, потрібно утримувати натиснутою клавішу **Ctrl**).

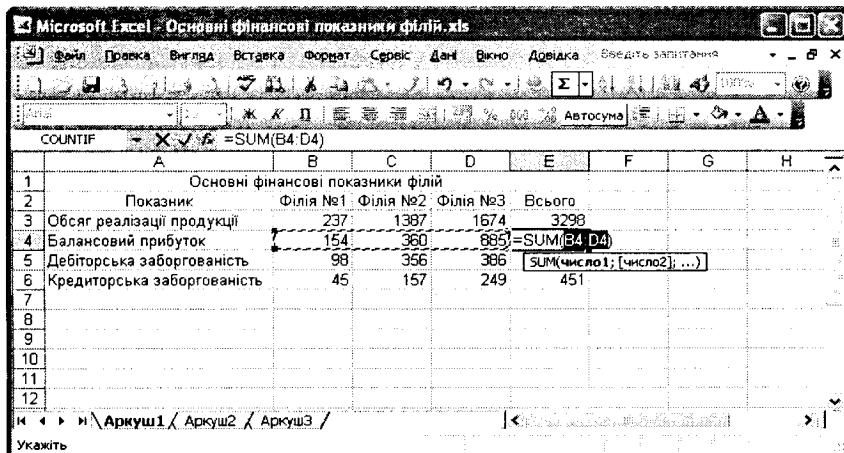


Рис. 7.11. Функція автосуми

Для застосування у формулах інших вбудованих функцій зручно користуватися майстром вставки функцій. Викликати його можна, окрім описаного вище способу, командою меню **Вставка/Функція** або відповідною піктограмою панелі інструментів **Стандартна**. У діалоговому вікні, що відкривається (рис. 7.12), потрібно:

- у списку **Категорія** вибрати групу функцій;
- у списку **Функція** відшукати функцію, користуючись її описом у нижній частині вікна;
- натиснути на кнопку **ОК**.

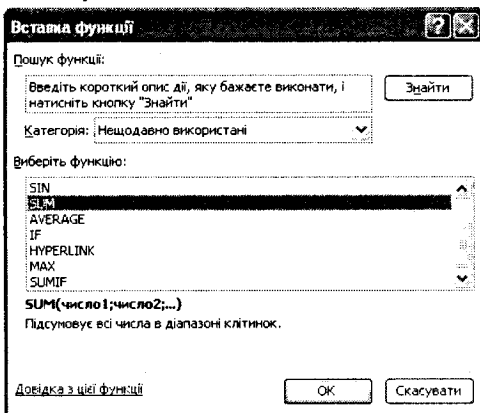


Рис. 7.12. Вставка функції (вибір функції)

Після цього ім'я функції з'явиться у правому полі рядка формул і відкривається ще одне діалогове вікно для задання аргументів (рис. 7.13).

Це можна зробити шляхом уведення потрібних даних з клавіатури або безпосереднім зазначенням діапазонів у таблиці за допомогою мишки описаним вище способом (у цьому випадку перед заданням кожного аргумента треба розмістити курсор уведення у відповідному полі діалогового вікна і натиснути на піктограму з червоною стрілкою, яка мінімізує вікно конструктора формул; повторне натискання на піктограму відновлює діалогове вікно). Завершують компонування формули натисканням на кнопку ОК.

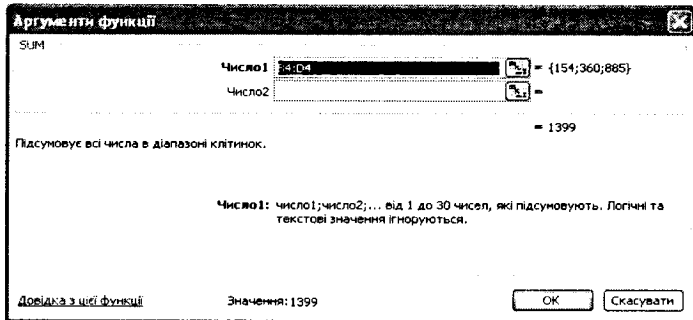


Рис. 7.13. Вставка функції (вказання аргументів)

При уведенні в клітинку формул слід дотримуватись наступних правил, котрі допоможуть уникнути помилок:

- Аргументи формул, як правило, розділяються символом (;);
- Адреси клітинок слід записувати в латинській розкладці клавіатури.
- Регістр не відіграє ніякої ролі;
- При використанні іменованих клітинок чи діапазонів аргументи вводяться в такій самій розкладці, що і при присвоєнні імен.

Якщо вставку функцій викликано уведенням в клітинку знаку = то назву функції потрібно вибирати не у діалоговому вікні **Вставка функції**, яке в цьому випадку не відкривається, а у лівому полі рядка формул (рис. 7.14).

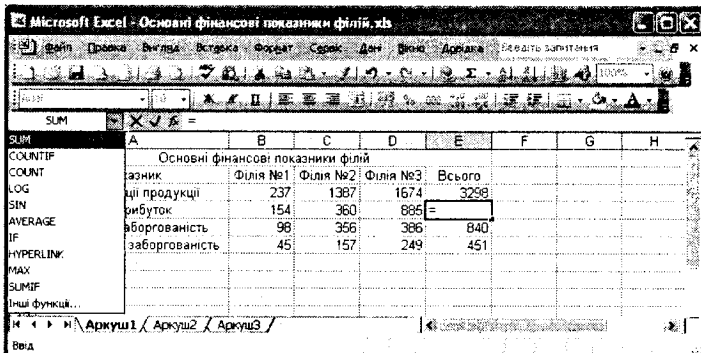


Рис. 7.14. Вибір функції з переліку нещодавно використаних

Приклад. Кошторис робіт

Нехай потрібно визначити ціну робіт, якщо фонд заробітної плати повинен становити 2000 грн., заплановані витрати на службові відрядження в сумі 200 грн. та матеріали в сумі 150 грн. Нехай відрахування на соціальне страхування становить 37% до статті "Заробітна плата". Накладні витрати, прибуток та інші прями витрати обчислюють за заданими відсотковими співвідношеннями до ціни робіт (рис. 7.15).

Для розв'язання такої задачі потрібно увести в клітинці таблиці такі дані (рис. 7.15). Використана в клітинці C4 абсолютна адреса \$C\$9 дає змогу увести формули в клітинки C7 та C8 шляхом копіювання з C4. Та ще ліпше присвоїти клітинці C9 ім'я, наприклад, *ціна*. Тоді формула в клітинці C4 мала б більш зрозумілий вигляд =ціна*B4/100 і її можна копіювати без проблем.

The screenshot shows a Microsoft Excel spreadsheet titled "Кошторис робіт.xls". The spreadsheet contains a table with the following data:

	A	B	C	D
1	Статті витрат	%	Сума	
2	Заробітна плата		2000	
3	Відрахування на соц. страхування	37	=C2*B3/100	
4	Накладні витрати	15	=\$C\$9*B4/100	
5	Матеріали		150	
6	Витрати на службові відрядження		200	
7	Інші прями витрати	2,2	=\$C\$9*B7/100	
8	Прибуток	10	=\$C\$9*B8/100	
9	Ціна		=SUM(C2:C8)	

Рис. 7.15. Уведення даних та формул

Потрібно звернути увагу на те, що формули в клітинках C4, C7-C9 містять рекурсію (значення клітинки C9 є сумою клітинок, значення яких залежать від C9). Після спроби увести ці формули з'явиться повідомлення "Неможливо обчислити формулу..." та пропозиція виявити та виправити формули з циклічними посиланнями. За замовчуванням Excel вважає такі обчислення помилковими. Дати дозвіл на рекурсію можна командою **Сервіс/Параметри** на сторінці опцій **Обчислення** у рамці **Ітерації**.

Результат обчислень за заданими формулами відображений на екрані (рис. 7.16).

Кількість цифр після десяткової крапки для стовпця C у цих прикладах задана піктограмою **Збільшити розрядність** панелі інструментів

Форматування. Знак десяткового роздільника (, або,) встановлюють настройкою компоненти **Мова і регіональні стандарти** утиліти **Windows Панель керування**.

	A	B	C	D	E	F
1	Статті витрат	%	Сума			
2	Заробітна плата		2000,00			
3	Відрахування на соц. страхування	37	740,00			
4	Накладні витрати	15	636,68			
5	Матеріали		150,00			
6	Витрати на службові відрядження		200,00			
7	Інші прямі витрати	2,2	93,38			
8	Прибуток	10	424,45			
9	Ціна		4244,51			

Рис. 7.16. Результат обчислень кошторису робіт

Наведений приклад можна доповнити відомостями щомісячних нарахувань заробітної плати працівникам. Такі відомості зручно розмістити на окремих аркушах, присвоївши кожному ім'я відповідного місяця: січень, лютий, березень. Нехай у кожній з цих відомостей клітинка "Всього нараховано" має абсолютну адресу D10. Тоді в кошторисі формула в клітинці C2 набуде вигляду =січень! D 10+лютий! D 10+березень! D 10

7.3.3. Робота з клітинками таблиці

Якщо у різних клітинках таблиці повинні міститись однакові дані, то немає потреби щоразу їх заново вводити – достатньо скористатись потужними засобами Excel'у для автоматичного заповнення та обміну даними. Скажімо, вміст виділеної ділянки можна вилучати або копіювати в буфер обміну командами **Правка/Вирізати (Ctrl+X)** та **Правка/Копіювати (Ctrl+C)** чи відповідними піктограмами панелі **Стандартна** з тим, щоб потім мати змогу вставити у потрібне місце таблиці командами **Правка/Вставити (Ctrl+V)** або **Правка/Спеціальна вставка** (чи піктограмою **Вставити**). У виділеній частині можна розмістити дані початкових (у сенсі напрямку заповнення) клітинок за допомогою директив меню **Правка/Заповнити: Вниз** – донизу дані верхніх клітинок, **Праворуч** – праворуч лівих, **Вгору** – угору нижніх, **Ліворуч** – ліворуч правих, **Прогресія** – у вигляді прогресії вказаного типу із заданим кроком.

Копіювати або переносити інформацію зручно за допомогою мишки, курсор якої на полі робочого аркуша може набувати, окрім звичайного вигляду "білий хрест", ще два: "чорний хрест" та "стрілка". Вигляд курсору "стрілка" (коли він є на межі виокремлення або активної клітинки) дає змогу "захопити" виділену ділянку або клітинку і перемістити її в інше місце. Якщо ж утримувати натиснутою ще й клавішу **Ctrl**, то дані копіюватимуться. У сусідні клітинки дані зручніше копіювати курсором мишки "чорний хрест" (коли курсор розміщений у правому нижньому куті виокремлення) з утримуванням натиснутою клавіші **Ctrl**. Якщо ж клавішу відпустити, то у випадку, коли вмістом виділеної ділянки є числа, Excel будуватиме зростаючий або спадний ряд (змінити його стандартні властивості можна у вікні **Прогресія**, яке активізують командою **Правка/Заповнити/Прогресія**).

Для вставлення у таблицю нових рядків використовують команду **Вставка/Рядки**, причому їх вставиться стільки, скільки рядків було виокремлено перед виконанням команди, а розміщаються вони перед першим виділеним. Стовпці вставляють аналогічно командою **Вставка/Стовпці** після виокремлення потрібної їх кількості. Команда ж **Вставка/Аркуш** – дає змогу вставити цілий аркуш.

Окремі клітинки вставляють командою **Вставка/Клітинки** також після виокремлення потрібної їх кількості. Ця команда відкриває вікно **Вставка** (рис. 7.17), де треба зазначити спосіб виконання операції:

- **клітинки зі зсувом вправо** – виділені клітинки зсуваються праворуч;
- **клітинки зі зсувом вниз** – виділені клітинки зсуваються донизу;
- **рядок** – вставляються рядки перед першою виділеною клітинкою;
- **стовпець** – вставляються стовпці перед першою виділеною клітинкою.

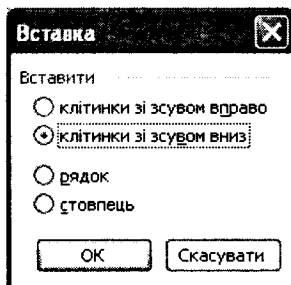


Рис. 7.17. Вибір способу вставлення клітинок

Для вилучення виділених клітинок призначена команда **Правка/Видалити** (рис. 7.18), яка відкриває вікно, де потрібно вказати спосіб вилучення:

- **клітинки зі зсувом вліво** – клітинки, що залишилися, зсуваються ліворуч;

- клітинки зі зсувом вгору – клітинки зсуваються вгору;
- рядок – вилучаються цілі рядки;
- стовпець – вилучаються цілі стовпці.

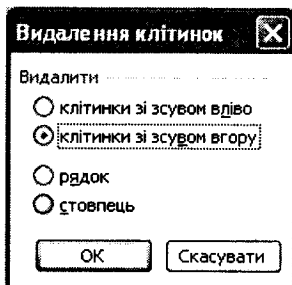


Рис. 7.18. Вибір способу вилучення клітинок

Вилучити вміст виділеної ділянки можна за допомогою команди **Правка/Очистити** з такими підкомандами:

- **Усі** – вилучити всю інформацію;
- **Формати** – тільки задані користувачем формати;
- **Вміст** – тільки вміст: числа, текст, формули (аналог – клавіша **Delete**);
- **Примітки** – тільки примітки.

Автоматично задані Excel висоту рядка та ширину стовпця можна змінити командами **Формат/Рядок/Висота** (рис. 7.19) та **Формат/Стовпець/Ширина** у відповідних діалогових вікнах.



Рис. 7.19. Вказання висоти рядка та ширини стовпця

Для швидшого та наочного задавання потрібних розмірів рядка або стовпця треба мишкою захопити межу відповідного заголовка (букви **A, B, C...** або цифри **1, 2, 3...**) і перемістити її. Команда **Формат/Стовпець/Автодбір ширини** або подвійне натискання мишкою на правій межі заголовка стовпця автоматично задає його ширину відповідно до розміру найдовшого запису.

7.3.4. Форматування клітинок

Для задання параметрів форматування клітинок використовують команду **Формат/Клітинки...**, яка активізує однойменне діалогове вікно з шістьма сторінками опцій: **Число**, **Вирівнювання**, **Шрифт**, **Межа**, **Візерунки** і **Захист**.

На сторінці **Число** (рис. 7.20) задають формат зображення даних у активній або виділених клітинках таблиці. Для цього у списку **Числові формати** треба вибрати відповідну категорію форматів, а у полях праворуч задати параметри, спостерігаючи за результатом їх застосування у рамці **Зразок**.

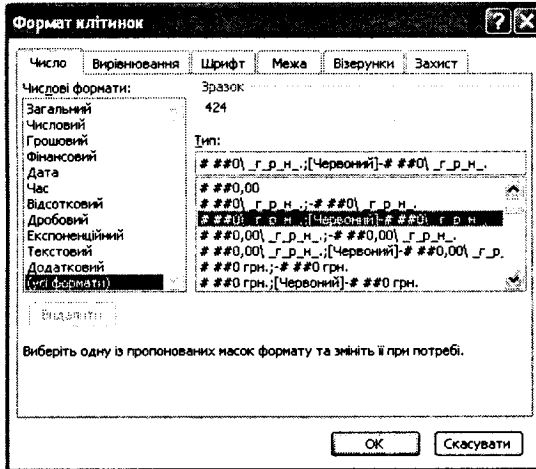


Рис. 7.20. Встановлення формату відображення даних

Крім вибору із великої кількості наявних форматів, можна також задати власний формат у полі **Тип** (попередньо вибравши категорію **усі формати**). Основні правила запису форматів такі:

- якщо формат від'ємного числа відрізняється від формату додатнього, то записують два формати через стандартний розділювач списків (;) (у разі потреби між ними можна вписати спеціальний формат для нульового значення);
- 0 – символ підстановки для цифри з обов'язковою індикацією нуля у випадку, коли немає цифри;
- # – символ підстановки для цифри без індикації нулів;
- ? – символ підстановки для цифри із заміною незначущих нулів пробілами;
- . або, – визначає позицію десяткового розділювача;
- % – число множиться на 100 і додається знак %;
- ' – символ-розділювач для тисяч;
- E+ або E– – для зображення чисел із плаваючим десятковим розділювачем; у разі задавання E– відображається тільки від'ємний знак порядку;
- : – + \$ <пробіл> <текст> – використовують у форматі довільно, вони будуть на вказаних місцях;
- [Колір] – задає колір об'єкта; основні значення: **Чорний**, **Білий**, **Червоний**, **Зелений**, **Синій**, **Жовтий**.

Приклад формату користувача:

##0,00 грн.;[Червоний]-# ##0,00 грн.; [Зелений]0

Вміст клітинки	Результат форматування
12300000	12 300 000,00 грн. (чорним кольором)
0	0 (зеленим кольором)
-1250,6	-1 250,60 грн. (червоним кольором)

Сторінку **Вирівнювання** використовують для задавання опцій вирівнювання вмісту клітинок (рис. 7.21). На ній потрібно:

- у полі **по горизонталі** задати спосіб горизонтального вирівнювання;
- у полі **відступ** – відступ тексту від лівого краю клітинки;
- у полі **по вертикалі** – спосіб вертикального вирівнювання;
- у рамці **Орієнтація** – напрям тексту;
- увімкнути опцію **переносити по словах**, якщо потрібно розмістити текст клітинки у кількох рядках;
- увімкнути опцію **автодобр ширини** для автоматичної зміни розміру шрифту так, щоб текст завжди вмщався в одному рядку клітинки;
- увімкнути опцію **об'єднання клітинок** для об'єднання виділених клітинок в одну.

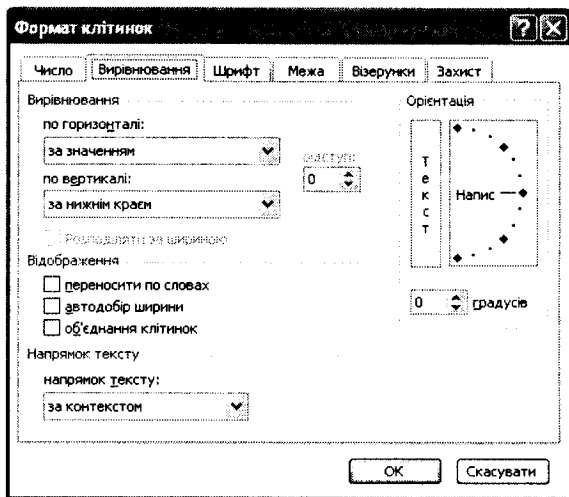


Рис. 7.21. Вирівнювання вмісту клітинок

На сторінці опцій **Шрифт** задають параметри шрифту (рис. 7.22):

- у полях **Шрифт**, **Накреслення** та **Розмір** відповідно гарнітуру, накреслення та розмір;
- у полі **Підкреслення** – стиль підкреслення;
- у полі **Колір** – колір тексту.

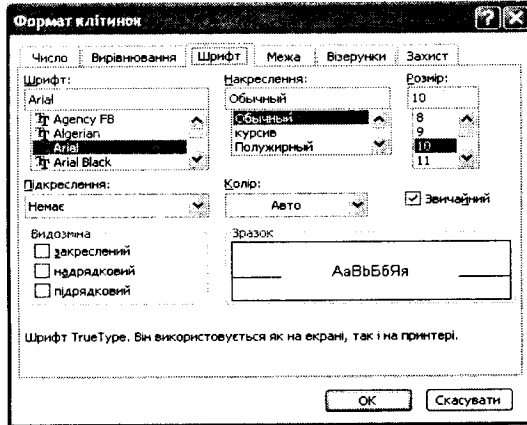


Рис. 7.22. Задання параметрів шрифту

Увімкнення опцій рамки **Видозміна** задає деякі текстові ефекти (закреслення, надрядковий та підрядковий шрифт), а опція **Звичайний** дає змогу повернутись до стандартних параметрів шрифту, заданих у стилі **Звичайний**. Зазначимо, що в режимі уведення та редагування тексту у вікні **Формат клітинок** доступна лише ця сторінка опцій.

На сторінці **Межа** задають межові лінії для виділених клітинок (рис. 7.23):

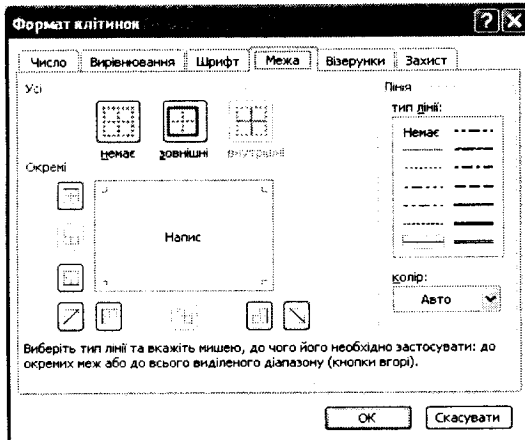


Рис. 7.23. Налаштування межових ліній

- у рамці **Лінія** визначають тип та колір межових ліній;
- у рамці **Усі** – загальний шаблон застосування вибраних ліній;
- у рамці **Окремі** – для застосування вибраних ліній до оформлення окремих меж можна використовувати піктограми або натискання мишкою на потрібних елементах у віконці.

На сторінці **Візерунки** (рис. 7.24) визначають спосіб затінювання клітинок:

- у полі **Колір** – колір тла;
- у полі **Візерунок** – спосіб та колір затінення.

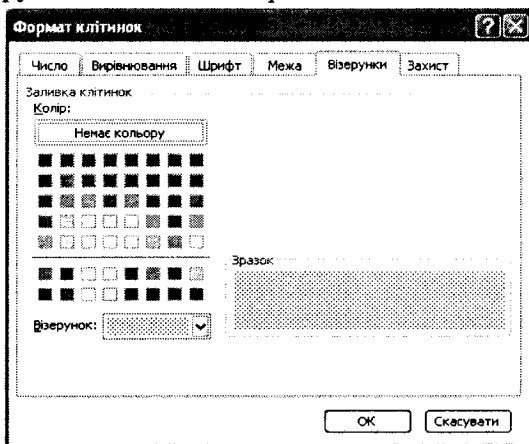


Рис. 7.24. Вибір способу затінювання

Нарешті, на сторінці **Захист** (рис. 7.25) задають режим захисту клітинок:

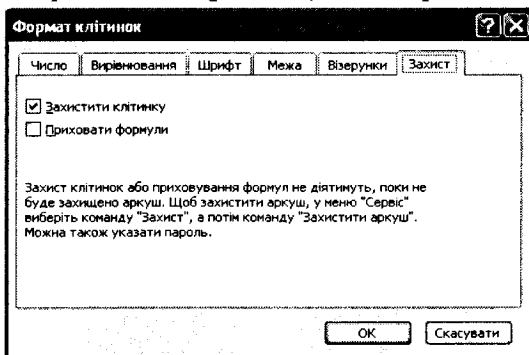


Рис. 7.25. Сторінка захисту аркуша

- опція **Захистити клітинку** забороняє модифікацію, копіювання та вилучення клітинок;
- опція **Приховати формули** дає змогу не відображати формули у рядку формул після активізації клітинок.

Однак, названі опції працюють лише у тому випадку, коли задано режим захисту аркуша. Це роблять за допомогою команди **Сервіс/Захист/Захистити аркуш** в однойменному діалоговому вікні (рис. 7.26), де встановлюють дозволи користувачам виконувати з захищеним аркушом ті чи інші дії.

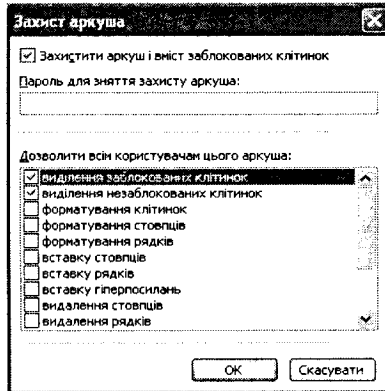


Рис. 7.26. Захист аркуша

Знімають режим захисту аркуша командою **Сервіс/Захист/Зняти захист аркуша**.

Швидко змінити шрифт, його розмір та накреслення, кольори вмісту клітинки та тла, спосіб вирівнювання та вигляд меж, формат даних для виділеного діапазону клітинок можна за допомогою відповідних піктограм панелі **Форматування**. Заданий формат клітинки можна перенести на іншу клітинку або групу клітинок піктограмою **Формат за зразком** панелі інструментів **Стандартна**. Встановити формат виділених клітинок можна також за допомогою засобів автоформатування Excel, виконавши команду **Формат/Автоформат** і у діалоговому вікні, що відкривається (рис. 7.27), вибрати готовий шаблон форматування.

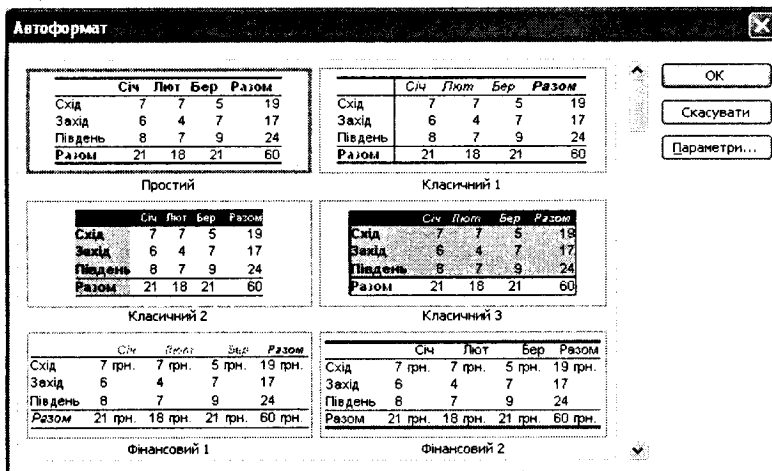


Рис. 7.27. Діалогове вікно Автоформат

Для збереження набору параметрів форматування з метою повторного використання потрібно оформити їх у вигляді стилю. Це можна зробити в діалоговому вікні **Стиль**, яке активізують командою **Формат/Стиль** (рис. 7.28).

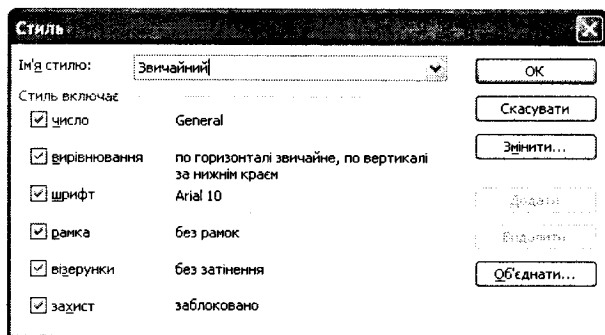


Рис. 7.28. Робота зі стилем

Щоб застосувати наявний стиль до виділених клітинок таблиці, потрібно вибрати його ім'я у списку **Ім'я стилю** і натиснути на кнопку **ОК**. Вибраний стиль можна частково модифікувати у вікні **Стиль**, вмикаючи потрібні опції у рамці **Стиль включає**. Натискання ж на кнопку **Змінити** викликає вже відоме вікно **Формат клітинок**, де можна задати всі бажані параметри форматування. Щоб створити свій власний стиль, треба у полі **Ім'я стилю** задати його ім'я, натиснути на кнопку **Додати** і потрібним чином модифікувати. Кнопка **Видалити** призначена для вилучення вибраного стилю, а кнопка **Об'єднати** дає змогу копіювати стилі з інших відкритих книг.

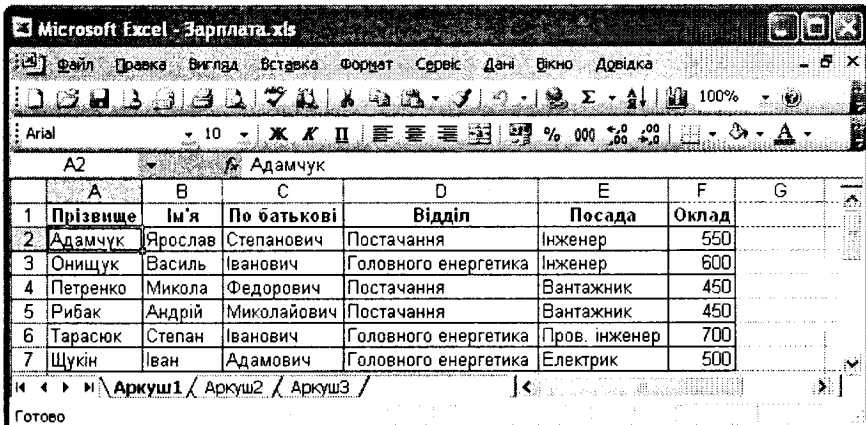
7.4. Бази даних в Excel

7.4.1. Створення списків. Використання форм

База даних в Excel (фірма Microsoft вживає термін *список*) – це послідовний набір рядків з однаковою структурою інформації у стовпцях. Такі рядки можна вважати записами бази даних. Кожен із стовпців, який використовують для зберігання даних, розглядають як поле даних, тобто окремий складовий елемент запису. Отже, певна ділянка суміжних клітинок електронної таблиці може бути базою даних, що містить до 65536 записів (перший рядок електронної таблиці відводять для назв полів), кожен з яких може мати не більше 256 полів. Для більшості практичних застосувань такі розміри бази даних є цілком достатніми.

Розглянемо процес створення та роботу з базою даних на прикладі інформації про працівників підприємства. Нехай кожен запис про працівника складається з таких полів: *Прізвище, Ім'я, По батькові, Відділ, Посада,*

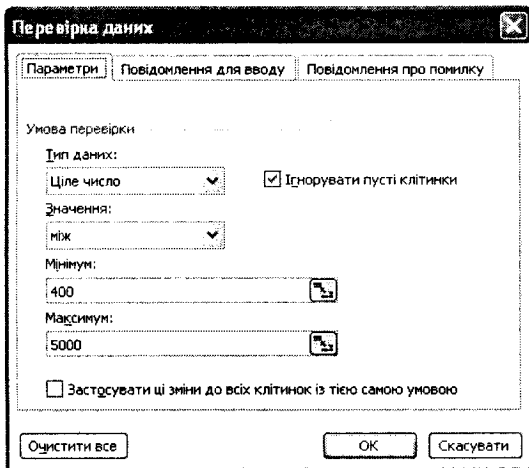
Оклад. Назви цих полів уводять в клітинки першого рядка таблиці. В наступні рядки уводять інформацію про працівників (рис. 7.29).



	A	B	C	D	E	F	G
1	Прізвище	Ім'я	По батькові	Відділ	Посада	Оклад	
2	Адамчук	Ярослав	Степанович	Постачання	Інженер	550	
3	Онищук	Василь	Іванович	Головного енергетика	Інженер	600	
4	Петренко	Микола	Федорович	Постачання	Вантажник	450	
5	Рибак	Андрій	Миколайович	Постачання	Вантажник	450	
6	Тарасюк	Степан	Іванович	Головного енергетика	Пров. інженер	700	
7	Шукін	Іван	Адамович	Головного енергетика	Електрик	500	

Рис. 7.29. База даних працівників

Щоб запобігти помилковому введенню даних, за допомогою команди **Дані/Перевірка** визначають параметри контролю для виділених клітинок або діапазонів (рис. 7.30).



Перевірка даних

Параметри | Повідомлення для вводу | Повідомлення про помилку

Умова перевірки

Тип даних: Ціле число Ігнорувати пусті клітинки

Значення: між

Мінімум: 400

Максимум: 5000

Застосувати ці зміни до всіх клітинок із тією самою умовою

Очистити все | ОК | Скасувати

Рис. 7.30. Параметри контролю введення

На сторінці опцій **Параметри**

- у полі **Тип даних** зазначають тип даних, що містяться у клітинках;
- у полі **Значення** – спосіб порівняння введених даних з контрольними значеннями;

- у полях **Мінімум**, **Максимум** та інших, що з'являються після вибору різних способів порівняння, контрольні значення;
- вмикають опцію **Ігнорувати пусті клітинки** для ігнорування порожніх клітинок.

Якщо умови перевірки даних повторно редагують, то опція **Застосувати ці зміни до всіх клітинок з тією самою умовою** дає змогу зробити аналогічні зміни в усіх клітинках з тими ж вихідними умовами. Кнопка **Очистити все** призначена для очищення всіх заданих полів на всіх сторінках вікна.

Для випадку, коли введені дані не задовольняють заданих умов, можна задати вигляд та текст повідомлення про помилку. Це роблять на сторінці **Повідомлення про помилку**, де:

- у полі **Вид** визначають вигляд повідомлення;
- у полі **Заголовок** задають заголовок вікна повідомлення;
- у полі **Повідомлення** – текст повідомлення.

Приклад задання параметрів та відповідне вікно повідомлення про помилку зображено на рис. 7.31.

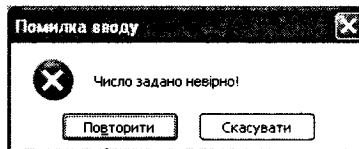
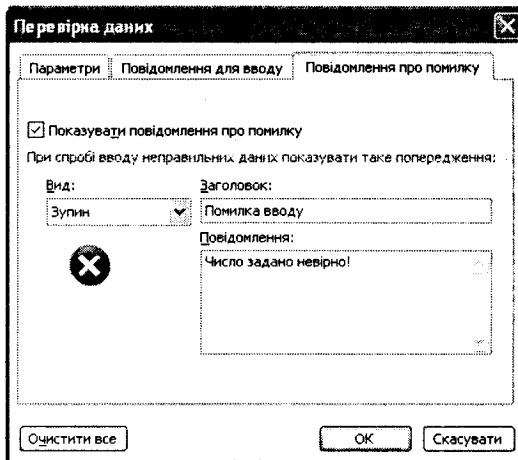


Рис. 7.31. Налаштування та відображення повідомлення про помилку

В аналогічний спосіб на сторінці **Повідомлення для вводу** можна задати текст підказки для введення, який висвічуватиметься у невеличкому вікні після активізації відповідних клітинок (рис. 7.32).

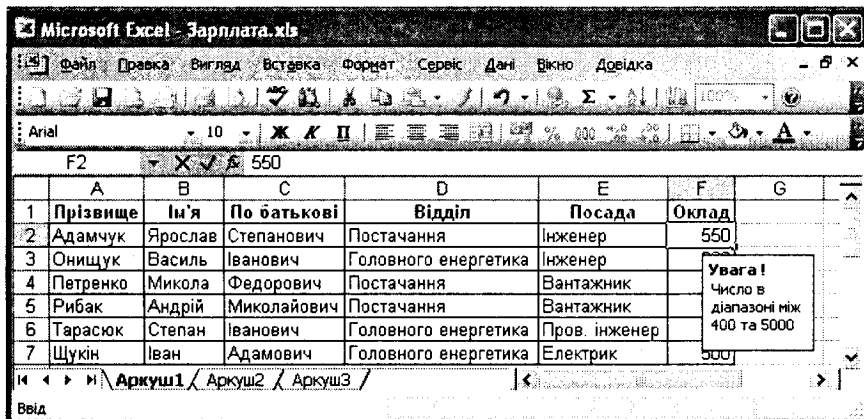


Рис. 7.32. Підказка для введення

Водночас є інший, зручніший, спосіб уведення інформації в базу даних – форма. Форму даних викликають командою меню **Дані/Форма**. Вона має вигляд діалогового вікна, де кожен запис бази даних наведений у вигляді списку поіменованих полів (рис. 7.33).

The screenshot shows the 'Forma даних' dialog box for 'Аркуш1'. It contains the following fields and buttons:

- Прізвище: Тарасюк
- Ім'я: Степан
- По батькові: Іванович
- Відділ: Головного енергетика
- Посада: Пров. інженер
- Оклад: 700

Buttons on the right side include: Створити, Видалити, Знайти назад, Знайти далі, Умови, and Закрити. A status bar at the top right indicates '5 з 6'.

Рис. 7.33. Вікно форми даних

Форма даних дає змогу додавати нові та вилучати активні записи (кнопками **Створити** та **Видалити** відповідно), зручно та наочно коригувати наявні дані. Переміщатися за записами бази можна за допомогою вертикальної смуги прокручування, клавіш **↑ (Shift+Enter)**, **↓ (Enter)** або кнопок **Знайти назад** та **Знайти далі**, а для переходу до потрібного поля, крім мишки, використовують клавішу **Tab** та комбінацію клавіш **Shift+Tab**. Натиснувши на кнопку **Умови**, можна задати критерії відбору записів за значеннями конкретних полів або заданими умовами з використанням

Для повернення у режим звичайного відображення таблиці треба виконати ту саму команду, якою був увімкнений режим автофільтра.

Складніші критерії фільтрування та пошук інформації в базі чи її частині реалізує команда **Дані/Фільтр/Розширений фільтр**.

7.4.3. Сортування даних

Excel також дає змогу швидко сортувати записи бази даних, використовуючи не більше трьох критеріїв одночасно. Командою меню **Дані/Сортування** викликають відповідне діалогове вікно (рис. 7.36), дезначають поля бази даних, за якими послідовно (від верхнього до нижнього) виконуватиметься сортування за зростанням чи спаданням. Значення опції **Діапазон даних** визначають спосіб задавання полів: за їхніми назвами (з рядком заголовка) чи за ідентифікаторами стовпців (без рядка заголовка). Кнопку **Параметри** використовують для задання додаткових параметрів сортування.

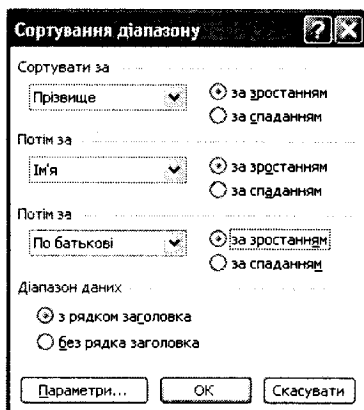


Рис. 7.36. Налаштування сортування даних

7.4.4. Використання команди Підсумки

Використання команди **Дані/Підсумки** допомагає впорядкувати список шляхом групування записів з виведенням проміжних результатів, середніх значень та іншої допоміжної інформації. Дана команда дозволяє виводити результуючу суму в верхній або нижній частині списку і полегшує підсумовування числових стовпців. Список відображається у вигляді структури, що дозволяє згорнути і розгорнути окремі розділи.

Розглянемо роботу команди **Підсумки** на прикладі даних про продаж автомобілів (рис. 7.37).

	A	B	C	D	E	F	G	H	I
1	Місяць	Продавець	Форд	Фіат	ВАЗ	Таврія	Всього		
2	Квітень	Бергман	45800	12000	98000	12800	168600		
3	Квітень	Іванов	32600	25000	84000	27670	169270		
4	Квітень	Карась	67000	11000	73000	50080	201080		
5	Травень	Бергман	32000	10000	67800	32880	142680		
6	Травень	Іванов	22000	36900	48700	80900	188500		
7	Травень	Карась	80000	31400	20000	120000	251400		
8	Червень	Бергман	100000	99000	12600	34500	246100		
9	Червень	Іванов	70000	120000	37770	67900	295670		
10	Червень	Карась	50000	58000	58000	23600	189600		

Рис. 7.37. Дані про продаж автомобілів

Включення проміжних підсумків у список здійснюється так:

1. Список доцільно відсортувати за полем, що містить групи. Наприклад, за місяцем, прізвищем тощо.
2. Виконати команду Дані/Підсумки. Відкриється діалогове вікно **Проміжні підсумки** (рис. 7.38).

Проміжні підсумки

При кожній зміні в:

Місяць

Використовувати функцію:

Сума

Додати підсумки до:

Місяць

Продавець

Форд

Замінити поточні підсумки

Закінць сторінки між групами

Підсумки під даними

Прибрати все ОК Скасувати

Рис. 7.38. Діалогове вікно **Проміжні підсумки**

3. Вибрати зі списку **При кожній зміні в** групу, для якої визначаються проміжні підсумки.
4. Вибрати зі списку **Використовувати функцію** одну з функцій (Сума – додавання чисел у групі, Кількість – підрахунок кількості заповнених клітинок у групі, Середнє – середнє арифметичне чисел у групі,

Максимум – визначення найбільшого числа у групі, Мінімум – визначення найменшого числа у групі, Добуток – добуток чисел у групі, Кількість чисел – кількість клітинок, котрі містять числові дані у групі, Зсунене відхилення – розрахунок стандартного відхилення за генеральною сукупністю, Незсунене відхилення – розрахунок стандартного відхилення за групою, Зсунена дисперсія – розрахунок дисперсії за генеральною сукупністю, Незсунена дисперсія – розрахунок дисперсії за групою).

5. Вибрати зі списку Додати підсумки до стовпці для розрахунку проміжних підсумків. Для вибору стовпця необхідно поставити відмітку проти його назви.

Структура списку після виконання команди Підсумки (рис. 7.39) дозволяє переглядати різні частини списку з допомогою кнопок, розміщених в лівому полі аркуша.

1	Місяць	Продавець	Форд	Фіат	ВАЗ	Таврія	Всього
2	Квітень	Бергман	45800	12000	98000	12800	168600
3	Квітень	Іванов	32600	25000	84000	27670	169270
4	Квітень	Карась	67000	11000	73000	50080	201080
5	Квітень Підсумок		145400	48000	255000	90560	538950
6	Травень	Бергман	32000	10000	67800	32880	142680
7	Травень	Іванов	22000	36900	48700	80900	189500
8	Травень	Карась	80000	31400	20000	120000	251400
9	Травень Підсумок		134000	78300	138500	233780	582580
10	Червень	Бергман	100000	99000	12600	34500	246100
11	Червень	Іванов	70000	120000	37770	67900	295670
12	Червень	Карась	50000	58000	58000	23600	189600
13	Червень Підсумок		220000	277000	108370	126000	731370
14	Загальний підсумок		499400	403300	499870	450330	1852900

Рис. 7.39. Дані про продаж автомобілів з проміжними та загальним підсумками

Кнопки у верхній частині поля, визначають кількість рівнів даних що відображаються і використовуються для виконання таких дій:

- 1 – Вивести тільки загальний підсумок;
- 2 – Вивести тільки загальний підсумок і проміжні підсумки;
- 3 – Вивести всі дані.

Кнопки з позначками (+) і (-) призначені для розгортання і згортання окремих груп.

7.4.5. Зведені таблиці

Найбільш потужним засобом обробки даних в Excel є команда **Зведена таблиця**. В результаті її виконання створюється нова таблиця. Зведені таблиці надають користувачу можливості:

- представити інформацію із традиційних списків у більш зручному вигляді;
- для побудови зведених таблиць використати запити до баз даних або інших джерел зовнішніх даних.

Зведена таблиця – це плоска або об'ємна (складається з декількох сторінок або шарів) прямокутна таблиця котра дає можливість виконати складний аналіз великих масивів даних. В такій таблиці поряд із звичайними списками можуть використовуватися дані з іншої зведеної таблиці, а також запити до зовнішніх даних. Для підсумкових розрахунків можна вибирати потрібну функцію зведення, або інший метод обчислення.

Обчислення здійснюють для одного, або декількох полів вихідного списку, визначених як поля даних. Інші поля вихідного списку можуть бути використані для групування даних в рядках і стовпцях зведеної таблиці.

Після виконання команди **Дані/Зведена таблиця** відкривається діалогове вікно майстра зведених таблиць і діаграм (рис. 7.40), котре складається з трьох кроків. Призначення кнопок в нижній частині майстра очевидне.

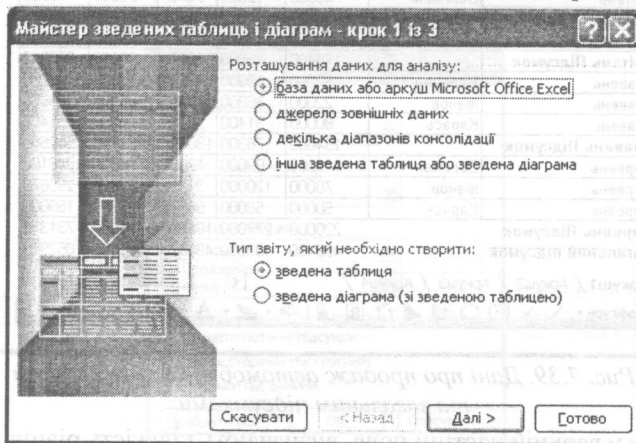


Рис. 7.40. Перший крок Майстра зведених таблиць і діаграм

Створимо зведену таблицю використовуючи дані продаж автомобілів з вибором в якості джерела даних аркуша Excel.

На першому кроці потрібно вибрати тип джерела даних. На другому кроці потрібно вказати діапазон даних (рис. 7.41). Цю процедуру виконують користуючись кнопкою **Огляд** і кнопкою згортання – розгортання

вікна **Діапазон**. У виділеному діапазоні мусять бути заголовки стовпців. Якщо перед запуском майстра курсор знаходився в області списку, то Excel може виділити його автоматично.

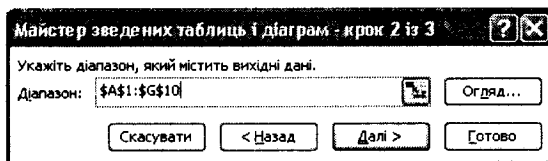


Рис. 7.41. Другий крок Майстра зведених таблиць і діаграм

На третьому кроці (рис. 7.42) визначається місце розташування зведеної таблиці, її структура (опція **Макет**) та параметри (опція **Параметри**).

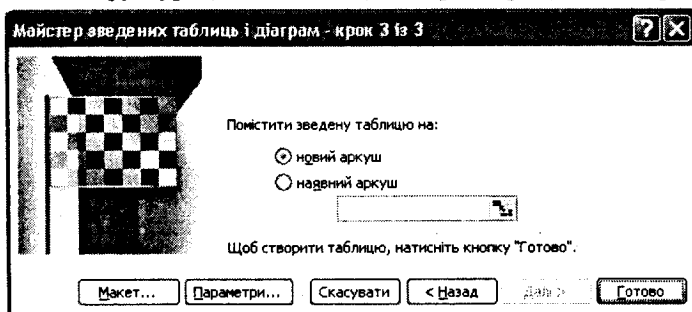


Рис. 7.42. Третій крок Майстра зведених таблиць і діаграм

Структуру зведеної таблиці задають перетягуванням кнопок з назвами полів в області **Рядок**, **Стовпець**, **Сторінка**, **Дані** (рис. 7.43).

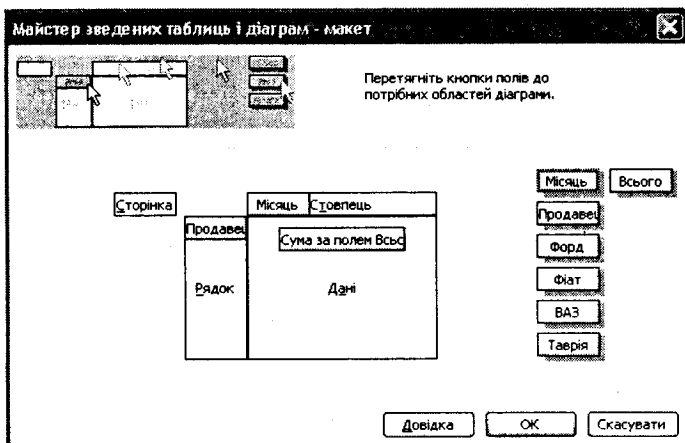


Рис. 7.43. Побудова макету зведеної таблиці

Вибравши опцію **Параметри**, можна замінити деякі параметри зведеної таблиці (рис. 7.44).

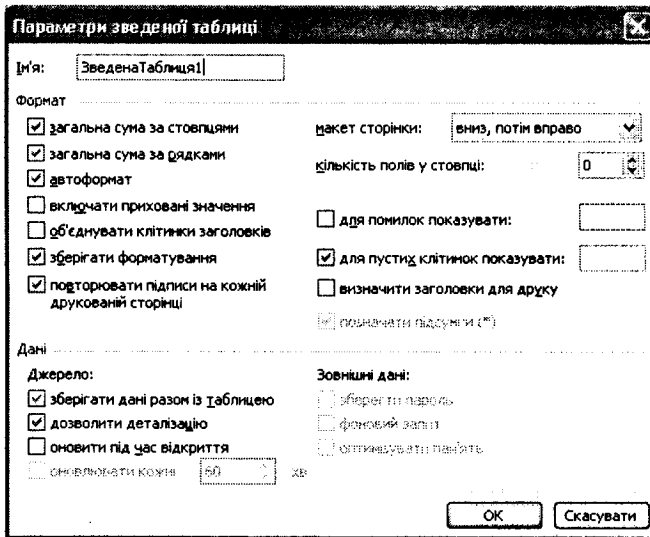


Рис. 7.44. Задавання параметрів зведеної таблиці

Результатом таких дій буде зведена таблиця щомісячної виручки від продажу автомобілів кожним з продавців (рис. 7.45).

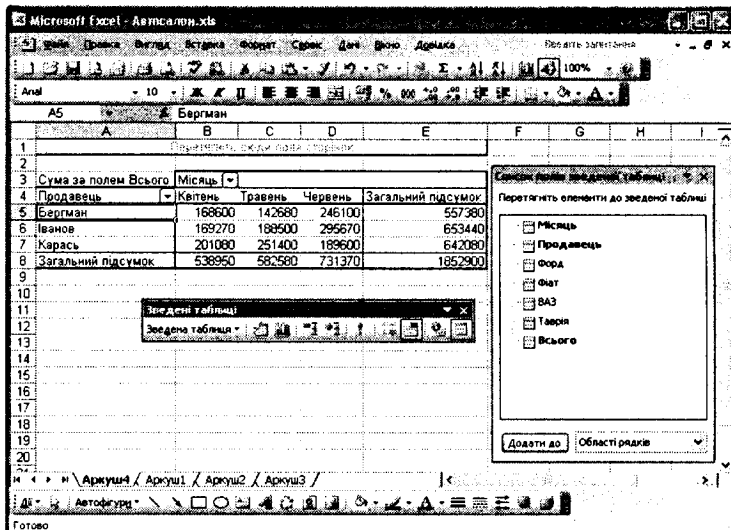


Рис. 7.45. Зведена таблиця щомісячної виручки від продажу автомобілів

7.4.6. Формати файлів баз даних

Базу даних, створену засобами Excel, можна зберегти не тільки у стандартному форматі книги (.xls), а й у звичному для баз даних форматі, наприклад, dbf-файлі. У цьому випадку програма коректно визначає типи полів (текст, числове, дата) та їхні довжини (для тексту) на підставі наявних даних. Для створення dbf-файлу достатньо вибрати значення **DBF 4 (dBASE IV) (*.dbf)** у полі **Тип файлу** діалогового вікна **Збереження документа**, яке викликається командою **Файл/Зберегти як...**

Excel дає змогу також працювати з dbf-файлами та файлами баз даних системи керування базами даних (СКБД) Microsoft Office Access. Для відкриття таких файлів у полі **Тип файлу** діалогового вікна **Відкриття документа** треба, відповідно, вибрати значення **Файли dBase (*.dbf)** або **Бази даних Access (*.mdb; *.mdb)**. Іншим способом відкриття таких файлів є використання команди **Дані/Імпорт зовнішніх даних/Імпорт даних....**

Для побудови запитів до баз даних, створених спеціалізованими СКБД (наприклад, SQL Server, MS Access, Paradox, FoxPro), в Excel є спеціальний засіб – програма MS Query, яку активізують командою **Дані/Імпорт зовнішніх даних/Створити запит....** Перевага використання запитів полягає в тому, що в Excel переноситься лише потрібна (відфільтрована) частина записів з бази даних.

7.5. Імпорт даних з текстових файлів

Потужні засоби програми Excel для аналізу даних та представлення результатів можна використовувати й тоді, коли файли з даними вже є як результат роботи тих чи інших спеціалізованих програмних засобів. Наявність цілої низки конверторів з інших форматів дає змогу імпортувати (за допомогою команди **Файл/Відкрити**) у таблицю Excel найрізноманітніші дані, які мають структурований характер (рис. 7.46).

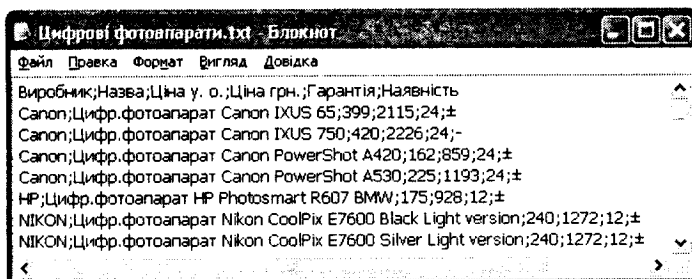


Рис. 7.46. Текстовий файл зі структурованими даними

Перетворення даних переважно відбувається в автоматичному режимі, однак у деяких випадках (наприклад, якщо імпортований файл має текстовий формат) програма не може сама визначити спосіб конвертації і викликає спеціальний програмний засіб *Майстер імпорту текстів*, який працює у режимі діалогу. Для того, щоб дані в рядках тексту можна було впорядкувати за стовпцями, вони повинні бути або відокремлені пробілом, символом табуляції, комою чи іншим спеціальним символом, або міститись у стовпцях фіксованої ширини.

В першому діалоговому вікні власне і визначають спосіб організації даних у файлі, тип кодування та місце початку імпортування (рис. 7.47):

- у рамці **Формат вихідних даних** вибирають потрібне значення опції: з **розділювачами** – дані відокремлені символами, **фіксованої ширини** – поля даних мають указану ширину;
- у полі **Почати імпорт з рядка** зазначають рядок тексту, з якого треба розпочати імпортування даних;
- у полі **Формат файлу** задають спосіб кодування символів тексту.

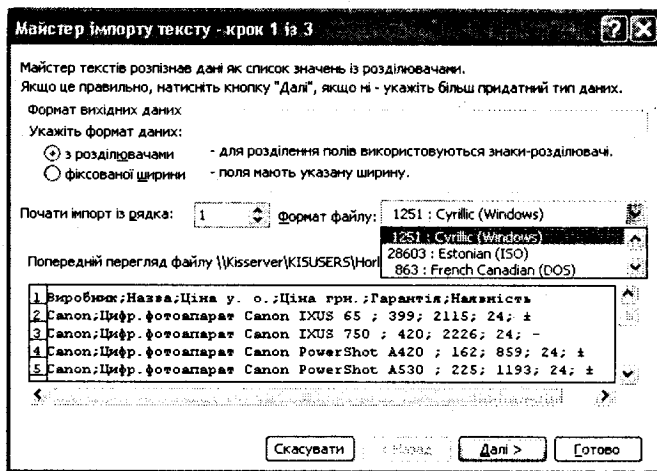


Рис. 7.47. Перший крок Майстра імпорту текстів

Результат перетворення даних у таблицю згідно із заданими параметрами можна спостерігати у нижній частині вікна. Кнопка **Далі** дає змогу перейти до наступного діалогового вікна, а кнопка **Готово** – розпочати імпортування з використанням стандартних значень для незаданих параметрів.

Вигляд другого вікна Майстра імпорту залежить від способу відокремлення даних у текстових рядках. Якщо це зроблено за допомогою символів-розділювачів (рис. 7.48), то:

- у рамці **Розділювачі** зазначають символи, які треба використати для відокремлення даних у рядках;

- у полі **обмежувач рядків** можна задати символи, які обмежують окремі дані (у цьому випадку символ-розділювач між ними вважається елементом тексту).

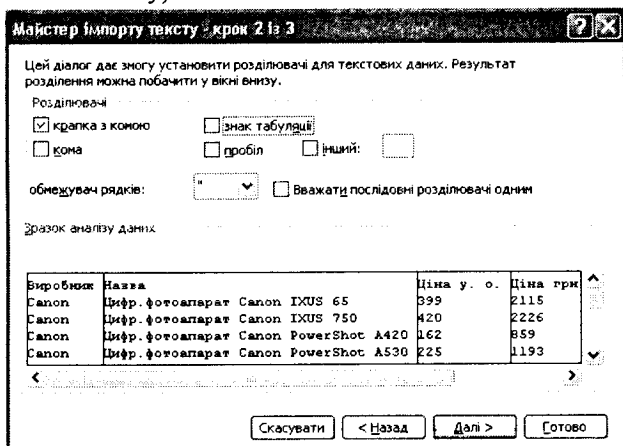


Рис. 7.48. Другий крок Майстра імпорту текстів (з розділювачами)

Якщо ж дані містяться у стовпцях фіксованої ширини, то у другому діалоговому вікні Майстра імпорту можна змінити вигляд таблиці, переміщуючи, додаючи (натисканням мишкою) або вилучаючи (подвійним натисканням мишкою) розділювачі стовпців.

У третьому діалоговому вікні для виділених стовпців можна задати окремі формати даних: загальний, текстовий або дата, а також вилучити зайві стовпці, вибравши значення пропустити стовпець (рис. 7.49).

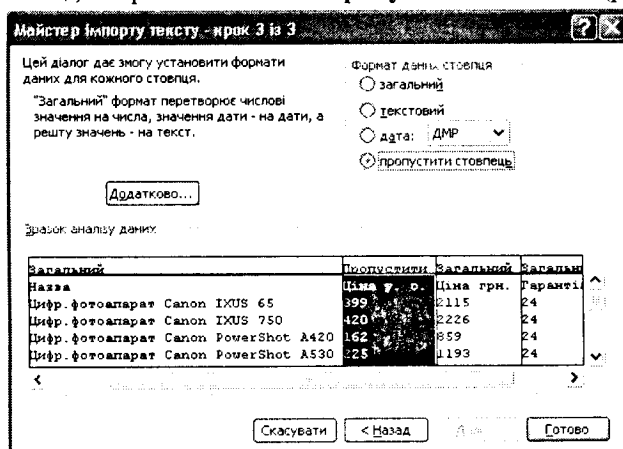
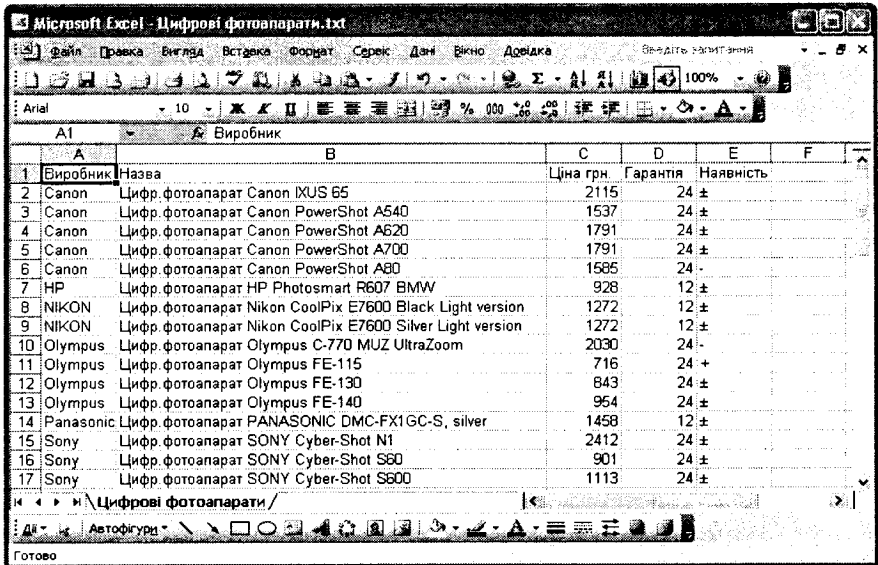


Рис. 7.49. Третій крок Майстра імпорту текстів

У всіх діалогових вікнах кнопка **Назад** дає змогу повернутися до попереднього вікна. Кнопка **Готово** ініціює процес перетворення даних у таблицю Excel (рис. 7.50).



А	В	С	Д	Е	Ф
Виробник	Назва	Ціна грн.	Гарантія	Наявність	
Canon	Цифр. фотоапарат Canon IXUS 65	2115	24 ±		
Canon	Цифр. фотоапарат Canon PowerShot A540	1537	24 ±		
Canon	Цифр. фотоапарат Canon PowerShot A620	1791	24 ±		
Canon	Цифр. фотоапарат Canon PowerShot A700	1791	24 ±		
Canon	Цифр. фотоапарат Canon PowerShot A80	1585	24 ±		
HP	Цифр. фотоапарат HP Photosmart R607 BMW	928	12 ±		
NIKON	Цифр. фотоапарат Nikon CoolPix E7600 Black Light version	1272	12 ±		
NIKON	Цифр. фотоапарат Nikon CoolPix E7600 Silver Light version	1272	12 ±		
Olympus	Цифр. фотоапарат Olympus C-770 MUZ UltraZoom	2030	24 ±		
Olympus	Цифр. фотоапарат Olympus FE-115	716	24 +		
Olympus	Цифр. фотоапарат Olympus FE-130	843	24 ±		
Olympus	Цифр. фотоапарат Olympus FE-140	954	24 ±		
Panasonic	Цифр. фотоапарат PANASONIC DMC-FX1GC-S, silver	1458	12 ±		
Sony	Цифр. фотоапарат SONY Cyber-Shot N1	2412	24 ±		
Sony	Цифр. фотоапарат SONY Cyber-Shot S80	901	24 ±		
Sony	Цифр. фотоапарат SONY Cyber-Shot S800	1113	24 ±		

Рис. 7.50. Текстові дані імпортовано у таблицю Excel

7.6. Ділова графіка

Можливості табличного процесора були б неповними, якби вони обмежувалися здатністю швидко виконувати великі обсяги обчислень. У багатьох випадках зображення даних у вигляді діаграм або графіків дає змогу ефективніше проаналізувати підготовлені таблиці та виконані розрахунки.

Для створення діаграми потрібно мати хоча б одну послідовність даних певного (числового) типу. Більшість діаграм мають дві осі: горизонтальну **X** та вертикальну **Y**. Вісь **X**, як звичайно, є віссю категорій, її використовують для нанесення визначень категорій або часових інтервалів. Уздовж осі **Y** відкладають значення величин з вибраної послідовності даних.

Діаграми в Excel створюють у режимі діалогу зі спеціальним програмним засобом *Майстер діаграм* (рис. 7.51), який викликають командою **Вставка/Діаграма** або піктограмою **Майстер діаграм** панелі **Стандартна** після виокремлення клітинок таблиці, що містять потрібні послідовності даних. У першому діалоговому вікні визначають загальний вигляд діаграми. На сторінці **Стандартні** або **Нестандартні**:

- у списку **Тип** вибирають тип діаграми;
- у віконці **Вид** – різновид вибраного типу;

- натискають на кнопку **Перегляд результату** і утримують її для перегляду реальної діаграми у вікні **Зразок**.

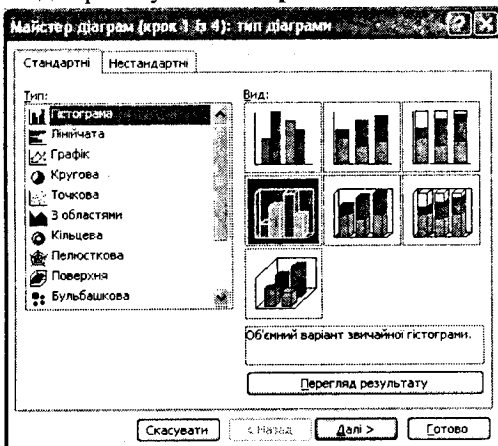


Рис. 7.51. Перший крок Майстра діаграм

Кнопки **Назад**, **Далі** та **Готово** виконують наступні функції: **Назад** – повернутися один крок назад щоб вибрати інший варіант; **Далі** – перейти до наступного етапу; **Готово** – дозволяє майстрові виконати всі наступні кроки, тобто задати всі параметри діаграми за замовчуванням.

У другому діалоговому вікні також є дві сторінки опцій. На сторінці **Діапазон даних** (рис. 7.52):

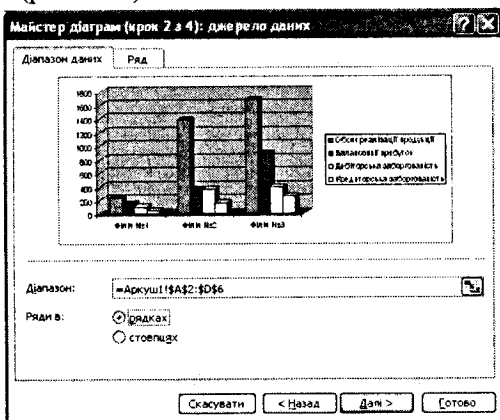


Рис. 7.52. Другий крок Майстра діаграм (Діапазон даних)

- у полі **Діапазон** задають (модифікують) наявний діапазон (набір діапазонів);
- вибирають потрібне значення опції **Ряди в:** **рядках** – послідовності даних містяться у рядках таблиці, **стовпцях** – у стовпцях.

На сторінці **Ряд** (рис. 7.53):

- у списку **Ряд** вибирають послідовність даних, для якої у полі **Ім'я** можна задати ім'я (або вказати на клітинку, звідки його взяти), а у полі **Значення** – модифікувати діапазон клітинок;
- кнопка **Додати** дає змогу додати порожню послідовність даних, а **Видалити** – вилучити вибрану послідовність;
- у полі **Підписи осі X** можна задати підписи до категорій (груп значень) або вказати (зокрема, мишкою) клітинки, звідки ці підписи братимуться.

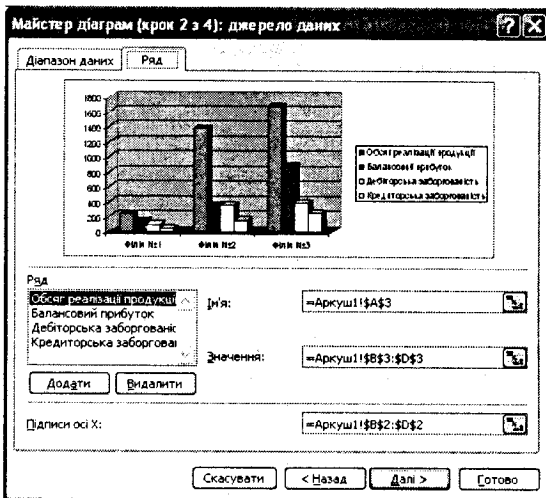


Рис. 7.53. Другий крок Майстра діаграм (Ряд)

Третє діалогове вікно містить шість сторінок: **Назви**, **Осі**, **Сітка**, **Легенда**, **Підписи даних** і **Таблиця даних**.

На сторінці **Назви** (рис. 7.54) задають надписи:

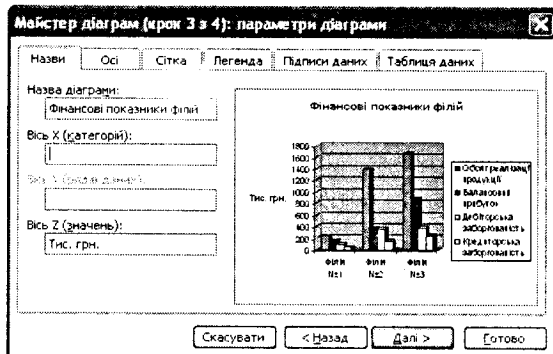


Рис. 7.54. Третій крок Майстра діаграм (Назви)

- у полі **Назва діаграми** – до всієї діаграми;
- у полях **Вісь X**, **Вісь Y** та **Вісь Z** – до відповідних осей.

Сторінка **Осі** (рис. 7.55) призначена для візуалізації або приховання (шляхом увімкнення/вимкнення відповідних опцій) значень та надписів головних осей діаграми, причому для осі категорій X можна вибрати три способи відображення:

- **автоматична** – відображаються позначки за замовчуванням;
- **категорії** – у ролі позначок використовують назви категорій;
- **вісь часу** – позначками є значення часу (ціна поділки осі X дорівнюватиме певному проміжку часу). Останній спосіб використовують, якщо дані діаграми містять дати.

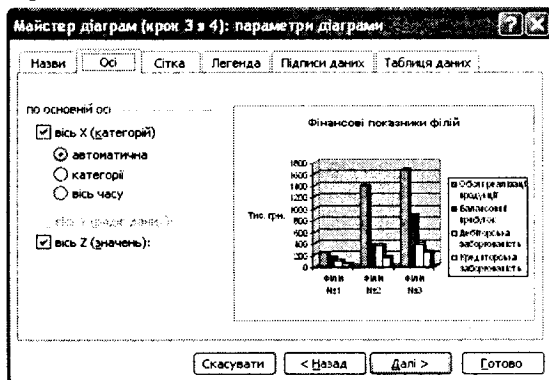


Рис. 7.55. Третій крок Майстра діаграм (Осі)

На сторінці **Сітка** для кожної осі у відповідних рамках можна задати режим візуалізації на діаграмі основних (**основні лінії**) та проміжних (**проміжні лінії**) ліній сітки (рис. 7.56).

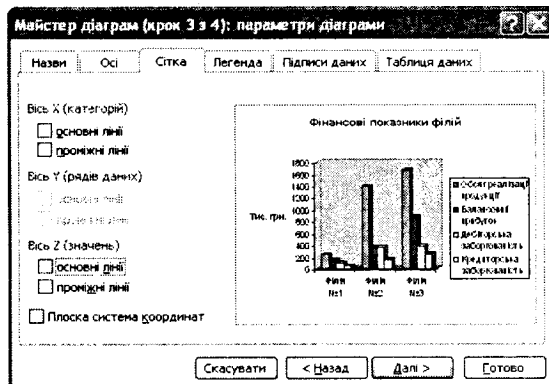


Рис. 7.56. Третій крок Майстра діаграм (Сітка)

Щоб відобразити або приховати на діаграмі легенду, треба на сторінці **Легенда** увімкнути або вимкнути опцію **Додати легенду**. Легенду додають до діграми у тому випадку, коли на діаграмі відображаються різні ряди даних. Задавши на другому кроці майстра побудови діаграми назву кожному ряду даних у легенді замість написів Ряд1 і т.д., отримують підписи для кожного ряду. У нашому прикладі це написи “Обсяг реалізації продукції”, “Балансовий прибуток” та інші. У рамці **Розташування** можна змінити місцезнаходження легенди на робочому полі діаграми (рис. 7.57).

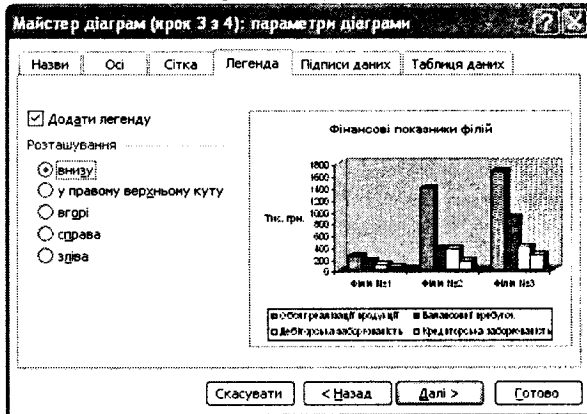


Рис. 7.57. Третій крок Майстра діаграм (Легенда)

Елементи послідовностей даних на діаграмі можуть бути підписані своїми значеннями, якщо на сторінці **Підписи даних** відмітити потрібні значення опції **Включити до підписів**. Опція **Ключ легенди** дає змогу додати до підписів кольоровий символ з легенди (рис. 7.58).

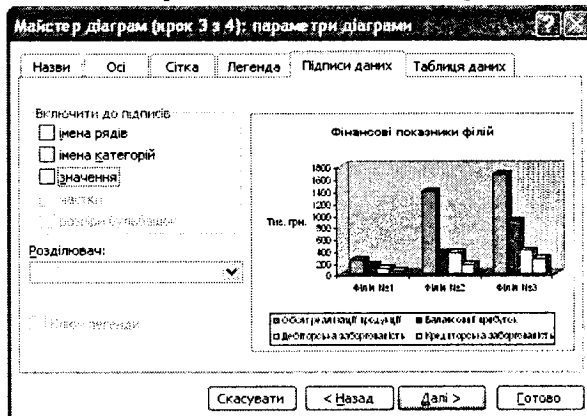


Рис. 7.58. Третій крок Майстра діаграм (Підписи даних)

І, нарешті, на сторінці **Таблиця даних** можна приєднати до діаграми таблицю її вихідних даних, увімкнувши опцію **Таблиця даних**. За допомогою опції **Ключ легенди** у клітинки таблиці можна додати кольорові символи з легенди (рис. 7.59).

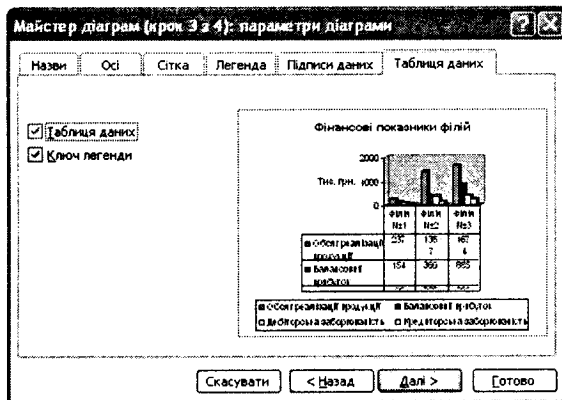


Рис. 7.59. Третій крок Майстра діаграм (Таблиця даних)

Останнє діалогове вікно *Майстра діаграм* дає змогу визначити місце розміщення діаграми у книзі за допомогою відповідних значень опції **Розмістити діаграму на аркуші: окремо** – в окремому аркуші, назву якого треба задати у полі праворуч, **на явному** – на одному з наявних аркушів книги, вибраному в полі праворуч (рис. 7.60).

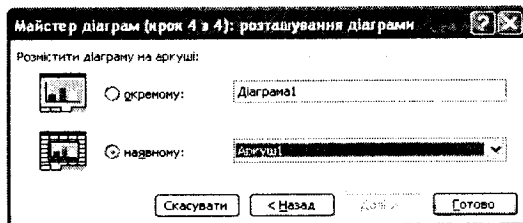


Рис. 7.60. Четвертий крок Майстра діаграм

Після натискання на кнопку **Готово** діаграма буде вставлена у визначене місце книги (рис. 7.61).

Для редагування діаграми потрібно виокремити один із її елементів (область діаграми, назва діаграми, область побудови діаграми, назви осей, легенду тощо) і скористатись відповідними командами меню **Діаграма**.

Натискання над одним із елементів діаграми правої кнопки мишки дає змогу викликати контекстне меню, після виконання першої команди якого з'явиться діалогове вікно для форматування цього елемента.

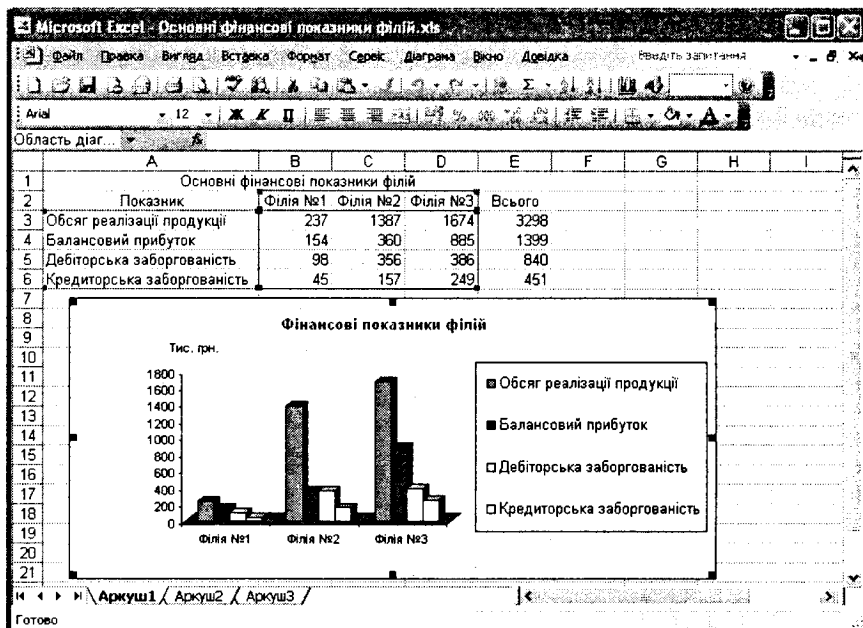


Рис. 7.61. Результат роботи Майстра діаграм

Можливості Excel щодо редагування діаграм настільки великі, що їх описові можна присвятити окреме видання. Тому назвемо лише деякі важливі моменти роботи з діаграмами:

- будь-які зміни значень у клітинках таблиці, які використані для побудови діаграми, миттєво відображаються на самій діаграмі;
- зміна значення будь-якого елемента послідовності даних на діаграмі спричиняє автоматичну зміну значення у відповідній клітинці таблиці;
- додати до діаграми нову послідовність даних можна транспортуванням за допомогою мишки виділених у таблиці клітинок в область діаграми;
- послідовності даних необов'язково повинні розміщуватися в сусідніх рядках чи стовпцях; у цьому випадку для їх виокремлення треба утримувати натиснутою клавішу **Ctrl**.

7.7. Друкування аркуша

Перш ніж друкувати аркуш, потрібно задати параметри друкованої сторінки. Це роблять за допомогою команди **Файл/Параметри сторінки...**, яка відкриває однойменне діалогове вікно (рис. 7.62) з чотирма сторінками опцій: **Сторінка**, **Поля**, **Колонтитули** та **Аркуш**.

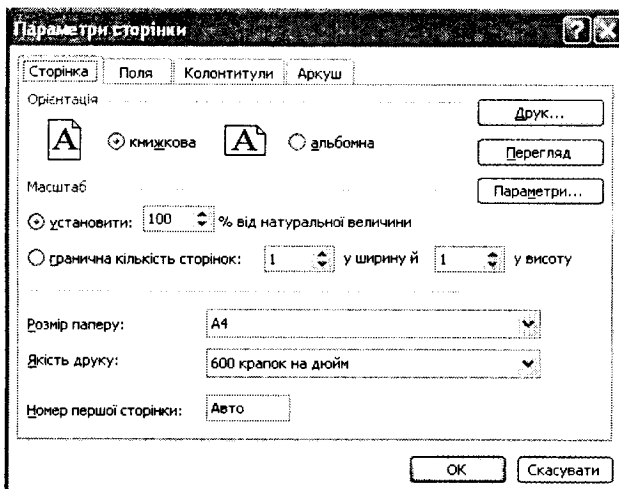


Рис. 7.62. Налаштування параметрів сторінки (Сторінка)

Сторінка опцій **Сторінка** призначена для задання загальних параметрів:

- у рамці **Орієнтація** задають орієнтацію друкованої сторінки: **книжкова** – поздовжня, **альбомна** – поперечна;
- у рамці **Масштаб** можна змінити масштаб друкованого фрагмента (опція **установити** із зазначенням у полі праворуч нових розмірів у відсотках до початкових) або задати кількість сторінок у ширину та висоту, на яких треба розмістити друковану ділянку, відповідно її змасштабувавши (опція **гранична кількість сторінок** із зазначенням потрібної кількості у полях праворуч);
- у полі **Розмір паперу** визначають розміри сторінки, на яку виводиться друкований фрагмент;
- у полі **Якість друку** – якість друкування;
- у полі **Номер першої сторінки** задають номер для першої сторінки, що друкуватиметься;
- за допомогою кнопки **Друк...** відкривають однойменне діалогове вікно для задавання параметрів друкування і власне друкування;
- кнопка **Перегляд** дає змогу переглянути сторінки перед їх друкуванням;
- кнопка **Параметри...** призначена для зміни параметрів принтера.

На сторінці **Поля** задають відступи від країв друкованої сторінки (рис. 7.63):

- у полях **ліве**, **праве**, **верхнє** та **нижнє** – відповідно ліворуч, праворуч, зверху та знизу;
- у полях **верхнього колонтитула** та **нижнього колонтитула** – відповідно зверху до верхнього та знизу до нижнього колонтитулів.

У рамці **Центрувати на сторінці** визначають режими центрування друкованої ділянки на сторінці по горизонталі (опція **горизонтально**) та вертикалі (опція **вертикально**).

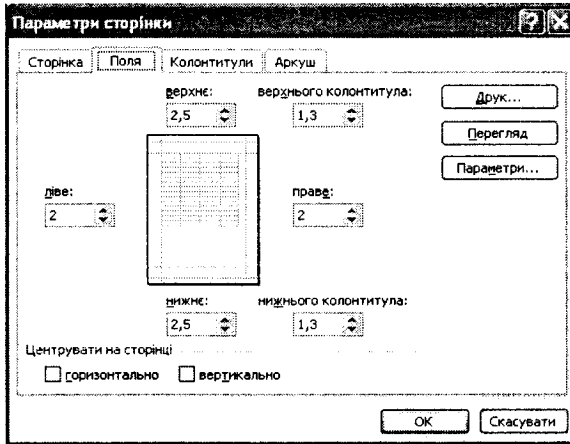


Рис. 7.63. Налаштування параметрів сторінки (Поля)

Сторінка **Колонтитули** призначена для роботи з колонтитулами (рис. 7.64).

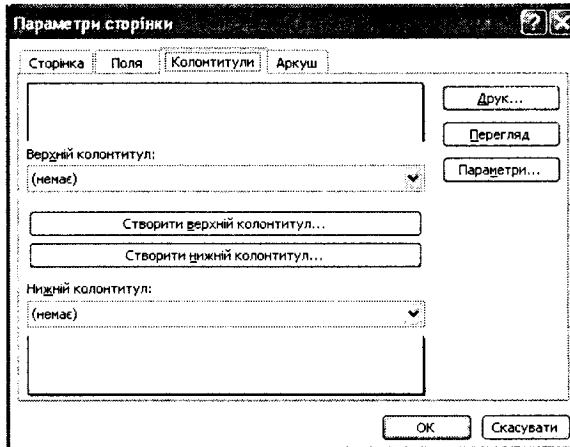


Рис. 7.64. Налаштування параметрів сторінки (Колонтитули)

Застосувати один із вбудованих колонтитулів можна за допомогою списків **Верхній колонтитул** для розміщення у верхній частині сторінки або **Нижній колонтитул** для розміщення внизу сторінки, вибравши потрібний і спостерігаючи результат у вікнах зверху або внизу. Значення **немає** дає змогу відмовитись від колонтитула.

Для редагування колонтитула або створення нового (якщо вибрано значення немає) використовують кнопки **Створити верхній колонтитул** та **Створити нижній колонтитул**. Після натискання на них активізується діалогове вікно (рис. 7.65), де у полях **Зліва**, **У центрі** та **Справа**, які відповідають лівій, центральній та правій ділянці колонтитула, треба увести потрібний текст та вставити поля, користуючись піктограмами у центральній частині вікна.

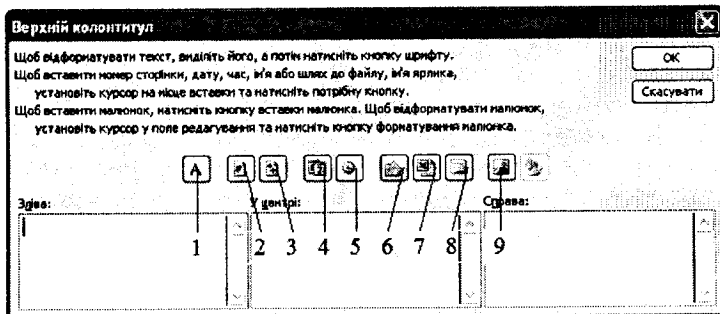


Рис. 7.65. Піктограми: 1 – параметри шрифту; 2 – вставка номера сторінки; 3 – вставка кількості сторінок; 4 – вставка дати; 5 – вставка часу; 6 – вставка шляху та імені файлу; 7 – вставка імені файла; 8 – вставка імені аркуша; 9 – вставка графічного зображення

На сторінці **Аркуш** (рис. 7.66) визначають, що саме треба друкувати:

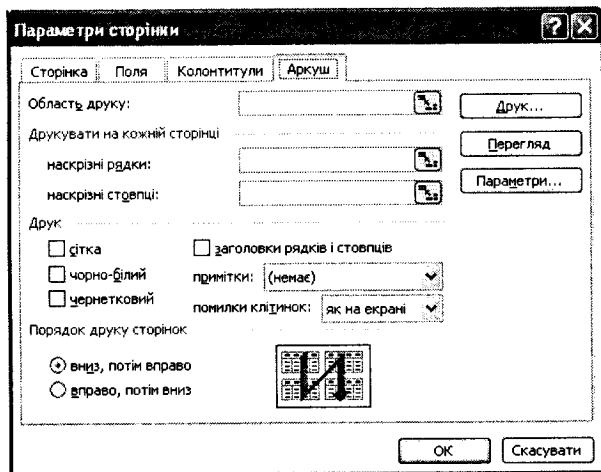


Рис. 7.66. Налаштування параметрів сторінки (Аркуш)

- у полі **Область друку** задають друковану ділянку – набір діапазонів клітинок, вміст яких друкуватиметься;

- у рамці **Друкувати на кожній сторінці** – рядки (поле **наскрізні рядки**) та стовпці (поле **наскрізні стовпці**), які повинні бути надруковані на кожній сторінці;
- у рамці **Друк** задають додаткову інформацію для друкування:
 - сітка** – друкувати розмежувальні лінії таблиці;
 - чорно-білий** – друкувати таблицю як чорно-білу;
 - чернетковий** – друкувати у режимі чернетки (без графічних об'єктів та сітки);
 - заголовки рядків і стовпців** – друкувати всі заголовки;
 - примітки** – режими друкування приміток;
 - помилки клітинок** – режими друкування помилок;
- у рамці **Порядок друку сторінок** задають спосіб виведення багато-сторінкового документа: **вниз, потім вправо** – зверху вниз і потім зліва направо, **вправо, потім вниз** – зліва направо і потім зверху вниз.

Після задавання параметрів сторінки таблицю можна друкувати. Це роблять за допомогою команди **Файл/Друк...** або кнопки **Друк** вікна **Параметри сторінки**, які відкривають однойменне діалогове вікно (рис. 7.67).

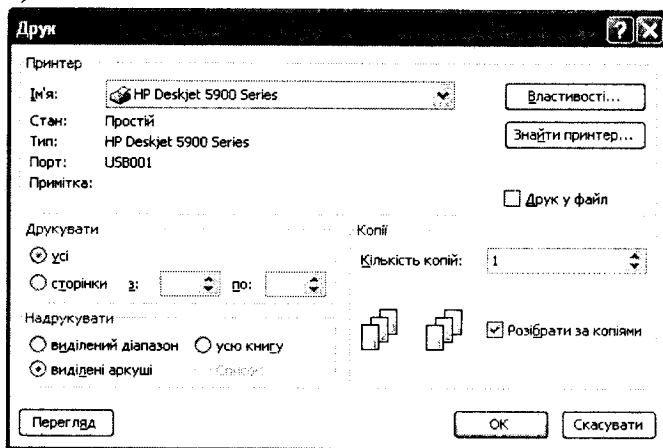


Рис. 7.67. Налаштування параметрів друку

- у рамці **Принтер** у полі **Ім'я** вибирають один із встановлених у системі принтерів, параметри якого можна змінити за допомогою кнопки **Властивості...**, і вмикають опцію **Друк у файл** у разі потреби спрямувати друкування у файл (потім цей файл можна роздрукувати, наприклад, командою DOS `copy <ім'я файлу> prn` навіть на комп'ютері, де немає Excel, але є принтер із зазначеним ім'ям);
- задають потрібне значення опції **Друкувати**: **усі** – всю друковану ділянку, **сторінки** – зазначений у полях **з і по** діапазон сторінок;

- задають потрібне значення опції **Надрукувати: виділений діапазон** – тільки виділені клітинки, **виділені аркуші** – виділені (з утримуванням клавіш **Ctrl** або **Shift**) аркуші книги, **усю книгу** – всі аркуші книги;
- у рамці **Копії** задають кількість копій (поле **Кількість копій**), вмикаючи у разі потреби опцію **Розібрати за копіями**, яка дає змогу друкувати одна за одною копії всього документа (інакше послідовно друкуються копії окремих сторінок).

Кнопка **Перегляд** призначена для попереднього перегляду документа у вигляді, найбільш наближеному до того, який буде одержано в результаті друкування.

7.8. Макрокоманди

Макрокоманда (макрос) в Excel – це набір команд програми, записаних мовою **Visual Basic**, які можна виконувати в автоматичному режимі. Макрокоманди є чудовим засобом прискорити виконання однотипних операцій. Їх можна використовувати для заповнення таблиці серіями даних однакової структури, побудови однотипних діаграм, імпорту в Excel багатьох однотипних файлів (наприклад, варіантів розрахунків) тощо.

Найпростішим і наочним способом створення макрокоманди є записування послідовності виконуваних команд за допомогою макрорекордера. Для запуску цього програмного засобу потрібно виконати команду **Сервіс/Макрос/Почати запис**, після чого відкриється діалогове вікно **Запис макросу** (рис. 7.68), де потрібно:

- у полі **Ім'я макросу** задати ім'я макросу;
- у полі **Сполучення клавіш** – "гарячу" комбінацію клавіш для швидкого запуску макрокоманди;
- у полі **Зберегти в** – місце для збереження макросу: у спеціальній книзі **Personal.xls** (**Особиста книга макросів**), в активній (**Ця книга**) або окремій книзі (**Нова книга**);
- у полі **Опис** – коментар.

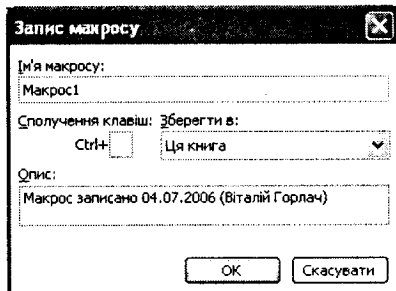


Рис. 7.68. Запис макросу

Процес записування макрокоманди розпочинається після натискання на кнопку **ОК** і завершується виконанням команди **Сервіс/Макрос/Зупинити запис** або натисканням на однойменну піктограму службової панелі, яка візуалізується на час записування. Запуск макрокоманди здійснюють через комбінацію клавіш або кнопкою **Виконати** у діалоговому вікні **Макрос** (рис. 7.69).

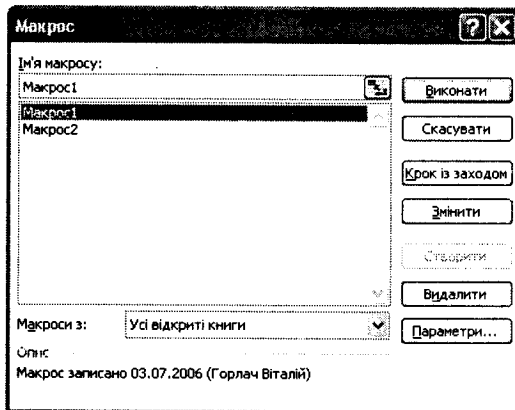


Рис. 7.69. Діалогове вікно для роботи з макросами

Макрокоманди можна редагувати або створювати самостійно шляхом написання команд мовою Visual Basic. Для цього використовують відповідно кнопки **Змінити** та **Створити** (яка стає доступною після задавання нового імені макросу) згаданого вище діалогового вікна **Макрос**.

Для досвідчених користувачів існує ще один спосіб. Виконанням команди **Сервіс/Макрос/Редактор Visual Basic** запустити редактор Visual Basic і створити процедуру (Procedure), задавши її тип (Sub, Function чи Property) і межі дії (Public чи Private); користувацьку форму (UserForm); модуль (Module) і наповнити його текстом програми; модуль класу (Class Module).

Додатки

1. Головне меню програми Excel

Команди меню **Файл**

Створити... – створення нової книги.

Відкрити... – відкриття наявної книги.

Закрити – завершення роботи з книгою.

Зберегти – збереження книги.

Зберегти як... – збереження книги під іншим іменем.

Зберегти як веб-сторінку... – збереження книги у форматі HTML.

Зберегти робочу область... – збереження інформації про всі відкриті книги.

Пошук файлів... – запуск вікна швидкого, звичайного або розширеного пошуку файлів.

Дозвіл – організація дозволів на роботу з документом:

Необмежений доступ – дозвіл на роботу з книгою без обмежень;

Не розповсюджувати... – задання обмежень на розповсюдження книги;

Обмежити дозвіл... – обмеження дозволу на роботу з книгою.

Параметри сторінки... – задавання параметрів сторінки для друкування.

Область друку – задавання діапазону друкування аркуша:

Установити – визначення ділянки аркуша, призначеної для друкування;

Прибрати – відміна ділянки аркуша для друкування.

Попередній перегляд – перегляд документа перед друкуванням.

Друк – друкування книги, аркуша або виділеного фрагмента.

Надіслати – оперативне розсилання книги:

Повідомлення – відправлення книги або поточного аркуша як повідомлення електронної пошти;

Повідомлення (на рецензію)... – відправлення книги як вкладення у повідомлення електронної пошти із проханням прорецензувати;

Повідомлення (як вкладення)... – відправлення книги як вкладення у повідомлення електронної пошти;

За маршрутом... – відправлення книги кільком адресатам заданим маршрутом;

Папка Exchange... – відправлення документа у вказану папку програми Microsoft Office Outlook;

Учасник мережних зборів – відправлення спільної книги учасникам мережних зборів їх організатором;

Факс користувачеві служби факсів Інтернету... – відправлення книги як факсу користувачеві зазначеної служби.

Властивості – задавання характеристик книги.

Вихід – завершення роботи з програмою Excel.

Команди меню Правка

Скасувати – відміна останньої операції.

Повторити – повторення останньої відміненої операції.

Вирізати – вилучення виділеного блоку в буфер.

Копіювати – копіювання блоку в буфер.

Буфер обміну Office... – відкривання області завдань **Буфер обміну** для роботи з його вмістом.

Вставити – уставлення вмісту буфера в аркуш.

Спеціальна вставка... – уставлення вмісту буфера особливим чином.

Вставити як гіперпосилання – уставлення вмісту буфера як гіперпосилання.

Заповнити – автоматичне заповнення виділеної ділянки:

Вниз – вмістом першого рядка вниз уздовж ділянки;

Праворуч – вмістом лівого стовпця праворуч уздовж ділянки;

Вгору – вмістом останнього рядка вгору уздовж ділянки;

Ліворуч – вмістом правого стовпця ліворуч уздовж ділянки;

По аркушах... – вмістом виділеної ділянки активного аркуша (якщо вибрано кілька аркушів);

Прогресія... – вниз або праворуч з певною закономірністю;

Вирівняти – щільне заповнення текстом виділеної ділянки.

Очистити – вилучення:

Усі – всього вмісту клітинок разом із заданими форматами;

Формати – форматів;

Вміст – вмісту;

Примітки – приміток.

Видалити... – вилучення клітинок, рядків або стовпців.

Видалити аркуш – вилучення аркуша.

Перемістити/копіювати аркуш... – переміщення або копіювання аркуша.

Знайти... – пошук за зразком.

Замінити... – заміна за зразком.

Перейти... – виокремлення клітинок за заданими умовами.

Зв'язки... – редагування зв'язків.

Об'єкт – редагування об'єкта.

Команди меню **Вигляд**

Звичайний – звичайний режим перегляду таблиці.

Розмітка сторінки – посторінковий режим перегляду таблиці.

Область завдань – увімкнення/вимкнення області завдань.

Панелі інструментів – увімкнення/вимкнення відображення панелей інструментів.

Рядок формул – увімкнення/вимкнення відображення рядка формул.

Рядок стану – увімкнення/вимкнення відображення рядка статусу.

Колонтитули... – редагування колонтитулів.

Примітки – увімкнення/вимкнення режиму відображення приміток.

Подання... – створення різних способів відображення таблиці.

На весь екран – увімкнення/вимкнення повноекранного режиму перегляду.

Масштаб... – вибір масштабу відображення документа.

Команди меню **Вставка**

Клітинки... – вставлення клітинки.

Рядки – вставлення рядка.

Стовпці – вставлення стовпця.

Аркуш – вставлення нового аркуша в книгу.

Діаграма... – вставлення діаграми.

Символ... – вставлення відсутніх на клавіатурі та спеціальних символів.

Розрив сторінки – вставлення кінця сторінки.

Функція... – вставлення функції.

Ім'я – операції з іменами клітинки або групи клітинок:

Надати – визначення нового імені;

Вставити – вставлення наявного імені в формулу;

Створити – створення імені з наявного тексту;

Застосувати – заміна адреси на наявне ім'я;

Заголовки діапазонів – створення імен на базі виділених діапазонів.

Примітка – вставлення примітки.

Малюнок – вставлення ілюстрації:

Картинки... – з графічної бібліотеки ClipArt;

З файлу... – з файлу;

Зі сканера або файлу... – безпосередньо зі сканера або камери;

Автофігури – з колекції фігур заданих типів;

Об'єкт WordArt... – як об'єкта WordArt;

Організаційна діаграма – деревовидної організаційної діаграми.

Організаційна діаграма... – вставлення організаційної діаграми заданого типу.

Об'єкт – вставлення об'єкта.

Гіперпосилання – вставлення гіперпосилання.

Команди меню **Формат**

Клітинки... – форматування виділених клітинок.

Рядок – форматування виділених рядків:

Висота... – задавання висоти рядка;

Автодобр висоти – автоматичне підбирання висоти рядка;

Приховати – приховання рядків;

Відобразити – відміна режиму **Приховати**.

Стовпець – форматування виділених стовпців:

Ширина... – задавання ширини;

Автодобр ширини – автоматичне підбирання ширини;

Приховати – приховання стовпців;

Відобразити – відміна режиму **Приховати**;

Стандартна ширина... – задання стандартної ширини стовпців у аркуші.

Аркуш – задавання/зміна параметрів аркуша:

Перейменувати – зміна назви;

Приховати – приховання аркуша;

Відобразити... – відміна режиму **Приховати**;

Фон... – вставлення ілюстрації як тла для аркуша;

Колір ярлика... – задавання кольору ярлика для аркуша.

Автоформат... – автоформатування виділеної ділянки аркуша.

Умвне форматування... – форматування виділених клітинок за виконання певних умов.

Стиль... – робота зі стилями.

Команди меню **Сервіс**

Орфографія... – перевірка орфографії.

Довідкові матеріали... – відкриття однойменної області завдань для пошуку потрібної довідки.

Контроль помилок... – пошук помилок у даних, списках та формулах.

Спільна робоча область... – увімкнення режиму спільного користування даними у мережі.

Доступ до книги – керування доступом до книги.

Виправлення – робота з виправленнями:

Виділити виправлення... – виокремлення змін, зроблених у аркуші;

Прийняти/відхилити виправлення... – пошук виправлень з можливістю їх підтвердження або відхилення.

Порівняти й об'єднати книги... – порівняння та об'єднання поточної та вибраної книги.

Захист – встановлення захисту від внесення змін:

Захистити аркуш... – для аркуша;

Дозволити зміну діапазонів... – за винятком вказаних діапазонів;

Захистити книгу... – для книги;

Захистити книгу та надати спільний доступ... – для книги спільного користування у мережі.

Спільна робота – організація спільної роботи над документом через Microsoft Windows NetMeeting;

Почати збори – відкривання незапланованих зборів у мережі;

Веб-обговорення – спільне обговорення документа з використанням однойменної панелі інструментів.

Підбір параметра... – підбирання значень для одержання бажаного результату внаслідок обчислень за формулою.

Сценарій... – створення та керування сценаріями аналізу даних.

Залежності формул – перевірка даних:

Впливаючі клітинки... – графічне відображення клітинок, значення яких впливає на значення активної клітинки;

Залежні клітинки... – графічне відображення клітинок, значення яких залежить від значення активної клітинки;

Джерело помилки – графічне відображення клітинок, значення яких може спричинити помилку;

Прибрати всі стрілки – відміна всіх графічних відображень;

Обчислити формулу – покрокове обчислення формули в окремому діалоговому вікні;

Показати вікно контрольного значення – встановлення спостереження за формулами та їх результатами у окремому вікні;

Режим перевірки формул – увімкнення режиму відображення у клітинках формул;

Панель залежностей – виклик панелі інструментів перевірки залежностей.

Макрос – робота з макросами:

Макроси... – виклик діалогового вікна для роботи з макросами;

Почати запис... – початок запису нового макросу;

Безпека... – задання рівня безпеки під час запуску макросу;

Редактор Visual Basic – відкривання вікна редактора Microsoft Visual Basic для роботи з макросами;

Редактор сценаріїв – відкривання вікна редактора Microsoft Script Editor для роботи з веб-сценаріями.

Надбудови... – виклик менеджера додаткових можливостей Excel.

Параметри автозаміни... – вибір параметрів автозаміни, автоформату при вводі даних та роботи з старт-тегами.

Настройка... – налаштування панелей інструментів та пунктів меню.

Параметри... – налаштування параметрів Excel.

Команди меню **Дані**

Сортування – сортування даних.

Фільтр – фільтрування даних:

Автофільтр – автоматичне створення фільтра;

Відобразити все – відмінити фільтр;

Розширений фільтр – фільтрування за критеріями користувача.

Форма... – автоматичне створення діалогового вікна для уведення та редагування даних.

Підсумки... – створення підсумкових клітинок.

Перевірка... – задавання допустимих значень для виділених клітинок.

Таблиця підстановки... – створення таблиці підстановки.

Текст за стовпцями... – розділення тексту на стовпці.

Консолідація... – консолідація числових даних.

Група й структура – робота з структурою даних – групами і підсумками:

Приховати подробиці – приховання групи рядків або стовпців, що містять деталізовані дані;

Відобразити подробиці – відображення прихованих груп рядків або стовпців;

Групувати... – групування виділених клітинок;

Розгрупувати... – розгруповання групи клітинок;

Створення структури – автоматичне створення структури даних;

Видалити структуру – вилучення структури;

Настройка... – задавання параметрів структури.

Зведена таблиця... – створення підсумкової таблиці.

Імпорт зовнішніх даних – приєднання зовнішніх даних:

Імпорт даних... – запуск майстра підключення до зовнішнього джерела даних;

Створити веб-запит... – запуск наявного запиту в Інтернеті;

Створити запит... – запуск наявного запиту до зовнішньої бази даних;

Змінити запит... – модифікація наявного запиту;

Властивості діапазону даних... – задавання параметрів даних, одержаних у результаті виконання запиту;

Параметри – задавання додаткових параметрів для складних запитів.

Список – робота зі списками даних:

Створити список... – створення списку даних;

Змінити розмір списку... – зміна діапазону даних існуючого списку;

Рядок підсумків – увімкнення/вимкнення відображення рядка підсумків;

Перетворити на діапазон – ліквідація списку як окремого об'єкту;

Опублікувати список... – розміщення списку в мережі на вузлі SharePoint Services;

Переглянути список на сервері – перегляд списку, опублікованого на вузлі SharePoint;

Скасувати зв'язування списку – скасування зв'язку списку з вузлом SharePoint;

Синхронізувати список – внесення локальних змін у список на вузлі SharePoint та навпаки;

Скасувати зміни та оновити – видалення локальних змін у списку та завантаження поточної копії списку з вузла SharePoint;

Приховувати межі неактивних списків – увімкнення/вимкнення відображення меж неактивних списків.

XML – робота із зовнішніми даними у xml-форматі:

Імпорт... – імпорт даних з XML-файлів;

Експорт... – експорт даних у формат XML;

Оновити XML-дані – оновлення даних, що зберігаються у;

Джерело XML... – відкриття області завдань для роботи з картами XML;

✦ **Властивості карти XML...** – налаштування властивостей карти XML;

Змінити запит... – модифікація наявного запиту;

Пакети розширення XML... – приєднання пакетів розширення XML.

Оновити дані – перерахунок підсумкової таблиці.

Команди меню **Діаграма** (тільки для активної діаграми)

Тип діаграми... – зміна типу діаграми.

Вихідні дані... – модифікація вихідних даних діаграми.

Параметри діаграми... – зміна стандартних параметрів вибраного типу діаграми.

Розташування... – зміна місця розміщення діаграми.

Додати дані... – додавання вмісту виокремлених клітинок до вихідних даних діаграми.

Додати лінію тренду... – додавання та зміна ліній тренду.

Об'ємний вигляд... – задання кута перегляду об'ємної діаграми.

Команди меню **Вікно**

Нове – відкриття нового дочірнього вікна.

Розташувати... – упорядкування дочірніх вікон у головному вікні програми за власним розсудом.

Порівняти поруч з... – розташування поруч дочірніх вікон у головному вікні програми з метою їх порівняння.

Приховати – увімкнення режиму приховання книги.

Відобразити... – відміна режиму **Приховати**.

Розділити – розділення аркуша на частини.

Закріпити області – фіксація частин аркуша.

Команди меню **Довідка**

Довідка: Microsoft Excel – виклик довідкової системи Excel.

Показати помічника – виклик "помічника" для надання корисних порад.

Microsoft Office Online – підключення до домашньої сторінки веб-вузла Microsoft Office Online.

Зверніться до нас – підключення до сторінки підтримки користувачів веб-вузла Microsoft Office Online.

Перевірити оновлення – перевірка наявності оновлень програми на веб-вузлі Microsoft.

Знайти й відновити... – увімкнення автоматичного пошуку та виправлення помилок в усіх файлах Microsoft Office.

Активувати продукт... – активація ліцензії на програму.

Параметри відзвигів користувачів... – налаштування параметрів служб підключення до вузла Microsoft Office Online, програми вдосконалення програмного забезпечення Microsoft, спільної роботи з документами.

Про програму Microsoft Office Excel – інформація про програму.

2. Основні функціональні клавіші Excel та їх дія

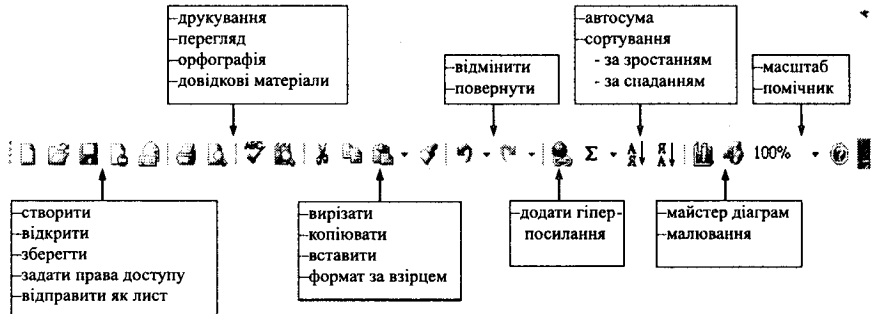
		SHIFT	CTRL	ALT	CTRL+ SHIFT	ALT+ SHIFT
F1	Довідка	Інтерактивна підказка		Вставити аркуш з діаграмою		Встав. новий аркуш
F2	Редагувати активну клітинку	Редагувати примітку клітинки		Зберегти як		Зберегти

		SHIFT	CTRL	ALT	CTRL+SHIFT	ALT+SHIFT
F3	Вставити ім'я в формулу	Вставити функцію у формулу	Присвоїти ім'я		Створити імена за вмістом клітинок	
F4	Повторити останню дію	Повторити останній перехід	Закрити вікно	Вихід		
F5	Перейти до	Відшукати	Відновити розмір вікна книги			
F6	Перейти до наступної ділянки вікна	Перейти до попередньої ділянки вікна	Перейти до наступної книги		Перейти до попередньої книги	
F7	Перевірка орфографії		Перемістити вікно книги			
F8	Вмк./вимк. режиму розширення виокремлення	Вмк./вимк. режиму доповнення виокремлення	Змінити розмір вікна	Викликати вікно діалога Макрос		
F9	Переобчислити всі таблиці	Переобчислити активний аркуш	Згорнути вікно книги			
F10	Перейти до рядка меню	Вивести контекстне меню	Відновити розмір вікна			
F11	Створити діаграму	Вставити новий аркуш	Вставити аркуш для макроса Excel 4.0	Відкрити вікно редактора Visual Basic		
F12	Зберегти як	Зберегти	Відкрити файл		Друкувати	

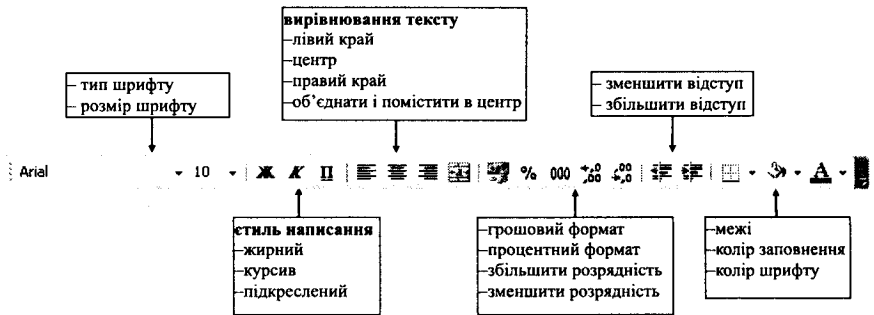
3. Додаткові поєднання клавіш та їх дія

Клавіші	Дія	Клавіші	Дія
Ctrl+C	Копіювати	Ctrl+Ins	Копіювати
Ctrl+X	Вирізати	Shift+Del	Вирізати
Ctrl+V	Вставити	Shift+Ins	Вставити
Ctrl+N	Створити нову книгу	Ctrl+F	Знайти
Ctrl+O	Відкрити книгу	Ctrl+G	Перейти до
Ctrl+S	Зберегти книгу	Ctrl+A	Виділити аркуш
Ctrl+W	Закрити книгу	Ctrl+P	Друкувати аркуш

4. Панель інструментів Стандартна



5. Панель інструментів Форматування



6. Часто живвані функції Excel

Математичні функції

ABS(число) – дає модуль (абсолютне значення) числа. Аргумент: дійсне число, абсолютне значення якого потрібно знайти.

Приклади:

ABS(2) дає 2.

ABS(-2) дає 2.

Якщо клітинка A1 містить -16, тоді ABS(A1) дає 16.

SQRT(число) – дає значення квадратного кореня. Аргумент: *число* – число, для якого обчислюється квадратний корінь. Якщо *число* від'ємне, то функція SQRT дає значення помилки #NUM!.

Приклади:

SQRT(16) дає 4.

SQRT(-16) дає #NUM!.

Якщо клітинка A1 містить -16, тоді SQRT(ABS(A1)) дає 4.

ROUND(число;кількість розрядів) – дає число, заокруглене до заданої кількості десяткових розрядів. Аргументи: *число* – число, яке потрібно заокруглити; *кількість розрядів* – кількість десяткових розрядів, до якої потрібно заокруглити число (якщо *кількість розрядів* більше 0, то число заокруглюється до заданої кількості десяткових розрядів праворуч від знака десяткового розділювача (як звичайно,

коми); якщо *кількість розрядів* дорівнює 0, то число заокруглюється до найближчого цілого, якщо *кількість розрядів* менше 0, то число заокруглюється до заданої кількості десяткових розрядів ліворуч від знака десяткового розділювача).

Приклади:

ROUND(2,15; 1) дає 2,2.

ROUND(-1,475; 2) дає -1,48.

ROUND(21,5;-1) дає 20.

Якщо клітинка A1 містить 2,149, тоді **ROUND**(A1; 1) дає 2,1.

TRUNC(число;кільк_розрядів) – дає число, утворене відкиданням певної кількості десяткових розрядів числа. Аргументи: *число* – число, в якому потрібно відкинути розряди; *кільк_розрядів* – необов'язковий аргумент, який визначає кількість розрядів, що залишаються. У разі використання лише першого аргументу *число* (значення за замовчуванням аргументу *кільк_розрядів*– 0 (нуль)) – дає ціле, утворене відкиданням дробової частини.

Приклади:

TRUNC(8,9) дає 8.

TRUNC(-8,9) дає -8.

TRUNC(21,596; 2) дає 21,59.

SUM(число1;число2;...) – дає суму всіх аргументів. Аргументи: *число1*; *число2*;... – це від 1 до 30 аргументів (аргументом може бути число, адреса клітинки або діапазон клітинок), для яких треба визначити результат їх додавання. У цьому випадку:

- 1) числа, логічні значення і текстові величини, що уведені безпосередньо в список аргументів, перетворюються в числа і додаються;
- 2) якщо аргументом є діапазон клітинок або адреса клітинки, то додаються лише числові дані (порожні клітинки, логічні значення, тексти та значення помилки ігноруються);
- 3) аргумент, що є значенням помилки, спричинює помилку.

Приклади:

SUM(3;2) дає 5.

Якщо клітинки A2, B2 та C2 містять числа 5, 15 та 30, тоді

SUM(A2:B2) дає 20,

SUM(B2:C2; 15) дає 60.

SUMIF(діапазон;критерій;діапазон_для_суми) – дає суму діапазону клітинок, специфікованих заданим критерієм. Аргументи: *діапазон* – це діапазон клітинок, для яких формулюють критерій; *критерій* – критерій у формі числа, виразу або тексту, який визначає, які клітинки додавати (наприклад, критерій можна записати у вигляді "32", ">32", "яблука"); *діапазон_для_суми* – це діапазон клітинок, значення яких підсумовуються (додаються лише ті клітинки, для яких відповідне значення в аргументі *діапазон* задовольняє критерій. Якщо аргументу *діапазон_для_суми* немає, то підсумовуються клітинки в аргументі *діапазон*).

Приклад:

Нехай клітинки A1:A4 містять такі назви категорій товарів: "папір", "папір", "аркуші", "олівці". Нехай клітинки B1:B4 містять коди товарів відповідних категорій, а клітинки C1:C4 – кількість кожного товару, наявного в магазині: 10, 25, 490, 120. Тоді кількість паперу, що є в магазині, **SUMIF**(A1:A4;"папір";C1:C4) дорівнює 35.

» Статистичні функції

MAX(число1;число2;...) – дає найбільше значення зі списку аргументів. Аргументи: *число 1*; *число2*;... – це від 1 до 30 чисел (аргументом може бути число, адреса клітинки або діапазон клітинок), серед яких треба відшукати найбільше.

Логічні значення та текст ігноруються. Якщо аргументи не містять чисел, то функція **MAX** дає 0 (нуль).

Приклади:

MAX(3;2) дає 3.

Якщо клітинки A2:C2 містять числа 5, 15 та 30, тоді

MAX(A2:C2) дає 30,

MAX(A2:C2;50) дає 50.

MIN(число1;число2;...) – дає найменше значення зі списку аргументів. Аргументи: *число1;число2;...* – це від 1 до 30 чисел (аргументом може бути число, адреса клітинки або діапазон клітинок), серед яких треба відшукати найменше.

Якщо аргументи не містять чисел, то функція **MIN** дає 0 (нуль).

Приклади:

MIN(3; 2) дає 2;

Якщо клітинки A2, B2 та C2 містять числа 5, 15 та 30, тоді

MIN(A2:C2) дає 5,

MIN(A2:C2; 0) дає 0.

AVERAGE(число1;число2;...) – дає середнє арифметичне своїх аргументів. Аргументи: *число1;число2;...* – це від 1 до 30 чисел (аргументом може бути число, адреса клітинки або діапазон клітинок), для яких треба відшукати середнє.

Якщо аргументом є адреса клітинки або діапазон клітинок, у яких містяться тексти, логічні значення або порожні клітинки, то такі значення ігноруються, однак клітинки з нульовими значеннями враховуються.

Приклади:

Якщо клітинки A1:A5 мають ім'я *Оцінки* і містять числа 5, 4, 4, 3 і 4, тоді

AVERAGE(A1:A5) дає 4.

AVERAGE(*Оцінки*) дає 4.

AVERAGE(A1:A5;5) дає 4,5.

AVEDEV(число1;число2;...) – дає середнє абсолютних значень відхилень чисел від їхнього середнього арифметичного (міра розкиду множини даних). Аргументи: *число1;число2;...* – від 1 до 30 чисел (аргументом може бути число, адреса клітинки або діапазон клітинок), для яких треба відшукати середнє абсолютних відхилень.

Якщо аргументом є адреса клітинки або діапазон клітинок, у яких містяться тексти, логічні значення або порожні клітинки, то такі значення ігноруються, однак клітинки з нульовими значеннями враховуються.

Приклад:

Якщо клітинки A1:A5 містять числа 5, 4, 4, 3 і 4, то:

AVEDEV(A1:A5) дає 0,4.

COUNT(число1;число2;...) – дає кількість клітинок із числами або кількість чисел в списку аргументів.

Приклад:

Якщо клітинки A1:A5 містять 5, 4, "не з'явився", 3 і 4, тоді

COUNT(A1:A5) дає 4.

COUNTIF(діапазон;критерій) – дає кількість непустих клітинок в заданому діапазоні, які відповідають заданій умові.

Приклад:

Якщо клітинки A1:A5 містять 5, 4, "не з'явився", 3 і 4, тоді

COUNTIF(A1:A5;4) дає 2.

Логічні функції

IF(*лог_вираз*; *значення_якщо_істина*; *значення_якщо_хибність*) – дає одне значення, якщо вказана умова в результаті обчислення дає значення TRUE (істина), і інше значення, якщо FALSE (хибність). Аргументи: *лог_вираз* – це будь-яке значення або вираз, які набувають значень TRUE або FALSE; *значення_якщо_істина* – це значення, яке повертається, якщо *лог_вираз* дає TRUE; *значення_якщо_хибність* – це значення, яке повертається, якщо *лог_вираз* дорівнює FALSE.

Приклад:

IF(A1>0;"додатне число";**IF**(A1=0;"нуль";"від'ємне число")) дає текстовий рядок "додатне число", якщо у клітинці A1 число більше від 0, текстовий рядок "нуль", якщо у клітинці A1 число дорівнює 0 і текстовий рядок "від'ємне число", якщо у клітинці A1 число менше від 0.

Функції дат

DATE(*рік*; *місяць*; *день*) – дає число, що відповідає даті *день.місяць.рік* (тобто перетворює дату у внутрішній числовий формат Excel). Аргументи: *рік* – число, яким задають рік; *місяць* – число, яким задають місяць у році; *день* – число, що відповідає номеру дня у місяці.

Приклади:

DATE(1998,7,16) дає 35992.

DATE(1998,14,16) дає 36207 (відповідає 16.02.1999).

DATE(98,7,36) дає 36012 (відповідає 5.08.1998).

DATEVALUE(*дата*) – дає число, що відповідає даті, яку задано аргументом *дата*. Якщо дата задана помилково, дає #VALUE!. Аргумент: *дата* – це текст вигляду "16 лип 1998", "16.07.98" або "16/07/98".

Приклади:

DATEVALUE("16 лип 1998") дає 35992.

DATEVALUE("1.1.2000") дає 36526.

DATEVALUE(2000) дає #VALUE!.

DAY(*дата*) – дає порядковий номер дня у місяці для дати, яку задано аргументом *дата*. Аргумент: *дата* – це дата у текстовому або числовому вигляді ("16 лип 1998", "16.07.98", "16/07/98" або 35992).

Приклади:

DAY("16 лип 1998") дає 16.

DAY(35992) дає 16.

WEEKDAY(*дата*, *тип*) – дає порядковий номер дня тижня, який задано аргументом *дата*. Аргументи: *дата* – це дата у текстовому або числовому вигляді; *тип* – це число 1, 2 або 3, яке визначає нумерацію днів тижня (необов'язковий аргумент, відсутність якого задає *тип* = 1):

1 – дні нумеруються таким чином: 1 – неділя, ..., 7 – субота;

2 – дні нумеруються таким чином: 1 – понеділок, ..., 7 – неділя;

3 – дні нумеруються таким чином: 0 – понеділок, ..., 6 – неділя.

Приклади:

WEEKDAY("16 лип 1998";2) дає 4 (четвер).

WEEKDAY("16 лип 1998";3) дає 3 (четвер).

MONTH(*дата*) – дає порядковий номер місяця у році для дати, яку задано аргументом *дата*. Аргумент: *дата* – це дата у текстовому або числовому вигляді.

Приклади:

MONTH("16 лип 1998") дає 7 (липень).

MONTH(30) дає 1.

MONTH(367) дає 1.

YEAR(дата) – дає рік для дати, яку задано аргументом *дата*. Аргумент: *дата* – це дата у текстовому або числовому вигляді.

Приклади:

YEAR("16 лип 1998") дає 1998.

YEAR(35992) дає 1998.

TODAY() – дає активну дату у числовому форматі. Аргументів немає.

Функції для роботи з текстом

CONCATENATE(текст1;текст2;...) – об'єднує декілька текстових рядків в один. Аргументи: *текст1, текст2,...* – це від 1 до 30 елементів тексту, які об'єднуються в один елемент тексту. Елементами тексту можуть бути текстові рядки, числа або посилання, які вказують на одну клітинку.

Приклад:

Якщо клітинки A1:D1 містять такі дані "Петренко", "Василь", "Миколайович" та "1980", тоді

CONCATENATE(A1;" ";B1;" ";C1;" народився у ";D1;" році.) дає текстовий рядок "Петренко Василь Миколайович народився у 1980 році."

Запитання для самоперевірки

1. Які завдання найкраще вирішувати за допомогою табличного процесора?
2. Що таке книга? Аркуш?
3. Який захист для книг пропонує Microsoft Excel?
4. Як зберігати інформацію про книги, з якими потрібна одночасна робота?
5. Охарактеризуйте призначення пунктів головного меню Excel.
6. Що таке клітинка? Як керувати курсором при навігації по клітинках?
7. Які є два способи адресації комірок? Яка між ними різниця?
8. Що таке абсолютне та відносне адресування комірок?
9. Чи можна присвоювати імена клітинкам?
10. Як відбувається введення даних в клітинку?
11. Використання формул для обчислень в Excel.
12. Як здійснювати виділення комірок за допомогою миші та клавіатури?
13. Як працювати з виділеними клітинками за допомогою буфера обміну?
14. Як додати до таблиці рядки та стовпці?
15. Як змінити висоту рядка та ширину стовпця таблиці?
16. Як задати формат зображення даних у клітинках?
17. Як задати шрифт та вирівнювання вмісту комірок?
18. Як задати способи облямування та затінення комірок?
19. Як визначити режим захисту комірок?
20. Як створити стиль для параметрів форматування?
21. Які основні принципи роботи з базами даних в Excel?
22. Як використати форму даних для введення даних?
23. Як здійснюється сортування інформації в базі даних?
24. Як здійснюється пошук інформації в базі даних?
25. Як працює Майстер імпорту тексту?
26. Як працює Майстер діаграм?
27. Як задають параметри друкованої сторінки?
28. Як задають параметри друкування?
29. Назвіть принципи роботи з макрокомандами.

8. РОБОТА З БАЗАМИ ДАНИХ.

Microsoft Access

📖 План викладу матеріалу

1. Створення бази даних
2. Таблиці
 - 2.1. Створення таблиць за допомогою майстра Table Wizard
 - 2.2. Створення таблиць в режимі Design View (Режим конструктора)
 - 2.3. Властивості полів
 - 2.4. Ключові поля
 - 2.5. Режим таблиць
 - 2.6. Введення зовнішніх таблиць
 - 2.7. Міжтабличні зв'язки
3. Запити QBE
 - 3.1. Запити на вибірку
 - 3.2. Запити з обчислювальними полями
 - 3.3. Запити для впорядкування записів в таблиці
 - 3.4. Групувальні запити
 - 3.5. Запити з умовами вибору (параметрами)
 - 3.6. Запити на створення таблиці
 - 3.7. Перехресні запити
 - 3.8. Запити вилучення
 - 3.9. Запити на оновлення записів в таблицях
 - 3.10. Запити на додавання записів
 - 3.11. Запити, що здійснюють вибірку записів, що повторюються
 - 3.12. Запити, що здійснюють вибірку записів, які не мають підпорядкованих
 - 3.13. Властивості запитів

↪ Ключові терміни розділу

▼ Access	▼ Ключові поля
▼ Записи	▼ Зв'язки
▼ Запити	▼ SQL
▼ Звіти	▼ QBE

СКБД – це програми, за допомогою яких можна створювати бази даних (БД), здійснювати введення інформації, а також отримувати потрібну інформацію з БД. Існує багато різних систем керування базами даних. З їх допомогою можна створювати структури, зручні для користувачів, вводити в них необхідні елементи керування. До таких систем можна віднести **FoxPro**, **Paradox**, **Oracle** та ін. Далі розглянемо створення таблиць і запитів (мова QBE). Створення звітів, роботу з формами, макросами та модулями, сторінки доступу детально розглянуто в книзі [3].

8.1. Створення бази даних

Після запуску MS Access на екран виводиться стартове діалогове вікно, яке дає можливість відкрити одну з баз даних або почати створення нової бази. Це вікно має такий вигляд (рис. 8.1). За допомогою цього вікна можна здійснити доступ до всіх створених об'єктів, або почати їх створення.

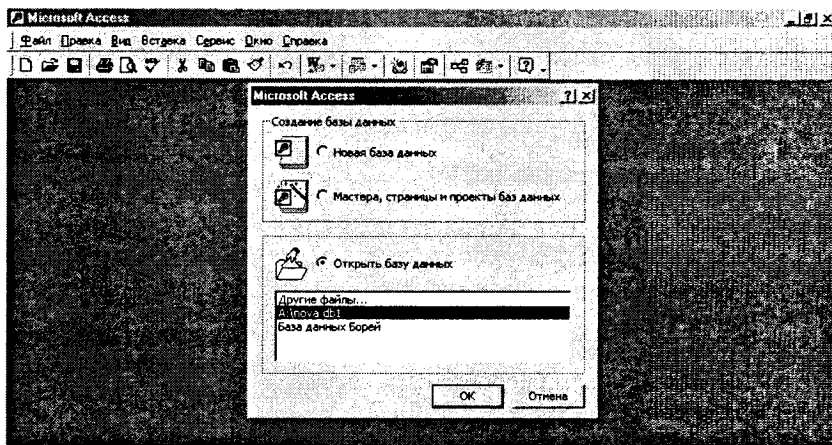


Рис. 8.1

Після запуску Access потрібно:

1) вибрати команду Blank Database (Нова база даних) з групи перемикачів;

1) після відкриття діалогового вікна File New Database (файл нової БД) обрати ім'я і місцезнаходження бази даних у вікні (рис. 8.2);

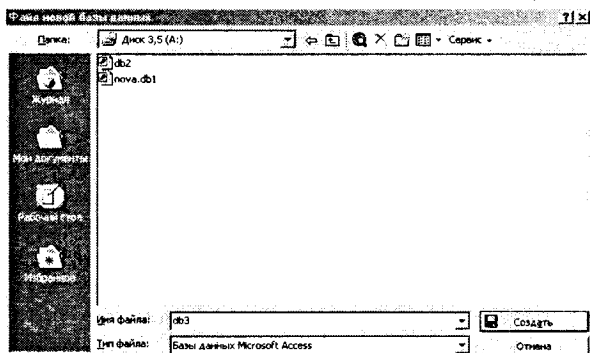


Рис. 8.2

3) натиснути кнопку Create (Створити). На екрані з'явиться вікно Database (База Даних) (рис. 8.3).

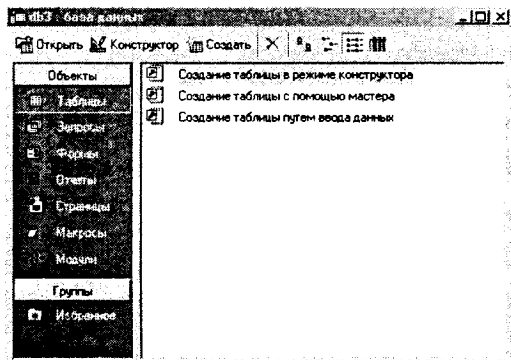


Рис. 8.3

На лівій панелі цього вікна зображені елементи керування для виклику всіх семи типів об'єктів програми. На правій панелі зображено список таблиць, що є в складі бази даних, а також елементи керування для створення нової таблиці.

Якщо діалогове вікно (рис. 8.1) не виводиться, то почати створення нової БД можна за допомогою команд File (Файл), Create (Створити), або кнопки New Database (Створити БД) на панелі інструментів. На екрані з'явиться вікно Create (Створення) (рис. 8.4).

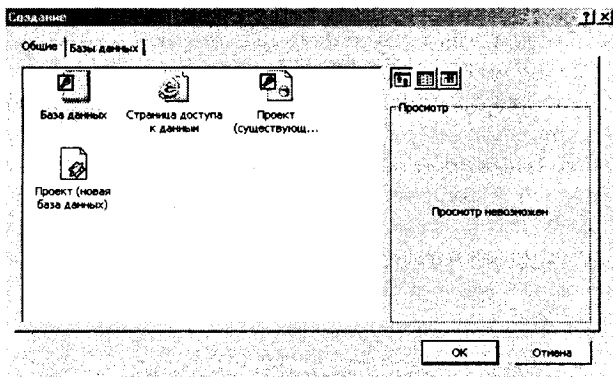


Рис. 8.4

У ньому обираємо закладку “Загальні”, а в ній об’єкт “База даних”. На екрані з’явиться вікно “Файл нової Бази даних” (рис. 8.2).

8.2. Таблиці

Створювати таблиці в Access можна трьома способами.

По-перше, можна скористатися майстром для побудови таблиць на основі зразків Access.

По-друге, створювати таблиці в режимі конструктора. Цей метод дає можливість контролювати властивості полів таблиці.

По-третє, інформація вводиться в таблицю і надається можливість системою Access визначити тип інформації на основі введених даних.

8.2.1. Створення таблиць за допомогою майстра Table Wizard

Для створення таблиці за допомогою майстра потрібно у вікні Database (База даних) два рази клацнути на кнопці Create Table by Using Wizard (Створення таблиці за допомогою майстра). На екрані з'явиться діалогове вікно, що містить описи різних типів таблиць баз даних. Таблиці поділяються на дві групи Business (Ділові) та Personal (Особисті) (рис. 8.5).

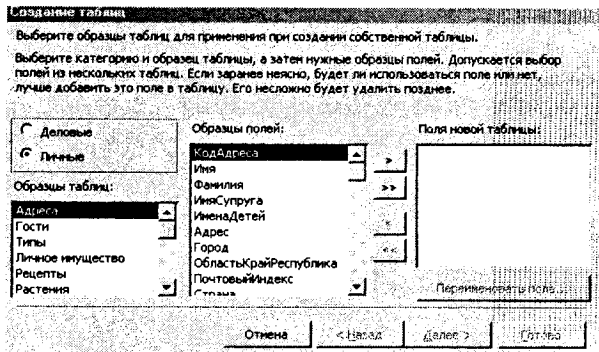


Рис. 8.5

Потрібно переглянути список таблиць і знайти варіант, який найбільш підходить для створюваної БД. Клацаючи на кнопках ">" і ">>", обрати потрібні поля. Додавати поля можна також і з інших таблиць. Після визначення всіх полів потрібно клацнути на кнопці Next (Далі). На екрані з'явиться діалогове вікно, в яке можна ввести назву таблиці (рис. 8.6).

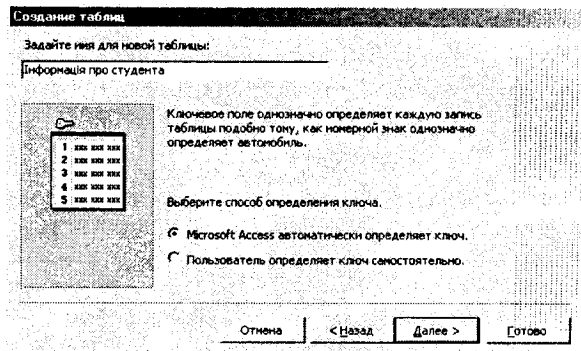


Рис. 8.6

Після введення назви таблиці необхідно позначити варіант створення первинного ключа. Якщо ключ будемо створювати самостійно, то на екран буде виведено вікно, що буде мати наступний вигляд (рис. 8.7).

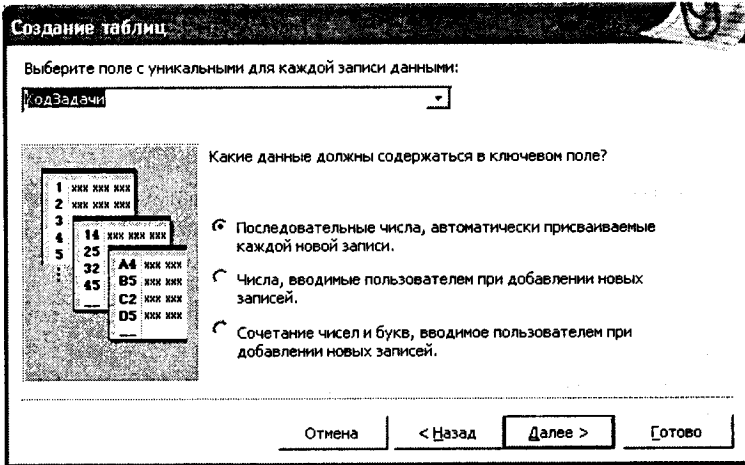


Рис. 8.7

Коли майстер автоматично визначає первинний ключ для таблиці, він переглядає поля, відшукуючи полу типу Autonumber. Перше знайдене поле такого типу використовується в ролі первинного ключа. Якщо майстер не знаходить ні одного подібного поля, він створює нове поле з назвою, що складається з назви таблиці і букви ID.

Якщо ключ буде створюватися самостійно, майстер відображає на екрані список томів таблиці. Далі користувач повинен виділити поле, яке буде мати унікальне значення і вибрати його тип. Майстер може формувати один з трьох типів первинного ключа: номер, що створюється Access автоматично, номер, що створюється користувачем, і текстовий рядок, створений вручну. Для продовження роботи потрібно натиснути кнопку Next (Далі).

Якщо БД складається з декількох таблиць, майстер Table Wizard запитав, чи зв'язана дана таблиця з іншими таблицями. Якщо зв'язків немає, то у вікні майстра Table Wizard можна змінити вигляд таблиці за допомогою Design View (Режим конструктора) і ввести дані за допомогою таблиці.

Якщо зв'язки є, то потрібно виділити назву таблиці і клацнути на кнопці Relationship (Зв'язки). Далі потрібно вибрати тип відношення, який найкраще підходить для двох таблиць. Якщо було вибрано зв'язок один-до-багатьох, то натискування кнопки ОК приведе до перевірки натискування полів первинного ключа в обох таблицях для перевірки того,

чи існує він в обох таблицях. Якщо ключові поля не знайдені, майстер автоматично добавить поля первинного ключа поля первинного ключа в іншу таблицю.

Приклад. Діючи згідно наведеного вище, створимо наступну таблицю: список студентів групи (рис. 8.8).

Код	Прізвище імя	По-батькові	Адреса	Поштовий індекс	Домашній тел	Дата народження
1	Андрійчук Віра Андрівна		Федьковича 38/45	790018	358170	21.06.1984
2	Борйчук Олег Богданович		Наукова 75/25	790053	911517	12.12.1984
3	Вербіцька Ірина Ігорівна		Кошова 19/1	790014	446568	11.10.1984
4	Винницька Ольга Романівна		С. Петлюри 47/38	290063	446546	13.11.1984
5	Гельнер Володимир Юрійович		Червоної Калини 53/71	790070	2210141	14.01.1984
6	Домагальська Марія Степанівна		Замарстинівська 11а	790058	785784	14.05.1984
7	Домагальська Анастасія Степанівна		Фурманська 25/5	790058	544564	15.04.1984
8	Іванік Юрій Олександрович		Ст. Бандери 14/7	790082	641916	15.08.1984
9	Жолінський Тарас Євгенович		Пасична 73/52	790032	354898	29.03.1984

Рис. 8.8

8.2.2. Створення таблиць в режимі Design View (Режим конструктора)

Якщо потрібно змінити таблицю після створення в режимі майстра, то потрібно у вікні (рис. 8.1) вибрати “Створення таблиці в режимі конструктора”. В результаті на екрані висвітиться вікно (рис. 8.9).

Рис. 8.9

Стовпці Field Name (Назва поля), Data Type (Тип даних) і Description (Опис) описують кожне поле таблиці. Назви полів, які будемо використовувати, не повинні перевищувати 64 символи. В назвах полів не можна використовувати такі символи “!”, “.”, “[”, “]” і “””. Дозволяється використовувати в назві поля пропуск, але не на початку.

Після введення назви поля переходимо до вибору типу даних. Для цього необхідно натиснути кнопку вибрати тип даних із списку типів даних в стовпці **Data Type** (тип даних) натиснувши кнопку **^**.

В Access допускаються такі типи полів.

1. Text (Текст). Поле цього типу використовується для зберігання таких даних, як назви, адреси або числові дані які не будуть брати участь в обчисленнях (наприклад номери телефонів, поштові індекси і інше). Текстове поле не повинно перевищувати 255 символів. По замовчуванню поля містять 50 символів. Щоб можна було до поля ввести більшу кількість символів потрібно ввести відповідні зміни у властивості **Field Size** розмір поля.

2. Memo. Поля цього типу використовуються для зберігання даних, що мають довжину більшу ніж 255 символів. Поле Мемо може містити до 65535 символів. Такі поля не можуть бути індексовані або посортовані. Поле не може бути ключовим. Для зберігання форматованих текстів і довгих документів замість поля Мемо потрібно використовувати поля **OLE**.

3. Number (Числа). Цей тип даних зберігає числові величини. Використовується для задання даних, які можуть використовуватися в числових обчисленнях. Розмір поля, може приймати 1, 2, 4, 8 і 12 байт.

4. Date time (Дата час). Поля такого типу використовуються для часу і дати. Довжина поля 8 байтів. В цьому полі можна задавати дані від 1 січня 100 року до 31 грудня 9999 року включно, а час з точністю 3,33 мілісекунди. Використання цього формату, а не числового, дає можливість здійснювати правильне впорядкування за датою і часом. За допомогою властивості формат поля можна задати спосіб зображення – дати і часу, тільки дати, тільки часу та ін.

5. Currency (Грошовий). Цей тип використовується для задання грошових величин. Обчислення в цьому форматі проводяться з точністю до 15 знаків в цілій частині і до 4 знаків в дробовій частині. Довжина поля 8 байтів спосіб зображення грошової інформації залежить від налаштувань Регіональні установки Windows.

6. Autonumber (Лічильник). Цей тип даних зберігає унікальне число, яке автоматично додається до кожного запису в момент включення цього запису до таблиці. Числа, згенеровані через поле типу **Autonumber**, не можна знищити, а також модифікувати. Як правило, лічильник використовується для створення первинного ключа таблиці.

7. Yes/No (Логічний). Цей тип даних можна використовувати для даних, що можуть приймати одне з двох можливих значень Так/Ні, True/False або Вкл./Викл. Логічним полям у формах як правило відповідають прапорці. Розмір поля 1 байт.

8. OLE Object (Об'єкт OLE). Поле цього типу зберігає посилання на інший об'єкт, наприклад, документ Word, Excel, рисунки і інші об'єкти

утворені в інших програмах при використанні протоколу OLE. Розмір полів об'єктів OLE обмежені розміром вільного місця на диску.

Для вставки в поле таблиці об'єкту OLE потрібно в меню "Вставка" вибрати пункт об'єкт на екрані з'явиться діалогове вікно вставка об'єкту. Через це вікно можна створити новий об'єкт, або вибрати готовий із файлу. Після натискування кнопки **створити новий** і вибору типу об'єкту натискування кнопки **ОК** приведе до запуску програми, що відповідає даному типу і в якій можна створити потрібний об'єкт. Після закриття програми створений об'єкт буде додано в таблицю. В такий спосіб файл фізично додається в файл бази даних. Якщо об'єкт існує то його можна приєднати або додати до файлу бази даних. Якщо об'єкт OLE додається то він втрачає зв'язок зі своїм прототипом і існує незалежно від нього.

9. Hyperlink (Гіперпосилання). Виділений іншим кольором текст або рисунок на якому потрібно клацнути щоб перейти до визначеного місця в тексті, WWW-сторінки в Інтернеті. Гіперпосилання можуть також вказувати на Telnet, FTP або дискусійні групи.

Поле такого типу може складатися із чотирьох частин тексту, що висвічується на екрані; адреси, що вказує шлях до файлів, або Web-сторінок; додаткових адрес, що вказує положення в файлі або на екрані і екранна підказка. Гіперпосилання можна задавати безпосередньо в полі таблиці, але крім такого способу в Access гіперпосилання можна задавати за допомогою майстра. Для цього потрібно вибрати в меню "Вставка" пункт **Гіперпосилання**. На екрані з'явиться вікно **Додавання гіперпосилань** (рис. 8.10).

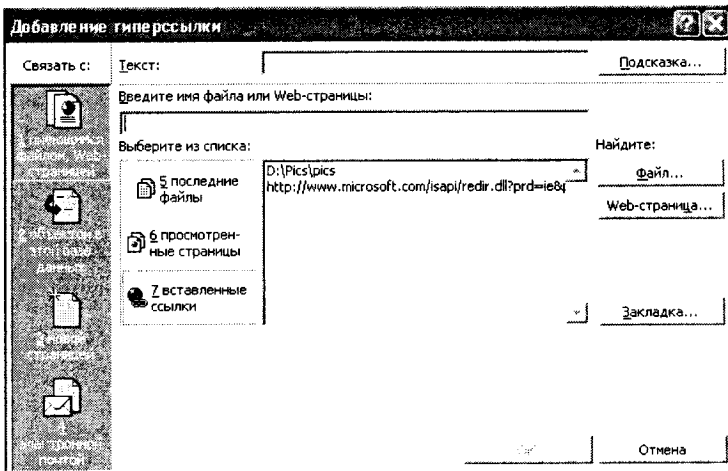


Рис. 8.10. Додавання гіперпосилань

У цьому вікні потрібно включити «Зв'язки з файлом чи Web-сторінкою». У вікні додавання гіперпосилання. Екранний текст можна ввести в поле введення Текст. Назву файлу або Web-адресу можна ввести у відповідні поля. Якщо ми будемо вставляти адресу Web-сторінки, то потрібно натиснути на кнопку Web-сторінку. Після встановлення зв'язку у відповідне поле буде поміщена адреса URL. При зміні адреси буде змінюватися і вміст поля адреси. Додаткова адреса є необов'язковою. Вона вказує на певну частину Web-сторінки. Додаткова адреса додається за допомогою кнопки Закладка.

Четверту частину гіперпосилання можна додати за допомогою натискування на кнопці Підказка. У вікно, що з'явиться, потрібно ввести текст. Після натискування на кнопці ОК. Вікно додавання гіперпосилання закриється. Створене гіперпосилання додаться в таблицю.

10. Lookup Wizard (Майстер підстановок). Це останній в списку, що розкривається тип даних. Для того, щоб задати в таблиці таке поле потрібно:

1) в списку, що розкривається, вибрати тип майстер підстановок (Lookup Wizard). На екрані з'явиться діалогове вікно (рис. 8.11).

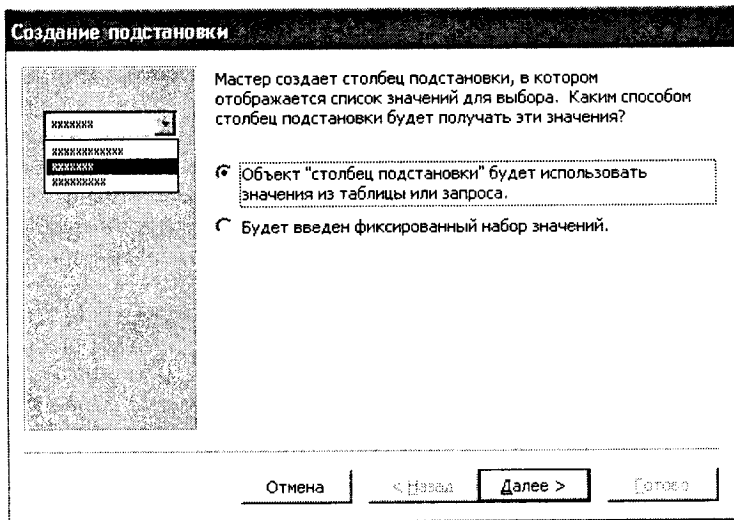


Рис. 8.11

В залежності від того, звідки будуть братися значення для підстановки (на основі полів іншої таблиці чи на основі фіксованого набору значень) вибрати потрібний пункт. Якщо буде вибрано фіксований набір значень то після натискування кнопки NEXT (Далі) на екрані з'явиться вікно (рис. 8.12).

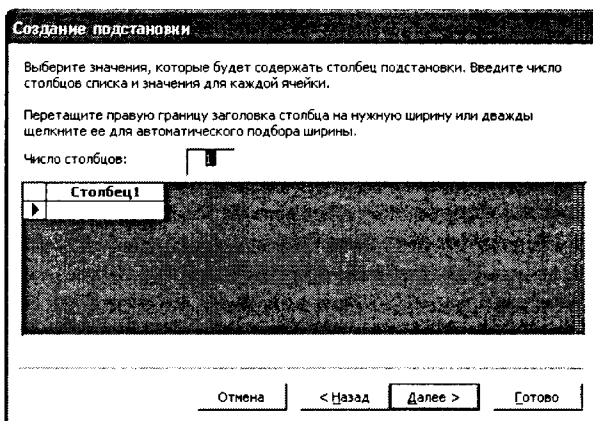


Рис. 8.12

Для підстановки можна розглянути декілька полів, для чого в вікні число стовпців, потрібно ввести їх кількість, а в стовпець список все можливих значень. Наприклад, для поля оцінка це можуть бути значення 2, 3, 4, 5. Після натискування кнопки **Next** (Далі) Майстер запропонує ввести назву стовпця підстановки. По замовчуванню це назва поля. Далі потрібно натиснути кнопку **Finish** в результаті буде створено набір даних, що повертаються після підстановки;

2) якщо вибрано об'єкт **“стовпець підстановки”**, буде використовуватися значення з таблиці, або запиту, то після натискання **Next** на екрані появиться вікно (рис. 8.13), в якому буде відображено список таблиць в базі даних. Далі, потрібно вибрати таблицю, що містить потрібні поля і натиснути кнопку **Next**. Після цього на екран буде виведено список полів вибраної таблиці (рис. 8.13).

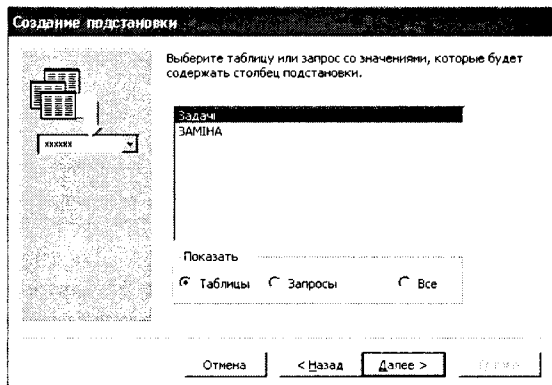


Рис. 8.13

В залежності від потреби можна вибирати одне і більше полів. Якщо ключ не вибрано, то він додається автоматично. Далі натискання кнопки **Next**, встановлює ширину полів, після натискування на кнопки **Finish** збережуться зміни.

Крім типів даних в Access для кожного поля можна задати довільний коментар, в якому можна описати призначення поля.

Для кожного поля можна також задати додаткові властивості полів, вибравши конкретні значення на закладці **General** (Загальна). Оскільки в Access параметри полів встановлюються по замовчуванню, тому змінювати їх можна лише у виняткових випадках.

8.2.3. Властивості полів

Розглянемо деякі властивості полів.

1. Розмір поля (Field size) задає розмір даних, що зберігаються в полі. Для поля з числовим типом даних значення властивості розмір поля буде таким:

Байт (Byte)	Поле займає 1 байт і може набувати цілочисельні значення від 0 до 255
Ціле (Integer)	займає два байти і може набувати значення від -32768 до 32768
Довге ціле (Long integer)	Поле займає 4 байти і може набувати значення з діапазону від -214783468 до 2147483468
З плаваючою крапкою (Single)	розмір поля 4 байти. Діапазон значень -3,4020823E38 до -1,401298E-45 від'ємних і від 1,401298E-45 до 3,402823E38
З плаваючою крапкою (Double)	Розмір поля – 8 байт від -1.79E308 до +1.79E308
Дійсне (Real)	

2. Формат поля (Format) – задає вигляд даних при виведенні на екран або друк. Формати можна задавати шляхом вибору із списку, що відкривається. В цьому списку задані такі формати:

Дата і час:

Повний формат дати (General Date)	Цей формат представляє комбінацію короткого формату дати і довгого формату часу 1/1/03 5:20:00 PM. Якщо в комірці зберігається тільки час, то висвічується тільки час – 7:35:00 PM. Якщо в комірці зберігається тільки дата, то висвічується тільки дата – 13/15/03
Довгий формат дати (Long Date)	Цей формат повністю відповідає установці у вікні Регіональні установки панелі керування Windows. Наприклад 21 листопада 2002 року

Середній формат дати (Medium date)	В цьому форматі дата містить число, перші три букви назви місяця і дві останні цифри року. Наприклад 30 gru-02
Короткий формат дати (Short Date)	Повністю відповідає однойменній установці у вікні Регіональні установки панелі Windows. Наприклад 13/25/03
Довгий формат дати (Long time)	відповідає однойменні установці Регіональні установки панелі керування Windows. Наприклад 7:18:01 PM
Середній формат часу (Medium time)	Цей формат містить години, хвилини і символи АМ/PM (12 годинний цикл). Наприклад 7:18 PM
Короткий формат часу (Short time)	Цей формат містить години і хвилини (24 годинний цикл)

Числові і грошові формати

Основні (General number)	Показує число так, як воно було введено
Грошовий (Currency)	Цей формат відповідає однойменній установці у вікні Регіональна установки панелі керування Windows
Євро (Euro)	Замість грошового символу підставляється символ
Фіксований (Fixed)	Цей формат відповідає формату чисел, вказаному в діалоговому вікні Регіональні установки
З розділенням розрядів (Standart)	Використовується фіксований формат з розділювачами розрядів.
Процентний (Percent)	Величина, що містить в полі, множиться на 100 і додається символ проценту
Експоненціальний (Scientific)	Використовується нормалізована форма запису чисел. Приклад. 5.55E+5

Формати полів таблиці можна задавати також користувачем за допомогою рядків, що містять комбінацію звичайних і керуючих символів. За допомогою такого формату можна по різному зображати інформацію, що зберігається в полях. Формат, який задає користувач, може складатися з чотирьох частин, що розділяються крапкою з комою (;). Перша частина описує формат для додатнього числа, друга для від'ємного, третя – для нульового значення і четверта частина – якщо число не введено. Всі чотири частини формату вказувати не обов'язково.

	Символ	
1.	Не задано	Відображає число без форматування

2.	(0)	Резервує місце на десяткову цифру. Забезпечує висвітлення цифри, якщо вона є, або нуля якщо її немає. Якщо число має менше цифр ніж є нулів у виразі форматування на екрані будуть висвітлені початкові і кінцеві нулі. Якщо число має більше цифр справа від десяткової крапки ніж є нулів у виразі форматування, то число буде заокруглено до стількох знаків десяткових знаків, скільки є нулів, у виразі форматування. Якщо число має більше цифр зліва від десяткової крапки ніж є нулів у виразі форматування то цифри будуть висвітлені без змін. Наприклад за форматами 00000.000 результат для числа 25,3579 буде таким 00025.380
3.	(#)	Резервує так само як (0) місце на десяткову цифру. Забезпечує висвітлення цифри або невисвітлення жодного знаку. Дія цього символу подібна до дії символу (0), за винятком початкових и кінцевих нулів, які не висвітлюються, якщо число має стільки само або менше цифр ніж число знаків # з будь-якої сторони від десяткової крапки у виразі форматування. Приклад: За форматом ###.### число 34,75 буде зображено у такій формі 34,75 , а число 37956,3789 - 37956,379
4.	(.)	Задає десяткову крапку. В деяких випадках для зображення крапки може використовуватися інший символ. Місце, що резервується на десяткову крапку, визначає скільки цифр буде висвітлюватися зліва и справа від десяткової крапки. Якщо вираз форматування містить стільки знаків (#) зліва від крапки то, число менше за одиницю розпочинається від десяткової крапки. Щоб зобразити нуль у числі на початку перед десятковою крапкою, потрібно використати (0). Приклад: За форматом #.## число 0,25 буде висвітлено у формі .25.
5.	(%)	Знак проценту. Значення поля множиться на 100 і в кінці додається знак проценту. Символ % проценту повинен використовуватися з іншими символами форматування. Приклад: За форматуючим виразом #.##% для числа 0,375 зображення буде мати наступний вигляд 3,75%.
6.	(,)	Зображає розділювач для тисячі. В деяких інсталяціях в ролі такого розділювача використовують крапку або пропуск. Такий розділювач відділяє більше як чотири цифри зліва від десяткового розділювача. Розділювач тисячі виконує дії в стандартний спосіб, якщо він знаходиться між символами десяткових цифр (0 або #). Наприклад, вираз форматування ##0,, може бути використаний для зображення 350 000 000 за допомогою числа 350.

7.	(E- E+ e- e+)	Ці символи дають можливість зображати числа в експоненціальному форматі. Наприклад, число 2450,01 в цьому форматі буде зображено так 2,45001E+3, а число 0,0000578 так: 5,78E-5.
8.	\$	Висвітлюється поданий знак грошової одиниці. Для висвітлення іншого знаку потрібно попередити його похилою лівою рисою (\) або помістити його в лапки (" ").
9.	(\)	Висвітлює наступний знак за похилою рисою в послідовності форматуючих знаків. Для того, щоб висвітлити знак, який самостійно виступає і має спеціальне значення, потрібно попередити його похилою лівою рисою (\). Знак риски не висвічується. Щоб висвітити похилу ліву риску потрібно використати два такі знаки \\. Знаки, що не висвічуються самостійно це знаки форматування дати і часу а, с, d, h, v, n, p, q, s, t, w, y, /, : , знаки форматування чисел #, 0, %, E, e, кома, крапка, а також @, &, <, >, !

1.	"ABC"	До значення поля додається рядок символів, знаків, які знаходяться між подвійними лапками. Для всіх типів полів можуть використовуватися такі символи
2.	\	
3.	!	(знак оклику) цей символ встановлює вирівнювання по лівій границі поля таблиці
4.	[Колір]	Прямокутні дужки повинні вказуватися обов'язково. Міняє колір символів на вказані. В Access можна використовувати наступні значення кольорів: 1. Чорний (Black) 2. Синій (Blue) 3. Зелений (Green) 4. Бірюзовий (Cyan) 5. Червоний (Red) 6. Ліловий (Magenta) 7. Жовтий (Yellow) 8. Білий (White) Наприклад: за форматом#, ##0.00[Жовтий]; ###0.00[Червоний] число, що зберігається, якщо воно додатне – буде надруковане жовтим кольором, і червоним, якщо воно від'ємне.

3. В Access можна використовувати також символи, які дають можливість користувачу визначити будь-який формат дати і години.

1.	: (двокрапка)	Розділювач години. В деяких регіональних установках цей розділювач може зображатися іншим знаком. Двокрапка відділяє години, хвилини і секунди при форматванні значень, що означає час
----	---------------	--

2.	/ (похила риска)	Розділювач дати. Цей розділювач відділяє день, місяць і рік при форматуванні значень, що означають дату
3.	@	За цим форматом висвітлюється повний формат дати так як це було розглянуто в попередньому пункті
4.	tttt	Цей формат повністю відповідає довгому формату часу
5.	d	Задає номер дня в місяці (від 1 до 31). Нуль не передує цифрі
6.	dd	Так само як d, задає день числом, в якому передує нуль перед цифрою. <i>Приклад: 01; 05; 31</i>
7.	ddd	Задає перші три букви назви дня тижня. <i>Наприклад: Пон; Вто; Сре; Чет; Пят; Суб; Нед</i>
8.	dddd	Задає повну назву дня тижня. <i>Наприклад: Понедельник; Вторник; Среда; Четверг; Пятница; Суббота</i>
9.	dddddd	Задає дату (день, місяць, рік), яка повністю відповідає встановленому формату короткоформатної дати. <i>Наприклад 05 /02 /03</i>
10.	ddddddd	Задає дату (день, місяць, рік), яка повністю відповідає встановленому формату довгоформатної дати
11.	W	Задає номер для читання (від 1 до 7) (1 для неділі, 7 для суботи)
12.	WW	Задає номер читання в році (від 1 до 54)
13.	m	Задає номер місяця в році, числом без передуючого нуля (1-12)
14.	mm	Задає номер місяця (від 01 до 12) з передуючим нулем. <i>Наприклад 03; 10</i>
15.	mmm	Надає перші три букви місяця. <i>Наприклад Янв, Фев, ...</i>
16.	mmmmm	Задає повну назву місяця. <i>Наприклад (Январь, ..., Декабрь)</i>
17.	q	Задає квартал року, в якому знаходиться дата (від 1 до 4)
18.	y	Задає номер дня в році (від 1 до 366)
19.	yy	Задає дві останні цифри року (від 00 до 99)
20.	yyyy	Задає повністю рік (від 100 до 99999)
21.	h	Задає годину (від 0 до 23) без передуючого нуля. <i>Наприклад 4; 21</i>
22.	hh	Задає годину (від 00 до 23) з передуючим нулем. <i>Наприклад 04; 21</i>

23.	n	Задає хвилини (від 0 до 59) без передуючого нуля. <i>Наприклад 5; 37</i>
24.	nn	Задає хвилини (від 00 до 59) з передуючим нулем. <i>Наприклад 05; 37</i>
25.	s	Задає секунди (від 0 до 59) без передуючого нуля
26.	ss	Задає секунди (від 00 до 59) з передуючим нулем
27.	AM/PM	Задає 12-годинний цикл для часу. Для годин до 12-ти годин висвітлює символи AM, а для годин після 12-ти годин до 23:59 висвітлює напис PM великими буквами
28.	am/pm	Працює так само як AM/PM, але символи am/pm висвітлюються малими буквами
29.	A/P	Так само як AM/PM, але символи висвітлюються A і P
30.	a/p	Так само як am/pm, але символи висвітлюються a і p
31.	AMPM	Встановлює 12-годинний цикл. В ролі символів AM або PM вказуються символи визначені у вікні регіональних установок (Regional setting) панелі керування Windows

Текстові або МЕМО поля.

Користувацький формат для текстових або Мемо полів може містити дві частини розділені ; крапка з комою. Перша частина описує формат якщо в поле введені будь-які текстові дані, а друга частина призначена для пустих Null рядків.

- > Переводить всі символи із вмістимого поля у верхній регістр
- < Переводить всі символи вмістимого поля в нижній регістр
- @ Потрібний текст (хоч би один символ)

Приклад. За форматом @; "не здано" комірки, що містять порожні рядки буде виведено слова "не здано". За форматом >@ текст з поля буде виводитись великими буквами.

Логічне поле.

Користувацький формат для логічних полів може містити три частини розділені крапка з комою (;). Перша частина не впливає на формування. Її можна пропустити, але крапка з комою є обов'язковою. Друга частина описує формат для значень ТАК (ІСТИНА, ВКЛЮЧЕНО), а третя – для значень НІ (НЕПРАВДА, ВИКЛЮЧЕНО).

3. Маска введення (input mask). Ця властивість використовується для полегшення введення даних, а також контролю інформації, що вводиться. Можна, наприклад, утворити маску для введення номеру телефону. В ACCESS маску введення можна створювати за допомогою майстра. Для запуску майстра потрібно клацнути на кнопці побудовувача (кнопка з трьома крапками справа від поля) властивості маска введення. У вікні, що появиться на екрані буде міститися список масок. Після вибору потрібної

маски можна перейти в поле введення. Проба для того, щоб випробувати роботу маски введення можна редагувати. Для цього натиснути на *СПИСОК* і у вікні, що з'явиться на екрані ввести зміни.

Маска введення так само, як деякі формати, може містити до трьох частин. В першій частині міститься опис маски. Опис маски – це рядок керуючих символів. Друга частина може містити 0 або 1. Порожні другі частини відповідають значенню 1. Якщо використати 0, то розділюючі символи, що були наведені вище, будуть зберігатися в полі разом із введеними даними.

В третій частині вказується символ маски. Цей символ означає знакомісце для введення інформації.

Маски введення можна створити, використовуючи наступні символи.

1.	0	Місце для цифри (від 0 до 9). Введення цифри обов'язкове. Знаки "+" і "-" не допускаються
2.	9	Дозволяє ввести цифру, пропуск, або можна нічого не вводити. Знаки "+" і "-" не допускаються
3.	#	Цифра або пропуск не обов'язкові. Допускається введення знаків "+" або "-"
4.	L	Знакомісце для букви (від A до Z від A до Я). Введення обов'язкове
5.	?	Так само як L.
7.	A	Знакомісце для букви або цифри. Введення необов'язкове
8.	&	Знакомісце для будь-якого символу або пропуску. Введення обов'язкове.
9.	C	Знакомісце для будь-якого символу або пропуску. Введення необов'язкове.
10.	.	Розділювач для десяткової частини і тисяч для чисел.
11.	, : - /	Дати і часу
12.	<	Переводить всі символи, що слідує за цим знаком в нижній регістр.
13.	>	Переводить всі символи, що слідує за цим знаком в верхній регістр.
14.	!	Забезпечує вирівнювання по лівому краю поля
15.	\	Забезпечує виведення знаку, що слідує за цим символом.
16.	""	Виводяться всі символи, що знаходяться в подвійних лапках.

Приклад:

Маска \(\0\-000"" 0 ""'00 ""'00 дає можливість вводити номери телефонів (8-245) 2-22 55 75

Маска \(\9\999""'9\ \99\ 99\ 99 дасть можливість ввести номер телефону у формі (8_245)2_32_45_78

Маска `\(9\999\)"9\ 99\ 99\ 99` дасть можливість ввести номер телефону у формі `(8_245)2_32_45_78`

За маскою `#9999` можна ввести `-247; +4979`

4. Підпис (Caption). Ця властивість є присутня у полів всіх типів. Вона може містити до 2048 символів символної інформації, яка задає заголовок стовпця. Якщо ця властивість не містить тексту, то для заголовка стовпця використовується назва поля.

5. Значення за умовчанням (Default Value). Цю властивість можна використовувати для всіх типів полів, за винятком полів типу лічильника і об'єктів типу OLE. В цій властивості задається значення, яке буде автоматично підставлено в поле при створенні нового запису. В цій властивості можна використовувати вирази, перед якими потрібно обов'язково вказувати знак `"="`. Значення виразу обчислюється на момент відкриття нового запису і підставляється у відповідне поле. Довжина цієї властивості не повинна перевищувати 255 символів.

6. Умова на значення (Validation Rule) і повідомлення про помилку. Ця властивість використовується до таких типів полів. Якщо умова на значення вказана, то при переході на інший запис або поле, буде виконуватись перевірка умови з використанням введеного в поточне поле значення. Якщо умова не виконується, то буде видано повідомлення про помилку згідно тексту, що міститься у властивості повідомлення про помилку. Умова на значення може містити 2048 символів, а повідомлення про помилку – до 255 символів. Наведемо приклади умов на значення:

1) `Like"C???"` вимагає, що введений текст повинен містити 4 символи і починатись з букви C;

2) Умова `>=#10|01|03#AND#<=30,01.03#OR>=#7|06|03#AND#&|07|03<` задає діапазон для дат, що будуть вводиться.

7. Обов'язкове поле (Required). Цю властивість можуть мати всі типи полів за виключенням лічильника. Можливі значення - **так** або **ні**. Якщо поле є обов'язковим, то в нього обов'язково потрібно ввести інформацію, інакше буде видано інформацію про помилку. За умовчанням встановлюється значення **ні**.

8. Пусті рядки (AllowZeroLenght) Властивості пусті рядки надаються текстовим полям, полям Memo і гіпертекстовим посиланням. Дозволяє або забороняє введення в текстове поле пустих рядків. Ця властивість може приймати значення **так** або **ні**. За умовчанням **ні**.

9. Індексоване поле (indexed). Ця властивість вказує на те, чи буде дане поле індексом. Вона задається для текстових, грошових, числових, логічних полів і полів типу дата-час. Можливі значення **ні**, **так**.

10. Стиснення Юнікод (Unicode compression). Це поле включає або виключає стиснення юнікод для текстових полів, полів Memo і гіперпосилань. Текстова інформація в MS Access зберігається в кодуванні unicode.

Це означає, що на кожен символ відводиться не один байт, а два байти. Кодування unicode дозволяє використовувати символи різних національних форматів. В Юнікод всього можна закодувати до 65535 символів. Якщо стискування включено, то всі символи, перший байт яких дорівнює нулю, будуть стискуватися, а при зчитуванні – повертатися в початковий стан.

10. Число десяткових знаків (Decimal places) вказує кількість знаків після коми в зображенні числа для числових і грошових полів.

11. Нові значення (New values). Ця властивість використовується тільки для лічильника. Вона вказує на те, які значення будуть підставлятися системою в поле при створенні нового запису – послідовні чи випадкові. За умовчанням підставляються послідовні значення.

8.2.4. Ключові поля

В попередньому розділі (§ Ключі відношень) дано означення різних типів ключів відношень.

В Access найпростіший спосіб створення ключа – це додавання додаткового поля типу лічильник. Після натискування кнопки із зображенням ключа на панелі інструментів, в лівій частині назви поля з'явиться зображення ключика. Це буде означати, що це поле є ключовим. В Access таке поле заповнюється автоматично.

Для встановлення ключового поля із окремих полів таблиці потрібно спочатку виділити, а потім натиснути на піктограму і так само як попередньому в попередньому випадку, зліва від назви поля з'являються ключики. Це означає, що ці поля утворюють складовий ключ (рис. 8.14).

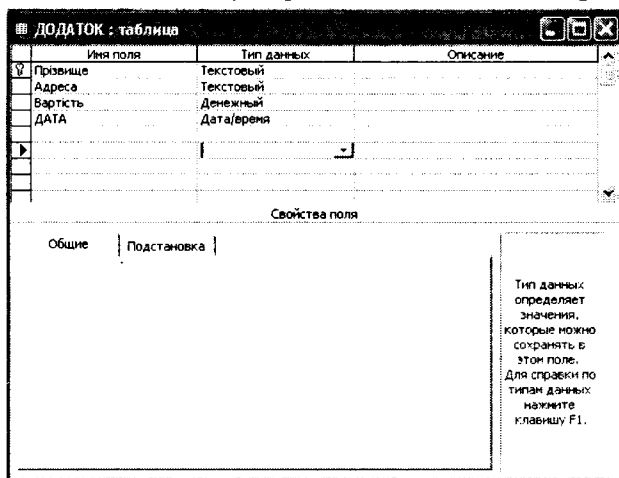


Рис. 8.14. Ключі

Аналогічно встановити ключові поля можна, використавши ліву кнопку миші (контекстне меню).

Для прискорення сортування і пошуку записів використовуються індекси. Індекси так само як і ключі можуть бути простими і складними. В одній таблиці може бути багато індексів, але їх велика кількість сповільнює роботу. Для ключового поля автоматично будується індекс.

Для створення простого індекса потрібно вибрати поле і встановити властивість:

Індексне поле в одне з двох значень: **Ні** (співпадіння не допускаються) або **Так** (співпадіння допускаються). В результаті отримаємо індекс з назвою, що співпадає з назвою поля.

Для створення і редагування індексів потрібно вибрати пункт меню **Вид (View)** далі **індекси**, або натиснути на кнопку **індекси** панелі інструментів. На екрані з'явиться діалогове вікно (рис. 8.15), в якому зі списку полів вибрати поле, для якого будемо будувати індекс, і, якщо потрібно, то задати порядок сортування і властивості індекса.

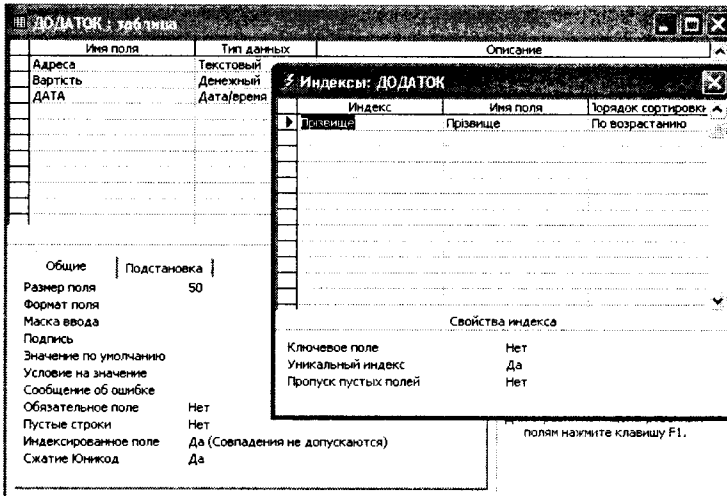


Рис. 8.15. Індекс

Зауваження. Назва складового індексу задається тільки для першого індексу і тільки в цьому полі можливі зміни властивостей індексу. Порядок сортування можна задавати для кожного поля окремо.

8.2.5. Режим таблиць

Режим таблиць – це один із способів створення таблиць в Access. Для того щоб створити таблицю, потрібно у вікні “Створення таблиці” вибрати пункт режим таблиці. На екрані з'явиться вікно (рис. 8.16).

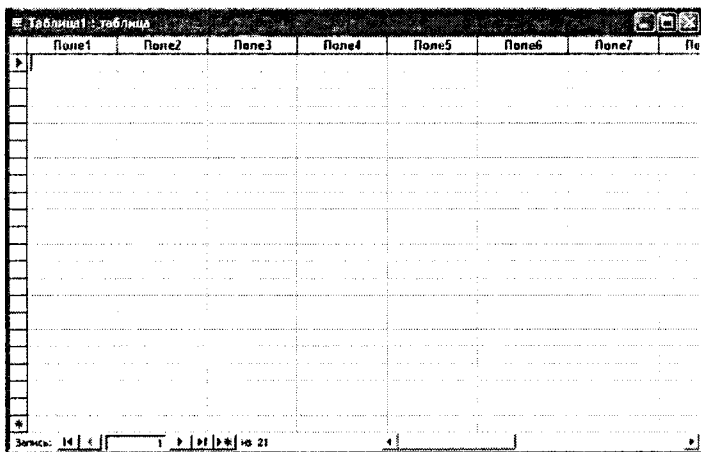


Рис. 8.16. Режим таблиці

Після відкриття у цьому вікні буде 10 стовпців і 21 запис. Назви стовпці мають назви поле 1, поле 2 та ін. Кожне з цих полів можна перейменувати. Для перейменування стовпця потрібно клікнути правою кнопкою мишки на його заголовку. На екрані з'явиться контекстне меню з якого потрібно вибрати пункт перейменувати стовпець. В цю ж таблицю можна вносити дані, нові записи будуть додаватися автоматично. Нове поле можна додати через контекстне меню. Під час запису в таблицю Access проаналізує інформацію і створить відповідний тип даного. Стовпці, в які не введена інформація, в таблицю не включаються.

8.2.6. Введення зовнішніх таблиць

Ще одним способом створення таблиць є імпорт і приєднання таблиць. За допомогою цих команд можна імпортувати або приєднати таблицю з іншого файлу бази даних, лист Excel, інші файли.

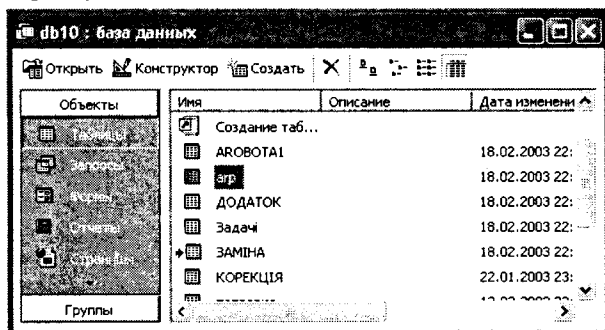


Рис. 8.17. Приєднання таблиці

При виконанні імпорту таблиць таблиця фізично поміщається в файл активної бази даних. Початкова таблиця залишається незмінною під час приєднання таблиці, у файлі відкритої БД буде знаходитися тільки посилення. Змінювати структуру приєднаної таблиці за допомогою конструктора не можна. У вікні бази даних приєднана таблиця зліва від назви буде мати стрілку.

Для здійснення імпорту таблиць потрібно в меню файл вибрати пункт Зовнішні дані. Приєднання здійснюється так сама як і імпорт, але замість пункту Імпорт потрібно вибрати пункт зв'язок з таблицями.

8.2.7. Міжтабличні зв'язки

Після розбиття даних на таблиці а також виділення ключових полів потрібно задати інформацію про те, як зв'язані таблиці між собою, для того, щоб забезпечити логічну цілісність бази даних. Зв'язок між таблицями дає можливість виключити викидання або зміну даних в ключовому полі головної таблиці, якщо з тим полем зв'язані будь-які поля іншої таблиці, а також при викиданні або зміні даних головної таблиці здійснити автоматично викидання або зміну відповідних даних в полях зв'язаних таблиць. Одна з таблиць вважається головною, а друга зв'язаною. Головна таблиця – та, яка бере участь у зв'язках своїм ключовим полем. Зв'язки між таблицями можуть бути трьох видів: один-до-багатьох, один-до-одного, багато-до-багатьох. Всі ці типи зв'язків розглянуті в розділі (ER діаграми). В Access для кожного виду зв'язків існує три способи об'єднання. Вибраний тип об'єднання визначає результат вибірки даних із зв'язаних таблиць.

Внутрішнє об'єднання (Inner Join). В даному випадку в результуючу таблицю входять тільки ті записи, зв'язані поля яких співпадають. Ця операція повністю співпадає з операцією природного з'єднання реляційної алгебри.

Ліве зовнішнє з'єднання (Left Join). З'єднуються всі записи таблиці з боку "один" і тільки ті записи з боку "багатьох", значення зв'язаного поля яких співпадають із значення відповідного поля першої таблиці. Тобто при лівому з'єднанні створюються кортежі внутрішнього з'єднання і додаються до всіх рядки лівого операнда, для яких не має відповідних рядків в правому операнді. Поля кортежів результуючої таблиці будуть приймати нульові значення. Праве зовнішнє з'єднання (Right join). Виконується аналогічно до лівого, але до результатів внутрішнього з'єднання додаються рядки для рядків правого операнда, для яких не було відповідного рядка в лівому операнді.

Для того, щоб встановити зв'язки між таблицями, потрібно закрити всі закриті таблиці і в меню Сервіс вибрати пункт Схема даних. На екрані з'явиться вікно Схема даних. У це вікно потрібно додати таблиці, між якими ми хочемо встановити зв'язок.

Додавання таблиць здійснюється за допомогою меню Зв'язки – додати таблицю або відповідну кнопку панелі інструментів. У вікні, що з'явиться потрібно два рази клікнути на назвах таблиць, які хочемо зв'язати. Вікно додавання таблиць потрібно закрити. Далі для зв'язку полів потрібно перетягнути ключове поле однієї таблиці до поля іншої таблиці. В результаті отримаємо зв'язки між таблицями (рис. 8.18).

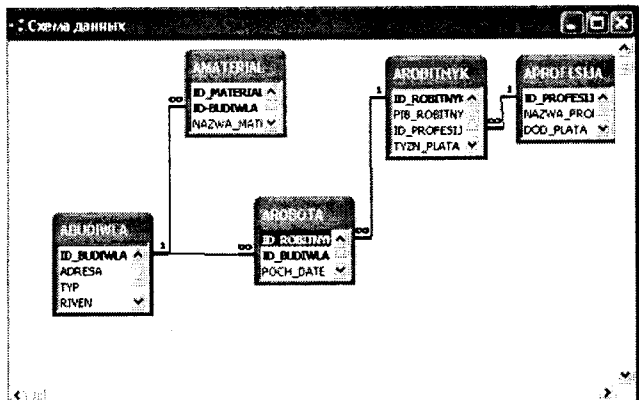


Рис. 8.18. Зв'язки

Якщо перетягнуто поле, яке не є ключовим, до іншого поля, яке теж не буде ключовим, то утворений зв'язок буде не визначений. Він не буде гарантувати цілісності даних. Такі зв'язки на кінцях лінії, що зв'яже дві таблиці не мають символів. Після встановлення зв'язків через меню Змінити зв'язки (або контекстне меню), на екрані з'явиться вікно Зміна зв'язків (рис. 8.19).

The dialog box 'Изменение связей' contains the following information:

- Таблица/запрос:** ABUDIWLA
- Связанная таблица/запрос:** AMATERIAL
- Field mapping:** ID_BUDIWLWA (from ABUDIWLA) is linked to ID-BUDIWLWA (from AMATERIAL).
- Options:**
 - Обеспечение целостности данных
 - каскадное обновление связанных полей
 - каскадное удаление связанных записей
- Тип отношения:** один-ко-многим
- Buttons:** OK, Отмена, Объединение..., Новое...

Рис. 8.19. Зміна зв'язків

В нижній частині цього вікна висвічується інформація про тип зв'язку. Крім того, у цьому вікні розмішені три прапорці, які задають способи забезпечення цілісності даних. Після включення опцій забезпечення цілісності даних стають доступними дві нові опції які дають можливість ввести ще деякі додаткові обмеження на таблицю. Цілісність даних – це множина правил, яка забезпечує допустимі між таблицні зв'язки і не дає можливості на випадкові зміни, або викидання зв'язаних даних.

Якщо після включення цих опцій знищимо запис у головній таблиці, або змінимо одне із значень ключа, то Access автоматично виконає зміни в підпорядкованій таблиці, щоб забезпечити цілісність даних. Для визначення способу з'єднання способу таблиць на кнопці об'єднання вікна зміна зв'язку. На екрані появиться діалогове вікно (рис. 8.20), в якому можна вибрати один із способів об'єднань.

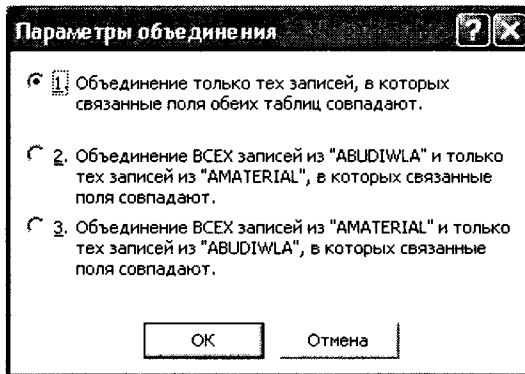


Рис. 8.20. Параметри об'єднань

8.3. Запити QBE

Query-by-Example – це графічна версія реляційної мови SQL. Ця мова створювалася в сімдесяті роки на фірмі IBM. Ця мова отримала широке розповсюдження і в даний час реалізована в більшості СКБД.

Назва запиту відповідає способу його створення. При створенні нового запиту в режимі конструктора відкривається вікно, у верхній половині якого вказується список полів таблиці, які необхідно включити до запиту. Нижня половина вікна – це таблиця **QBE**, в якій задаються умови для запиту. В рядках умови необхідно задати значення полів, які необхідні для запиту.

В таблиці є такі рядки.

Поле – у цьому рядку розміщаються назви полів. У випадку обчислювального поля, в клітинці розміщається вираз, який після обчислення

стає значенням цього поля. Клітинки, що знаходяться нижче належать до цього поля.

Сортування – визначає спосіб впорядкування записів по відповідному полю.

Виведення на екран – визначає, чи буде включатися поле до динамічного набору даних.

Умова вибору – вміщує критерії згідно з якими записи вибираються у динамічний набір даних.

8.3.1. Запити на вибірку

Запити на вибірку містять формулювання запиту до бази даних, який визначається як набір критеріїв для вибірки з однієї або декількох таблиць. Такі запити є найбільш поширеними.

Створення запиту на вибірку можна здійснити шляхом натискання кнопки **запит** у вікні відкриття БД. Далі у відкритому вікні натиснути кнопку **“Створити”** (*Создать*). Створити запит можна власноруч, або за допомогою майстра.

Для того щоб створити новий запит, потрібно у діалоговому вікні *Створення запиту* вибрати мишкою кнопку *Новий запит*. Відкриється вікно **Запит вибірки** і з’явиться діалогове вікно *Додавання таблиці* (рис. 8.21). Мишкою виділяється потрібна таблиця і виконується команда *Додати*.

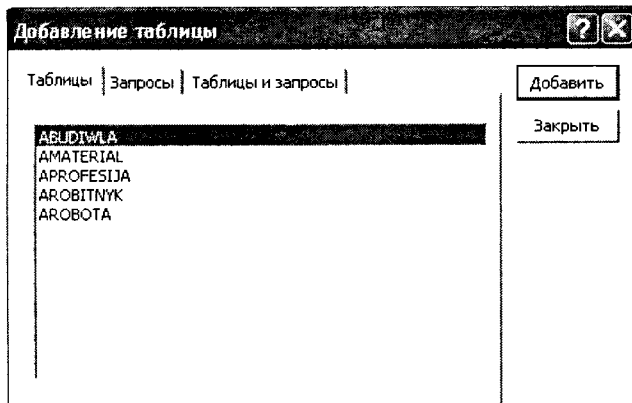


Рис. 8.21. Додавання таблиць

Після того, як вибрана таблиця буде додана, в нижній частині з’явиться таблиця **QBE**. В рядок поле таблиці **QBE** потрібно задати поля, які необхідно отримати в результаті виконання запиту. Нижче буде задаватися назва таблиці, а далі під полем вибираються сортування і умова. Далі **Меню Файл** і підменю **вибір зберегти** запам’ятовується запит.

Приклад. З таблиці AROBITNYK вибрати тих працівників хто має код професії ID_PROFESIJA = 302. Цей запит в таблиці QBE буде мати наступний вигляд (рис. 8.22).

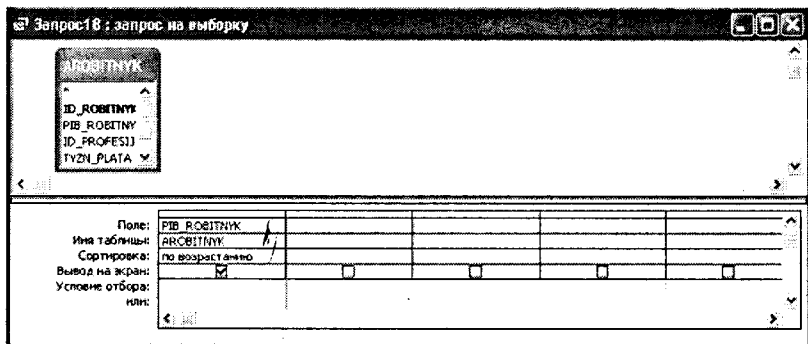


Рис. 8.22. Запит AROBITNYK

8.3.2. Запити з обчислювальними полями

У таблицях недоцільно зберігати дані, які можна отримати за допомогою обчислень з інших полів. Для цього можна використовувати запити з обчисленням. Для цього використовуються поля з обчисленнями. В обчислювальних полях можна здійснювати дії із значеннями різних полів того самого запису, або відповідних записів із різних таблиць. В добре спроектованій базі даних зберігаються тільки основні дані, з яких завдяки обчисленням можна отримати нові дані. Це дає можливість зберігати місце на диску і кожен раз після виконання запиту отримувати нові результати обчислень.

Наприклад; нехай потрібно порахувати, скільки одиниць зарплати отримають робітники за 4 тижні роботи разом з доплатою. Для створення такого запиту таблиці AROBITNYK, APROFESIJA перенесемо у вікно бланку запитів, далі виберемо на сітці QBE потрібні поля PIB_ROBITNYK; TYZN_PLATA таблиці AROBITNYK і DOD_PLATA таблиці PROFESIJA

$$[TYZN_PLATA] * 4 + [DOD_PLATA]$$

Щоб додати в таблицю стовпчик потрібно в рядку *Поле* сітки QBE вписати вираз

$$[TYZN_PLATA] * 4 + [DOD_PLATA]$$

При переході до іншої рубрики вираз буде виконуватися автоматично. Цьому стовпчику буде дана назва "вираз 1". Стовпчику можна задавати власну назву, для цього перед виразом потрібно написати назву після якої поставити дві крапки. Наприклад,

$$ZARPLATA: [TYZN_PLATA] * 4 + [DOD_PLATA]$$

На сітці QBE цей запит матиме наступний вигляд (рис. 8.23).

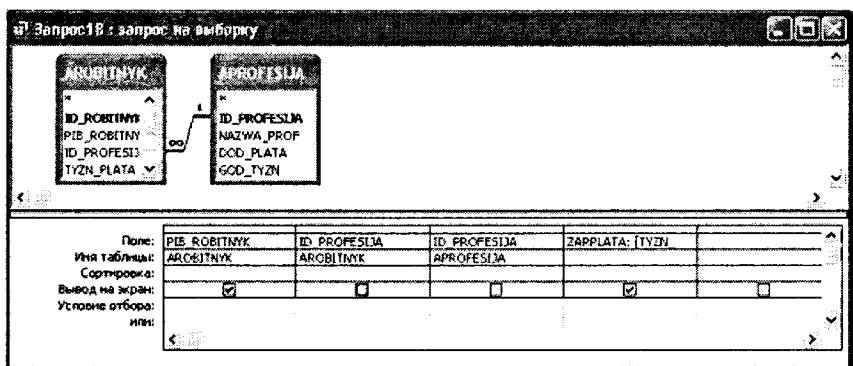


Рис. 8.23

В результаті запиту отримаємо таку таблицю (рис. 8.24).

PIB_ROBITNYK	ZARPLATA
Петренко М.	94,00 грн.
Іванчук М.	98,00 грн.
Савенко Р.	98,00 грн.
Коваленко П.	110,80 грн.
Сидоренко В.	112,00 грн.
Ільницький М.	114,00 грн.
Борис М.	116,00 грн.
Петренко С.	104,00 грн.
Бойчук Є.	120,00 грн.
Войтович Л.	120,00 грн.
Лабунець М.	122,00 грн.
Микитин Л.	130,00 грн.
Ситник Ф.	132,00 грн.

Запис: 1 из 15

Рис. 8.24

8.3.3. Запити для впорядкування записів в таблиці

За допомогою запитів записи таблиці можна сортувати за відповідними полями в такій послідовності, яка потрібна користувачу. Для того щоб створити такий запит потрібно у вікні БД натиснути кнопку **Запит**, далі вибрати створення в режимі конструктора. З вікна “**Додавання таблиці**” вибрати таблицю, яку потрібно відсортувати. Виберемо таблицю AROBITNYK і впорядкуємо за алфавітним порядком за прізвищами по зростанню. Для цього в рядку **Поле** таблиці QBE задамо поле PIB_ROBITNYK, нижче в рядку **Сортировка** виберемо режим сортування

“по возрастанию” в рядку “Вывод на экран” увімкнемо прапорець. В таблиці QBE запит буде мати наступний вигляд (рис. 8.25).

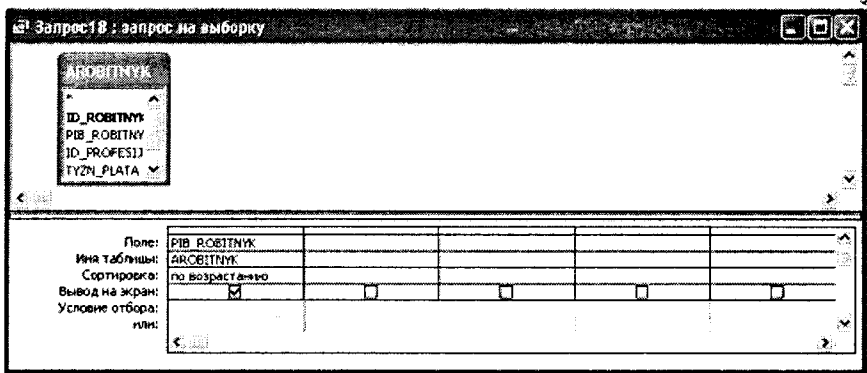


Рис. 8.25. Сортування

Виконання цього запиту у результаті дасть список робітників в алфавітному порядку.

8.3.4. Групувальні запити

Групування використовуються для об'єднання записів, які мають однакові значення в полях, що групується в один запис. Створювати запит можна також вказуючи поля для групування записів і виконання відповідних обчислень. Такі запити можна створювати, як з однієї, так і декількох таблиць, які зв'язані між собою. Наприклад потрібно створити запит для двох зв'язаних таблиць ABUDIVLA та AMATERIAL, в якому згрупуємо дані по типу будівлі (поле TYP) і знайдемо кількість записів для кожного типу. Для цього натиснімо кнопку запис у вікні бази даних і виберемо за допомогою конструктора з вікна додавання таблиць додavimo таблиці ABUDIVLA та AMATERIAL. Далі в таблиці QBE виберемо поля TYP, NAZVA, MATERIAL, ТА KILKIST. Після цього в меню вид виберемо пункт Групповая операция. Після чого в бланку запитів QBE з'явиться рядок Групповая операция проти кожного поля. Далі в стовпчику того поля по якому буде здійснюватися групування потрібно натиснути на клітинку групповая операция. Після розкриття списку на екрані буде видано перелік функцій за допомогою яких можна виконувати групувальні операції. Вибрати потрібну операцію. В нашому випадку візьмемо функцію COUNT. Таблиця QBE з цим запитом буде мати такий вигляд (рис. 8.26).

Оскільки групування відбулося, то кількість записів в кожному рядку є одиниця.

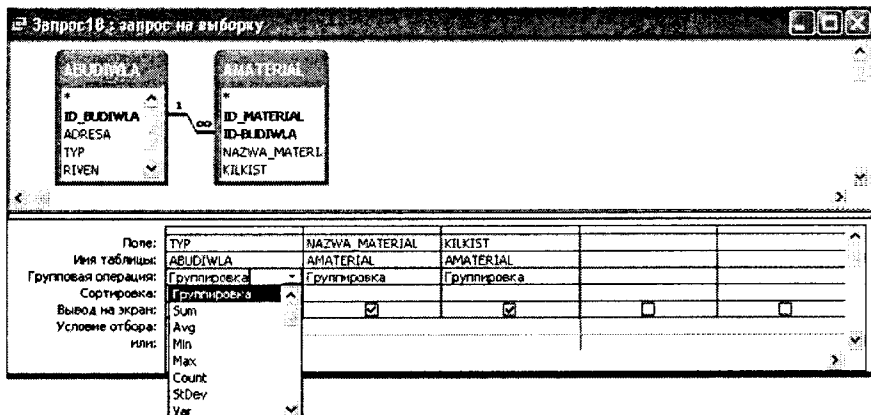


Рис. 8.26. Групування

В Access можна використовувати наступні групувальні функції.

- SUMM.** Обчислення суми значень поля;
- AVG.** Обчислення середнього арифметичного значення поля;
- MIN.** Пошук мінімального значення поля;
- MAX.** Пошук максимального значення поля;
- COUNT.** Підрахунок кількості значень у полі;
- StDev.** Обчислення середньоквадратичного відхилення для значень поля;
- Var.** Обчислення дисперсії значень поля;
- First.** Вибір значення з першого запису;
- Last.** Вибір значення з останнього запису, для встановлення поточного порядку сортування.

8.3.5. Запити з умовами вибору (параметрами)

У розглянутих запитах для умов відбору використовувались константи, але замість констант у виразах умов відбору можна використовувати змінні (параметри). Параметри – це довільні текстові рядки в квадратних дужках. Для того, щоб встановити умову вибірки використовуваться вирази, в яких можна використовувати операції порівняння “>”, “<”, “>=”, “<= “, “=”, логічні операції __ AND, OR, а також службові слова розглянуті в SQL.

Оператори	Значення оператора	Приклад	Значення прикладу
<i>Оператори порівняння</i>			
=	Рівність	=Бойчук	Тільки ім'я Бойчук
>	Більше ніж	>5000	

<	Менше ніж	<05.04.2003	Менше ніж (більше ніж) 5 квітня 2003
>=	Більше рівне	>=M	Більше або рівно літері M
<=	Менше рівне	<=05.04.2003	Більше або рівно 5 травня 2003 року
<>	Не рівно	<>UA	Не рівно PL
Between	Мід двома значеннями	Between 15 and 25	Число від 15 до 25
In	Список значень	In(UA, GB, USA)	UA, GB, lub USA
Is Null	Поле порожнє	Is Null	Запис, який <i>не має</i> значення в полі
Is Not Null	Поле не є порожнє	Is Not Null	Запис, який <i>має</i> значення в полі
Like	Описує зразок	Like c*	запис починається з букви "с"
Оператори логічні			
And	Обидва значення спраджуються	>=1 And <=10	між 1 і 10
Or	Одне зі значень справджується	UA or USA	UA або USA
Not	хибно	Not Like ma???	Записи, які не розпочинаються літерами "ma" з трьома іншими знаками
Спеціальні знаки			
?	Одинарний знак	8?-791	Номер телефону з будь яким другим знаком
*	знаки	(8245)*	Текст, що починається з (8245). Наприклад номер факсу.
[назва поля]	інші поля із запиту	<[TYZN_PLATA]	записи, в яких значення поля є менші від значення в полі „TYZN_PLATA”

Наприклад, для того, щоб вибрати робітників TYZN_PLATA яких більша 15.00 і менша 20.00 інтервал відбору можна записати у такому вигляді: TYZN_PLATA задати умову > 15.00. В рядку умови відбору BETWEEN [введіть нижню границю зарплати] AND [введіть верхню границю зарплати].

Під час запуску на виконання запиту, що містить параметри, на екран видається діалогове вікно, в яке потрібно ввести значення параметрів. Введені таким чином у вікно значення параметра підставляються в умову відбору.

Запити такого типу мають переваги, тому, що зручніше задавати значення для умов відбору при запуску запиту на виконання, а не в таблиці QBE в режимі конструктора.

Запити з параметрами зручно використовувати в ролі джерела даних для форми. В формі створюється елемент керування, що має назву параметра. Якщо в запиті використовується форма, то діалогове вікно для введення параметра не відображається. Введення параметра здійснюється у відповідному елементі керування форми.

Система Access автоматично визначає тип даних, що введені. Але користувач має можливість сам задавати тип параметру через діалогове вікно **Параметри**. Це вікно можна вивести на екран через контекстне меню (права клавіша мишки) і підменю Параметри. На екрані появиться вікно Параметри Запиту (рис. 8.27).

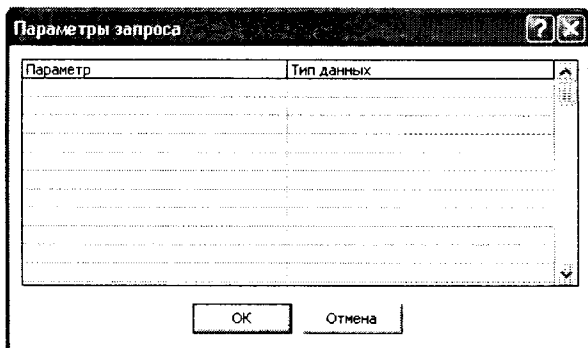


Рис. 8.27

8.3.6. Запити на створення таблиці

Запити на створення таблиці від звичайних запитів на вибірку відрізняються тим, що замість тимчасового зберігання даних утворюють в результаті нову таблицю, в якій зберігаються результати запиту.

Особливим випадком застосування запитів, що створюють таблиці, є запам'ятовування зібраних з різних таблиць даних в одній допоміжній таблиці. В деяких випадках створення таких тимчасових таблиць є необхідне, коли система Access сильно сповільнює роботу під час виконання запитів, побудованих на запитах, для яких джерелами є об'єкти-запити. В новій таблиці тип даних, розмір і поля унаслідуються з полів-джерел, інші атрибути приймають стандартні установки

Для того, щоб створити такий запит потрібно:

1. У вікні бази даних викликати потрібний запит в режимі конструктора запитів. В нашому випадку (Запит РОБ_ПОЧ_ДАТА_БУД_ПРОФ,

що створює таблицю про робітників, їх професії та будівлі, на яких вони працюють.

2. Виберемо таблиці, AROBOTA, AROBITNYK, APROFESIA та ABUDIWLA. В таблиці QBE це запит буде мати такий вигляд (рис. 8.28).

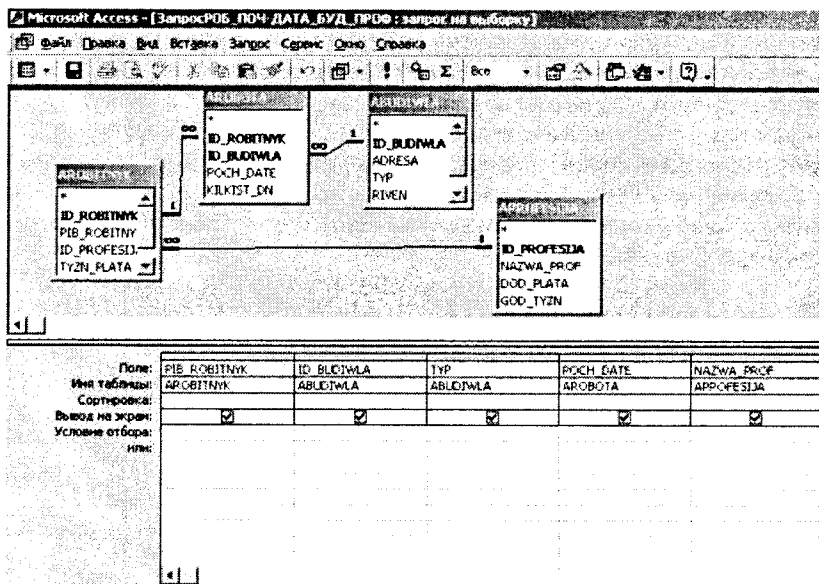


Рис. 8.28. Запит РОБ_ПОЧ_ДАТА_ПРОФ

3. У спадному меню Make Table Query натиснемо праву кнопку мишки. На екрані з'явиться вікно (рис. 8.29), куди потрібно ввести назву таблиці.

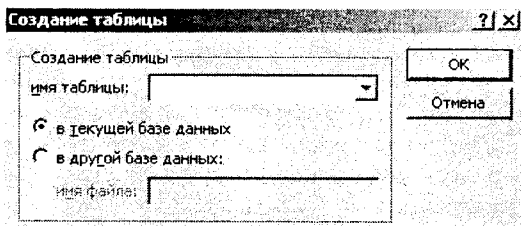


Рис. 8.29. Создание таблицы

Після введення назви отриману таблицю можна використати для створення запитів і запуску цього запиту на екран буде видано підказку про підтвердження створення нової таблиці. Потрібно натиснути кнопку ТАК. Далі натиснувши кнопку мишки на таблиці можна переглянути утворену таблицю. Вона має вигляд, зображений на рис. 8.30.

PIB	ROBITNYI	ID_BUDIWLA	TYP	POCH_DATE	NAZWA_PROF
Волошин М.		700	Університет	15.01.2002	Електрик
Волошин М.		720	Офіс	26.01.2002	Електрик
Іванчук М.		705	Будинок	21.01.2002	Електрик
Іванчук М.		720	Офіс	26.01.2002	Електрик
Савенко Р.		700	Університет	16.01.2002	Електрик
Савенко Р.		720	Офіс	26.01.2002	Електрик
Коваленко П.		700	Університет	25.01.2002	Штукатур
Коваленко П.		720	Офіс	10.02.2002	Штукатур
Сидоренко В.		720	Офіс	10.02.2002	Штукатур
Ільницький М.		705	Будинок	23.01.2002	Штукатур
Борис М.		700	Університет	25.01.2002	Штукатур
Борис М.		720	Офіс	10.02.2002	Штукатур
Петренко С.		705	Будинок	05.02.2002	Маляр
Петренко С.		720	Офіс	12.02.2002	Маляр
Бойчук Є.		705	Будинок	01.05.2002	Маляр
Бойчук Є.		720	Офіс	12.02.2002	Маляр
Войтович Л.		700	Університет	20.01.2002	Паркетчик
Ситник Ф.		705	Будинок	06.02.2002	Столяр

Рис. 8.30

Наприклад, використаємо отриману таблицю для пошуку робітників, хто розпочав роботу 15.01.2002 р. Цей запит в таблиці QBE має наступний вигляд (рис. 8.31).

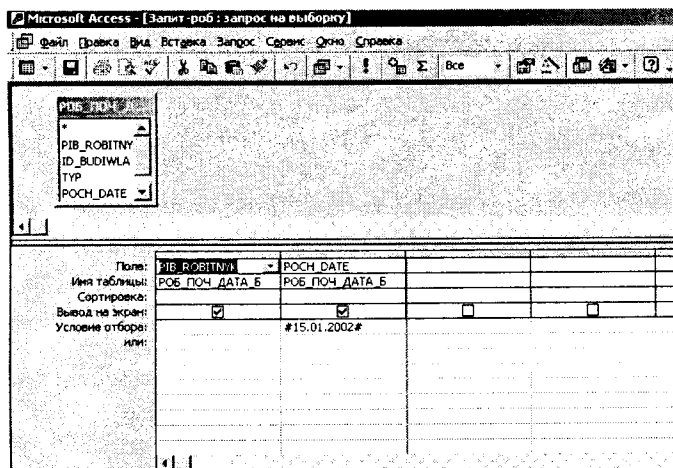


Рис. 8.31

Діалогове вікно для задання параметрів може задавати користувач у зручній для нього формі, яке дасть можливість введення або вибирання відразу багатьох параметрів.

8.3.7. Перехресні запити

Перехресні запити – це один з різновидностей запитів на вибірку. Ці запити призначені для більш глибокого аналізу інформації, що зберігається в

таблицях. Результати перехресного запиту групуються по двох наборах даних, один з яких розміщений в лівому стовпці (стовпцях) таблиці, а другий у верхньому рядку. В решті стовпців таблиці розміщуються результати статистичних розрахунків (Sum, Count та ін.), які виконуються над даними.

Припустимо, що потрібно створити запит за датами закінчення робіт на будовах окремими робітниками. Для цього створимо новий запит.

1. У вікні натиснемо кнопку Запрос \ Создать і перейдемо в режим конструктора. Додамо до нього такі таблиці: AROBITNYK, AROBOTA, ABUDIWLA.

2. Натиснути праву кнопку мишки і в контекстному меню вибрати пункт Тип запиту \ Перехрестний бланк запиту QBE, перетягнути поля PIB_ROBITNYK з таблиці AROBITNYK і TYP з таблиці ABUDIWLA. В наступне поле введемо: DATA:[POCH_DATE]+[KILKIST_DN]

3. Для поля PIB_ROBITNYK встановимо властивість Перекрестная таблица в заголовки строк, а властивість Сортирування по зростанню. Для поля TYP Перекрестная таблица встановимо в значення Заголовки столбцов. Для поля Data властивість Перекрестная таблица потрібно встановити в Значение. В останнє поле – Групповая операция в MAX. Цей запит в таблиці QBE буде мати такий вигляд (рис. 8.32).

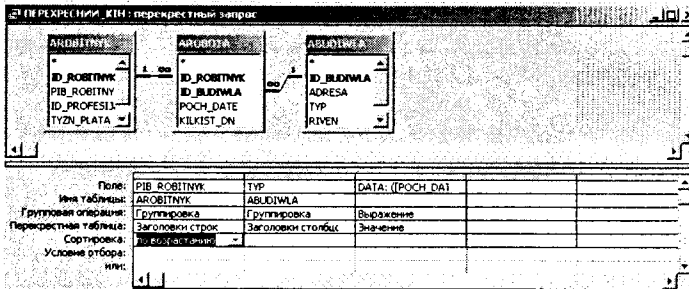


Рис. 8.32

У результаті виконання цього запиту отримаємо таблицю (рис. 8.33).

PIB_ROBITNYK	Будинок	Офіс	Університет
Бойчук Є.	06.05.2002	17.02.2002	
Борис М.		15.02.2002	04.02.2002
Войтович Л.			25.01.2002
Волошин М.		03.02.2002	25.01.2002
Іванчук М.	23.01.2022	03.02.2002	
Ільницький М.	26.01.2002		
Коваленко П.		15.02.2002	04.02.2002
Петренко С.	10.02.2002	17.02.2002	
Савенко Р.		03.02.2002	26.01.2002
Сидоренко В.		15.02.2002	
Ситник Ф.	08.02.2002		

Рис. 8.33

8.3.8. Запити вилучення

У базі даних може зберігатися велика кількість даних, які в певний момент є неактуальними. Наявність таких даних в базі сильно сповільнює роботу системи. Очевидно, що деякі з неактуальних даних на всякий випадок потрібно зберігати протягом якогось часу. Однак найкраще зберігати такі дані в окремих таблицях бо навіть в іншій базі даних. Після архівації потрібно усунути непотрібні дані з поточних таблиць. Найшвидше усунення непотрібних даних можна виконати за допомогою запиту на видалення, який дає можливість викидати з таблиць цілі рядки на підставі будь-яких заданих критеріїв.

Єдиним обмеження на викидання рядків зв'язане з порушенням цілісності даних. Наприклад, для таблиці зі сторони 1 задекларовано зв'язок один до багатьох. Не можна викидати кортежі, які мають свої посилання в таблиці до сторони багато, якщо не включена опція каскадного викидання.

За допомогою запитів вилучення можна вилучати певні записи з таблиці. Це можна робити вручну за допомогою команд Правка\Удалить. Але вилучення можна робити також за допомогою запитів. Для надійності спочатку створюють запит вибірку, за допомогою якого переглядають ті запити які можуть бути вилучені. Якщо в результаті запиту вибірки присутні лише ті запити, які підлягають вилученню, то такий запит-вибірку можна перетворити в запит-вилучення.

Розглянемо, як перетворити запит вибірку у запит вилучення. Нехай потрібно вилучити робітників з кодом професії 302. Для цього в головному меню Запрос необхідно спочатку вибрати Удаление, а потім виконати Выполнить. Після виклику команди Выполнить на екрані з'явиться вікно, в якому буде вказана кількість записів, що можуть бути вилучені.

Щоб створити запит на викидання потрібно:

1) спочатку створимо запит на вибірку, який буде містити таблиці, з яких потрібно видалити записи. В нашому випадку таблицю *ROP_DATY_PRO*;

2) Далі після натискування правої кнопки мишки на екрані з'явиться випадаюче меню в якому потрібно вибрати Тип запроса \ Удаление, після виконання цієї команди в бланку запиту QBE з'явиться рядок Викидання (delete), а в клітинках Условиє;

3) За допомогою мишки символ *(зірочка) і списку полів таблиці потрібно перемістити бланк QBE. В рядку Удаление в стовпці цього поля з'явиться фраза Из (from). Так можуть бути викинуті всі поля таблиці.

У рядку таблиці QBE Викидання під назвами полів з'явиться значення Умова (QUERY) в рядок умова відбору для цих полів потрібно задати необхідні умови. Цей запит QBE буде мати наступний вигляд (рис. 8.34):

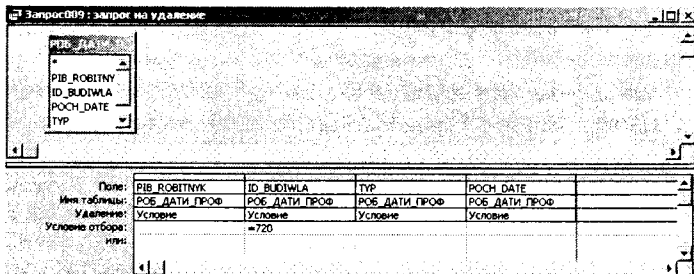


Рис. 8.34

4) запит записів, що видаляються, можна попередньо переглянути, натиснувши кнопку зображення запиту (QUERYVIEW) на панелі інструментів. Щоб повернутися в режим конструктора, потрібно цю кнопку натиснути ще раз;

5) для видалення записів потрібно запустити цей запит на виконання.

8.3.9. Запити на оновлення записів в таблицях

Запити на оновлення записів – це процедури автоматичного перетворення даних у таблиці. Виконання запитів на оновлення спричинює зміни в початкових таблицях. Запити на вибірку таких змін не здійснюють.

Загальна схема створення таких запитів є така:

1) в режимі конструктора створити новий запит і додати до нього потрібні таблиці;

2) у випадаючому меню після натискання лівої клавіші мишки вибрати Тип запроса / Запрос на обновление;

3) у бланк QBE запиту додаються з таблиць ті поля, які потрібно змінити. В рядку Обновление вказуються вирази, значення яких присвоюються вибраним полям. Крім того, у бланк QBE потрібно задати ті поля, за якими необхідно провести відбір, задавши умови. Наприклад, робітникам, що мають професію маляра, збільшити зарплату на 10% (код професії дорівнює 302). Таблиця QBE в цьому випадку буде мати такий вигляд (рис. 8.35).

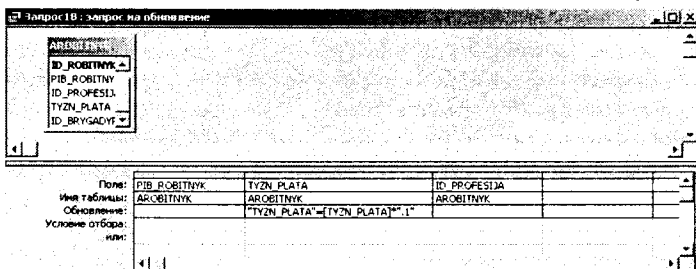


Рис. 8.35

Після запуску запиту на виконання на екрані з'явиться діалогове вікно в якому вказуються кількість запитів, що оновлюється та запитання про продовження операції оновлення (рис. 8.36).

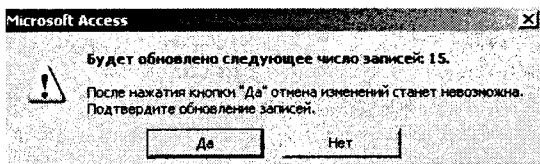


Рис. 8.36

8.3.10. Запити на додавання записів

Запити на додавання записів – це такі запити, які доповнюють існуючі таблиці даними, вибраними з інших таблиць, або запити, які додають нові записи. При виконанні такого типу запитів повинна зберігатись узгодженість типу даних, що додаються з типом полів таблиці до якої додаються дані. Також повинно зберігатись значення ключа цієї таблиці. Таблиця в яку додаються записи може бути як у файлі поточної бази даних так і у зовнішньому файлі бази даних. Записи додаються в кінець цільової таблиці. Запит на додавання записів створюється в такий спосіб:

1. Створимо спочатку запит на вибірку з однієї або декількох взаємоз'язаних таблиць. Для цього з таблиць AROBOTA, AROBITNYK, APROFESIYA та ABUDIVLYA виберемо поля, які хочемо додати до таблиці РОБ_ПОЧ_ДАТА_БУД_ПРОФ. Це поля РІВ_РОБИТНИК, РОСН_ДАТА, ТУР, NAZVA_PROFESIYI.

Access не вимагає, щоб у вбудованому запиті кількість полів була рівна кількості полів в таблиці, куди додаються дані. Можна записувати дані тільки до вибраних полів, не беручи до уваги інших. Важливим є те щоб в запиті було поле що відповідає полю ключа таблиці куди додаються дані. Якщо головним ключем є поле типу лічильника, то в такій ситуації Access сам визначає значення головного ключа.

2. Після натиснення правої клавіші миші у випадяючому меню виберемо Тип запиту / Добавление (APPEND QUERY). Після цього відкриється діалогове вікно Добавление (рис. 8.37).

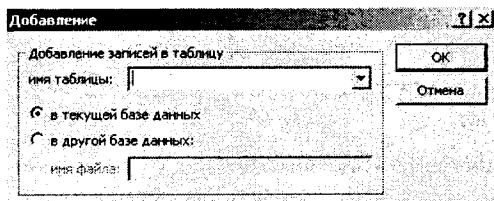


Рис. 8.37

Так само як у випадку запиту, що творить таблицю, тут також потрібно ввести назву таблиці, куди будуть додаватися дані. Шаблон QBE для запиту, що додає замість рядка Показати має рядок Добавление. Цей рядок заповнимо назвами полів, які відповідають відповідним полям таблиці, в яку буде здійснюватись додавання. Крім цього, як і в попередньому запиті на оновлення в бланк QBE можуть бути включені поля, за якими здійснюється відбір. Ці вносяться в рядок Условие отбора (Criteria).

3. Для вказування в рядку Додавання (Append To) назв полів таблиці - результату потрібно в кожній комірці рядка відкрити список і вибрати потрібну назву. Якщо вибрані поля мають однакові назви, то назви полів у рядок Додавання (Append To) вносяться автоматично. Запит на додавання в таблиці QBE буде мати такий вигляд (рис. 8.38).

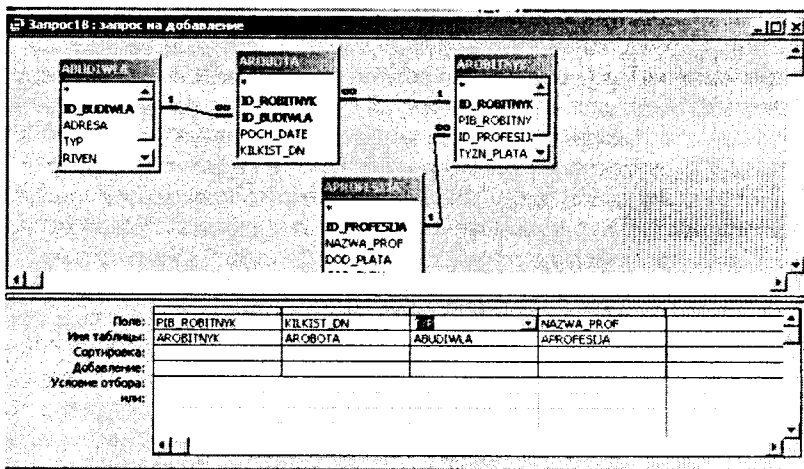


Рис. 8.38

4. Для попереднього перегляду записів, які плануються додати в таблицю потрібно натиснути кнопку Зображення запиту (Query View) на панелі інструментів. Повернутися назад в режим конструктора здійснюється за допомогою цієї ж кнопки.

5. Для додавання записів потрібно виконати Запит натиснути (!) на панелі інструментів. На екрані появиться діалогове вікно про кількість запитів, що оновлюються і про продовження оновлення.

8.3.11. Запити, що здійснюють вибірку записів, що повторюються

Цей запит визначає, чи містить таблиця значення, що повторюються, в одному або декількох полях. Побудувати такий запит можна за допомогою майстра. Для цього:

1. У вікні бази даних натиснути кнопку Запит\Створити. Відкриється вікно (рис. 8.39).

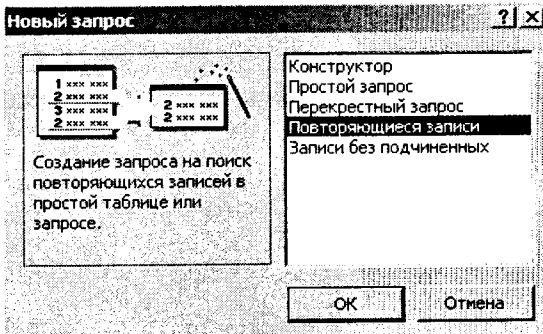


Рис. 8.39. Запити, що повторяться

У ньому зі списку вибрати пункт **Повторяющиеся запросы**. Після чого на екрані з'явиться перше вікно майстра (рис. 8.40).

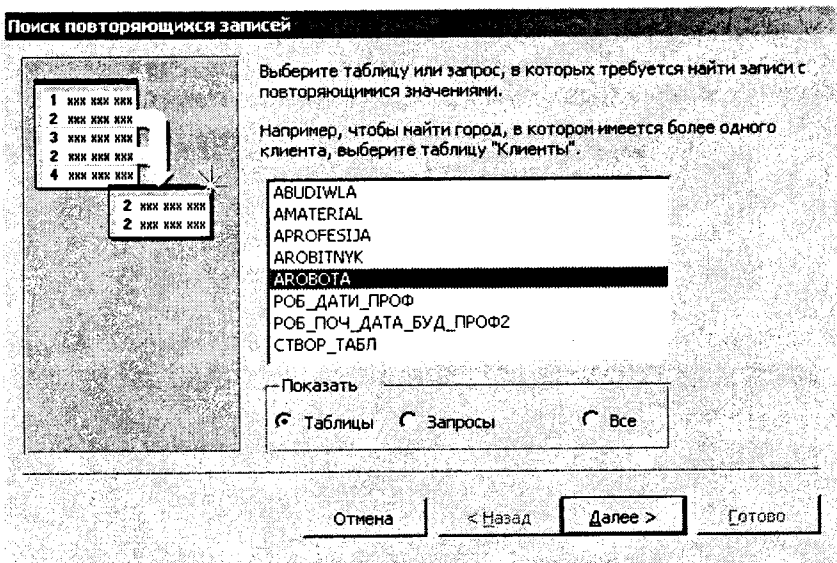


Рис. 8.40

Після вибору відповідної таблиці (AROBOTA) натиснути кнопку **Далее**.

На екрані появиться друге вікно майстра (рис. 8.41).

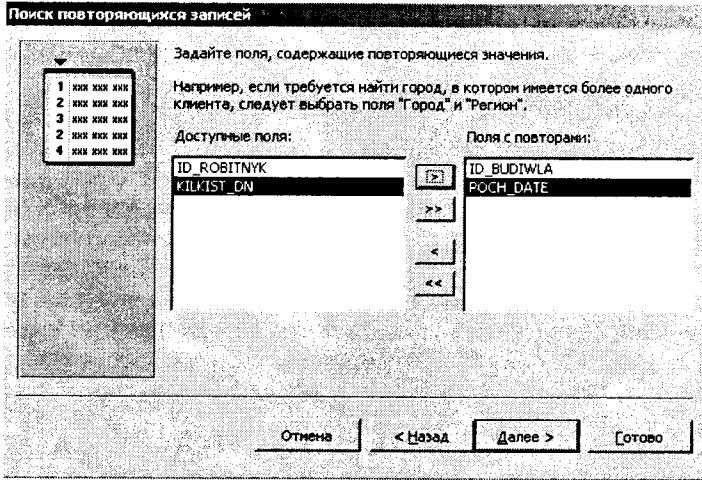


Рис. 8.41

Зі списку доступних полів вибрати ті значення, які будемо використувати для вибору дублікатів (ID_BUDIWLA, POCH_DATE)

У третьому вікні майстра потрібно натиснути Далее, а у четвертому Готово. Після чого система створить запит **Поиск повторений** для таблиці AROBOTA, в якій будуть вибрані записи, що повторюються (рис. 8.42).

Поиск повторений для AROBOTA : запрос на выборку			
ID_BUDIWLA	POCH_DATE	Повторы	
700	25.01.2002	2	
720	26.01.2002	3	
720	10.02.2002	3	
720	12.02.2002	2	

Рис. 8.42

Цей запит в таблиці QBE буде мати наступний вигляд (рис. 8.43).

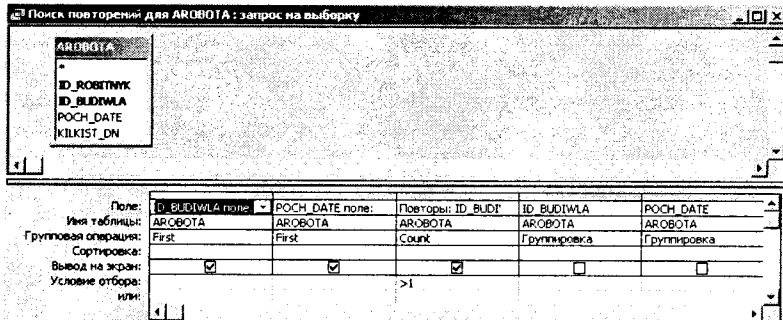


Рис. 8.43

Запит отриманий майстром, можна підкоректувати за допомогою конструктора.

8.3.12. Запити, що здійснюють вибірку записів, які не мають підпорядкованих

Цей запит дає можливість знаходити всі записи вказаної таблиці, які не мають зв'язаних записів в іншій таблиці. Цей запит створюється так само, як в попередньому пункті.

Для цього у вікні бази даних потрібно натиснути кнопку Запит\Створити. У вікні, що відкривається (рис. 8.44), потрібно із списку вибрати пункт Записи без підчинених.

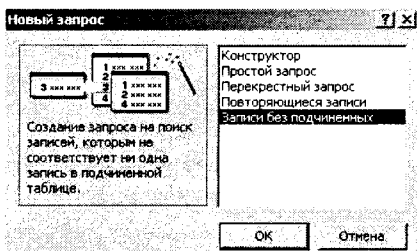


Рис. 8.44

На екрані появиться вікно (рис. 8.45).

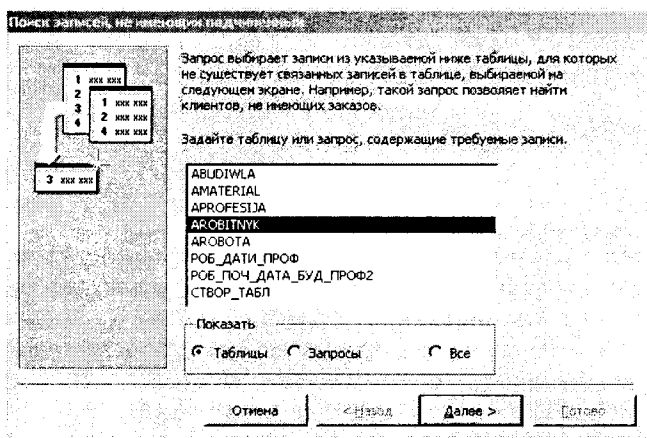


Рис. 8.45

На першому кроці потрібно вибрати таблицю зі сторони один – голівну таблицю (AROBITNYK).

Після натискування кнопки Далее на екрані появиться вікно (рис. 8.46), в якому потрібно вибрати таблицю зі сторони багато (AROBOTA).

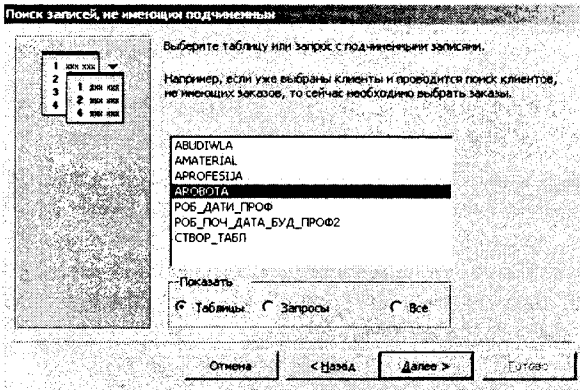


Рис. 8.46

2. Після натискання кнопки **Далее** на екрані появиться вікно, що має такий вигляд (рис. 8.47).

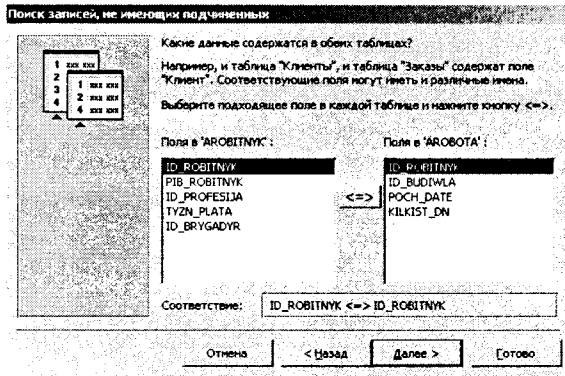


Рис. 8.47

Далі виберемо відповідні поля з обидвох таблиць. Достатньо клікнути на назві поля на одній стороні вікна, на другій стороні, що йому відповідає, а в кінці натиснути на кнопку \Leftrightarrow .

Після цього виберемо поля, які повинні бути відображені в результаті запиту: ID_ROBITNYK, PIB_ROBITNYK, ID_PROFESIJA. В результаті отримуємо (рис. 8.48):

ID_ROBITNYK	PIB_ROBITNYK	ID_PROFESIJA
40302	Лабунец М.	401
40303	Микитин Л.	401
50302	Сидор І.	503
50303	Борисюк І.	503

Рис. 8.48

8.3.13. Властивості запитів

1. Властивості полів запитів. Поля, що отримані в результаті запиту успадковують властивості полів базової таблиці. Якщо змінюються властивості полів то відповідні поля запитів також змінюються. В тих випадках, коли властивості поля запиту було задано в режимі конструктора, то зміна властивості цього поля в базовій таблиці не впливає на нього.

Значення обчислювальних полів, які відображаються в підсумковій таблиці, але не зберігаються в базовій таблиці, не успадковують властивостей полів, що входять в їх опис. Для того, щоб відкрити вікно властивостей поля запиту в конструкторі, потрібно встановити в цьому полі курсор і з контекстного меню вибрати пункт **Властивості**.

У цьому вікні перераховуються тільки ті властивості, які можна змінити.

2. Властивості запитів. Властивості запитів дають можливість задавати запитам певні правила виконання. Для того, щоб відкрити вікно властивостей запиту, потрібно під час створення запиту в режимі конструктора з контекстного меню вибрати пункт **Властивості (Properties)**. На екрані з'явиться вікно Властивості запитів (рис. 8.49).

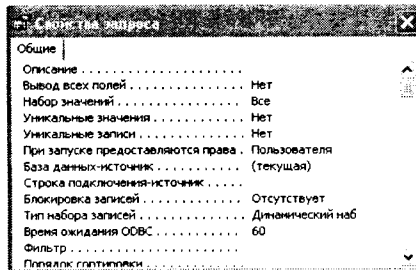


Рис. 8.49. Властивості запитів

Дамо коротку характеристику властивостей запитів.

Опис (Description). Цю властивість можна використовувати для всіх типів записів. В рядку цієї властивості задається опис запиту. Максимальна довжина такого рядка 255 символів. По замовчужанню встановлюється порожній рядок. Заданий опис виводиться у вікні бази даних, якщо об'єкти розміщені в ньому у вигляді таблиць.

Виведення всіх полів (Output All Fields). Ця властивість використовується для запитів на вибірку записів, додавання записів і на створення таблиці. Вона може приймати два значення Так (Yes) або Ні (No). Якщо ця властивість приймає значення Так, то в результуючу таблицю попадуть всі поля з базової таблиці. По замовчужанню встановлюється значення Ні.

× **Набір значень (Top Values).** Ця властивість може використовуватися для запитів на вибірку, на додавання запитів і на створення таблиці. Ця властивість може приймати із списку, що випадає одне з наступних значень: Все, 5, 25, 100, 5% і 25%. Значення Все означає, що будуть виводитися всі

записи, якщо задано ціле число n , то це означає, що будуть виводитися n перших записів, якщо вказано ціле число n і знаком процента, то це означає, що будуть виводитися перші $n\%$ записів. Крім значень зі списку, що випадас, можна задавати будь-яке ціле число (можна також із знаком $\%$). За умовчанням ця властивість приймає значення Все.

Унікальне значення (Unique Galues). Ця властивість може використовуватися, так само як попередня для запитів на вибірку, додавання записів і на створення таблиці. Вона може приймати значення TAK або HI. Значення TAK відповідає заданню в SQL-операторі опції DISTINCT, що означає, що виводитися будуть тільки унікальні значення із результуючого набору даних. За умовчанням встановлюються значення HI.

Унікальні записи (Unique Records). Ця властивість може використовуватися для запитів на вибірку, додавання і викидання записів, а також на створення таблиці. Вона може приймати два значення – TAK або HI. Значення TAK відповідає заданню в SQL-операторі опції DISTINCTROW. Це означає, що будуть виводитися тільки унікальні значення з базових таблиць. Унікальність записів при цьому буде перевірятися для всіх полів таблиці, а не тільки для полів, що включені в запит. Властивості *унікальні значення* і *унікальні записи* не можуть одночасно приймати значення TAK. За умовчанням встановлюються значення HI.

При запуску *надаються права (RunPermissions)* цю властивість можна використовувати для всіх запитів, крім запиту на об'єднання. Вона може приймати значення – Користувача (User's) і Власника (Owner's). Цю властивість можна задати в багатокористувацькому оточенні, щоб перевизначити існуючі права користувача. Значення Власника означає, що користувачу надається набір прав власника на перегляд і запуск запиту. За умовчанням встановлюється значення Користувача.

База даних джерело (Source Database). Так само як попередню властивість, цю властивість можна використовувати для всіх типів запитів, крім запиту на об'єднання. Вона вказує яка база даних є джерелом – поточна чи зовнішня. У випадку використання зовнішньої бази даних в цій властивості вказується шлях до її файлу. За умовчанням встановлюється *поточна*.

Рядок підключення – джерело (Source ConnectStr). Ця властивість доступна для всіх типів записів, крім запиту на об'єднання. Значенням цієї властивості є рядок, що містить інформацію про підключення зовнішньої бази даних. Ця властивість має сенс тільки для баз даних інших форматів (Dbase, Paradox, FoxPro та ін.). При підключенні бази даних MS Access ця властивість ігнорується. За умовчанням встановлюється порожній рядок.

Блокування записів (RecordLocks). Властивість можна використовувати для всіх типів запитів. Вона вказує рівень блокування записів, якщо два користувачі роблять спробу одночасно редагувати той самий запис. Можливі такі значення – *Відсутня (NoLocks)*, *Всіх записів (All Records)* і *Змінюваного запису (edited record)*. Розглянемо окремо кожне значення:

Відсутня (NoLocks) – Якщо два користувачі роблять спробу зберегти зміни, що вносять в один і той же запис, то Access видає попереджуваче повідомлення тому користувачу, який робить спробу зробити це другим. Другий користувач може відмінити свої зміни, або записати пізніше.

Всіх записів (AllRecords) – Всі записи базових таблиць і запитів блокуються на час роботи запиту. Користувачі можуть переглядати записи, але не можуть переглядати записи, але не можуть їх редагувати або викидати.

Змінюваного запису блокується сторінка, що містить запис, який редагується. Блокування знімається в момент переходу користувача на інший запис. Значення за умовчанням для запитів на зміну встановлює **Змістований запис (Edited record)**, для всіх решта – **Відсутня (NoLocks)**.

Тип набору записів (RecordSetType). Цю властивість можна використовувати для запитів на вибірку і перехресних записів. Властивість може приймати такі значення.

Динамічний набір (DynaSet), Динамічний набір (неузгоджений) (Dynaset (Inconsistent Updates)) і Статичний набір (Snapshot). Вона вказує на можливість редагування базових таблиць запиту.

Динамічний набір – в цьому випадку можна редагувати одну базову таблицю або декілька базових таблиць, що мають зв'язок один-до-одного.

Динамічний набір (неузгоджений) – редагування допускається в будь-якому випадку.

Статичний набір – редагувати можна за будь-яких умов.

За умовчанням встановлюється **Динамічний набір**.

Час очікування ODBC (ODBC TimeOut). Властивість можна використовувати для всіх типів запитів задається кількість секунд, які Access буде очікувати до видання повідомлення про помилку (time-out Error). За умовчанням ця властивість встановлюється в 60.

Фільтр (Filter). Ця властивість використовується для запитів на вибірку і об'єднання. Значенням властивості є рядок, подібний до речення WHERE SQL-оператора, але без ключового слова WHERE. Для запитів ця властивість особливого значення не має оскільки всі необхідні умови можна задати в самому запиті.

Порядок сортування (OrderBy). Властивість можна використовувати для запитів на вибірку об'єднання. Вона може містити список полів і порядок їх сортування. Значенням властивості є рядок подібний до речення ORDER BY SQL – оператора, але без ключового слова ORDER BY.

Максимальне число записів (MaxRecords). Ця властивість використовується так само до тих запитів, як і попередня. Вона є цілим довгим числом (Long integer) і вказує максимальну кількість записів, що повертаються в базу даних Access (mdb-файл) із бази даних в бази даних ODBC або в проект Access (adp-файл) із SQL-бази. Записи повертаються в порядку, який вказано в реченні ORDER BY.

Назва підтаблиці (Subdatasheet Name). Цю властивість можна використовувати для всіх типів запитів, крім запитів на викидання. Назву підтаблиці можна вибрати із випадającego списку, що має назви всіх до-ступних таблиць і відповідних записів.

Підпорядковані поля (LinkchildFields) і Основні поля (Link MasterFields). Використовується до запитів такого ж типу, як і попередня властивість. Тут вказуються первинний і зовнішній ключі.

Висота підтаблиці (SubdatasheetHeight). Властивість можна використувати для всіх типів запитів, крім запитів на вилучення. Тут можна задавати числовий вираз, що буде визначати висоту розгорнутої підтаблиці.

Розгорнута підтаблиця (SubdatasheetExpanded). Використовується в таких запитах, як і попередня властивість. Вона може набувати два значення **Так** або **Ні**.

Таблиця-одержувач (DestinationTable), База даних-одержувач (DestinationDb) і Рядок підключення одержувач (DesconnectStr). Цю властивість можна використовувати для запитів на створення таблиці і додавання записів. В цих властивостях вказується таблицях. В цих властивостях вказується таблицях, база даних і її рядок підключення в які додаються записи. Властивості подібні за своїм змістом з властивостями: Таблиця-джерело, База даних-джерело і Рядок підключення-джерело.

Використати транзакцію (UseTransaction). Цю властивість можна використовувати для всіх запитів, що виконують зміни (викидання, додавання, оновлення і створення таблиці). Вона може приймати два значення **Так** або **Ні**. Якщо значення встановлено в **Так**, то запит виконується як транзакція. За умовчанням встановлюється значення **Ні**.

Зупинка при помилці (FailOnError). Цю властивість можна використовувати лише для запитів на викидання і оновлення. Можливі значення **Так** або **Ні**. Задає поведінку запиту при виникненні помилки. В залежності від значення цієї властивості при виникненні помилки запит або переривається, або не переривається. За умовчанням встановлюється значення **Ні**.

Заголовки стовпців (ColumnHeadings). Використовується для перехресного запиту. Значенням цієї властивості є список заголовків стовпців. Властивість використовується для зміни порядку або обмеження кількості стовпців в перехресному запиті.

Запитання для самоперевірки

1. Способи створення таблиць в MS Access.
2. Які ви знаєте типи запитів? Їхні особливості.
3. Створити базу даних про студентів факультету. В базі даних передбачити такі таблиці: СТУДЕНТ (ПІБ; дата народження; фотографія; адреса, телефон) та УСПІШНІСТЬ (предмети; оцінка). Ввести не менше 15 записів.
4. Створити запит, що знаходить середній бал кожного студента.
5. Видрукувати звіт про успішність. Здійснити групування за предметами.

9. СТВОРЕННЯ ПРЕЗЕНТАЦІЙ. Microsoft PowerPoint

📖 План викладу матеріалу:

1. Поняття презентації.
2. Початок роботи з Microsoft PowerPoint.
3. Головне вікно PowerPoint.
4. Підготовка презентації.
5. Приклад створення презентації.
6. Режим сортувальника. Демонстрація.

↔ Ключові терміни розділу

✓ Презентація	✓ Ефекти анімації
✓ Слайд	✓ Сортувальник слайдів
✓ Microsoft PowerPoint	✓ Перехід слайда

9.1. Поняття презентації

На перший погляд сучасні комп'ютерні презентації нагадують дуже популярні колись діафілми, діапозитивами яких тепер є комп'ютерні слайди, а сам комп'ютер виконує роль діапроектора. Роль презентацій у сучасних умовах ведення бізнесу постійно зростає: навіть декілька років тому на великих виставках лише деякі експонати супроводжувались демонстраційним матеріалом, тепер без нього неможливо уявити жоден поважний стенд.

Термін *презентація* походить від англійського «*presentation*» – показ. Зазвичай презентація є слайд-фільмом із рекламно-інформаційним змістом, орієнтованим на певну категорію глядачів.

Застосовують презентації не лише у торгово-економічній діяльності, а й з навчальною метою, без них не обходиться захист курсових, дипломних, наукових робіт. Крім того, бурхливо розвивається ще одна галузь застосування – презентації у всесвітній інформаційній мережі **Internet**.

Серед програм для створення презентацій досить поширеною є **PowerPoint** із пакета **Microsoft Office**.

9.2. Початок роботи з Microsoft PowerPoint

Ця програма володіє рисами, притаманними усім складовим Microsoft Office, тому більшість прийомів роботи нам уже знайомі. Наприклад,

редагування тексту тут відбувається аналогічно Word. Існує декілька варіантів запуску PowerPoint: запуск безпосередньо виконавчої програми будь-яким із передбачених у Windows способів або вибором готового шаблону презентації **Создать документ Office** (рис. 9.1):

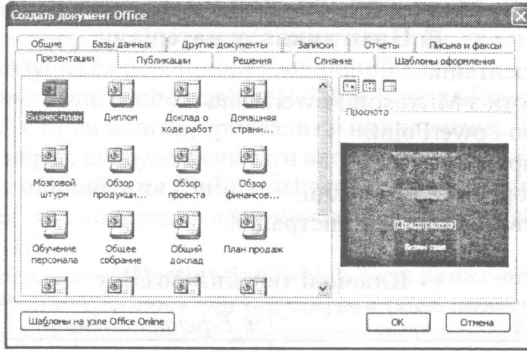


Рис. 9.1

Після вибору бажаного шаблону і натискання кнопки **ОК** запускається вікно програми з обраним шаблоном (рис. 9.2).

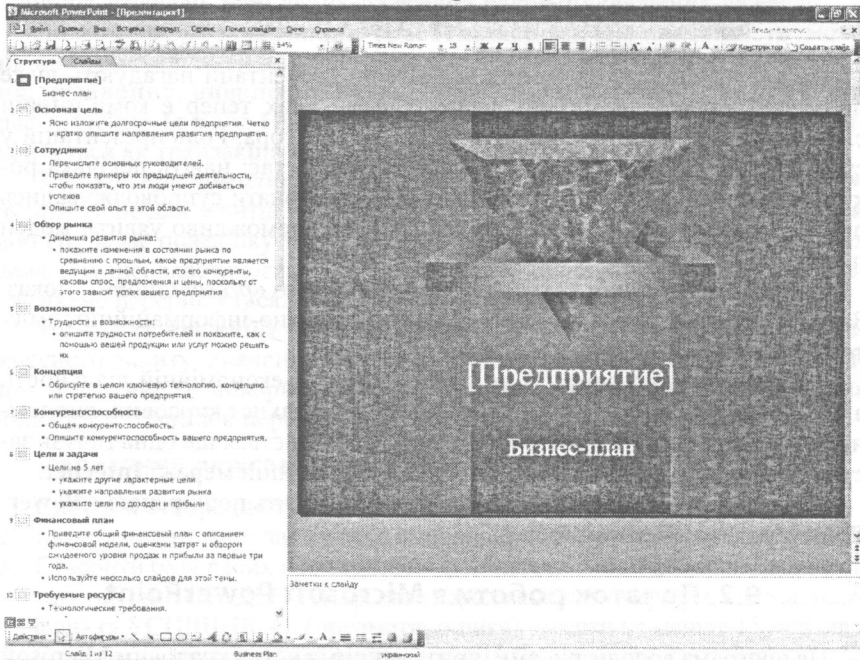


Рис. 9.2

Шаблон визначає місцезоташування об'єктів на слайді (тексту, рисунків, таблиць, діаграм тощо). Для розміщення об'єктів на слайді передбачені порожні рамки, що зветься контейнерами.

У випадку безпосереднього запуску виконавчого файлу **powerpnt.exe** з'явиться вікно програми з розміткою за замовчуванням (рис. 9.3). У полі праворуч доступне меню відкриття файлів, з якими працювали останнім часом, та меню **Мастера создания презентации**.

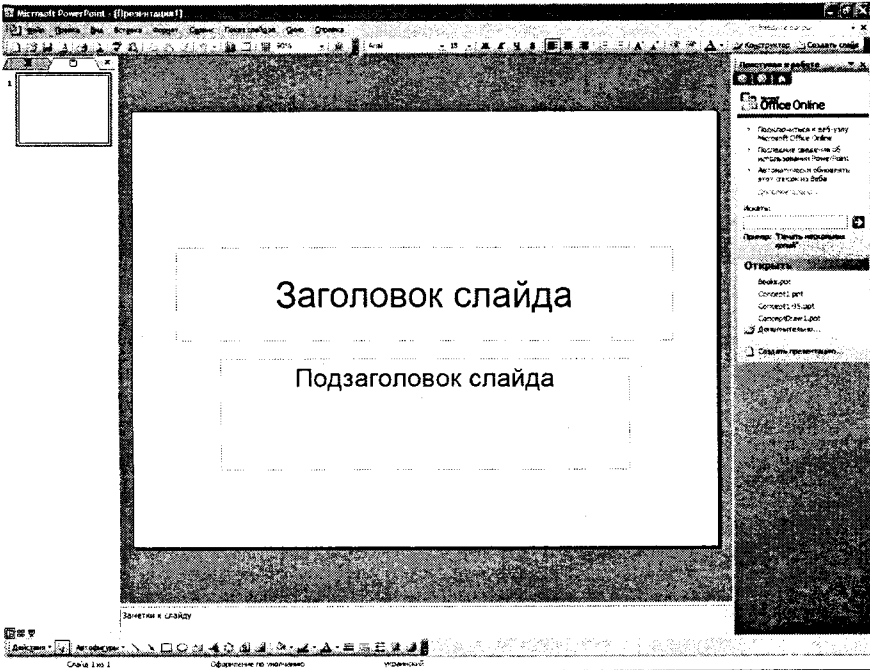


Рис. 9.3

На першому кроці **Мастера создания презентации** (рис. 9.4) до вибору користувача пропонується створити нову презентацію, вибрати шаблон оформлення чи створити презентацію на базі вже існуючої (файл формату PowerPoint має розширення **.ppt**).

При виборі **Мастера автосодержания** (рис. 9.5) останній задасть ряд стандартних запитань: про вид презентації, її стиль, бажані параметри тощо. Кожен із чотирьох кроків діалогу повинен завершуватись натисанням кнопки **Далее**, в кінці роботи Майстра слід натиснути кнопку **Готово**. Майстер видасть декілька слайдів, які, на його думку, повинні відповідати Вашим задумам. Зрозуміло, у переважній більшості випадків, так буде не завжди, отримані слайди слід буде відредагувати.

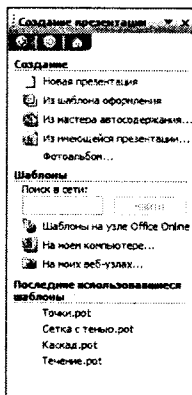


Рис. 9.4

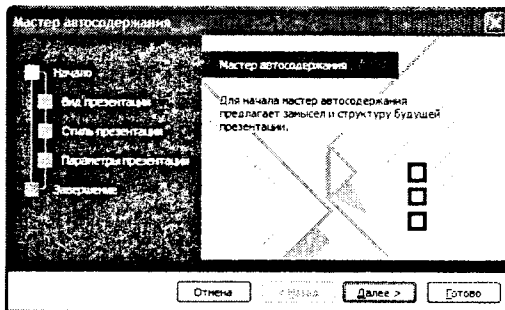
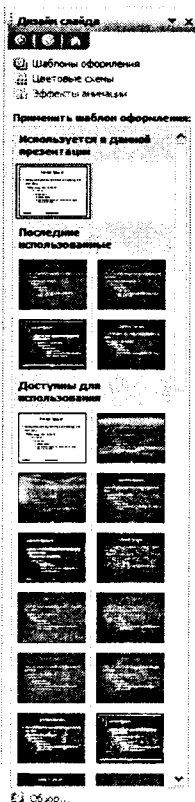


Рис. 9.5



Е4 Облор...

Рис. 9.6

У процесі роботи з **Мастером** создания презентації також можна вибрати **Из шаблона оформления**. В області задач відобразяться готові шаблони презентацій (рис. 9.6). Тут також можна задати варіанти взасмодії кольорів на слайді в закладці **Цветовые схемы** та анімаційні ефекти в закладці **Эффекты анимации**.

Після вибору стилю оформлення процес підготовки шаблона вважається завершеним. Зауважимо, що для можливості скористатися закладками панелі задач необхідно, щоб у діалоговому вікні **Сервис/Параметры** у закладці **Вид** (рис. 9.7) був встановлений прапорець **область задач при запуске**. За замовчуванням він там є.

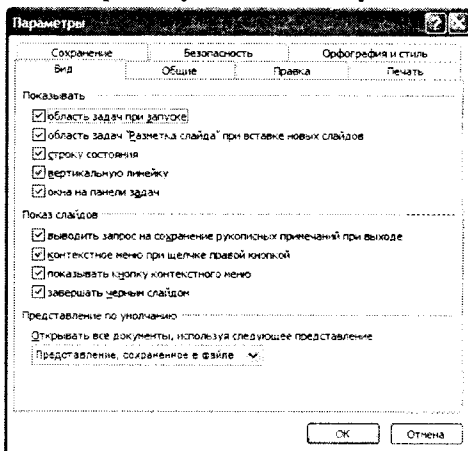


Рис. 9.7

9.3. Головне вікно PowerPoint

Головне вікно програми мало чим відрізняється від вікон інших програм, що працюють у середовищі Windows. Тут теж присутні *заголовок, головне меню* з 9-ти пунктів: **Файл, Правка, Вид, Вставка, Формат, Сервіс, Показ слайдов, Окно, Справка**, також є *панелі інструментів, робоче поле, область задач, стрічка стану* тощо.

Ліворуч кожного слайда розташоване поле, яке закладками перемикається на *режим структури* або *режим слайдів*. На рис. 9.2 показана структура презентації (активна перша закладка). Із закладки видно, що шаблон створення бізнес-плану містить понад 10 слайдів з готовими заголовками та визначеною структурою подання матеріалу. Режим структури необхідний для ознайомлення зі структурою презентації та її зміни. Самі слайди у значно зменшеному вигляді показані у режимі слайдів. У цьому режимі активна друга закладка (наприклад, рис. 9.3).

Режим сортувальника необхідний для визначення порядку черговості слайдів, редагування ефектів та часових інтервалів їх появи. Пересування слайда тут здійснюється мишею, для швидкого переходу в режим редагування слайду достатньо двічі натиснути по ньому мишею.

Внизу під слайдом присутнє текстове *поле для нотаток*, куди можна ввести коментар до слайда.

Режим показу здійснює демонстрацію презентації.

На рис. 9.8 показано, які панелі інструментів активовано за допомогою контекстного меню панелі (те саме можна зробити і викликом з меню **Вид/Панелі інструментов**). Внизу під контекстним меню знаходиться панель **Рисование**. Кнопка **Действия** дає змогу повернути/зсунути об'єкти, визначити порядок взаємного розташування і накладання об'єктів, та багато інших корисних опцій. Кнопка **Автофигуры** містить дев'ять груп різних об'єктів: від ліній, стрілок до виносков та керуючих кнопок. Решта кнопок служать для додання на слайд надписів, рисунків, діаграм, об'єктів **WordArt**, задання кольору, ефекту тіні тощо.

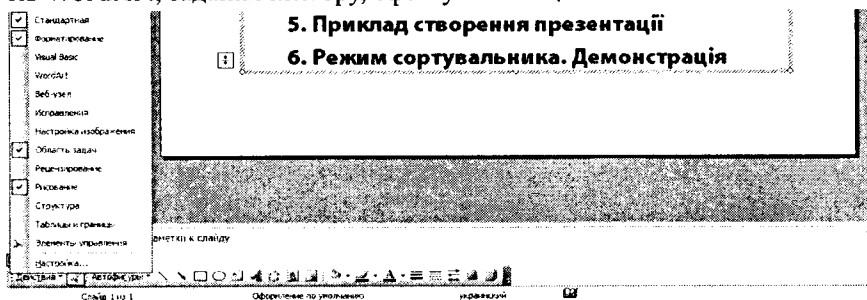


Рис. 9.8

Верхні панелі інструментів (**Стандартная і Форматирование**) не становлять особливого інтересу для користувача, вже знайомого із продуктами сімейства MsOffice: тут продубльовані команди меню **Файл**, можна встановити масштаб відображення, налаштувати шрифт, колір тексту, задати нумерований чи маркований список тощо.

9.4. Підготовка презентації

У випадку порожньої презентації потрібні об'єкти створюють за допомогою піктограм панелі інструментів або з меню **Вставка**. На слайді зі створеною розміткою текстові чи графічні блоки можна «клонувати»: виділивши об'єкт, вибрати **Правка/Дублювати** або просто натиснути **Ctrl+D**. При цьому блок повинен бути виділеним ніби «по краях» – всередині не повинен знаходитись курсор вводу.

Після вибору об'єкта слід вказати його місцезрештування на полі слайда, окреслюючи мишею його межі. Якщо це текстовий блок (**Надпись**), то, натиснувши мишею всередині блоку, можна розпочати набір тексту. Для редагування його параметрів застосовують меню **Формат**, команди контекстного меню чи панель інструментів. Відбувається все це аналогічно засобам Word, у тому числі й форматування самого блоку. Для цього викликають діалогове вікно **Формат Автофигуры**, виділивши блок (без курсора вводу всередині) і клацнувши двічі по його краю (або вибрати з меню **Формат/Прототип**). На закладці **Надпись** (рис. 9.9) встановлюють необхідні параметри.

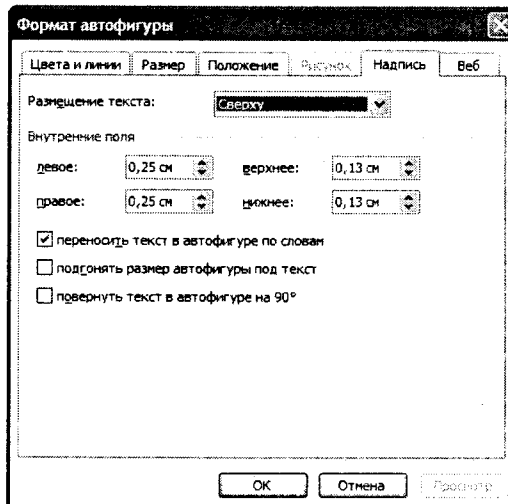


Рис. 9.9

Для того, щоб змінити вигляд маркера у маркованому списку, слід виділити абзаци у режимі введення тексту або виділити блок та виконати команду меню **Формат/Список**. Відкриється діалогове вікно **Список** із закладками **Маркированный** (рис. 9.10) та **Нумерованный**. У ньому слід вибрати необхідний символ і натиснути **ОК**.

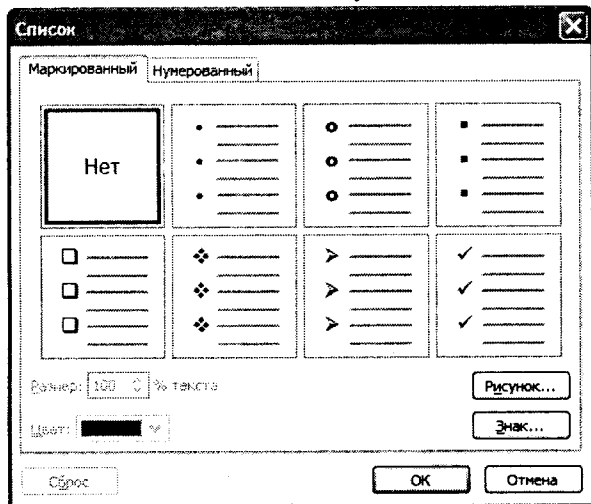


Рис. 9.10

У багатьох випадках при створенні презентації потрібно вставити туди таблицю чи графік. Це можна зробити, обравши з меню **Вставка/Таблица**. Відкриється діалогове вікно **Вставка таблицы** (рис. 9.11), у якому слід задати кількість стовпців та рядків.

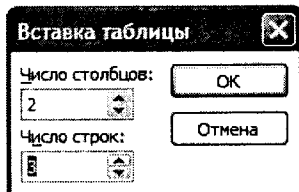


Рис. 9.11

Після цього на полі слайда буде розміщено контейнер таблиці із заданою кількістю рядків та стовпців.

При виборі **Вставка/Диаграмма** до презентації буде додано стандартну діаграму і вікно **Таблица данных** (рис. 9.12). Введення даних тут відбувається аналогічно подібним діям у програмі Excel. У слайді зберігається динамічне посилання на додаток **Microsoft Graph** для зміни формату діаграми, редагування даних тощо.

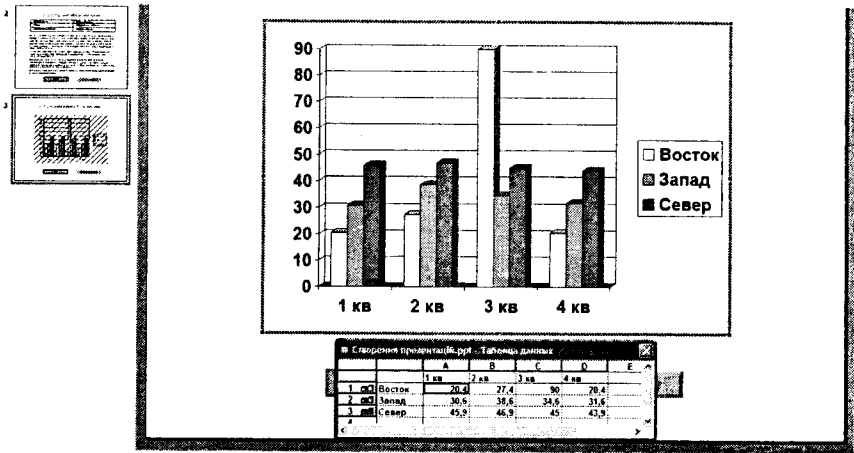


Рис. 9.12

Редагування полів та значень у Microsoft Graph (рис. 9.13) відбувається за схемою, традиційно прийнятою в MsOffice.

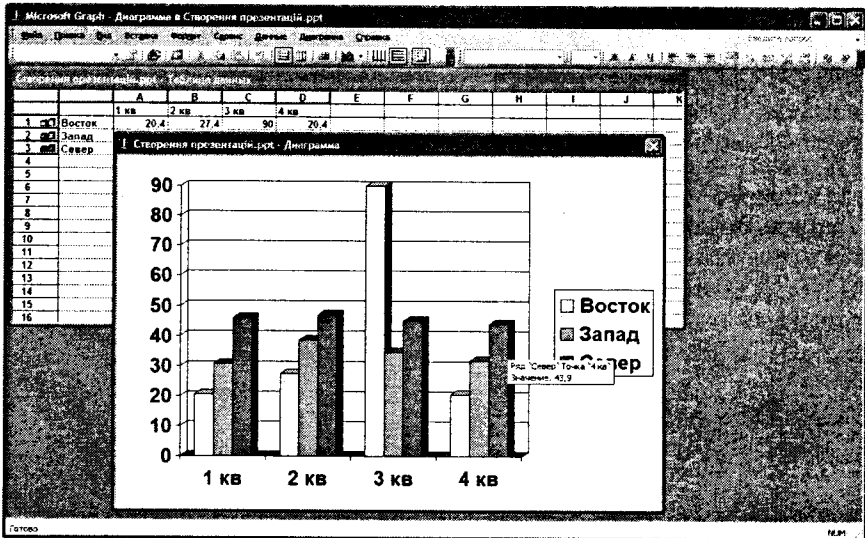


Рис. 9.13


Для вставки у презентацію діаграм інших типів слід вибрати **Вставка/Схематическая Диаграмма** або на панелі **Рисование** натиснути  **Добавить диаграмму** или **организационную диаграмму**. Відкриється діалогове вікно **Библиотека диаграмм** (рис. 9.14), у якому слід обрати потрібний тип діаграми і натиснути **ОК**.



Рис. 9.14

Для організаційної діаграми, як і для вставлених таблиць, графіків, зберігається дія команди **Формат автофигуры**, інших параметрів роботи PowerPoint, у т.ч. параметрів анімації, тощо.

При натисканні на кнопку **Добавить Картинку** в області задач створюється вікно **Коллекция клипов** (рис. 9.15) з готовими рисунками.

Кнопка **Добавить Рисунок** завантажує діалогове вікно **Добавление рисунка** (рис. 9.16) для пошуку і вставки рисунка з файлів графічних форматів.

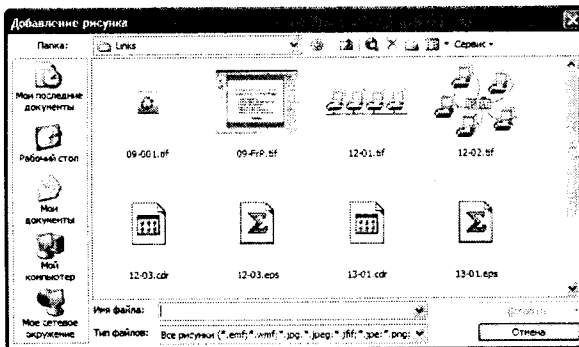


Рис. 9.16

До презентацій можна додати звуковий- та відео- супровід. Засоби для цього знаходяться у меню **Вставка/ Фильмы и звук**. Можна вставити фільми і звуки з колекції, інші, котрі містяться на диску, а також записати через мікрофон власний звуковий супровід.

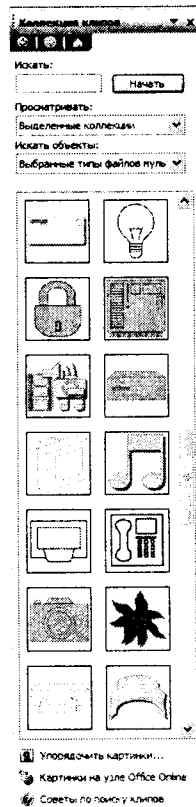


Рис. 9.15

9.5. Приклад створення презентації

Наступний приклад ілюструє створення презентації можливостей програми PowerPoint. Інформативно це буде реалізовано за матеріалами даного розділу. Титульна сторінка презентації оформлена у вигляді заголовка, змісту та кнопки виходу (рис. 9.17).

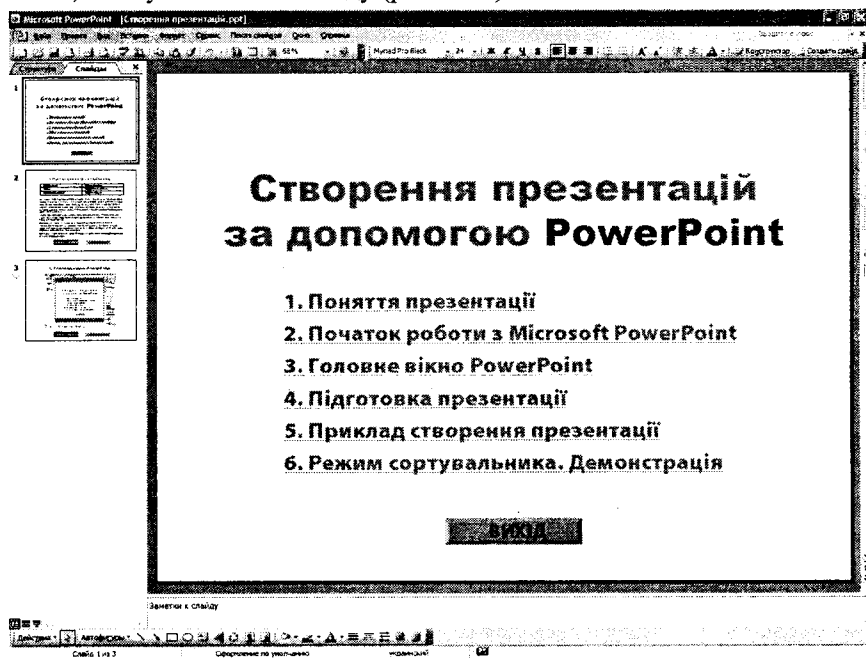


Рис. 9.17

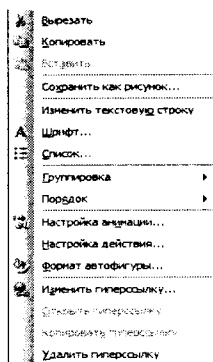


Рис. 9.18

Перші два елементи – текстові. Варто розглянути детальніше створення кнопки. Її вибрано з панелі **Рисование/Автофигуры/Управляющие кнопки**. За допомогою миші визначено розміри кнопки, потім у діалоговому вікні **Настройка действия**, що відкрилось, задано ефект звуку при натисканні кнопки у полі **Звук** та її керуючу дію у полі **Действие по щелчку мыши** – **Завершить показ**. У контекстному меню кнопки (рис. 9.18) є опції **Изменить текстовую строку** для задання напису на кнопці, **Настройка анимации** для налаштування ефектів анімації та **Настройка действия** для зміни керуючої дії кнопки. Крім того, присутня стандартна опція **Формат автофигуры** тощо.

Вигляд діалогового вікна **Настройка действия** ілюструє рис. 9.19.

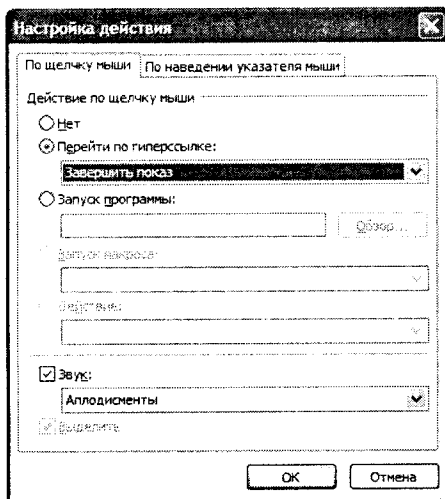


Рис. 9.19

Діалогове вікно **Настройка действия** можна викликати також і для фрагментів тексту. Для прикладу з рис. 9.17 слід виділити перший абзац змісту і задати для нього дію переходу на наперед створений слайд, що відповідає йому за змістом. Для цього у списку **Перейти по гиперссылке** слід вибрати **Слайд...** Відкриється діалогове вікно **Гиперссылка на слайд** (рис. 9.20), де й слід обрати потрібний і натиснути кнопку **OK**.

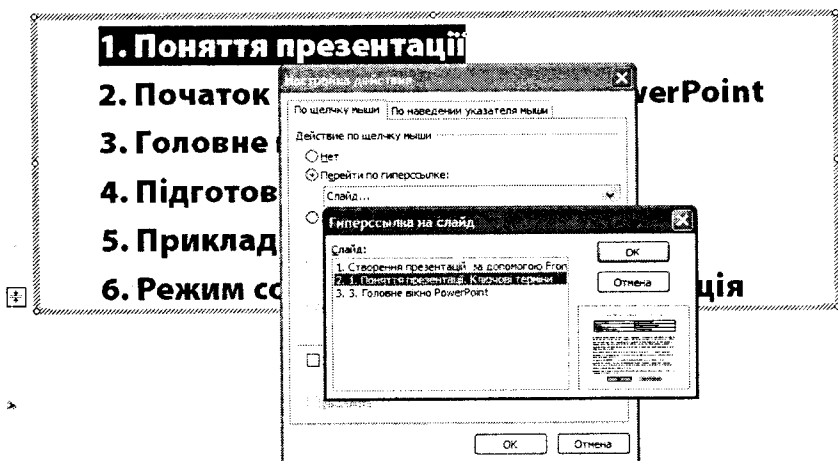


Рис. 9.20

У вибраному слайді вище текстового блоку було вставлено таблицю (див. рис. 9.11). Після заповнення таблиці було виділено весь текст і застосовано команду **Формат/Список** (див. рис. 9.10).

Слайд 2 із заголовком **Поняття презентації. Ключові терміни** зображено на рис. 9.22. Просування слайдами тут реалізовано двома кнопками. Кнопка **На головну** має керуючу дію переходу **Первый слайд**. Кнопка ліворуч зі стрілкою – це ще один різновид з панелі **Управляющие кнопки** і має керуючу дію переходу **Следующий слайд**.

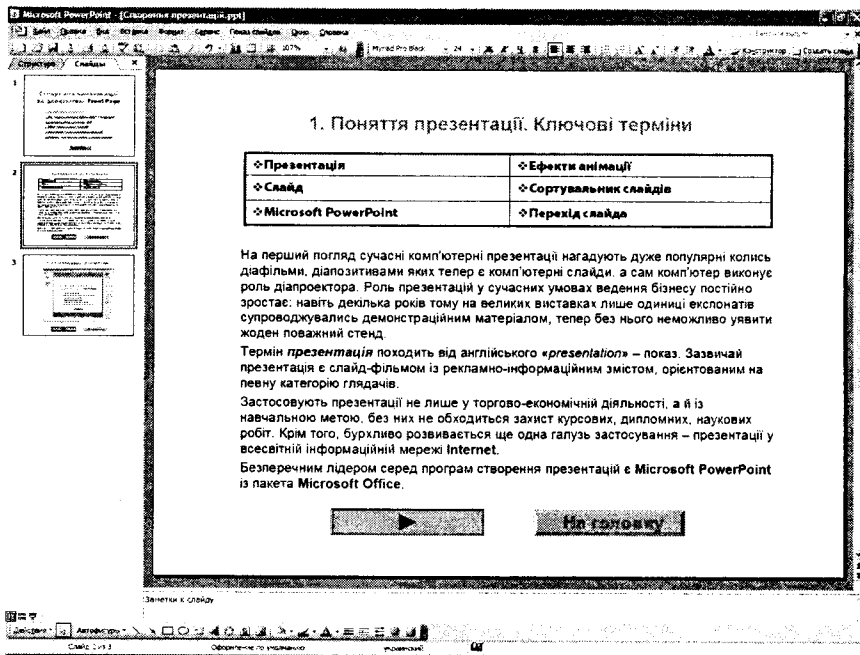


Рис. 9.22

Діалогове вікно **Налаштування анімації** присвоює об'єкту ефекти промальовування. Викликається воно з меню **Показ слайдов** чи контекстного меню об'єкта. В області задачі створюється вікно **Налаштування анімації** з кнопкою-списком **Добавить эффект**. Можна обрати такі значення: **Вход**, **Выделение**, **Выход**, **Пути перемещения**. Вибір **Вход/Другие эффекты** викликає діалогове вікно **Добавление эффекта входа**, зображене на рис. 9.23. Після вибору бажаного ефекта стають активними поля його параметрів: **Начало**, **Направление** та **Скорость**, у яких можна задати певні значення. Стрілками вниз можна визначити порядок появи ефектів під час презентації, активізувати автоперегляд ефекта при його виборі та запустити презентацію на виконання з поточного місця.

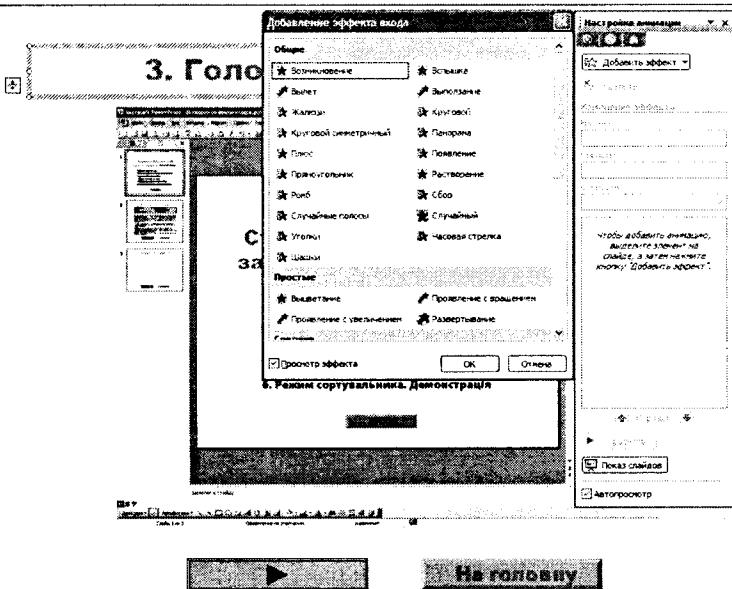


Рис. 9.23

Створивши необхідне інформаційне підґрунтя презентації, розглянемо детальніше її монтування у режимі сортувальника та параметри демонстрації.

9.6. Режим сортувальника. Демонстрація

Кілька слайдів у режимі сортувальника ілюструє рис. 9.24.

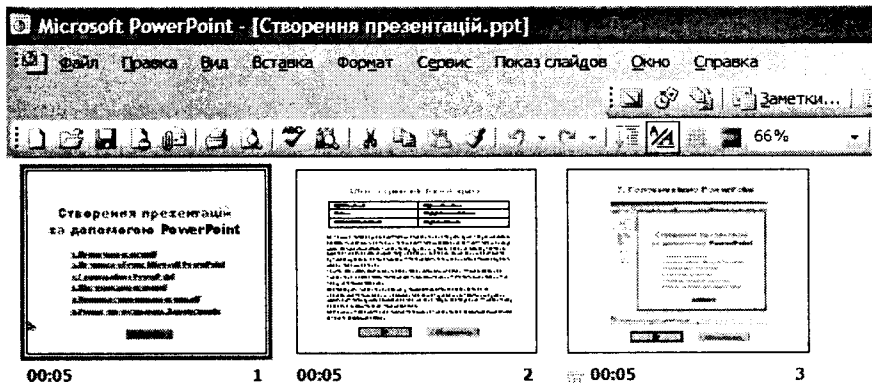


Рис. 9.24

Активний слайд завжди обведений рамкою. У цьому режимі слайди можна вилучати, змінювати їх почерговість, «клонувати» тощо. Також можна задати деякі параметри, котрі безпосередньо стосуватимуться слайд-фільму.

Перехід слайдів задається з меню **Показ слайдов/Смена слайдов**. В області задач з'являється вікно **Смена слайдов** (рис. 9.25), у якому можна задати ефект переходу та його параметри.

Слайд можна не включати в перегляд, виділивши його і виконавши **Показ слайдов/Сховати слайд** чи скористатись перемикачем у контекстному меню слайда. Якщо ж потрібна презентація лише з декількома вибраними слайдами, то слід виконати **Показ слайдов/Произвольный показ** і в діалоговому вікні, що відкривається, натиснути **Создать**. В діалоговому вікні **Задание произвольного показа** (рис. 9.26) слід вибрати необхідні слайди.

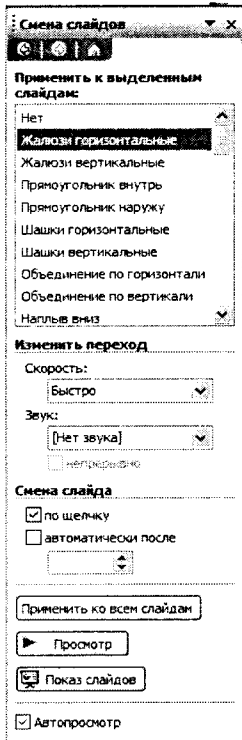


Рис. 9.25

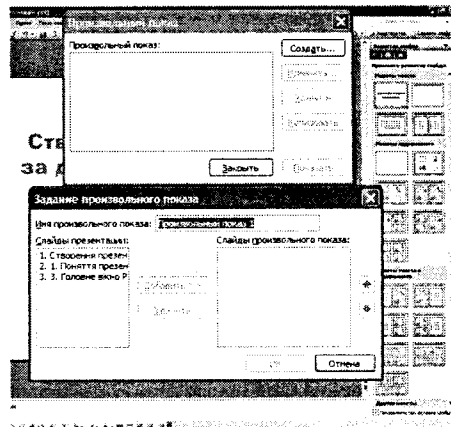


Рис. 9.26

Задати час промальовування слайда експериментально можна, виконавши **Показ слайдов/Настройка времени**. У режимі показу з'явиться вікно **Репетиция** (рис. 9.27), у якому слід встановити потрібні значення. Для спрощення процедури можна використати такі клавіатурні скорочення:

- N (*Next*, наступний) – показ наступного слайда;
- P (*Previous*, попередній) – показ попереднього слайда;
- H (*Hidden*, прихований) – показ прихованого слайда;
- ESC (*Escape*, порятунок) – припинити показ слайдів.



Рис. 9.27

Ці клавіші можна застосовувати і під час презентації, за винятком варіанта, коли презентація запускається автоматично з неперервним циклом до натиснення **ESC**. Цей режим можна задати в полі **Параметри показа** діалогового вікна **Настройка презентации** (рис. 9.28), яке можна викликати, виконавши з меню **Показ слайдов/Настройка презентации**.

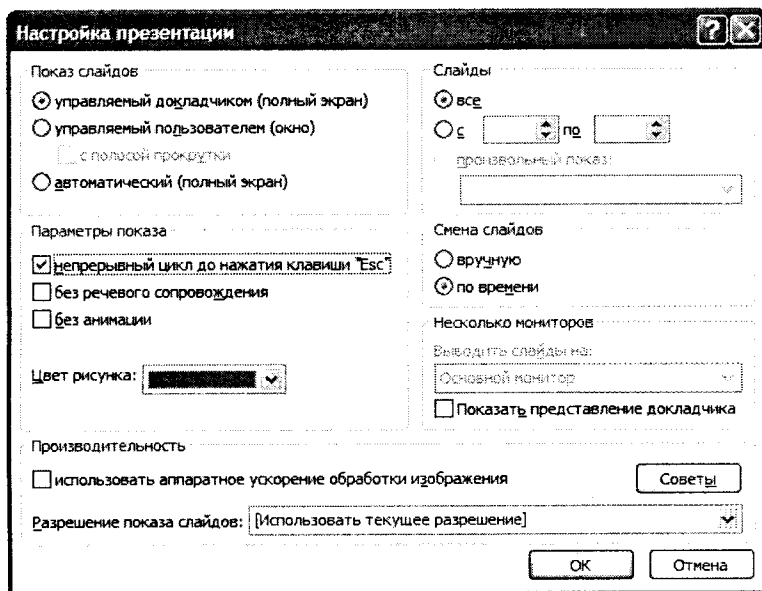


Рис. 9.28

Зауважимо, що після виконання вищепописаних дій під кожним слайдом з'явиться підпис із часом тривалості його промальовування. На даному етапі презентація вже готова до запуску.

Запустити її можна багатьма способами. Для цього можна просто натиснути **F5**, виконати команди меню **Показ слайдов/Начать Показ** чи **Вид/Показ слайдов**.

У варіантах показу **Управляемый докладчиком (полный экран)** та **Управляемый пользователем (окно)** (див. рис. 9.28) в допомогу останнім надається контекстне меню у лівому нижньому куті сторінки. У ньому продубльовані команди керування показом, а також є можливість змінити курсор миші на *перо*, *ручку* чи *фломастер*, щоб підкреслити чи обмалювати щось в процесі коментування слайда. Якщо відбулась модифікація зображення під час презентації, при виході з режиму показу програма видасть запит на збереження змін.

Іноді простіше запускати презентацію без запуску PowerPoint: у контекстному меню **.ppt**-файла вибрати **Показать**.

І останнє. Є можливість демонструвати презентацію не просто без запуску PowerPoint, а й за відсутності цього програмного продукту серед програмного забезпечення комп'ютера. Для цього готову презентацію слід зберігати у спеціальному форматі **Демонстрація PowerPoint** (він має розширення **.pps**), виконавши **Файл/Сохранить как** і в діалоговому вікні **Сохранение документа** (рис. 9.29) у полі **Тип файла** вибрати **Демонстрация PowerPoint**.

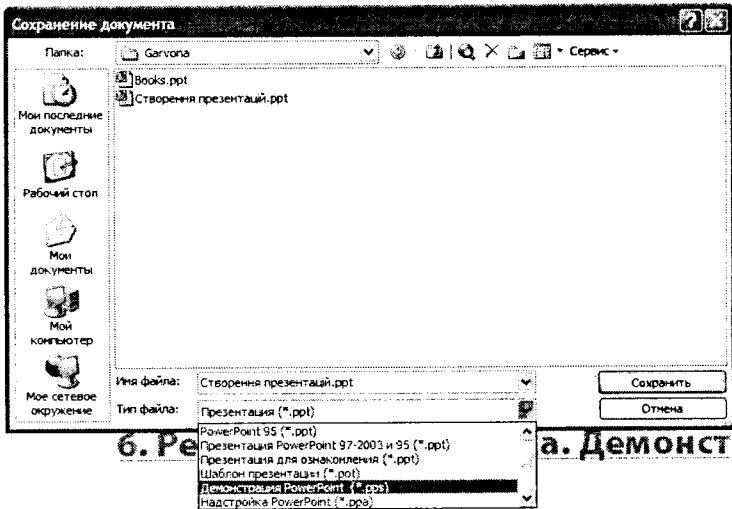


Рис. 9.29

Запитання для самоперевірки

1. Що таке презентації? З якою метою їх застосовують?
2. Як створювати презентації, використовуючи шаблони? Що таке авто-розмітка документа?
3. Які режими роботи передбачені у PowerPoint?
4. Якими способами можна вставити у презентацію таблиці, діаграми, рисунки, організаційні діаграми?
5. Як надати елементові презентації керуючої дії?
6. Як задати ефекти промальовування об'єкта презентації?
7. Якими засобами задаються способи переходу одного слайда в інший?
8. Як експериментально задати час промальовування слайда?
9. Які задати налаштування для показу презентації?
10. У якому форматі слід зберігати презентацію, щоб мати змогу демонструвати її за відсутності PowerPoint серед програмного забезпечення?

10. КОМП'ЮТЕРНА ГРАФІКА

📖 План викладу матеріалу:

1. Основні визначення, поняття та пристрої комп'ютерної графіки.
2. Растровий графічний редактор Adobe Photoshop.
3. Векторний графічний редактор CorelDRAW.

↔ Ключові терміни розділу

✓ <i>Зображення</i>	✓ <i>Перо</i>
✓ <i>Візуалізація</i>	✓ <i>Олівець</i>
✓ <i>Вузол</i>	✓ <i>Контур</i>
✓ <i>Дуга</i>	✓ <i>Округлення</i>
✓ <i>Область</i>	✓ <i>Згладжування</i>
✓ <i>Заповнення фігур</i>	✓ <i>Розміри</i>
✓ <i>Растр</i>	✓ <i>Масштаб</i>
✓ <i>Спектр</i>	✓ <i>Тонування</i>
✓ <i>Палітра кольорів</i>	✓ <i>Тіні</i>
✓ <i>Форма</i>	✓ <i>Роздільна здатність</i>

10.1. Основні визначення, поняття та пристрої комп'ютерної графіки

Сучасна комп'ютерна графіка – це галузь сучасних технологій з найшвидшими темпами розвитку. Методи комп'ютерної графіки широко застосовуються в розробці різних продуктів, навчальних програм, створенні музичних кліпів і телевізійної реклами, кінофільмах, аналізі даних, наукових дослідженнях, медичних процедурах, а також в інших застосуваннях. В цих прикладних застосуваннях використовуються різноманітні технології і апаратні рішення. На даний час ґрунтовно досліджені основні елементи геометричних перетворень та опису кривих та поверхонь. Достатньо досліджені, але продовжують розвиватися методи растрового сканування, відсічі, вилучення невидимих ліній та поверхонь, методи врахування кольорових ефектів та ефектів прозорості. В самостійну область комп'ютерної графіки виділяються питання про формати файлів графічних зображень, взаємні переходи між цими форматами, питання архівування та розархівування графічних файлів. Більша частина сучасних досліджень в галузі комп'ютерної графіки пов'язана з підвищенням ефективності, реалістичності і швидкості генерації зображень. Щоб отримати реалістичне зображення природної сцени, графічна програма повинна імітувати ефекти реального заломлення світла від фізичних об'єктів. Таким чином,

в комп'ютерній графіці спостерігається тенденція до впровадження в графічні алгоритми оптимізаційних апроксимацій фізичних принципів, щоб краще імітувати складні взаємодії між об'єктом і світловим середовищем.

Одна з перших областей застосування комп'ютерної графіки – це демонстрація простих *графіків* даних. Побудова графіків до цих пір залишається найбільш поширеним прикладом застосування комп'ютерної графіки, але сьогодні можна дуже просто створювати *діаграми* для звітів та презентацій, які демонструють складні взаємозв'язки даних, використовуючи слайди, плакати або мультиплікацію. Графіки і діаграми, як правило, використовуються для узагальнення фінансових, статистичних, математичних, наукових, технічних і економічних даних при оформленні наукових доповідей, адміністративних звітів, інформаційних бюлетенів і публікацій інших видів. Зараз існує багато комерційних графічних програм, які перетворюють зображення на екрані у фільм, слайди або плакати, які використовуються для презентацій або архівного зберігання. Типовими прикладами графічного зображення даних є лінійні графіки, гістограми, секторні діаграми, площинні діаграми, контурні та інші схеми, що показують взаємозв'язки між декількома параметрами в двох-, трьохмірному або більшого розміру просторі. Трьохмірні графіки і діаграми використовуються для того, щоб показати додаткову інформацію про параметри, хоча деколи ними користуються просто, щоб надати інформації більш вражаючого ефекту або показати залежність даних у більш наочній формі.

Комп'ютерна графіка широко використовується в процесах проектування, а саме в технічних і архітектурних системах. В наш час ці системи називаються системами *автоматизованого проектування* (АП, computer-aided design – CAD) або *автоматизованого проектування і складання креслень* (АПСК, computer-aided drafting and design – CADD) і використовуються для проектування будинків, автомобілів, кораблів, космічних апаратів, комп'ютерів, текстильних виробів, побутової техніки та іншої продукції. В цих галузях проектування предмет спочатку зображається у вигляді каркасної моделі, яка показує загальну форму і характерні особливості об'єкта. Коли проектування об'єкта завершується, накладаються умови реалістичного освітлення і замальовуються поверхні, в результаті отримуємо зображення, яке показує завершальний зовнішній вигляд продукту.

Одне з найновіших застосувань комп'ютерної графіки – це створення *віртуальної реальності*, в якій користувач може взаємодіяти з предметами в трьохмірному просторі. Архітектурний проект можна перевірити, здійснюючи віртуальний обхід по кімнатах або навколо будинку, щоб краще оцінити загальне враження від розробленого проекту. Анімація віртуальної реальності часто використовується для навчання керуванню транспортним засобом або аналізу ефективності різних моделей складових засобу та зручності розміщення елементів управління ним.

Створення графічних відображень наукових, технічних, медичних даних і процесів – це ще одна з нових областей застосувань комп'ютерної графіки, яку ще називають *науковою візуалізацією*. А термін *ділова (комерційна) візуалізація* використовують у зв'язку з наборами даних, які відносяться до торгівлі, промисловості та інших ненаукових сфер. Дослідникам, аналітикам та іншим спеціалістам часто доводиться працювати з великими масивами інформації або вивчати проходження процесів високої складності. Під час комп'ютерного моделювання часто видаються файли, що містять не одну тисячу значень. Також аналогічно супутникові камери та інші записуючі пристрої накопичують файли великих обсягів швидше, ніж їх можна обробити. Перегляд цих значних наборів цифр з метою визначення тенденцій і взаємозв'язків – невтішне заняття. Але якщо ці дані перетворити у візуальну форму, тоді зразу ж можна відслідкувати загальні тенденції і закономірності.

Існують багато різноманітних видів даних і ефективність схеми візуалізації залежить від характеристик цих даних. Набори даних можуть складатися зі скалярних величин, векторів, тензорів високого порядку або задавати будь-яку комбінацію цих видів даних. Крім цього, ці дані можуть бути координатами об'єктів двох-, трьохмірного простору або простору більш високого порядку. Кольорове кодування – це тільки один зі способів візуалізації даних. До інших методів візуалізації відносяться контурні графіки, діаграми, ізолінії, замальовування поверхонь однакових значень, а також спеціально розроблені форми задання різних типів даних.

В освітніх цілях часто використовуються комп'ютерні моделі фізичних, фінансових, політичних, соціальних, економічних та інших систем. Моделі фізичних процесів, фізіологічних функцій, демографічних тенденцій або устаткування з кольоровим кодуванням допомагають краще зрозуміти принцип дії систем. Для деяких *навчальних* застосувань розроблено спеціальне устаткування. До прикладу таких спеціалізованих систем можна віднести тренажери для практичних занять або навчання капітанів суден, пілотів літаків, водіїв транспорту та авіадиспетчерів. Особливо часто такі тренажери застосовують для вивчення поведінки в критичних ситуаціях.

Методи комп'ютерної графіки зараз широко використовують як в *поліграфії*, так і в *образотворчому мистецтві (комп'ютерне мистецтво)*. Сьогодні художникам доступні різноманітні комп'ютерні методи і засоби, в тому числі і спеціальне устаткування, комерційні програмні продукти, настільні видавничі та анімаційні системи, що дають можливість проєктувати форми і описувати рух об'єктів. Щоб намалювати картину, художник може користуватися сумісно програмами трьохмірного моделювання, накладанням текстури, малювання і програмним забезпеченням автоматизованого проєктування. Цими методами „синтезу” зображень

також користуються для створення логотипів та інших елементів дизайну, макетів сторінок, що містять текст і графічні зображення, телевізійної реклами. В поліграфії часто використовують фотореалістичні методи створення малюнків.

При створенні *телевізійної комерційної продукції* також користуються методами комп'ютерної анімації. Такі рекламні ролики створюються кадр за кадром, де кожний кадр малюється і зберігається як окремий файл зображення. На кожному наступному кадрі положення об'єкта трошки зміщується, що викликає сприйняття руху в анімаційному фільмі. Коли намальовані всі кадри анімаційної послідовності, вони об'єднуються у фільм або зберігаються у відеобуфері для перегляду. Часто при цьому один об'єкт плавно перетворюється в інший.

В кінофільмах і музичних кліпах використовувати методи комп'ютерної графіки стало звичайним явищем. Деколи графічні зображення доповнюють гру живих акторів і реальні сцени. Існують фільми, які повністю зроблені за допомогою методів комп'ютерного малювання і анімації. Методами комп'ютерної графіки користуються також для моделювання гри самого актора. За допомогою цифрових файлів, в яких записані риси обличчя актора, анімаційна програма може видати частину фільму, в якій буде міститися створена за допомогою комп'ютера репліка цієї людини. Під час зйомок кінофільму у випадку хвороби актора або неможливої участі можна скористатися методами моделювання і замінити актора у деяких сценах.

Зміна або інтерпретація вже існуючих зображень, таких як фотографії або відеоплівки, називається *обробкою зображень*. Ці методи застосовуються для покращення якості малюнку, його аналізу або розпізнавання візуальних шаблонів в застосуваннях. Спочатку фотографію потрібно оцифрувати і перетворити у файл зображення. Далі за допомогою цифрових методів можна перемістити окремі частини зображення, збільшити контрастність кольорів або покращити якість ретушування. Ці методи інтенсивно використовуються в поліграфії для аналізу фотографій Землі, отриманих з супутника, телескопічних записів розподілу зірок в галактиках тощо.

В медичних застосуваннях обробка зображень і комп'ютерна графіка часто сумісно використовуються для моделювання і вивчення фізичних функцій, посилення зображень в томографії, проектування штучних органів, планування і виконання хірургічних операцій. За допомогою комп'ютерних методів створення зображень отримують двохмірні поперечні розрізи тіла. Далі ці розрізи переглядаються, над ними за допомогою графічних методів виконуються визначені дії з метою моделювання справжніх хірургічних процедур і апробації різних хірургічних втручань.

В даний час майже стандартно прийнято супроводжувати програмне забезпечення *графічним інтерфейсом* (GUI). Головний елемент графічного інтерфейсу – це блок керування вікнами, який дозволяє користувачу бачити на екрані багато прямокутників, так званих вікон. В кожному вікні екрану можуть відображатися різні процеси, які видають графічну або неграфічну інформацію і для активізації окремих вікон можуть застосовуватися різні методи. Інтерфейси також служать для зображення меню і піктограм для вибору вікон, опції обробки або значення параметра. Перевага піктограм в тому, що вони займають менше місця на екрані, ніж відповідний текстовий опис, і при вдалому дизайні їх призначення можна зрозуміти швидше. Часто вікно можна перетворити в піктограму, а в меню помістити списки як текстових описів, так і піктограм.

Попередньо зроблено огляд деяких застосувань, в яких використовується комп'ютерна графіка, таких як побудова графіків даних, автоматизоване проектування, віртуальна реальність, наукова візуалізація, освіта, мистецтво, розваги, обробка зображень і графічні користувацькі інтерфейси. Зрозуміло тут не названо ще багато інших областей, в яких значну роль відведено графічним методам обробки інформації.

При такому розвитку комп'ютерної графіки для спеціалістів, які працюють в цій області, постійно пропонують все нові і нові терміни. Часто нові терміни та поняття є, як правило, похідними від вже раніше використовуваних, які мають чітке визначення і зміст. Більше того, все більше проникнення комп'ютерної графіки в інші галузі призводить до того, що термінологія відповідної області мимоволі вливається в термінологію комп'ютерної графіки.

Оскільки немає єдиного набору термінів в комп'ютерній графіці, то застосуємо лише встановлену термінологію, яка застосовується більшістю спеціалістів.

У Державному стандарті України ДСТУ 2939-94 дається таке визначення: *комп'ютерна графіка* – це сукупність методів і способів перетворення за допомогою комп'ютера даних у графічне зображення і графічного зображення у дані.

В комп'ютерній графіці основним об'єктом є *зображення*, яке містить будь-які *графічні дані*, призначені для відображення (*візуалізації*) на пристрої виводу. Ці зображення можуть використовуватися з різною метою. Вони можуть бути технічним кресленням, ілюстрацією до книги, діловою діаграмою, архітектурним видом конструкції або проєктованого будинку, рекламною продукцією або кадром до фільму. Для обробки зображень вимагається введення початкової інформації у вигляді візуального представлення. Ввід зображень в пам'ять комп'ютера відбувається за допомогою спеціальних технічних пристроїв: сканера і дигітайзера.

Сканер – це пристрій для зчитування графічної і текстової інформації. Сканер дозволяє створити електронну копію зображення для наступної її обробки засобами комп'ютера. Як правило, розрізняють три типи сканерів: ручні, планшетні та барабанні. Результат сканування представляється у вигляді файлу, який може бути записаний в різних форматах. Графічні формати відрізняються між собою за способами кодування графічної інформації. Найбільш популярні **графічні формати** – BMP, PCX, GIF, TIFF, EPS. Розмір файлу, необхідного для зберігання зображення, залежить від величини зображення, роздільної здатності сканера, а також від кількості відтінків кольору. В більшості випадків якість зображення є доброю в тому випадку, якщо роздільна здатність сканера удвічі більша, ніж роздільна здатність пристрою виводу.

Дігітайзер – це пристрій планшетного типу, призначений для вводу графічних даних у цифровій формі. Дігітайзер складається з електронного планшета і курсора. Він має свою власну систему координат і при русі курсора по планшету координати перехрестя його ліній передаються в комп'ютер. Розміри планшета дігітайзера змінюються від формату А4 до А0, змінною є також кількість кнопок на курсорі.

Пристрої виводу зображень призначені для виводу графічних даних у вигляді, зручному для користувача комп'ютера. За допомогою **плотера (графопобудовувача)** можна отримати більш якісні чорнобілі та кольорові зображення. Для виводу використовуються також різні **принтери**, в тому числі і лазерні. Результати роботи можуть бути представлені у вигляді відеофільмів, записаних на дисках, роздрукованих у формі звітів або відправлених по мережі у зовнішні комп'ютерні системи.

Графічні дані традиційно розділяють на два типи: **векторні і растрові**.

Векторні дані використовуються для задання прямих, многокутників і кривих за допомогою визначених контрольних точок (вершин). Вершини з'єднуються лініями програмно за відповідними правилами. З векторними даними завжди пов'язана інформація про атрибути (колір і товщина точки, ліній) і набір правил, які допомагають програмі з'єднати ці точки, тобто намалювати об'єкти.

Растрові дані задають набір числових значень, що визначають колір окремих пікселів. **Пікселі** – це кольорові точки, розміщені на регулярній сітці, які формують зображення. Сам термін **растр (raster)** асоціюється з електронно-променевою трубкою і вказує на те, що при формуванні зображення на трубці створюють прообрази стрічок (т.зв. **стрічки розгортки**). У більшості випадків, коли мова йде про графічні дані, отримані, наприклад, з пристроїв вводу, то мається на увазі термін **растрове зображення**.

Програмне забезпечення комп'ютерної графіки здійснюється спеціалізованими системами, які забезпечують проведення малювання чи побудов, перетворення, редагування та виведення графічної інформації.

Далі розглянемо найбільш вживані з професійних засобів програмного забезпечення для обробки графічних даних растрового і векторного типів.

10.2. Растровий графічний редактор Adobe Photoshop

10.2.1. Вікно програми. Робота з документами

У наш час де-факто стандартом растрового графічного редактора для професійної роботи є **Adobe Photoshop**. Інші растрові редактори, ті ж **Paint** або **Imaging**, хоч і є простими в освоєнні, проте примітивні з точки зору можливостей. Photoshop, у першу чергу, характеризується потужними можливостями: тут не тільки можна створювати власні документи, формувати у них зображення засобами програми, редагувати файли відсканованих зображень, комбінувати те й друге, а й здійснювати багато інших речей. Опишемо можливості програми на прикладі **Adobe Photoshop 8.0**. Головне вікно програми наведено на рис. 10.1.

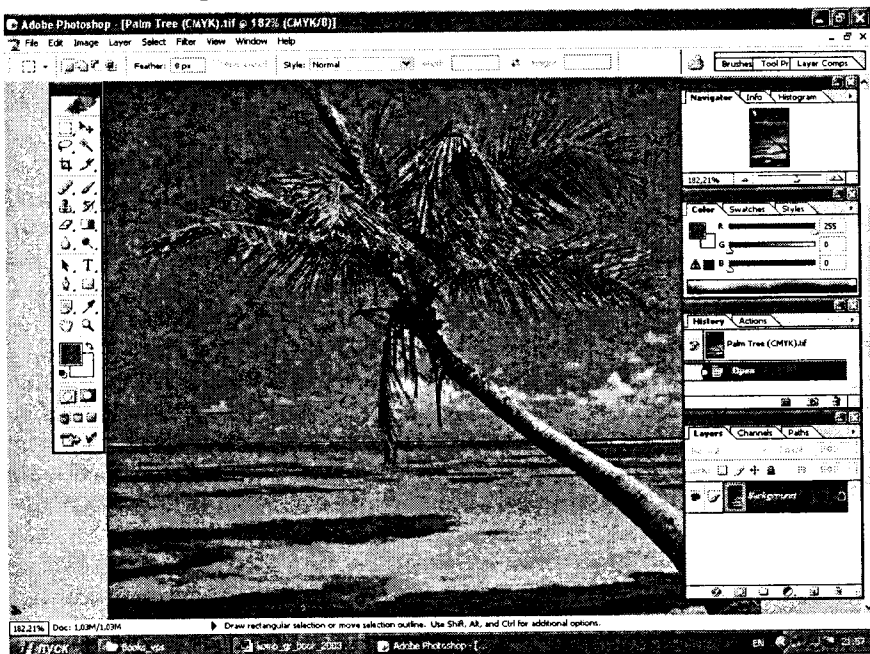


Рис. 10.1

Як видно з рисунка, головне вікно представлено не у максимізованому вигляді, фактично воно збігається з робочим полем. Головне меню містить дев'ять пунктів: **Файл (File)**, **Редактирование (Edit)**, **Изображение (Image)**, **Слой (Layer)**, **Выделение (Select)**, **Фильтр (Filter)**, **Вид (View)**, **Окно (Window)**, **Помощь (Help)**. Ліворуч робочого поля розміщена панель інструментів, праворуч – плаваючі панелі.

10.2.1.1. Створення, збереження та відкриття документів

Для створення документа використовують команду **Файл/Новый (File/New)**. У діалоговому вікні (рис. 10.2) задають: у полі **Имя (Name)** – ім'я; у полях **Ширина (Width)** та **Высота (Height)** – висоту і ширину (одиниці вибирають у полях праворуч); у полі **Разрешение (Resolution)** – розподільну здатність; у полі **Режим (Mode)** – колірну модель; у рамці **Содержимое (Contents)** – тіло документа: **Белый цвет (White)** – біле, **Фоновый цвет (Background Color)** – фонового кольору, заданого перед створенням документа, **Прозрачная основа (Transparent)** – прозоре.

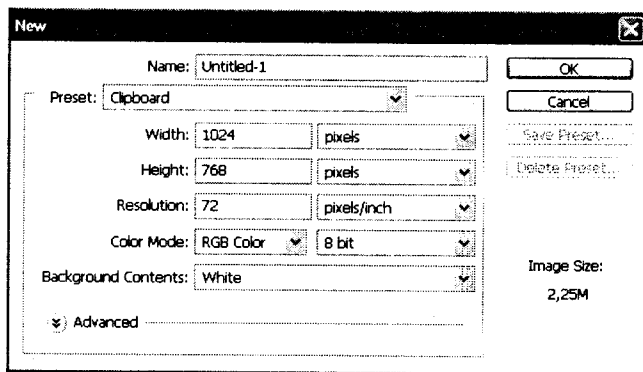


Рис. 10.2

Зберігають документ командою **Файл/Сохранить (File/Save)**. Якщо це робиться вперше, то відкривається діалогове вікно **Сохранить как (Save as)** (рис. 10.3), де: у полі **Папка (Folder)** вибирають каталог; у полі **Сохранить (Save as)** – формат; у полі **Имя файла (Save this document as)** задають ім'я файлу. Наступний запис цією командою проходить без діалогу.

Команда **Файл/Сохранить копию (File/As a Copy)** зберігає копію, однак робота продовжується далі з вихідним файлом. Якщо зображення має кілька шарів, то увімкнення опції **Выполнить сведение (Layers)** дає змогу звести всі шари в один, внаслідок чого об'єм файлу зменшиться.

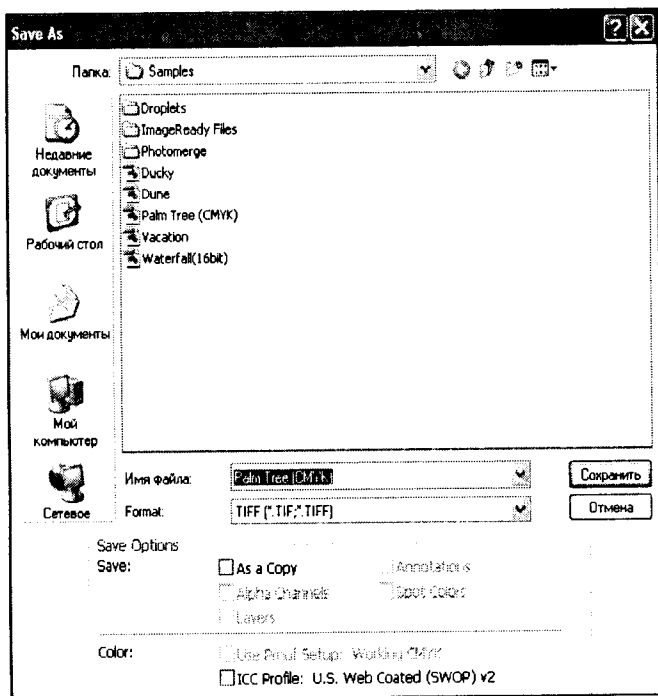


Рис. 10.3

Відкривають наявний документ командою **Файл/Открыть (File/Open)**. Слід також відзначити, що Photoshop є багатовіконним графічним редактором, тому можна відкривати одночасно кілька зображень, перехід до кожного з яких здійснюють вибором імені у меню **Окно (Window)**.

10.2.1.2. Зміна параметрів готових зображень

Якщо з якихось причин потрібно змінити розміри зображення, використовують команду **Изображение/Размер изображения (Image/Image Size)**. В однойменному діалоговому вікні (рис. 10.4) задають нові розміри зображення у відповідних полях рамки **Размерность (Pixel Dimension)** (у % до вихідного або пікселях) або **Размер печатного оттиска (Print Size)** (в лінійних одиницях), або змінюють роздільну здатність (поле **Разрешение (Resolution)**). Опція **Сохранить пропорции (Constrain Proportions)** призначена для автоматичного збереження пропорцій зображення. Увімкнення опції **Интерполяция (Resample Image)** призводить до перерахунку зображення методом, вказаним у полі праворуч, і зміни кількості пікселів, вимкнення ж залишає кількість пікселів незмінною.

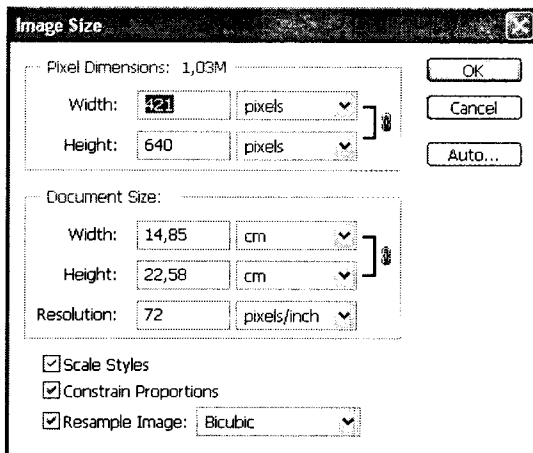


Рис. 10.4

Якщо відскановане зображення нахилене або перевернуте, то слід скористатися підкомандами команди **Изображение/Повернуть холст (Image/Rotate Canvas)**: **180° (180°)** повертає зображення на 180°, **90 по часовой (90° CW)** – на 90° за годинниковою стрілкою, **90 против часовой (90° CCW)** – на 90° проти годинникової стрілки, **Произвольно (Arbitrary)** – на довільний заданий кут, **Зеркально по горизонтали (Flip Horizontal)** – дає дзеркальне відображення по горизонталі, **Зеркально по вертикали (Flip Vertical)** – по вертикалі.

Обрізати зайві поля зображення можна за допомогою інструмента **Рамка (Crop)**: треба визначити мишею межу потрібних розмірів та розташування, двічі натиснути мишею всередині рамки чи натиснути **Enter**. Якщо ж потрібно додати до зображення вільне місце, використовують команду **Изображение/Размер холста (Image/Canvas Size)**: у діалоговому вікні, що відкриється, задають нові розміри зображення, а в області **Расположение (Anchor)** вказують, де слід розмістити попереднє зображення.

10.2.2. Панелі інструментів

Всі інструменти містяться на спеціальній панелі (рис. 10.5).

Деякі з них позначені маленьким трикутником і містять набір інструментів. Його можна відкрити, натиснувши на піктограму і зачекавши певний час. Для вибору інструмента з набору слід перемістити на нього курсор миші, не відпускаючи лівої клавіші.

Рис. 10.5 Це можна також зробити у панелі **Параметры (Options)**, для

швидкого відкриття якої треба двічі натиснути мишею на відповідному інструменті. При наведенні на інструмент висвічується підказка, якщо увімкнута опція **Подказки по инструментам (Show Tool Tip)** діалогового вікна **Установки (Preferences)**, що відкривається командою **Файл/Установки/Основные (File/Preferences/General)**.

10.2.2.1. Інструменти виділення

До інструментів вибору належать **Область (Rectangular)**, **Овальная область (Elliptical)**, **Горизонтальная строка (Single Row)**, **Вертикальная строка (Single Column)**, **Лассо (Lasso)**, **Многоугольное лассо (Polygon Lasso)**, **Магнитное лассо (Magnetic Lasso)** та **Волшебная палочка (Magic Wand)**.

Інструмент **Область (Rectangular)** виділяє прямокутні області, **Овальная область (Elliptical)** – еліптичні, **Горизонтальная строка (Single Row)** та **Вертикальная строка (Single Column)** – відповідно горизонтальні та вертикальні смуги товщиною 1 піксель.

Інструмент **Лассо (Lasso)** дає змогу виділяти області неправильної форми, нарисовані на екрані курсором, **Многоугольное лассо (Polygon Lasso)** – області у формі многокутників, вузли яких утворюються натисканням мишею (для завершення вводу слід натиснути двічі), інструмент **Магнитное лассо (Magnetic Lasso)** дозволяє виділяти області на основі розмежування кольорів.

Інструмент **Волшебная палочка (Magic Wand)** виділяє області на основі подібності кольорів. Близький за змістом інструмент **Рамка (Crop)** служить для обрізання країв зображення, а **Перемещение (Move)** використовують для переміщення вибраних областей. Практично в усіх режимах роботи (за винятком **Pen** та **Hand**) переміщення здійснюється при натисненій клавіші **Ctrl**, при **Ctrl+Alt** здійснюється переміщення з клонуванням, і при **Ctrl+Alt+Shift** маршрут переміщення проходить стро-го під кутом 90°.

10.2.2.2. Інструменти малювання

Основними інструментами малювання в Adobe Photoshop є **Заливка (Paint Bucket)**, **Градиент (Gradient)**, **Линия (Line)**, **Пипетка (Eyedropper)**, **Ластик (Eraser)**, **Карандаш (Pencil)**, **Аэрограф (Airbrush)** і **Кисть (Paintbrush)**. Інструменти **Заливка (Paint Bucket)**, **Линия (Line)**, **Кисть (Paintbrush)** та **Аэрограф (Airbrush)** використовують для малювання основним кольором. Інструмент **Ластик (Eraser)** малює кольором фону і служить для стирання частин зображення. Інструмент **Карандаш (Pencil)** імітує малювання олівцем як основним, так і фоновим кольорами. Інструмент **Градиент (Gradient)** дає змогу створювати переходи кольору між основним та фоновим кольорами і навпаки у варіантах, представлених у

списку. За допомогою інструмента **Пипетка (Eyedropper)** можна вибрати довільний колір із зображення і зробити його основним або фоновим.

10.2.2.3. Інструменти редагування

До інструментів редагування належать **Штамп (Clone Stamp)**, **Текстура (Pattern Stamp)**, **Палец (Smudge)**, **Размытие (Blur)**, **Резкость (Sharpen)**, **Осветлитель (Dodge)**, **Затемнитель (Burn)** та **Губка (Sponge)**. **Штамп (Clone Stamp)** – інструмент для клонування: ним вибирають деяку ділянку зображення і копіюють її в інше місце. Інструмент **Текстура (Pattern Stamp)** клонує наперед вибрану з палітри текстуру. Інструмент **Палец (Smudge)** “розмазує” кольори, створюючи ефект малювання аквареллю. Інструмент **Размытие (Blur)** розмиває різкі краї, а **Резкость (Sharpen)** – навпаки, виявляє на зображенні всі подробиці. Інструменти **Осветлитель (Dodge)** та **Затемнитель (Burn)** відповідно освітлюють або затемнюють ділянки зображення, а **Губка (Sponge)** збільшує або зменшує інтенсивність кольорів.

10.2.2.4. Інструменти створення і редагування контурів

Інструментами для роботи з контурами є **Перо (Pen)**, **Форма пера (Free Form Pen)**, **Перо + (Add anchor point)**, **Перо – (Delete anchor point)** і **Угол (Convert anchor point)**. Інструмент **Перо (Pen)** призначений для створення контурів, які зазвичай використовують для формування масок – електронних трафаретів, що захищають певні ділянки зображення від змін. Інструмент **Форма пера (Free Form Pen)** дає змогу виділяти контури для подальшого переміщення та редагування. За допомогою інструментів **Перо + (Add anchor point)** і **Перо – (Delete anchor point)** можна додавати і вилучати точки прив'язування з контурів. Інструмент **Угол (Convert anchor point)** перетворює криволінійні контури у прямолінійні і навпаки.

10.2.2.5. Текстові інструменти

Для введення тексту є два інструменти: **Горизонтальный Текст (Horizontal Type)** та **Горизонтальный Текст-маска (Horizontal Type Mask)**. Інструмент **Горизонтальный Текст (Horizontal Type)** відразу створює текстовий горизонтальний надпис у новому шарі, а **Горизонтальный Текст-маска (Horizontal Type Mask)** формує виділення у формі горизонтального тексту, яке можна заповнити довільним кольором. Інструменти **Вертикальный текст (Vertical Type)** та **Вертикальный текст-маска (Vertical Type Mask)** виконують аналогічні операції з вертикально розміщеним текстом.

10.2.2.6. Інструменти перегляду

Інструмент **Масштаб (Zoom)** збільшує або зменшує (якщо утримувати клавішу **Alt**) коефіцієнт масштабування зображення без зміни розмірів

вікна. Інструмент **Рука (Hand)** (клавіша **Spacebar**) дає змогу «прокрутити» зображення у його вікні і переглянути ті ділянки, які розміщені за межами екрану.

10.2.2.7. Елементи керування

У нижній частині панелі інструментів містяться кілька службових піктограм та кнопок. Піктограми **Основной цвет (Foreground Color)** та **Фоновый цвет (Background Color)** відображають поточні значення основного та фонового кольорів, а натискання на них викликає діалогове вікно **Палитра цветов (Color Picker)** для зміни відповідного кольору. Піктограма **Переключение основного и фонового цветов (Switching Colors)** міняє місцями основний та фоновий кольори, а **Основной и фоновый цвета по умолчанию (Default Color)** відновлює стандартні значення – чорний та білий.

Нижче містяться кнопки **Редактирование в стандартном режиме (Standard)** та **Редактирование в режиме быстрой маски (Quick Mask)**, які дають змогу перемикає між звичайним режимом роботи і режимом роботи з масками, у якому створюють, редагують та переглядають маски.

Внизу панелі інструментів є три кнопки керування структурою вікна. Ліва вмикає звичайний режим, середня розгортає вікно на весь екран, а права вмикає повноекранний (без меню) режим відображення.

Ще нижче розміщена кнопка переходу в програму **Adobe ImageReady (Edit in ImageReady)**, тоді поточний малюнок переміщується в активне вікно програми **Adobe ImageReady**.

10.2.2.8. Плаваючі панелі

Доступ до багатьох найбільш вживаних функцій програми можна отримати за допомогою плаваючих панелей. Їх налічується 16: **Кисти (Brushes)**, **Налаштування (Options)** (залежно від того, який інструмент активний, панель набуває назви цього інструмента), **Цвет (Color)**, **Образцы (Swatches)**, **Слои (Layers)**, **Каналы (Channels)**, **Пути (Paths)**, **Информация (Info)**, **Навигатор (Navigator)**, **Действия (Actions)**, **Параграф (Paragraph)** та **Символ (Character)**, **Гистограмма (Histogram)**, **История (History)**, **Подсчет слоя (Layer Comps)**, **Стили (Styles)**. Для візуалізації/приховання панелей використовують команди **Показать (Show)/Спрятать (Hide)** меню **Окно (Window)** з відповідними іменами.

Всі панелі розбиті на 5 груп, кожна з яких – це одне вікно із кількома вкладками, через які здійснюється доступ до самих панелей. Стандартне розміщення таке: **Навигатор (Navigator)**, **Информация (Info)**, **Гистограмма (Histogram)** – одна група, **Цвет (Color)**, **Образцы (Swatches)**, **Стили (Styles)** – друга, **Слои (Layers)**, **Каналы (Channels)**, **Пути (Paths)** – третя, **История (History)** та **Действия (Actions)** – четверта,

Параграф (Paragraph) та **Символ (Character)** – п'ята. Однак, захопивши корінець вкладки, можна перетягнути панель в іншу групу або взагалі поза групою.

На кожній панелі, крім стандартних кнопок згортання/розгортання та закриття, є піктограма з трикутною стрілкою, яка призначена для відкриття контекстного меню, пов'язаного з конкретною панеллю.

Панель **Кисти (Brush)** використовують для вибору розмірів та вигляду електронного “пензлика”. Панель **Настройки (Options)** надає доступ до всіх параметрів активного на даний момент інструмента (для її швидкого відкриття достатньо двічі натиснути на піктограмі потрібного інструмента на панелі інструментів). Панель **Цвет (Color)** дає змогу швидко змінити основний та фоновий кольори зі спектральної шкали, а панель **Образцы (Swatches)** – з набору зразків кольорів. Панель **Слои (Layers)** призначена для роботи з шарами зображення. За допомогою панелі **Каналы (Channels)** можна переглянути і відредагувати зображення в окремих кольороподільних каналах. Панель **Пути (Paths)** використовують для редагування контурів. У панелі **Информация (Info)** відображається різноманітна інформація стосовно зображення. Панель **Навигатор (Navigator)** дає змогу швидко змінити масштаб перегляду зображення шляхом переміщення спеціального повзунка. Панель **Действия (Actions)** використовують для запису і відтворення певних дій.

10.2.3. Створення виділених областей

Більшість команд Photoshop виконуються для виділених областей. Для їх створення використовують інструменти виділення, а для модифікації – команди меню **Выделение (Select)**. Застосування інструментів виділення з утримуванням клавіші **Shift** додає новостворену область виділення до наявної, з **Alt** – вирізає новостворену з наявної, з **Alt+Shift** – утворює їх перетин.

10.2.3.1. Виділення областей канонічної форми

Виділення областей канонічної форми здійснюють інструментами **Область (Rectangular)**, **Овальная область (Elliptical)**, **Горизонтальная строка (Single Row)** та **Вертикальная строка (Single Column)**. На панелі **Параметры (Options)** можна змінити стандартні параметри інструментів: у полі **Форма (Shape)** вибрати тип: **Прямоугольник (Rectangular)** – інструмент **Область (Rectangular)**, **Эллипс (Elliptical)** – інструмент **Овальная область (Elliptical)**, а також інструменти **Горизонтальная строка (Single Row)** та **Вертикальная строка (Single Column)**; у полі **Режим (Style)** задати форму областей: **Нормальный (Normal)** – довільної форми, **Заданные пропорции (Constrained Aspect Ratio)** – з пропорціями

сторін, заданими у полях **Ширина (Width)** та **Высота (Height)**, **Заданный размер (Fixed Size)** – фіксованого розміру, заданого у полях **Ширина (Width)** та **Высота (Height)** у пікселях (все це доступне лише для інструментів **Область (Rectangular)** та **Овальная область (Elliptical)**); у полі **Растушевка (Feather)** встановити ширину зони розтушування – пом'якшення границі виділення; опція **Сглаживание (Anti-aliased)** дає змогу згладити межі області овальної форми.

Інструмент **Область (Rectangular)** виділяє області прямокутної форми. Для цього слід підвести курсор у потрібне місце, натиснути на ліву кнопку миші і, не відпускаючи її, переміщенням курсора зобразити рамку (якщо в режимі **Нормальный (Normal)** утримувати натиснутою клавішу **Shift**, рамка матиме квадратну форму). Цілком аналогічним способом за допомогою інструмента **Овальная область (Elliptical)** виділяють області еліптичної форми. Інструменти **Горизонтальная строка (Single Row)** і **Вертикальная строка (Single Column)** виділяють відповідно рядок та стовпець шириною 1 піксель у місці натискання мишею.

У кожному наступному сеансі роботи з Photoshop програма використовує останні задані параметри інструментів. Для відновлення стандартних параметрів активного інструмента потрібно виконати команду локального меню панелі **Восстановить инструмент (Reset Tool)**, а всіх інструментів – команду **Восстановить все инструменты (Reset All Tools)**.

10.2.3.2. Области довільної форми

За допомогою інструментів **Лассо (Lasso)**, **Многоугольное лассо (Polygon Lasso)** та **Магнитное лассо (Magnetic Lasso)** можна виділяти області довільної форми. Параметрами інструментів є лише розтушування та згладжування, керування якими здійснюють за допомогою поля **Растушевка (Feather)** та опції **Сглаживание (Anti-aliased)** відповідно.

Інструмент **Лассо (Lasso)** виділяє області, нарисовані курсором однієї руки: потрібно натиснути на ліву кнопку миші, не відпускаючи її, обвести бажану ділянку і відпустити кнопку (перший та останній пікселі будуть з'єднані прямою лінією). Інструментом **Многоугольное лассо (Polygon Lasso)** виділяють області полігональної форми: треба поміщати курсор у вершини бажаної ділянки і натискати мишею (для завершення виділення досить двічі натиснути на ліву кнопку миші або ще раз ввести першу вершину). Інструмент **Магнитное лассо (Magnetic Lasso)** виділяє область на основі розмежування кольорів. Утримуючи натисненою ліву кнопку миші, слід протягувати маніпулятор по бажаній межі виділення, при цьому лінія ніби «примагнічується» до найближчого контрастного контуру. Ставити контрольні точки у процесі виділення можна, відпустивши її натиснувши кнопку миші. Приклад застосування інструмента наведено на рис. 10.6.



Рис. 10.6

10.2.3.3. Текстові області

Для створення текстових написів у Photoshop використовують інструменти: **Горизонтальний Текст (Horizontal Type)**, **Горизонтальний Текст-маска (Horizontal Type Mask)**, **Вертикальний текст (Vertical Type)** та **Вертикальний текст-маска (Vertical Type Mask)**. При використанні цих інструментів (натискання ними у потрібному місці зображення) відкривається група з панелей **Параграф (Paragraph)** та **Символ (Character)** (рис. 10.7):



Рис. 10.7

У панелі **Символ (Character)** потрібно задати: у першій стрічці – назву шрифту, далі можна вибрати гарнітуру тексту; у другій стрічці –

розмір, відстань між рядками, спосіб прилягання сусідніх букв, а також відстань між буквами; в наступному блоці задаються величина букви у відсотковому співвідношенні до реальних розмірів, а також висота індексів і колір **Color (Цвет)** шрифту; в блоці букв можна задати стиль шрифту: **Жирный (Bold)**, **Курсив (Italic)**, **Все верх (All Caps)**, **Малые верх (Small Caps)**, **Надстрочный (SuperScript)**, **Подстрочный (Sub-Script)**, **Подчеркнутый (Underline)**, **Зачеркнутый (Strikethrough)**; в останньому блоці вказується алфавіт країни, а також спосіб для згладжування контурів букв: **Sharp(Острый)**, **Crisp(Четкий)**, **Strong(Строгий)**, **Smooth(Плавный)**.

10.2.3.4. Виділення областей близьких кольорів

Найбільш потужним з усіх інструментів виділення вважається інструмент **Волшебная палочка (Magic Wand)**. Він дає змогу виділяти неперервну область, колір пікселів якої потрапляє у певний діапазон від того пікселя, на якому натиснули кнопку миші (приклад проілюстровано на рис. 10.8).



Рис. 10.8

Ширину діапазону задають у полі **Обратная чувствительность (Tolerance)** панелі **Параметры (Options)** у спеціальних одиницях, які можуть змінюватись від 0 до 255. Опції **Не грубо (Anti-aliased)**, **Непрерывный (Contiguous)** означають спосіб пошуку границі області, а опція **Использовать все слои (Use All Layers)** аналізує кольори не лише в активному шарі (як це робиться зазвичай), а й в інших видимих шарах.

Спосіб виділення простий: слід помістити курсор у місце з потрібним кольором і натиснути мишею. Таким чином можна виділяти ті області зображення, які вирізняються своїм кольором.

10.2.3.5. Модифікація виділення

Виділені області можна модифікувати за допомогою команд меню **Выделение (Select)**. Підкоманди команди **Модифікація (Modify)** використовують для зміни виділених фрагментів зображення: **Расширить (Expand)** розширює виділення, **Сжать (Contract)** – стискає, **Оптимизировать (Smooth)** – оптимізує, а **Граница (Border)** – виділяє межу поточного виділення.

Команда **Все выделено (All)** виділяє все зображення, **Снять выделение (Deselect)** відмінює виділення, а **Інверсія (Inverse)** – інвертує його (мінює місцями виділені та невиділені області).

10.2.4. Робота з виділеними областями

Якщо у зображенні є виділені області, то майже всі команди виконуватимуться лише для них. Виділені області можна вирізати, копіювати, вставляти в інші місця і т.д.

10.2.4.1. Робота з буфером обміну і шарами

Одним із найживаніших прийомів роботи із зображеннями є виділення певних фрагментів на одних зображеннях і перенесення їх в інші документи. Зробити це можна за допомогою буфера обміну (**Clipboard**). Команди для роботи з буфером містяться у меню **Редактирование (Edit)**. Команда **Скопировать (Copy)** копіює, а **Вырезать (Cut)** – вирізає вміст виділеної області в буфер (заповнивши виділення фоном, вибраним під час створення документа у рамці **Содержимое (Contents)**). Після цього стає доступною команда **Вклеить (Paste)**, яка вставляє вміст буфера у новий шар зображення, про що свідчить інформація на панелі **Слои (Layers)**.

Шари зображення можна уявити як прозорі кальки з елементами зображення, накладені один на другий. Редагування вмісту одного шару ніяк не впливає на вміст інших. Параметри вибраного (у панелі **Слои (Layers)**) шару задають на цій же панелі: повзунком **Непрозрачность (Opacity)** визначають прозорість шару від повної (0%) до нульової (100%), у спеціальному полі зліва визначають режим накладання кольорів, опція **Сохранить прозрачность (Preserve Transparency)** означає, що редагуванню підлягають лише непрозорі ділянки шару. Змінити всі ці параметри, а також перейменувати шар можна у діалоговому вікні **Параметры слоя (Options)**, яке відкривається однойменною командою меню **Слой (Layer)**.

Переміщенням шарів у панелі по вертикалі можна змінити порядок їх накладання, а натисканням на піктограму зі значком ока можна приховувати або знову показувати шар.

Слід зауважити, що команда **Скопировать (Copy)** копіює дані лише з активного шару. Якщо ж потрібно зробити це з усіх видимих шарів, використовують команду **Скопировать совмещенные данные (Copy Sample Merged)**.

10.2.4.2. Трансформування виділеної області

Для трансформування виділеної області за допомогою миші використовують підкоманди команди меню **Редактирование/Трансформирование (Edit/Transform)**: **Масштабирование (Scale)** дає змогу змінити розміри виділення (якщо утримувати клавішу **Shift**, то зі збереженням пропорцій), **Поворот (Rotate)** – повернути, **Наклон (Skew)** – нахилити, **Искажение (Distort)** – розтягнути в різних напрямках, **Перспектива (Perspective)** – додати перспективу. Задати точні параметри трансформування можна у відповідних полях діалогового вікна, яке відкривається підкомандою **Числовые значения (Numeric)**. Підтверджують трансформацію натисканням на клавішу **Enter**, відміняють – клавішею **Esc**.

10.2.4.3. Обробка виділених областей

Змінити яскравість та контрастність виділеної області можна підкомандою **Яркость/Контраст (Brightness/Contrast)** команди **Изображение/Коррекция (Image/Adjust)** (рис. 10.9).

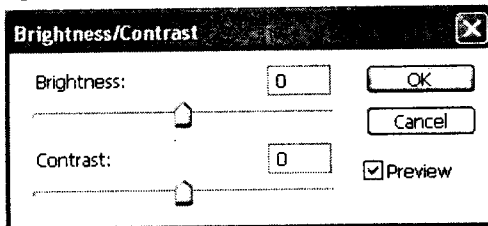


Рис. 10.9

Відповідні значення задають у полях **Яркость (Brightness)** та **Контраст (Contrast)** або встановлюють за допомогою повзунків. Перегляд результату забезпечує опція **Просмотр (Preview)**.

Підкоманда **Негатив (Invert)** інвертує виділену область; **Выровняют яркость (Equalize)** рівномірніше розподіляє значення кольорів по всьому діапазону – від чорного до білого (опція **Только выделенную область (Selected Area Only)** дозволяє корекцію лише у виділеній області, а **Все изображение на основе области (Entire Image Based on Area)** – у всьому зображенні); **Изогелия (Threshold)** перетворює кольорове або

напівтонеове зображення у чорно-біле (поріг яскравості визначають у діалоговому вікні); **Постеризация (Posterize)** зменшує кількість рівнів сірого кольору у зображенні. Цю кількість задають у діалоговому вікні **Уровни (Levels)** (рис. 10.10), яке викликається командою **Изображение/Коррекция/Уровни (Image/Adjust/Levels)**:

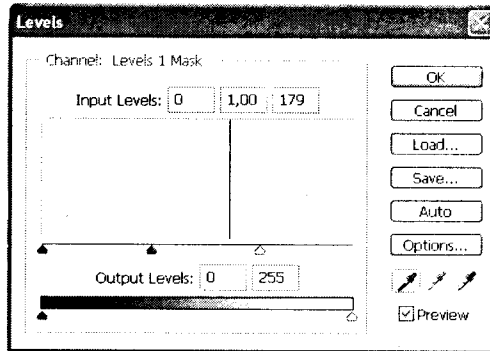


Рис. 10.10

Якщо виділених областей немає, зміни стосуватимуться всього зображення або шару.

Якщо параметр **Растушевка (Feather)** інструментів виділення відмінний від нуля, то після переміщення або вставлення виділеної області її краї переходитимуть у прозорий фон. Цього ж можна досягнути командою **Выделение/Растушевка (Select/Feather)**. Такий ефект використовують для створення віньсток.

10.2.4.4. Заповнення та обведення

Для заповнення виділеної області використовують команду **Редактирование/Выполнить заливку (Edit/Fill)**. Відкриється діалогове вікно **Заполнить (Fill)** (рис. 10.11):

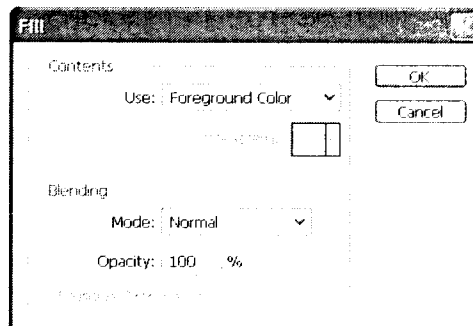


Рис. 10.11

У ньому задають: у полі **Стиль (Use)** – колір заповнення зі значень **Основной цвет (Foreground Color)** – основний, **Фоновый цвет (Background Color)** – фоновий, а також чорний, 50%-й сірий та білий; у рамці **Наложение (Blending)**: у полі **Непрозорність (Opacity)** визначають відсоток непрозорості заповнення; у полі **Режим (Mode)** – режим накладання кольорів (див. “Режими накладання кольорів” підрозділу “малювання”); опція **Сохранить прозорність (Preserve Transparency)** захищає від заповнення прозорі області. Швидко заповнити область основним непрозорим кольором можна за допомогою комбінації клавіш **Alt+Backspace**, а фоновим – **Backspace** або **Delete** (щоправда, лише у базовому шарі; в інших шарах обидві клавіші просто очищають (роблять прозорою) виділену область).

Обведення контура виділеної області здійснюють у діалоговому вікні **Обводка (Stroke)**, яке відкривається командою **Редактирование/Обводка (Edit/Stroke)**: у полі **Толщина (Width)** задають товщину контура, вибирають відповідне значення опції **Позиція относительно границы (Location)** зі значень **Внутри (Inside)**, **По центру (Center)** та **Снаружи (Outside)**, які визначають позицію контура відносно межі виділення (всередині, в центрі або зовні); смисл полів та опції рамки **Наложение (Blending)** аналогічний до відповідних параметрів заповнення. Вигляд даного діалогового вікна ілюструє рис. 10.12.

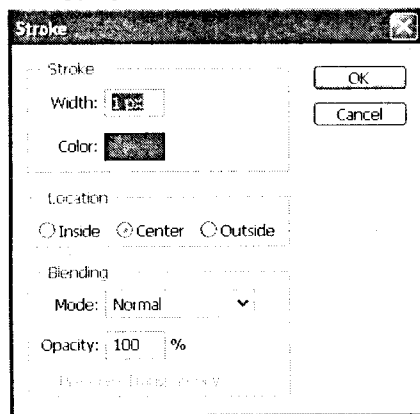


Рис. 10.12

Якщо виділення проводилося інструментом із ненульовим параметром **Растушевка (Feather)** або до нього застосовувалась команда **Выделение/Растушевка (Select/Feather)**, то після його заповнення краї виглядатимуть розмитими. Ця властивість у поєднанні з командами **Выделение/Модификация/Граница (Select/Modify/Border)**, вирізання та вставлення дає змогу створювати ефект сяйва навколо різних об'єктів.

Що стосується зміни основного та фонового кольорів, то це можна зробити кількома способами: у діалоговому вікні **Палитра цветів (Color Picker)**, яке відкривається після натискання на піктограми **Основной цвет (Foreground Color)** та **Фоновый цвет (Background Color)** панелі інструментів; на панелі **Синтез (Color)** зі спектральної шкали (основний – звичайним натисканням мишею, а фоновий – з утримуванням клавіші **Alt**); на панелі **Каталог (Swatches)** аналогічним чином з набору зразків кольорів; тим самим способом просто із зображення інструментом **Пипетка (Eyedropper)**, параметр якого **Размер образца (Sample Size)** вказує на спосіб визначення кольору: **Точка (Point Sample)** – колір пікселя, на якому було натиснуто, **В среднем 3 на 3 (3 by 3 Average)** – усереднений з площі 3x3 пікселі, **В среднем 5 на 5 (5 by 5 Average)** – усереднений з площі 5x5 пікселів.

Особливо зручним є вибір з панелі **Каталог (Swatches)**. Змінити стандартний набір кольорів можна за допомогою команд локального меню панелі **Загрузить каталог (Load Swatches)** (додає до наявного набору) або **Заменить каталог (Replace Swatches)** (заміняє наявний набір): у діалоговому вікні, що відкриється, потрібно відшукати підкаталог **Palettes** робочого каталога програми Photoshop і вибрати один із файлів з розширенням ***.aco**. Відновлюється стандартний набір командою **Восстановить каталог (Reset Swatches)**.

10.2.5. Малювання

За допомогою інструментів малювання у Photoshop можна створювати найрізноманітніші зображення. При цьому слід мати на увазі: якщо у зображенні є виділені області, то малювання відбувається тільки там.

10.2.5.1. Режими накладання кольорів

Вибраний режим накладання кольорів визначає, як взаємодітимуть між собою пікселі новостворюваного елемента зображення із наявними. Вибір режиму здійснюється у лівому верхньому полі палітри параметрів інструментів малювання зі значень: **Нормальный (Normal)** – наявний колір замінюється на накладуваний; **Растворение (Dissolve)** – новий колір випадковим чином замінює наявний (якщо прозорість ненульова); **Умножение (Multiply)** – множить значення обох кольорів і ділить на 255 (результат завжди темніший); **Осветление (Screen)** – освітлює наявний колір залежно від накладуваного; **Перекрытие (Overlay)** – кольори змішуються (темні затемнюються, світлі освітлюються); **Мягкий свет (Soft Light)** – дає ефект освітлення розсіяним нерізким світлом; **Жесткий свет (Hard Light)** – дає ефект освітлення пучком яскравого напрямленого світла; **Осветление основы (Color Dodge)** – освітлює наявний колір;

Затемнение основы (Color Burn) – затемнює наявний колір; **Замена темным (Darken)** – затемнює кольори, світліші від накладуваного; **Замена светлым (Lighten)** – освітлює кольори, темніші від накладуваного; **Разница (Difference)** – віднімає яскравість накладуваного від яскравості наявного; **Исключение (Exclusion)** – схожий на попередній, тільки кольори виходять м'якшими; **Цветовой тон (Hue)** – змінюється лише кольоровий тон наявного кольору, який береться від накладуваного; **Насыщенность (Saturation)** – змінюється лише насиченість наявного кольору; **Цветность (Color)** – змінюється насиченість і кольоровий тон наявного кольору; **Яркость (Luminosity)** – змінюється лише яскравість наявного кольору.

10.2.5.2. Малювання ліній

Лінії малюють інструментом **Линия (Line)** таким чином: потрібно помістити курсор у місце початку лінії, натиснути мишею, не відпускаючи кнопки, перемістити курсор у місце закінчення лінії і відпустити кнопку. Якщо при цьому утримувати клавішу **Shift**, то лінії малюватимуться під кутом, кратним 45 градусів. Товщину лінії (у пікселях) задають у полі **Толщина (Line Width)** панелі параметрів. Опції **В начале (Start)** та **В конце (End)** використовують для додання стрілок відповідно на початку та в кінці лінії, причому форму стрілки можна змінити у діалоговому вікні, яке відкривається після натискання на кнопку **Форма (Shape)**: у полях **Ширина (Width)**, **Длина (Length)** та **Кривизна (Concavity)** задають (у %) відповідно ширину, довжину та кривизну стрілки.

10.2.5.3. Вільне малювання

Для малювання ліній довільного вигляду основним кольором використовують інструменти **Карандаш (Pencil)**, **Кисть (Paintbrush)** та **Аэрограф (Airbrush)**. Процес малювання полягає у переміщенні курсора миші із натиснутою лівою кнопкою. Якщо утримувати натиснутою клавішу **Shift**, то можна малювати ламані: точки, що вводяться окремими натисканнями миші, будуть з'єднані прямими лініями. Малювання ж з утриманням цієї клавіші допускає лише прямі лінії під кутом 90°.

Розмір та форму “пензлика” вибирають на панелі **Кисти (Brushes)**. Змінити стандартний набір “пензлів” можна за допомогою команд локального меню панелі **Загрузить кисти (Load Brushes)** (додає до наявного набору) або **Заменить кисти (Replace Brushes)** (заміняє наявний набір): у діалоговому вікні, що відкриється, потрібно відшукати підкаталог **Brushes** робочого каталога програми Photoshop і вибрати один із файлів з розширенням **.abr**. Відновлюється стандартний набір командою **Восстановить кисти (Reset Brushes)**.

Інструмент **Карандаш (Pencil)** рисує лінії з чітко окресленими краями. Параметри інструмента (крім уже розглянутих) такі: увімкнення опції **Переход (Fade)** дає змогу створювати мазки кольорових переходів заданої у лівому полі довжини від основного кольору до прозорого (значення **Прозрачному (Transparent)** правого поля) або фоновому (значення **Фоновому (Background)**). Увімкнення опції **Автоматическое стирание (Auto Erase)** дає такий ефект: якщо на початку малювання під курсором є основний колір, то малювання здійснюється фоновим (у даний момент); якщо ж колір на початку інший, то інструмент малює основним кольором.

Інструмент **Кисть (Paintbrush)** малює подібно, однак для нього доступні “пензлики” з розмитими краями. Крім цього, увімкнення опції **Размытые края (Wet Edges)** дає змогу досягти ефекту використання акварелі: кольори у центрі ліній менш насичені, ніж на краях. Опція **Переход (Fade)** діє аналогічно до інструмента **Карандаш (Pencil)**.

Інструмент **Аэрограф (Airbrush)**, у свою чергу, подібний до інструмента **Кисть (Paintbrush)**, лише краї намальованих ним ліній виглядають іще м'якшими. У його параметрах повзунок **Непрозрачность (Opacity)** замінено на **Нажим (Pressure)**: чим більше це значення, тим щільнішою буде лінія.

10.2.5.4. Витирання фрагментів зображення

Під витиранням мається на увазі заповнення фоновим кольором у базовому шарі **Задний план (Background)** або прозорим в інших шарах. Тому одним із способів витирання є заповнення виділених областей фоновим кольором (у базовому шарі) або виконання команди **Редактирование/Очистить (Edit/Clear)**. Однак у більшості випадків потрібен більш гнучкий механізм витирання, який реалізується за допомогою інструментів **Ластик (Eraser)** та **Фоновый ластик (Background Eraser)**. Фактично їх використання зводиться до малювання фоновим або прозорим кольором.

У верхньому лівому кутку панелі параметрів інструмента міститься поле для вибору типу ластика: **Кисть (Paintbrush)**, **Аэрограф (Airbrush)**, **Карандаш (Pencil)** – аналоги відповідних інструментів рисування з майже всіма своїми параметрами (тільки опція **Переход (Fade)** змінює назву на **Обесцвечивание (Fade)**), **Ластик (Block)** – електронний аналог звичайного ластика (гумки) квадратної форми. Кнопка **Стереть слой (Erase Image)** (якщо шар єдиний – то **Стереть все (Erase Layer)**) витирає все зображення з активного шару. Опція **До сохраненной версии (Erase to Saved)** дає змогу не витирати зображення під інструментом, а відновлювати його старий вигляд, який береться з останньої збереженої версії файлу.

Повністю ж відновити останню збережену версію файлу можна командою **Файл/Восстановить (File/Revert)**.

Інструмент **Волшебный ластик (Magic Eraser)** поєднує в собі дію інструментів **Волшебная палочка (Magic Wand)** та **Очистить (Clear)**.

10.2.5.5. Інструменти заповнення

Інструмент **Заливка (Paint Bucket)** призначений для заповнення фрагментів зображення основним кольором. Розміри області для заповнення визначаються подібно до інструмента **Волшебная палочка (Magic Wand)**: зафарбовуються всі суміжні з натиснутим пікселі, колір яких потрапляє у заданий діапазон. Ширину цього діапазону задають у полі **Допуск (Tolerance)** панелі параметрів інструмента. Опція **Совмещенные данные (Sample Merged)** визначає, аналізувати пікселі лише активного шару (вимкнута) чи всіх видимих шарів (увімкнута).

Інструментом **Градиент (Gradient)** можна заповнювати все зображення (або виділену область) плавним переходом кольорів: слід помістити курсор у потрібне місце (початкову точку), натиснути ліву кнопку миші, не відпускаючи її, перемістити курсор у кінцеву точку і відпустити кнопку. Переходом кольорів буде заповнено область між початковою та кінцевою точками, решта області – однорідними кольорами залежно від вибраної гами. Приклад застосування інструменту ілюструє рис. 10.13.



Рис. 10.13

Параметри переходу визначають на панелі параметрів інструмента. Кольорову гаму переходу вибирають у полі **Стиль (Gradient)** зі списку значень, серед яких основними є: **Foreground to Background** – від основного кольору до фонового, **Foreground to Transparent** – від основного до прозорого, **Transparent to Foreground** – від прозорого до основного, **Spectrum** – повний спектр, **Transparent Rainbow** – кольори веселки з прозорими краями, **Transparent Stripes** – чередування ділянок основного та прозорого кольору, **Black, White** та інші перерахування кольорів – перехід від одного кольору до іншого. У полі нижче вибирають спосіб заповнення (**Линейный (Linear)** – перехід по прямій лінії, **Радиальный (Radial)** – перехід вздовж радіуса кола), опція **Маска (Mask)** дає дозвіл на використання прозорого кольору (інакше замість нього використовуватиметься основний), а **Настройка (Dither)** призначена для згладжування переходів кольорів. Кнопка **Редактор (Edit)** відкриває спеціальне діалогове вікно, де можна відредагувати параметри наявного типу заповнення або створити новий. Ділянка вікна у правому нижньому куті призначена для перегляду результатів застосування вибраних параметрів.

10.2.6. Редагування зображень

Інструменти редагування зображень дають змогу підкоригувати деякі фрагменти зображень. Розміри захоплених ділянок та режими модифікації залежать від вибору типу та розмірів пензлика на панелі **Кисти (Brushes)**. Наявність виділених областей дозволяє редагування тільки у них.

10.2.6.1. Інструменти тонування

Інструменти тонування містяться в одному наборі на панелі інструментів. **Осветитель (Dodge)** призначений для освітлення ділянок зображення, а **Затемнитель (Burn)** – для їх затемнення. Вибрати один із них можна (крім стандартного способу з панелі інструментів) на панелі параметрів у полі **Інструмент (Tool)**. У верхньому лівому полі панелі вибирають режим тонування: **Тени (Shadows)** – редагувати темні компоненти зображення, **Средние тона (Midtones)** – середні тони, **Света (Highlight)** – світлі компоненти. Збільшення параметра **Експозиція (Exposure)** посилює ефект освітлення для інструмента **Осветитель (Dodge)** та ефект затемнення для інструмента **Затемнитель (Burn)**.

Третій інструмент набору – **Губка (Sponge)** – використовують для підвищення або пониження насиченості (інтенсивності) кольорів. У верхньому лівому полі панелі інструментів тепер вибирають напрям зміни насиченості: **Уменьшить (Desaturate)** – зменшити, **Увеличить (Saturate)** – збільшити. Повзунок **Експозиція (Exposure)** для цього інструмента замінений на **Нажим (Pressure)**, який посилює або послаблює дію інструмента.

Спосіб використання інструментів тонування аналогічний до рисування: слід помістити курсор у потрібне місце зображення, натиснути на ліву кнопку миші і, не відпускаючи її, переміщувати інструмент по зображенню.

10.2.6.2. Інструменти фокусування та Палец (Smudge)

Інструменти фокусування впливають на контраст кольорів між сусідніми пікселями: **Размытие (Blur)** пом'якшує переходи кольорів, роблячи їх "розмитими", а **Резкость (Sharpen)** – навпаки, надає їм більшої чіткості та виокремлює деталі. На панелі параметрів у лівому верхньому полі задають режим накладання кольорів, у полі **Інструмент (Tool)** вибирають потрібний інструмент, а повзунком **Нажим (Pressure)** посилюють або послаблюють ефект використання інструментів. Вимкнута опція **Совмещенные данные (Sample Merged)** означає, що аналізуються лише пікселі активного шару, а увімкнена – пікселі усіх видимих шарів.

Дія інструмента **Палец (Smudge)** полягає у змішуванні кольорів, ніби їх розмазують пальцем. Крім цього, увімкнення опції **рисование пальцем (Finger Painting)** дає змогу додати до наявних ще й основний колір: рисування інструментом починається з основного кольору і переходить у наявні.

Спосіб використання описаних інструментів аналогічний до тонування та рисування.

10.2.6.3. Інструмент Штамп (Rubber Stamp)

Штамп (Rubber Stamp) – багатоцільовий інструмент, який можна використати для клонування фрагментів зображення, відновлення помилково відредагованих ділянок або застосування шаблону. На панелі параметрів інструмента, крім уже описаних опцій та полів, є поле **Вариант (Option)**, яке визначає спосіб застосування інструмента.

Значення **Клон (произвольно) (Clone (non-aligned))** дає змогу клонувати (копіювати) вибраний фрагмент у довільному місці зображення. Вибір фрагмента для клонування (чи, точніше, його центра) здійснюється простим натисканням мишею у потрібному місці зображення з утримуванням клавіші **Alt**. Після цього "малювання" у довільних місцях документа призводить до появи копій вибраного фрагмента (рис. 10.14).

Варіант **Клон (с выравниванием) (Clone (aligned))** відрізняється від попереднього тим, що клонування в різних місцях зображення призводить не до появи кількох копій, а до "проявлення" над зображенням різних місць вибраного фрагмента.

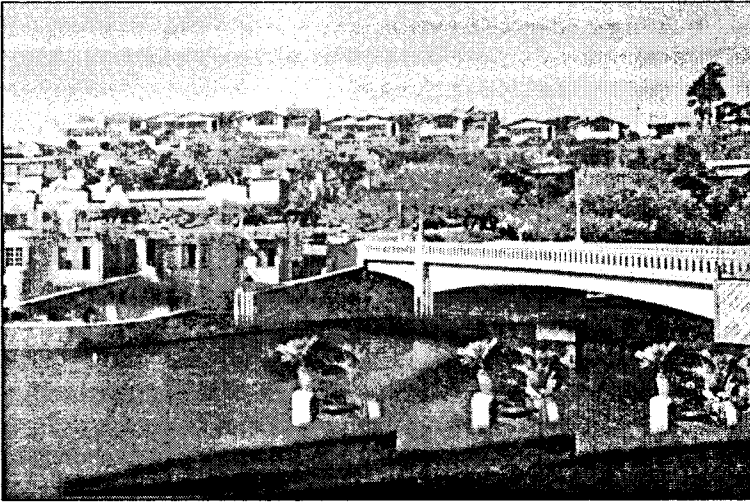


Рис. 10.14

Щоб застосувати варіанти інструмента **Образец (произвольно)** (**Pattern (non-aligned)**) та **Образец (с выравниванием)** (**Pattern (aligned)**), спочатку потрібно визначити шаблон для клонування – аналог вибраного фрагмента. Для цього відкривають потрібне зображення, виділяють прямокутну область (з нульовим параметром **Растушевка (Feather)**) і виконують команду **Редактирование/ Определить образец (Edit/Define Pattern)**. Подальше використання інструмента накладає на наявне зображення набір зображень-шаблонів, розміщених один за другим (мозаїкою), причому другий варіант (з вирівнюванням) вирівнює шаблони по горизонталі та вертикалі в усіх заповнених ділянках. Кілька зображень, спеціально створюваних як шаблони, можна знайти у підкаталозі **Patterns** робочого каталога програми Photoshop, а також у робочому каталозі Windows.

Для варіанту **Из снимка (From Snapshot)** теж треба попередньо визначити знімок (версію зображення або його виділеної області, що зберігатиметься у пам'яті) командою **Редактирование/Сделать снимок (Edit/Take Snapshot)** або **Редактирование/Сделать совмещенный снимок (Edit/Take Merged Snapshot)** (відповідно для зображень з активного шару та всіх видимих шарів). У цьому випадку використання інструмента відновлює зображення зі знімка. Аналогічно працює варіант **Из сохраненной версии (From Saved)**, який відновлює зображення з останньої збереженої версії документа. Варіант **Импрессионизм (Impressionist)** дає змогу створювати ефекти, властиві цьому стилю мистецтва.

10.2.6.4. Швидке редагування

Зображення переважно займають досить багато місця на диску та у пам'яті, і з часом може виявитись, що ресурсів комп'ютера недостатньо для проведення редагування. У такому випадку доцільно скористатись командою швидкого редагування, яка завантажує не все зображення повністю, а лише деякі його фрагменти. Недоліком цього способу є те, що він працює лише із документами у форматі **TIFF** (нестиснутому) або **Scitex CT**. Тому перед його застосуванням необхідно перетворити потрібні зображення у документи названих вище форматів.

Для того, щоб розпочати процес швидкого редагування, треба виконати підкоманду **Quick Edit** команди **Файл/Імпортувати (File/Import)** і вибрати потрібний файл. Після цього відкриється діалогове вікно із зображенням, на якому рамкою вказують фрагмент для редагування або, увімкнувши опцію **Сетка (Grid)** і задавши кнопками зі значками + та – кількість рядків та стовпців розбиття, потрібну комірку. Після натискання на кнопку **Да (ОК)** вказаний фрагмент буде завантажено. Для збереження внесених під час редагування змін використовують команду **Файл/Експортувати/Сохранить после Quick Edit (File/Export/Quick Edit Save)**.

10.2.7. Робота з контурами

Контури – це єдині у Photoshop об'єкти векторної графіки. Вони складаються з прямолінійних та криволінійних сегментів, створених спеціальним інструментом (рис. 10.15).

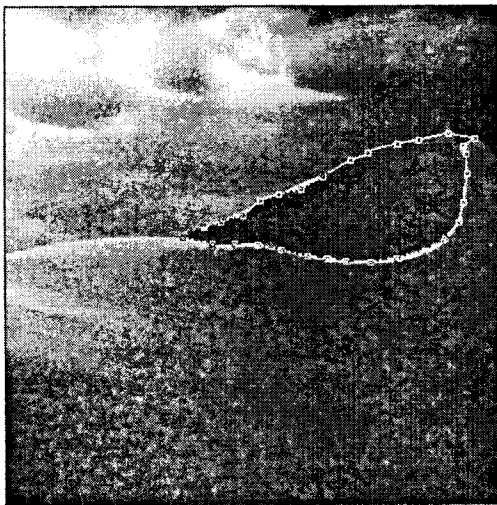


Рис. 10.15

Контури можна редагувати, заповнювати, обводити, а найголовніше – перетворювати у виділені області і навпаки.

10.2.7.1. Створення контурів

Контур утворюють введені мишею точки (вузли), які з'єднуються між собою прямолінійними або криволінійними відрізками. Якими саме – залежить від способу введення вузлів.

Першу точку контура вводять простим натисканням мишею у потрібному місці зображення. Щоб додати прямолінійний сегмент, треба перемістити курсор у місце, де має знаходитись наступний вузол, і знову натиснути мишею. Нововведений вузол буде з'єднаний із попереднім прямою лінією. Якщо під час цього утримувати натиснутою клавішу **Shift**, то кут нахилу сегмента буде кратним 45° .

Для додання криволінійного сегмента курсор поміщають у потрібне місце, натискають на ліву кнопку миші і, не відпускаючи її, починають далі переміщувати курсор. В результаті цього з'явиться напрямна лінія з двома напрямними точками, відстань між якими та кут нахилу визначають форму криволінійного сегмента. Досягнувши бажаного вигляду кривої, кнопку миші відпускають. Утримування під час цього клавіші **Shift** задає кут нахилу напрямної лінії, кратний 45° .

Інструмент **Перо (Pen)** має один параметр – опцію **Предварительный просмотр (Rubber Band)**, увімкнення якої дає змогу побачити попередній вигляд нового сегмента.

Щоб правильно завершити введення контура, його потрібно замкнути – ввести останній вузол на місці першого. Для цього необов'язково ідеально точно наводити курсор – якщо помістити його достатньо близько до початкового вузла, то введений вузол притягнеться до наявного, про що свідчить поява кружечка біля інструмента. Тепер можна вводити новий контур, який разом із уже введеним розглядатимуться як підконттури одного загального контура, котрий відображається на панелі **Контури (Paths)** подібно до окремого шару.

Для створення справді нового контура потрібно натиснути на піктограму **Создает новый контур (New Path)** у нижній частині панелі **Контури (Paths)** і сформувати його за допомогою інструмента **Перо (Pen)**. У вікні документа можна побачити лише один – активний – контур, вибраний на панелі **Контури (Paths)**, а, щоб приховати контури, достатньо не вибрати жодного, натиснувши мишкою на вільному місці внизу панелі. Для перейменування контура слід двічі натиснути мишкою на його відображення на панелі.

10.2.7.2. Редагування контурів

Рідко вдається відразу побудувати потрібний контур – переважно для досягнення поставленої мети його потрібно відредагувати. Всі інструменти

для редагування контурів містяться в одному наборі з інструментом **Перо (Pen)**. Найпростіше діє інструмент **Перо – (Delete anchor point)**, призначений для вилучення вузлів контура: потрібно навести курсор на зайвий вузол і натиснути мишею. **Перо + (Add anchor point)**, навпаки, додає вузли: треба навести курсор на сегмент контура, натиснути мишею і, якщо необхідно, перемістити курсор, не відпускаючи клавіші, – це дає змогу змінити кривизну створених сегментів. Наведення курсора на один із вузлів або напрямну точку перетворює інструмент **Перо + (Add anchor point)** на інструмент **Стрелка (Direct Selection)**.

Інструмент **Стрелка (Direct Selection)** використовують для вибору та переміщення всього контура (або його підконтура), групи вузлів, окремого вузла або напрямної точки. Натискання мишкою на сегменті контура виділяє його (однак вузли будуть невибраними – без чорного заповнення). Для вибору і переміщення одного вузла достатньо захопити його (тобто натиснути мишею і утримувати ліву клавішу) і перетягнути на потрібне місце (вибраний вузол набуває чорного заповнення). Таким же чином переміщують і напрямні точки, змінюючи кривизну сегмента, причому обидві точки залишатимуться на одній лінії з вузлом.

І, нарешті, інструмент **Угол (Convert anchor point)** дає змогу перетворювати вузли одного типу в інші. Наприклад, натискання інструментом на гладкому вузлі (у якому криволінійні сегменти неперервно переходять один в другий) перетворює його на кутовий, який не має напрямних точок. І навпаки, захоплення кутового вузла і подальше переміщення курсора перетворює вузол на гладкий.

10.2.7.3. Робота з контурами

Як уже було сказано, активні контури можна заповнювати, обводити та перетворювати у виділені області і навпаки. Якщо у контура є підконтури і якісь із них виділені, то робота проходитиме з ними, якщо ж виділених немає, то описані нижче команди стосуватимуться всього контура.

Для перетворення контура у виділену область використовують команду **Образовать выделенную область (Make Selection)** локального меню панелі **Контури (Paths)**. У діалоговому вікні, що відкривається, у полі **Радиус растушевки (Feather Radius)** потрібно встановити ширину зони розтушування і в разі необхідності увімкнути опцію **Сглаживание (Anti-aliased)**, яка дає змогу згладити межі області. Якщо у зображенні немає виділення, то після натискання на кнопку **Да (ОК)** утвориться виділена область з формою контура, якщо ж попередньо було щось виділено, то стануть доступними значення опції **Операция (Operation): Новая выделенная область (New Selection)** – ділянка, обмежена контуром, стає новою виділеною областю; **Добавить к выделенной области (Add to Selection)** – ділянка, обмежена контуром, додається до наявного виділення;

Вычесть из выделенной области (Substract from Selection) – ділянка виключається з наявного виділення; **Пересечение с выделенной областью (Intersect with Selection)** – виділяється область, що є перетином ділянки і наявного виділення.

Зворотну операцію – перетворення виділеної області в контур – виконують за допомогою команди **Образовать рабочий контур (Make Work Path)**, після чого у діалоговому вікні слід задати потрібне значення поля **Допуск (Tolerance)**, яке визначає точність перетворення області в контур (чим більше значення, тим точніше).

Заповнюють активні контури (або їх виділені підконтури) командою **Залить контур (Fill Path)** локального меню панелі **Контури (Paths)** (але не командою меню **Редактирование (Edit)**). Поля та опції діалогового вікна, що відкриється, аналогічні однойменним полям діалогових вікон **Заливка (Fill)** (команда **Редактирование/Выполнить заливку (Edit/Fill)**) та **Образовать выделенную область (Make Selection)** (описаного вище). Обведення контура виконують командою **Обвести контур (Stroke Path)** локального меню панелі **Контури (Paths)**, вибираючи в однойменному діалоговому вікні інструмент, яким буде здійснено обведення. Параметри інструмента та розміри пензлика встановлюють попередньо на відповідних панелях.

Активні контури або їх виділені сегменти можна копіювати (вирізати) і вставляти відповідними командами меню **Редактирование (Edit)**, а також дублювати командою **Создать копию контура (Duplicate Path)** локального меню панелі **Контури (Paths)** (але тільки для проіменованого контура). Натискання на клавішу **Delete** вилучає виділену точку (якщо вона є), наступне – виділений підконтур (якщо він є), а ще одне – весь контур.

10.2.8. Фільтри

Фільтри використовують для автоматичної обробки та редагування зображень. Будь-який фільтр працює лише з одним – активним – шаром, а якщо є виділена область – то лише з нею. Для повторення дії останнього застосованого фільтра слід виконати команду **Фільтр/Останній фільтр (Filter/Last Filter)**, а для послаблення дії фільтра – команду **Фільтр/Ослабить (Filter/Fade)**. У багатьох діалогових вікнах фільтрів є вікно для попереднього перегляду результатів застосування, у яких можна збільшувати чи зменшувати масштаб перегляду кнопками зі значками + та –, а також “прокручувати” зображення, захоплюючи та переміщуючи його; увімкнення ж опції **Прогляд (Preview)** дає змогу переглядати результат застосування фільтра безпосередньо на зображенні. Процес фільтрування завжди можна обірвати, натиснувши на клавішу **Esc**.

Усі фільтри розділені на групи: **Имитация (Artistic)**, **Размытие (Blur)**, **Штрихи (Brush Strokes)**, **Деформация (Distort)**, **Шум (Noise)**, **Оформление (Pixelate)**, **Визуализация (Render)**, **Резкость (Sharpen)**, **Эскиз (Sketch)**, **Стилизация (Stylize)**, **Текстура (Texture)**, **Видео (Video)**, **Другие (Other)**.

10.2.9. Друкування зображень

Для друкування активного зображення слід виконати команду **Файл/Печатать (File/Print)**, яка викличе діалогове вікно **Печать (Print)**, зображене на рис. 10.16.

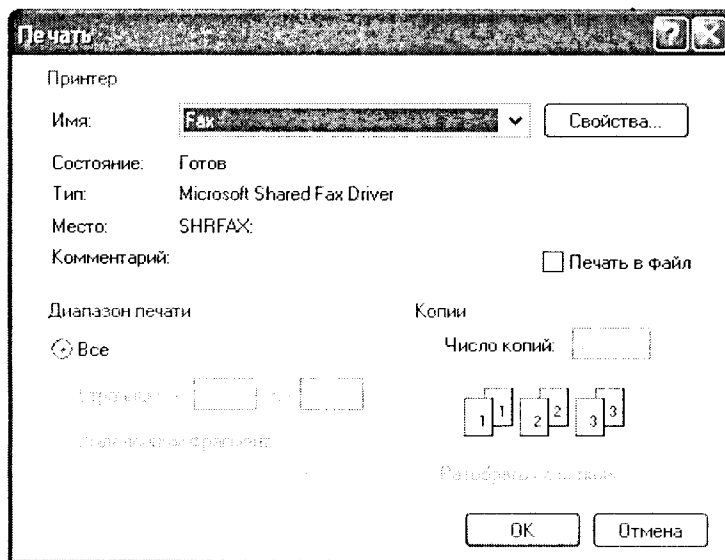


Рис. 10.16

У полі **Копии (Copies)** задається кількість копій, причому опція **Разобрать по копиям (Collate Copies)** визначає послідовність друкування – окремо кожну копію документа чи підряд копій кожної сторінки. Опція **Печать в файл (Print to File)** призначена для спрямування друкованого образу документа у файл. Якщо у зображенні є виділена область, то можна роздрукувати лише її, вибравши значення **Выделенный фрагмент (Selection)** опції **Диапазон печати (Print Range)**.

Змінити стандартні параметри друкування можна у діалоговому вікні **Параметры страницы (Page Setup)**, яка відкривається натисканням на кнопку **Установка (Setup)**. У полі **Принтер (Printer)** обирають принтер, у рамці **Бумага (Paper)** вибирають формат паперу та спосіб його подачі, а в рамці **Ориентация (Orientation)** – орієнтацію сторінок.

10.3. Векторний графічний редактор CorelDRAW

CorelDRAW – це пакет засобів для редагування зображень і видавничої діяльності, можливості роботи з різними графічними об'єктами для отримання якісних зображень, широкий спектр засобів для створення логотипів, емблем та інших елементів текстового оформлення.

CorelDRAW є зручним інструментом для малювання фігур. За його допомогою можна створити як прості ілюстрації, так і складні технічні креслення, а також кольорові художні твори. **CorelDRAW** містить корисний інструмент **Shape** (форма), який дає змогу легко змінити вигляд об'єкта так, як потрібно користувачу. Програма зображає всі об'єкти в спеціальному векторному вигляді. Саме тому під час роботи з об'єктом не псується якість зображення.

До головних можливостей програми належить малювання ліній і геометричних фігур, а також визначення вигляду їхнього контуру та характеру заповнення. У цій програмі є низка ефектів, за допомогою яких можна затінювати або підсвічувати об'єкти, виконувати їхній плавний перехід з одного в інший і змінювати форму всього об'єкта за допомогою задання спеціальної форми. Об'єкти, створені в **CorelDRAW**, можна переміщувати в інші прикладні програми за допомогою експортування відповідних файлів або використання буфера інформаційного обміну **Clipboard** системи **Windows**.

10.3.1. Інтерфейс CorelDRAW

Розглянемо роботу з пакетом на прикладі програми **CorelDRAW 11**. Як і у більшості програм, написаних для **Windows**, усі функції **CorelDRAW** оформлені у вигляді меню, діалогових вікон і кнопок інструментів.

Після запуску програми з'являється діалогове вікно **Welcome to CorelDRAW** зі стандартними пропозиціями щодо початку роботи (рис. 10.17):

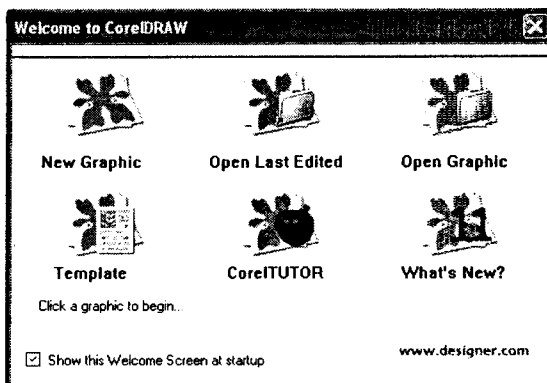


Рис. 10.17

Коротко розглянемо головні елементи вікна CorelDRAW 11 (рис. 10.18). Над вікном активної програми CorelDRAW є *рядок заголовка*. *Рядок меню* розміщений під заголовком і містить додаткові меню, призначені для керування програмою CorelDRAW. *Рядок стану* є внизу екрана. В ньому міститься інформація про поточний стан об'єктів CorelDRAW, наприклад, їхнє розміщення на екрані, розмір, кількість вибраних об'єктів і характеристики внутрішнього заповнення. *Панель керування* міститься безпосередньо під рядком головного меню. На ній є кнопки для зручного та швидкого виклику функцій CorelDRAW. У CorelDRAW можливо налаштувати панель керування, додаючи чи забираючи окремі кнопки до/з панелі за допомогою засобів налаштування CorelDRAW.

Панель властивостей – це контекстно-залежна командна панель, яка змінюється залежно від команд, кнопок і режимів роботи з документом.

Горизонтальна лінійка розміщена під панеллю властивостей, а *вертикальна* – уздовж панелі інструментів. За їхньою допомогою можна розташовувати об'єкти в зоні малювання і визначати їхні розміри. Шкалу лінійок визначає користувач. Зазвичай початок координат сторінки розміщують у його нижньому куті, щоб усі координати були додатніми.

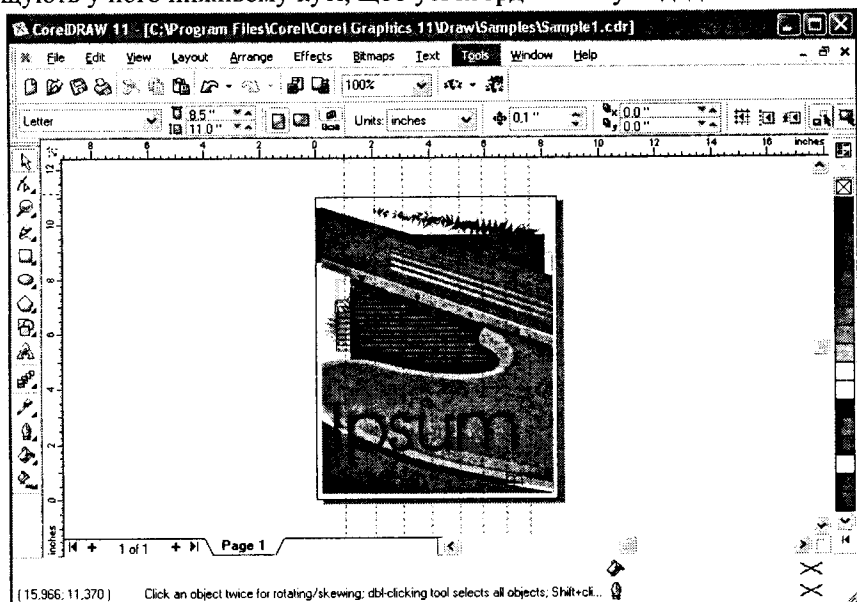


Рис. 10.18

Область малювання міститься в центрі екрана. Саме тут виконують редагування і вдосконалення малюнка. В середині зони малювання розміщений аркуш. Усе, що зображено на аркуші, можна надрукувати у вигляді документа.


Горизонтальну і вертикальну смуги перегляду використовують для переміщення по зоні малювання. Натискання на якійсь зі стрілок на кінцях смуг перегляду, приводить до переміщення зони малювання у відповідному напрямі. Для швидкого переміщення слід захопити і протягнути «повзунок».

CorelDRAW дає змогу створювати і редагувати багатосторінкові документи. Якщо в поточному документі CorelDRAW створити додаткові сторінки, то на горизонтальній смугі перегляду з'явиться вкладка з номером поточної сторінки, а також кількість сторінок у файлі.

Колірна палітра може бути розміщена вздовж вертикальної смуги перегляду. Її кольори використовують для задання кольору контуру і заповнень малюнків у зоні малювання.

10.3.2. Огляд панелі інструментів

Основою роботи в CorelDRAW, крім команди меню **Save**, є панель інструментів, без якої неможливо створити жодного малюнка. Розглянемо призначення інструментів цієї панелі.

 Інструмент **Pick** (Вибір) є головним для всіх операцій у CorelDRAW. Його використовують для вибирання і переміщення об'єктів, з якими потрібно працювати. Вибрати об'єкти можна по-різному, у більшості випадків достатньо натиснути кнопкою миші на об'єкті.



Інструмент **Shape** (Форма) призначений для внесення змін в окремі елементи об'єктів. Якщо вибрати цим інструментом об'єкт, то зовнішні його контури перетворяться у набір вузлів, які можна пересувати, вирівнювати, а також змінювати їхні робочі характеристики. Крім редагування вузлів, інструмент **Shape** використовують для редагування символів тексту і растрових зображень, заокруглення кутів прямокутників, виділення дуг і секторів з вибраного кола.



Інструмент **Zoom** (Масштаб) дає змогу змінювати масштаб зображення. Перший режим роботи перетворює курсор у лупу зі знаком "+" для збільшення потрібної частини об'єкта. Вибравши режим **Pan One-Shot**, можна захопити малюнок у певному місці і швидко перемістити його у потрібному напрямі.



Інструментом **Freehand** можна малювати криві довільної форми ніби звичайним олівцем. Режим **Bezier** дає змогу малювати гладкі криві Безьє,

потрібні в якісних ілюстраціях, а інструмент **Natural Pen** – лінії різної товщини. Якщо потрібно задати розміри об'єктів, то можна скористатись інструментом **Dimension**. Для плавного переходу між об'єктами застосовують допоміжні лінії останнього режиму роботи.



Інструмент **Rectangle** призначений для малювання прямокутників будь-яких розмірів та пропорцій.



Інструмент **Ellipse** дає змогу будувати різної форми еліпси.



Інструмент **Polygon** утворює випуклі многокутники та зірки з довільною кількістю вершин. Можна також малювати спіралі з різною кількістю витків та діаграмні сітки з різною кількістю комірок.



Інструмент **Basic Shapes** дозволяє утворювати основні типи форм, а також вказувати різні типи стрілок, з'єднання схем, правильних зіркоподібних фігур та виносних надписів.



Для введення тексту в малюнок використовують інструмент **Text**. Вибираючи відповідну піктограму в потрібному місці малюнка, задають різний за оформленням фігурний текст.



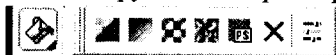
За допомогою інструменту **Interactive Blend** можна створювати і редагувати переходи між об'єктами. Якщо вибрати перші два режими, то виконається відповідно інтерактивне налаштування переходу об'єктів або інтерактивна деформація об'єктів безпосередньо у вікні документа. Якщо потрібно задати форму об'єкта для його редагування, то використовують наступний режим цього інструменту. Для побудови тривимірного зображення об'єкта можна застосувати режим **Interactive Extrude**: у ньому для задання точки збігу зображення використовують останню кнопку зі списку режимів інструменту. Інструмент **Interactive Transparency** робить об'єкти прозорими. Можна задавати рівень прозорості та неоднорідну прозорість, яка зробить створені зображення реальнішими і життєвішими.



Інструмент **Eyedropper** дає змогу вводити колір заповнення безпосередньо з об'єкта.



Інструментом **Outline** задають колір і стиль контуру вибраного об'єкта. Для контуру можна вибрати товщину лінії й одну з семи градацій чорного кольору або викликати діалогове вікно **Outline Pen** для доступу до елементів керування і параметрів задання іншого кольору контуру.



Інструмент **Fill** містить засоби зміни атрибутів заповнення вибраного об'єкта. За його допомогою можна регулювати кольори об'єкта, шаблони й інші характеристики заповнення.



Інструмент **Interactive Fill** дає змогу швидко, без допомоги діалогового вікна, змінювати тональне заповнення.

10.3.3. Друкування документа

Діалогове вікно **Печать (Print)** викликають однойменною командою з меню **Файл (File)** або комбінацією клавіш **Ctrl+P**.

У полі **Принтер (Printer)** першої вкладки **Общие (General)** (рис. 10.19) в списку **Имя (Name)** потрібно задати принтер, на якому друкуватиметься документ. Дані про вибраний принтер визначають параметрами **Состояние (Status)**, **Тип (Type)**, **Порт (Where)** і **Заметки (Comment)**. Якщо натиснути на кнопки **Свойства (Properties)**, то розкриється вікно з переліком властивостей вибраного принтера.

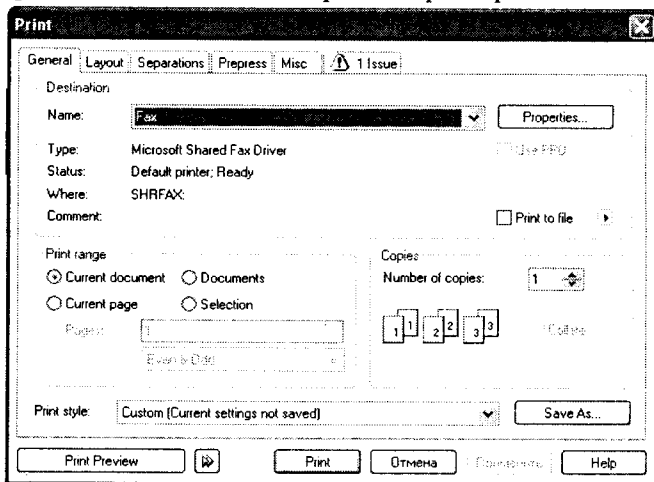


Рис. 10.19

У цьому ж полі можна використати **Печать в файл (Print to file)** і створити незалежний щодо CorelDRAW і комп'ютера файл для друкування документа. У цьому випадку стає доступною можливість **Для Mac (For Mac)**, яка дає змогу підготувати файл для друкування на платформі Macintosh.

У полі **Страницы (Print range)** задають, які сторінки або об'єкти треба друкувати. За замовчуванням є **Все (All)**. Можливість **Текущая (Current page)** дає змогу надрукувати лише поточну сторінку, **Выделение (Selection)** – виділений фрагмент документа. Якщо документ складається з кількох сторінок, то застосовують **Номера (Pages)** і зазначають у полі номери сторінок, які треба друкувати. Нижче можна додатково вибрати зі списку один із параметрів: **Нечётные страницы (Odd Pages)**, **Чётные страницы (Even Pages)**, **Чётные и нечётные страницы (Even and Odd Pages)**.

У полі **Копии (Copies)** можна задати кількість копій. Параметр **В порядок (Collate)** дає змогу друкувати сторінки копій у різному порядку.

На вкладці **Макет (Layout)** задають розташування рисунка.

Режим кольороподілу визначають параметром **Печатать цветоделение (Print separation)**, що на сторінці **Цветоделение (Separations)**.

Вкладка **Режимы печати (PrePress)** містить параметри, які треба налаштувати для друкування плівок (діапозитивів) для подальшого поліграфічного відтворення.

Для друкування документа на принтері у форматі **PostScript** потрібно налаштувати параметри на відповідній вкладці.

10.3.4. Малювання об'єктів

Розглянемо основи створення об'єктів та зміни їхньої форми. Усі об'єкти в документах CorelDRAW містяться на відповідних шарах (**layer**) і мають свої атрибути. **Атрибути** – це розміри документа, його початкові та кінцеві точки, наявність або відсутність контуру та заповнення. Атрибути можна змінювати. Відомості про них є в рядку стану (**Status Bar**), розміщеному внизу головного вікна програми. Він дає повну інформацію про об'єкт. Одним із головних елементів усіх малюнків у CorelDRAW є **криві**. Кожна крива має свій шлях, який визначає форму об'єкта. Коли перша і кінцева точки об'єкта збігаються, то кажуть, що шлях замкнений. Це означає, що в об'єкта є внутрішня область, яку можна заповнити кольором або орнаментом. Незамкнені області заповнювати не можна. Шлях можна розділити на сегменти, початок і кінець сегмента визначений вузлами. Вузли є основою векторних графічних конструкцій. Кожен вузол – це точка на площині, яка має пару координат (x, y) .

10.3.4.1. Малювання ліній

Лінії можна малювати у двох режимах: **Freehand** і **Bezier**. Для цього потрібно натиснути на чорному трикутнику внизу четвертої зверху кнопки

інструментів **Toolbox**. З'явиться допоміжна панель інструментів **Curve**, з якої вибирають потрібний режим.

☞ **Режим Кривая (Freehand)** нагадує малювання олівцем. Щоб створити за його допомогою послідовність прямих ліній, треба виконати такі дії:

- натиснути в точці, де буде початковий вузол;
- перемістити курсор миші туди, де повинен бути кінець сегмента;
- натиснути один раз, щоб помістити в цій точці кінець сегмента;
- натиснути у цьому ж місці ще раз, щоб задати нову початкову точку;
- повторити останні три кроки для збільшення кількості складових ламаної.

☞ У режимі **Кривая Безье (Bezier)** треба задавати тільки точки прив'язування, які **CorelDRAW** потім сполучає відрізками. Щоб створити послідовність відрізків прямих інструментом **Bezier**, треба виконати такі дії:

- натиснути мишею у точці, де буде початковий вузол;
- перемістити курсор миші туди, де повинен бути кінець сегмента;
- натиснути один раз, щоб розмістити у цій точці кінець сегмента;
- перемістити курсор миші;
- натиснути один раз, щоб розмістити у цій точці кінець сегмента;
- повторити останні чотири кроки.

Для того, щоб перетворити незамкнутий шлях у замкнутий, у будь-якому з цих режимів треба повторно натиснути на початковому вузлі.

Щоб намалювати криву лінію інструментом (**Кривая Freehand**), потрібно виконати такі дії:

- розмістити курсор у вигляді хрестика в початковій точці майбутньої лінії;
- натиснути і, утримуючи кнопку миші натиснутою, переміщати хрестик по сторінці так, ніби малюють олівцем;
- у разі помилки, якщо малювання не закінчене, не відпускаючи кнопки миші, натиснути й утримувати клавішу **Shift**. Повернутися назад уздовж траєкторії шойно намальованої лінії, лінія зникне;
- дійшовши до кінцевої точки кривої, відпустити кнопку миші.

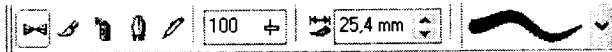
Намалювати криву лінію інструментом **Кривая Безье (Bezier)** можна двома способами. Перший спосіб:

- розмістити курсор у вигляді хрестика в початковій точці майбутньої лінії. Натиснути один раз кнопкою миші;
- перемістити курсор миші в потрібне місце;
- натиснути і перетягти хрестик. Під час перетягування утворюватиметься крива;
- натиснути на клавішу **Spacebar**, щоб закінчити малювання кривої.

Другий спосіб:

- натиснути у будь-якому місці сторінки. Перетягти курсор миші вбік від цієї точки;
- відпустити кнопку миші і перемістити курсор в інше місце;
- натиснути. Утвориться крива, що сполучає ці точки.

Для малювання ліній змінної товщини використовують інструмент **Перо (Natural Pen)**, розміщений на панелі інструментів **Curve (Кривая)**. Вибравши інструмент **Natural Pen**, одержимо таку панель властивостей:



Перші чотири кнопки дають змогу вибирати тип пера: **Fixed Width** (Фіксованої товщини), **Pressure** (З натиском), **Calligraphic** (Каліграфічне), **Preset** (Заготовка). Праворуч від кнопок розташоване текстове поле, де є значення максимальної товщини лінії або форми. Наступне текстове поле містить значення параметра **Nib Angle** (Нахил пера) і доступне лише для пера каліграфічного типу. В правій частині панелі є список заготовок пера, якими можна скористатися лише для пера **Preset** (Заготовка).

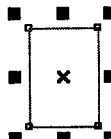
10.3.4.2. Малювання найпростіших ілюстраційних об'єктів

Найпростіші ілюстраційні об'єкти створюються інструментами **Прямоугольник (Rectangle)**, **Эллипс (Ellipse)**, **Многоугольник (Polygon)**, **Спираль (Spiral)**, **Диаграммная Сетка (Graph Paper)** або **Текст (Text)**. Усі ці об'єкти теж є кривими і мають свої характерні властивості.

Щоб намалювати прямокутник, треба виконати таке:

- натиснути на кнопці інструмента **Rectangle**. Курсор перетвориться у хрестик;
- розташувати хрестик у ділянці малювання, де буде кут прямокутника;
- перетягнути маніпулятор миші в довільному напрямі, з'являться контури об'єкта. Щоб намалювати квадрат, під час перетягування треба утримувати натиснутою клавішу **Ctrl**;
- відпустити кнопку миші.

Для того, щоб побудувати об'єкт із центральної точки назовні, потрібно натиснути й утримувати під час перетягування миші клавішу **Shift**. Намальований прямокутник матиме такий вигляд:



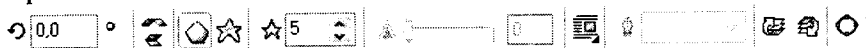
На екрані видно вузли об'єкта (чотири квадратики без заповнення) та маркери (чорні квадратики). Це означає, що об'єкт активний. Ухопивши мишею за один із чотирьох квадратиків, можна скруглювати кути прямокутника. Округлювати кожен кут окремо можна в панелі **Свойства (Properties)** на вкладці **Прямоугольник (Rectangle)**, відключивши опцію **Округление всех углов (Round All Corners)** і встановивши відповідні значення у чотирьох полях.

Еліпс малюють за аналогічною схемою. Його вузли не можна змінювати за допомогою інструмента **Shape**, однак перетягуючи єдиний вузол еліпса, можна створювати сектори і дуги. Для малювання кола утримують натиснутою клавішу **Ctrl** під час перетягування курсора. Щоб намалювати дугу або сектор, треба виконати такі дії:

- намалювати еліпс або коло;
- натиснути на піктограмі інструменту **Форма (Shape)**, на контурі еліпса з'явиться єдиний вузол;
- щоб намалювати сектор, треба натиснути на вузлі і перетягнути його в будь-якому напрямі, не виходячи за периметр еліпса. Коли сектор набуде потрібних розмірів, відпустити кнопку миші;
- щоб намалювати дугу, потрібно натиснути на вузлі і перетягнути його в будь-якому напрямі зовні периметра еліпса. Коли дуга набуде потрібних розмірів, відпустити кнопку миші.

CorelDRAW дає змогу задавати значення кутів початку та кінця дуги або сектора, а також координати точки вершини і напряму на панелі властивостей, коли об'єкт-еліпс активний.

Для малювання багатокутника використовують інструмент **Многоугольник (Polygon)**. Як і у випадку з прямокутником, щоб правильно намалювати багатокутник з центральної точки назовні, застосовують клавіші **Ctrl+Shift**. Якщо об'єкт-многокутник активний, то на панелі властивостей з'являються додаткові кнопки. За їхньою допомогою можна, наприклад, перетворити багатокутник у зірку – для цього треба натиснути на кнопці **Многоугольник/Звезда (Polygon/Star)**. Користуючись кнопкою **Number of points on Polygon**, можна задати кількість вершин, повзунком **Sharpness of Polygon** – змінювати кут у вершинах.



Перед малюванням спіралей треба налаштувати параметри на панелі властивостей. Тут можна задати, зокрема, кількість витків спіралі.



Для ознайомлення з її можливостями пропонуємо наступне:

1. Намалюйте криву, що складається з прямолінійних і криволінійних сегментів.
2. Виділіть два вузли одночасно. Натисніть на одному з вузлів, натисніть на клавішу **Shift** і, утримуючи її, натисніть на іншому. Можна також обвести рамкою один вузол, натиснути на клавішу **Shift** і обвести рамкою другий вузол.
3. Утворіть нові вузли. Для цього натисніть на сегменті – з'явиться маленька крапка. Натисніть на кнопці зі знаком "+", і з'явиться новий вузол.
4. Вилучати вузли та сегменти можна за допомогою кнопки зі знаком "-".
5. Виділіть два або більше вузлів. За допомогою кнопок вузли можна обертати і вирівнювати. Для вирівнювання виділених вузлів натисніть на кнопці **Вирівняйте узли (Align Nodes)**. З'явиться діалогове вікно **Вирівнювання узлів (Node Align)** (рис. 10.20), у якому можна задати потрібний варіант вирівнювання.

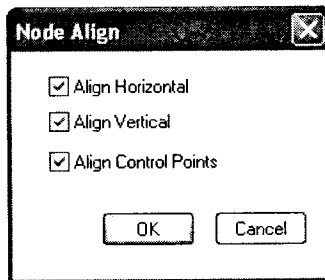


Рис.10.20

6. Виділіть початковий і кінцевий вузли. Сполучіть їх за допомогою кнопки **Join Two Nodes**. За допомогою кнопки **Auto-Close** можна провести між початковим і кінцевим вузлами з'єднувальний сегмент.
7. З'єднайте дві криві прямою лінією за допомогою кнопки **Extend Curve to Close**. Для цього попередньо треба ці дві криві скомбінувати.
8. Розділіть будь-який вузол на два за допомогою кнопки **Break Curve**.
9. За допомогою кнопки **Stretch** змініть розміри виділених сегментів кривої.
10. У режимі **Elastic Mode** можна змінювати поведінку двох і більше виділених вузлів у разі їхнього переміщення. Проміжок, на який переміщається вузол, залежить від того, який з виділених вузлів перетягають.
11. Нехай крива складається з кількох гілок. Відокремте лише одну гілку від кривої. Для цього досить виділити один з вузлів гілки кривої і натиснути на кнопці **Extract Subpath**.

12. Перетворіть прямолінійні сегменти в криволінійні за допомогою кнопки **To Curve**, а криволінійні в прямолінійні – кнопкою **To Line**.

Перетворювати гладкі або симетричні вузли в гострі можна за допомогою кнопки **Cusp**. Кнопку **Smooth** використовують для отримання гладких вузлів. Щоб отримати симетричний вузол, застосовують кнопку **Symmet**.

За допомогою кнопки **Auto-Reduce** виконують автоматичну мінімізацію кількості вузлів.

Щоб змінити обриси кривої за допомогою інструменту **Shape**, слід:

- натиснути на кривій, обрис якої треба змінити. Вузли стануть видимі;
- натиснути на вузлі, який треба перемістити, – він перетвориться на чорний квадратик (виділіть одночасно декілька вузлів);
- натискати на вибраних вузлах і перетягувати їх так, щоб крива набула бажаного обрису.

Обрис кривої можна змінювати за допомогою контрольних точок вузлів:

- натиснути на кривій, обрис якої треба змінити;
- натиснути на вузлі, який треба змінити. Контрольні точки вузла стануть видимі;
- натискати на контрольних точках і перетягувати їх так, щоб крива набула бажаного обрису.

Вигляд кривої можна змінювати, безпосередньо перетягуючи сегменти кривої. Для цього потрібно:

- натиснути на кривій, обрис якої треба змінити;
- натиснути на сегменті, форму якого треба змінити, і перетягти його так, щоб крива набула бажаного обрису.

Для засвоєння описаних можливостей намалюйте просту фігуру, у якій змінюйте форми сегментів.

1. Відкрийте новий рисунок. Переконайтеся, що можливість **Snap to Guidelines** у пункті **Layout** головного меню вимкнено.
2. Виберіть інструмент **Freehand** або **Bezier** і намалюйте замкнену ламану лінію, яка складається з чотирьох прямолінійних сегментів. Для полегшення малювання користуйтеся вертикальними і горизонтальними напрямними. Для цього Натисніть на вертикальній або горизонтальній лінійці і перетягніть напрямні на сторінку малювання. Переконайтеся, що ламана є замкненою.
3. Виберіть інструмент **Shape**.
4. Натисніть на кривій і виділіть перший сегмент.
5. На панелі властивостей натисніть на кнопку **В кривую (To Curve)**. **CorelDRAW** перетворить прямолінійний сегмент у криволінійний. Тепер сегмент 1 можна скривлювати.
6. Змініть обрис сегмента так, щоб він нагадував обрис серця.

7. Повторіть ті ж дії з сегментом 2.
8. Виділіть верхній лівий вузол. Рядок стану відображає, що вибраний вузол є гострим (**Cusp**).
9. На панелі властивостей натисніть на кнопку **Сглаживание (Smooth)**. **CorelDRAW** перетворить гострий вузол у гладкий.
10. Перетворіть правий верхній вузол у гладкий.
11. Перетворіть прямолінійні сегменти 3 і 4 у криволінійні. Змінійте їхні обриси, щоб вони нагадували серце.
12. За бажанням замалюйте серце будь-яким кольором.
Приклад ілюструє рис. 10.21.

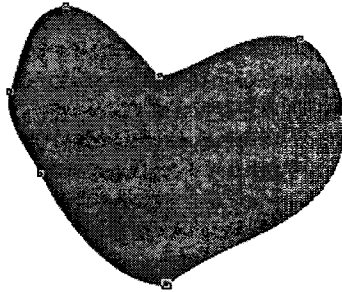


Рис. 10.21

13. Виберіть інструмент **Knife (Лезо)** і розмістіть його над обрисом серця. Курсор змінить орієнтацію.
14. Перетягайте курсор поперек рисунка ламаною траєкторією доти, доки не досягнете протилежного краю. Коли досягнете краю об'єкта, курсор знову змінить орієнтацію.
15. Відпустіть ліву кнопку миші, і серце буде розрізане.
16. Виділіть одну половину і пересуньте її вбік (рис. 10.22).

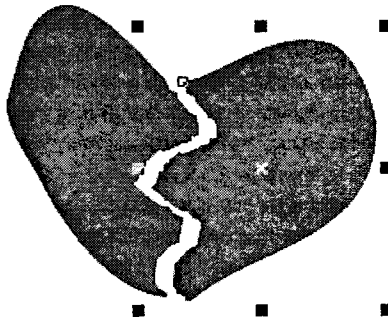


Рис. 10.22

Якщо вибрати інструмент **Eraser** (Гумка) з панелі інструментів **Shape**, то на панелі властивостей з'являться два елементи керування:



Перший параметр керує товщиною гумки, а другий – це кнопка **Auto-Reduce** (Зменшити кількість вузлів). Інколи після витирання кількість вузлів потрібно зменшити, оскільки замість витертих сегментів кривої автоматично домальовуються нові, які сполучають кінці розривів і можуть містити зайві вузли.

Навіть найпростіші об'єкти мають свої спеціальні властивості, однак їх легко позбутися. Для цього досить натиснути на кнопку **Преобразовать в кривую (Convert to Curves)** на панелі властивостей чи натиснути комбінацію клавіш **Ctrl+Q**. Цю команду можна відшукати також у пункті **Arrange** головного меню. Якщо застосувати її до будь-якого з об'єктів (**Rectangle, Ellipse, Polygon Text**), він перетвориться в криву. Крива розділиться на підшляхи та сегменти, до яких можна додавати і віднімати вузли, змінювати їх обриси.

10.3.5. Робота з об'єктами

10.3.5.1. Виділення об'єктів

Перш ніж працювати з об'єктом, його потрібно виділити. Тільки після цього **CorelDRAW** призначить об'єктові атрибути. Для того, щоб виділити окремий об'єкт, треба натиснути на піктограмі **Выбор (Pick)**, а далі – на потрібному об'єкті. Виділений об'єкт буде оточений маркерами (чорними квадратами), а його центр позначений чорним хрестиком.

Якщо на екрані зображено кілька об'єктів, розмішених дуже близько один від одного, що утруднює виділення окремого інструментом **Pick**, то користуються клавішею **Tab**. Виокремивши об'єкт інструментом **Pick**, а далі натиснувши на клавішу **Tab**, можна виділити наявні об'єкти в порядку розташування. Якщо натиснути на клавіші **Shift+Tab**, то можна змінити порядок виділення на протилежний.

Виділити одночасно декілька об'єктів можна по-різному. Один із способів полягає в покриванні їх шатром. Виконайте такі дії:

- намалуйте декілька об'єктів;
- натисніть на піктограмі **Pick**;
- натисніть у лівому верхньому куті уявної прямокутної ділянки, де розміщені усі потрібні об'єкти. Перетягніть курсор у правий нижній кут цієї ділянки. Під час перетягування навколо ділянки з'являтиметься пунктирна рамка (шатро). Стежте за тим, щоб об'єкти, які потрібно виділити, були повністю покриті шатром;

- відпустити кнопку миші. Усі об'єкти, які є всередині шатра, тепер виділені.

Якщо об'єкти розміщені на відстані один від одного, то їх можна виділяти за допомогою клавіші **Shift**:

- вибрати інструментом **Pick** потрібний об'єкт;
- натиснути й утримувати клавішу **Shift** під час натискання на інших об'єктах, які треба виділити.

Натисканням з утриманням клавіші **Shift** на вже виділеному об'єкті виділення відмінюють.

10.3.5.2. Переміщення об'єктів

Під час створення рисунків інколи потрібно переміщувати або повертати на деякий кут об'єкт чи групу об'єктів. Це можна робити різними способами. Найзручніший – перетягування мишею. Для цього треба натиснути в будь-якому місці об'єкта лівою кнопкою миші і, не відпускаючи кнопки, перетягти його на нове місце.

Для прецизійного переміщення об'єкта на точну відстань у потрібному напрямі застосовують інший спосіб: задають нові координати центра об'єкта (x, y) на панелі властивостей.

Ще інший спосіб полягає у використанні вкладки **Положение (Position)** докерного вікна **Преобразование (Transformation)**, яке можна активізувати з меню **Окно/ Пристыковываемые окна/Преобразование (Window/Dockers/Transformation)**. У цьому випадку треба виконати такі дії:

- виділити об'єкт;
- відкрити додаткове меню **Position** (його вигляд зображено на рис. 10.23,а);
- для переміщення об'єкта на сторінці ввести потрібні числа у вікнах **H** (горизонталь) і **V** (вертикаль);
- для переміщення об'єкта щодо його місця розташування відзначити позначкою **3** кнопку параметра **Относительное положение (Relative Position)** і ввести потрібні числа. Зазначимо, що за замовчуванням у задане місце переміщується центральна точка об'єкта. Щоб відмінити цей механізм, треба натиснути на спрямовану вниз стрілку, розташовану поряд із параметром **Relative Position**, і вибрати точку, щодо якої переміщуватиметься об'єкт;
- для переміщення об'єкта натиснути на кнопці **Применить (Apply)**, для переміщення «клонів» об'єкта натиснути на кнопці **Применить к дубликату (Apply To Duplicate)**.

Об'єкт можна пересувати на точно задану відстань. У цьому випадку користуються клавішами керування курсором. Спочатку треба виділити об'єкт, потім натиснути на клавішу, яка вказує на потрібний напрям руху. Для задання розміру переміщення об'єкта відкривають меню **Сервис/**

Параметри (Tools/Options) і в лівій частині вікна вибирають сторінку **Edit**. Тут, у діалоговому вікні **Сдвиг (Nudge)** вводять нове значення стрибка та суперстрибка (з натисненою клавішею **Shift**).

10.3.5.3. Повертання та нахилення об'єктів

Для того, щоб повернути або нахилити об'єкт, потрібно його виділити і повторно натиснути на ньому. Навколо об'єкта з'являться двонаправлені стрілки (маркери поворотів), центр повороту зображено кільцем з крапкою.

За допомогою цих маркерів об'єкт можна повертати та нахилити. Кут повороту або нахилу відобразяться в рядку стану. Якщо утримувати під час повертання клавішу **Ctrl**, то обертання виконуватиметься з кроком 15° (за замовчуванням).

Виділений об'єкт можна повертати, задавши кут повороту на панелі властивостей та використовуючи вкладку **Поворот (Rotation)** (рис. 10.23,б) вікна **Преобразование (Transformation)**. Це меню подібне на меню **Position**, однак тут є вікно для задання кута повороту.

10.3.5.4. Зміни розмірів та масштабу об'єктів

Масштабування об'єкта змінює його розміри, однак не впливає на форму. Витягування збільшує або зменшує тільки один вимір (його довжину або ширину). Для цього потрібно виконати такі дії за допомогою миші:

- виділити об'єкт, розміри якого треба змінити.
- розмістити курсор миші на одному з чотирьох кутових маркерів. Курсор перетвориться у двонаправлену стрілку;
- перетягати курсор від об'єкта, щоб його збільшити, або до центра об'єкта, щоб зменшити;
- відпустити кнопку миші.

Об'єкт змінить свої розміри, однак його форма буде попередньою. Щоб видовжити об'єкт за допомогою миші, потрібно виконати такі дії:

- виділити об'єкт, який треба змінити. Навколо нього з'являться маркери;
- розмістити курсор миші на одному з чотирьох бокових маркерів. Курсор перетвориться у двонаправлену стрілку;
- перетягати курсор від об'єкта, щоб його видовжити, або до центра об'єкта, щоб його стиснути;
- відпустити кнопку миші. Об'єкт змінить свою форму тільки в напрямку перетягування.

Змінювати розміри об'єктів можна за допомогою панелі властивостей, задаючи нові розміри по горизонталі і вертикалі, а також використовуючи вкладку **Размер (Size)** (рис. 10.23,г) вікна **Преобразование (Transformation)**.

Для масштабування та дзеркального відображення об'єктів можна скористатися як панеллю властивостей, так і використовуючи вкладку **Масштаб и отражение (Scale and Mirror)** (рис. 10.23,в) вікна **Преобразование (Transformation)**.

Рис. 10.23,д ілюструє вкладку **Наклон (Skew)** вікна **Преобразование (Transformation)**, яка застосовується для нахилу об'єктів.

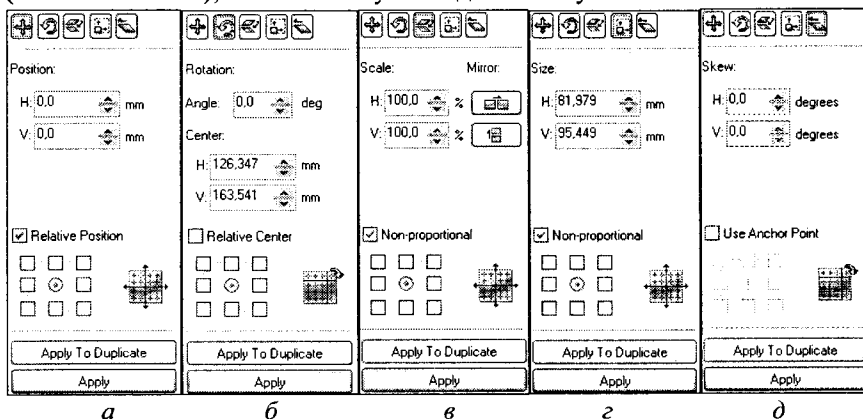


Рис. 10.23

10.3.5.5. Копіювання об'єктів

Є два головні види копій – дублювання та імітація. У випадку дублювання утворюється новий об'єкт – абсолютна копія оригіналу, яка не змінюється після подальших змін оригіналу. У разі імітації також утворюється копія оригіналу, проте подальші зміни оригіналу призводять до зміни копії (клону).

Для копіювання виділених об'єктів треба виконати такі дії:

- відкрити меню **Редактирование (Edit)** і вибрати **Дублировать (Duplicate) (Ctrl+D)**. На заданій відстані від оригіналу буде створена копія;
- або натиснути на клавішу + на допоміжній цифровій клавіатурі. Дублікат розміститься над оригіналом;
- або використати команди **Копировать (Copy) (Ctrl+C)** і **Вставить (Paste) (Ctrl+V)** буфера обміну Windows.

Дублювати об'єкт можна і під час його переміщення чи обертання.

Для цього потрібно:

- натиснути лівою кнопкою миші на об'єкті й, утримуючи її, перетягувати об'єкт на потрібне місце;
- не відпускаючи лівої кнопки миші, натиснути на праву. При утриманні лівої кнопки миші біля курсору з'явиться «←»;
- відпустити ліву кнопку, оригінал залишиться на місці, а копія переміститься на нове місце.

Імітація – це унікальний метод дублювання: тут зміна оригіналу приводить до зміни всіх імітованих об'єктів. Імітувати можна тільки оригінал. Щоб виконати цю процедуру, треба:

- виділити об'єкт, який потрібно імітувати;
- відкрити меню **Edit** і вибрати команду **Clone**. На тому ж місці утвориться імітація. Клон можна перемістити в інше місце.

Щоб одночасно змінити всі об'єкти, треба вибрати оригінал і зробити потрібні корективи. Однак якщо ви змінили будь-які атрибути імітації, то з тієї ж миті вона втрачає зв'язок з оригіналом: подальші зміни оригіналу на імітацію не вплинуть.

10.3.5.6. Об'єднання та групування об'єктів

Для об'єднання декількох об'єктів в один треба скористатися однією з двох команд **Групувати (Group)**, **Соединить (Combine)** з меню **Монтаж (Arrange)**.

Команда **Group** зв'яже всі об'єкти і водночас залишає їх незалежними. Вона дає змогу пересувати, розтягати, стискати і зафарбовувати їх як одне ціле. Згруповані об'єкти завжди можна розгрупувати. Щоб застосувати цю команду, потрібно виділити об'єкти та відкрити меню **Монтаж (Arrange)** і вибрати у ньому команду **Групувати (Group) (Ctrl+G)**.

Проілюструємо зручність цієї команди вправою.

1. Намалуйте невеликий еліпс і зафарбуйте його.
2. Натисніть двічі на об'єкті. Навколо нього з'являться двонаправлені стрілки і посередині кільце з крапкою, яка визначає центр обертання.
3. За допомогою курсора миші перетягніть це кільце з центра до периметра еліпса.
4. У вкладці **Поворот (Rotation)** вікна **Преобразование (Transformation)** задайте кут повороту 30° у полі **Угол (Angle)**. Натисніть на кнопку **Применить к дубликату (Apply To Duplicate)** одинадцять разів. Отримаєте квітку з дванадцятьма пелюстками.
5. Зробіть копію цього малюнка і помістіть її поряд.
6. Виділіть усі дванадцять пелюстків і виконайте команду **Group**. Рядок стану відобразить, що це група об'єктів (рис. 10.24).

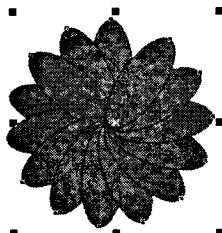


Рис. 10.24

Щоб розгрупувати об'єкти, треба застосувати команду **Разгруппировать (Ungroup) (Ctrl+U)** з меню **Монтаж (Arrange)**. Якщо група складається з кількох підгруп, то цю операцію слід повторити.

Якщо потрібно внести корективи в окремий об'єкт виділеної групи, то його виділяють як дочірній, натискаючи на ньому мишею при натисненій клавіші **Ctrl**.

Якщо ж об'єкти мають однакові атрибути і немає потреби працювати з кожним окремо, то їх доцільно об'єднати в один. Для цього треба застосувати команду **Соединить (Combine) (Ctrl+L)** з меню **Монтаж (Arrange)**. Тепер **CorelDRAW** сприймає ці об'єкти як одну криву з вузлами. Після об'єднання об'єктів, які не обов'язково повинні дотикатися, **CorelDRAW** перетворює їх у криві. Новому об'єкту буде присвоєно атрибути останнього виділеного об'єкта. Для того, щоб роз'єднати об'єкти, треба виконати команду **Разделить на части (Break Apart) (Ctrl+K)** з меню **Монтаж (Arrange)**.

10.3.6. Заповнення

Усі об'єкти в **CorelDRAW** мають контури і внутрішні ділянки, які можна зафарбовувати, тінювати або заповнювати фактурою. Єдина необхідна умова для заповнення – об'єкт повинен бути замкненим. Внутрішню частину замкненого векторного графічного об'єкта можна розглядати як окремий об'єкт. Якщо він має заповнення, то у разі зміни його форми заповнення кожного разу зафарбовує його внутрішню область.

Програма **CorelDRAW** передбачає сім типів заповнення: однорідне, градієнтне, двокольорове орнаментне, кольорове орнаментне, растровим орнаментом, текстурне і **PostScript**.

10.3.6.1. Однорідне заповнення

Головне призначення однорідного заповнення **Uniform Fill** – зафарбувати об'єкт однорідним кольором, відтінком або градацією сірого кольору.

За замовчуванням для заповнення всіх графічних об'єктів прийнято режим **No Fill**, а для текстових об'єктів – чорний колір. Щоб призначити колір, який сприйматиметься за замовчуванням, треба виконати такі дії:

- переконатися, що жоден об'єкт не виділений;
- натиснути на кнопку **Інструмент заливки (Fill Tool)** з панелі інструментів;
- натиснути на кнопку **Окно діалога заливки (Fill Color Dialog)**, відкриється вікно з заголовком **Однородная заливка (Uniform Fill)** для налаштування значень, які приймаються за замовчуванням. Задати прапорець **Графика (Grafic)**, як показано на рис. 10.25,а.
- натиснути на **ОК**. З'явиться діалогове вікно **Uniform Fill**. Натиснути на кнопці **More**, щоб розгорнути вікно;

- натиснути на кнопку **Color Viewers** і вибрати модель **СМЯК** зі списку **Model**;
- увести число 100 у лічильник параметра **К** і число 0 у лічильники параметрів **С**, **М** і **Y**. У текстовому полі **Name** (Ім'я) з'явиться ім'я кольору – **Чёрный (Black)** (рис. 10.25,б);
- натиснути на **ОК**. Тепер за замовчуванням колір заповнення буде чорний.

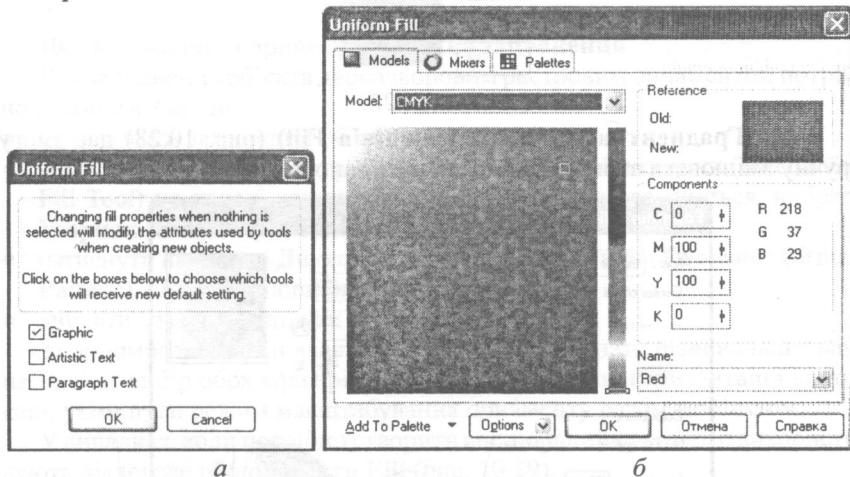


Рис. 10.25

10.3.6.2. Градієнтне заповнення

Інколи потрібно заповнювати об'єкти не одним кольором або відтінком, а плавним переходом кольорів від одного до іншого. У цьому випадку використовують градієнтне заповнення. Для ознайомлення з градієнтним заповненням пропонуємо виконати таку вправу.

- намалюйте великий прямокутник і виділіть його;
- виберіть інструмент **Інтерактивна заливка (Interactive Fill)**;
- натисніть на лівому боці прямокутника і перетягніть курсор миші на його правий бік. Утвориться напрямна градієнтного заповнення. Для отримання горизонтальної напрямної треба скористатися клавішею **Ctrl**. За замовчуванням уздовж напрямної утвориться перехід від чорного до білого кольору (рис. 10.26):

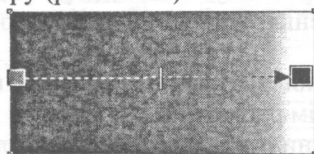


Рис. 10.26

На рис. 10.27 зображено фрагмент панелі властивостей у випадку градієнтного заповнення; чотири квадрати стосуються чотирьох типів такого заповнення: лінійного, радіального, конічного і квадратного.

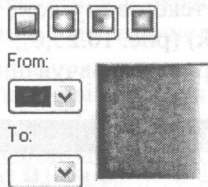


Рис. 10.27

Вікно **Градиентная заливка (Fountain Fill)** (рис. 10.28) дає змогу вручну змінювати атрибути градієнтного заповнення:

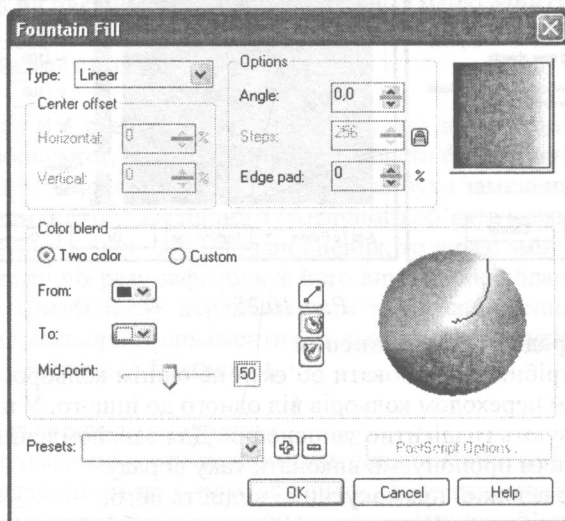


Рис. 10.28

- відкривши список **Type**, можна вибрати один з чотирьох типів градієнтного заповнення;
- увівши числа в лічильники **Center Offset**, можна змінити вертикальне і горизонтальне розташування центра радіального, конічного і квадратного заповнення (при лінійному заповненні лічильники не активні);
- уведення числа в лічильник **Angle** впливає на кутову орієнтацію всіх типів заповнення, крім радіального;
- у лічильнику проміжних градацій переходу кольорів **Steps** (Градацій) за замовчуванням буде число 256;

- уведення числа в лічильник **Edge pad** визначає довжину відрізка, на якому початковий і кінцевий колір градієнтного заповнення буде незмінним;
- у правому верхньому куті діалогового вікна є поле перегляду, у якому відображається напрям орієнтації лінії градієнта;
- розділ **Переход цвета (Color blend)** має два режими – **Два цвета (Two color)** і **Нестандартный (Custom)**. За замовчуванням діє **Two color**.

10.3.6.3. Двокольорове орнаментне заповнення

Для заповнення об'єкта двокольоровим растровим орнаментом потрібно виконати такі дії:

- намалювати замкнений об'єкт і виділити його;
- вибрати інструмент **Инструмент интерактивной заливки (Interactive Fill Tool)** на панелі властивостей. Зі списку, що розкриється, вибрати **Узорная заливка (Pattern Fill)**;
- натиснути на кнопки **Двухцветная узорная заливка (Two-color Bitmap Pattern)**, потім на кнопки вибору візерунка;
- вибрати зразок і виділити його.

За допомогою панелі властивостей можна виконати різні налаштування, такі як вибір обох кольорів, розмір орнаменту по горизонталі і вертикалі, увімкнути режим масштабування орнаменту разом з об'єктом.

У випадках, коли потрібно створити спеціальний орнамент, використовують діалогове вікно **Pattern Fill** (рис. 10.29).

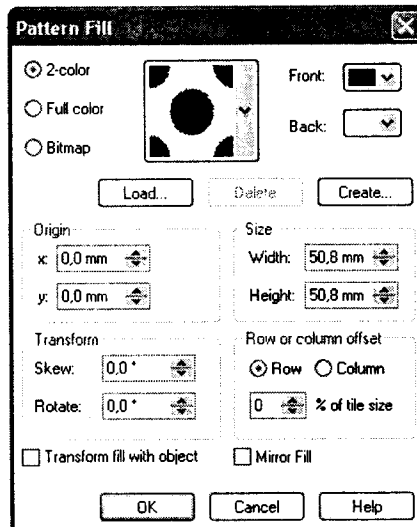


Рис. 10.29

Зверніть увагу на перемикач **Преобразовывать с объектом (Transform Fill With Object)**. Якщо натиснути на цей кнопці і перевести її в активний стан, то у разі зміни розмірів об'єкта за допомогою кутового маркера орнамент буде розтягуватися або стискатися пропорційно до цих змін.

10.3.6.4. Кольорове та растрове орнаментне заповнення

У кольоровому орнаменті можна застосувати будь-який колір, оскільки в кольорових зображеннях використано повний спектр кольорів СМЯК. У CorelDRAW є два типи кольорових орнаментів. Терміном **Full Color Pattern** позначено векторні орнаменти, терміном **Bit Pattern** – кольорові растрові.

Багатоколірне заповнення **Full Color** об'єктів виконують за допомогою діалогового вікна **Pattern Fill**, де вибирають другий рядок **Full Color**. Це заповнення подібне до двоколірного, однак тепер у діалоговому вікні і на панелі властивостей немає кнопок **Основной цвет (Front Color)** і **Фоновый цвет (Back Color)**.

Растровим (**Bitmap**) орнаментом об'єкти заповнюють так само, як і багатоколірним.

10.3.6.5. Текстульне заповнення

Текстура – це растрове зображення, яке накладається на початкове, відіграючи роль заповнення. Кожну текстуру можна модифікувати багато разів. Четвертою кнопкою ліворуч **Узорная заливка (Texture Fill)** (меню інструмента **Fill**) можна відкрити однойменне діалогове вікно (рис. 10.30):

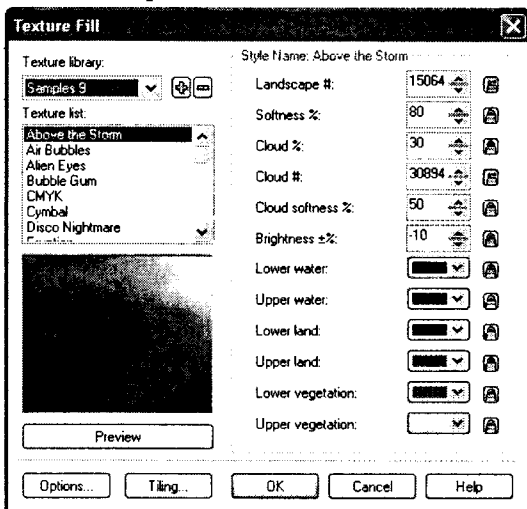


Рис. 10.30

У лівому верхньому куті діалогового вікна розміщено список **Texture Library**. Бібліотека **Styles** містить оригінали зразків різних текстур, а бібліотека **Samples** – варіанти цих оригіналів. Безпосередньо під списком бібліотек текстур є вікно списку назв різних текстур зі складу вибраної бібліотеки. Якщо натиснути мишею на назві будь-якої текстури, у вікні попереднього перегляду з'явиться її зображення. Діалогове вікно містить багато елементів керування параметрами текстур, за їхньою допомогою можна змінювати властивості вибраної текстури.

10.3.6.6. Орнаментне заповнення PostScript

Об'єкт можна також заповнити орнаментною текстурою **PostScript**. Для заповнення треба скористатися діалоговим вікном (рис. 10.31), яке відкривають п'ятою кнопкою ліворуч **Окно заливки PostScript (Post Script Fill Dialog)** меню інструменту **Fill**.

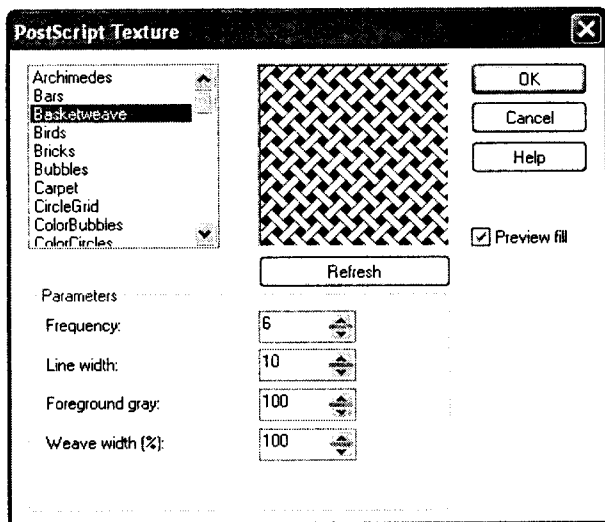


Рис. 10.31

У лівому верхньому куті вікна є список усіх заповнень. Щоб побачити, як виглядає вибране заповнення, треба увімкнути кнопку **Preview Fill** у діалоговому вікні. За допомогою параметрів, які є в діалоговому вікні, можна змінити вигляд заповнення. Щоб побачити ці зміни, треба щоразу користуватися кнопкою **Refresh**.

Зазначимо, що об'єкт заповнюється сполученнями букв "PS", а не самим заповненням. Справжнє заповнення об'єкта можна побачити лише у надрукованому вигляді.

10.3.7. Контури об'єктів

Виділеному об'єктові можна призначити контур. Перша кнопка меню інструменту **Інструмент контура (Outline Tool)** викликає одноім'яне діалогове вікно (рис. 10.32). Друга кнопка служить для вибору кольору контуру, третя видаляє контур, четверта-дев'ята здійснюють швидке присвоєння суцільного контуру товщиною відповідно **Волосяна лінія (Hairline)**, 0,5, 2, 8, 16, 24 пункти.

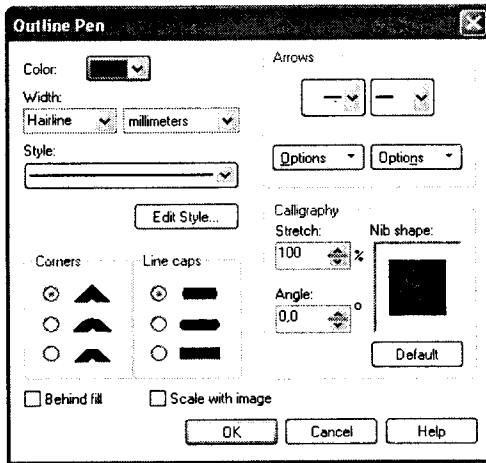


Рис. 10.32

Колір контуру зі списку можна задати у першому полі **Цвет (Color)** діалогового вікна, у полі **Толщина (Width)** задається товщина лінії контуру, у полі **Стрелки (Arrows)** – форму стрілок, **Угли (Corners)** – кутів; у полі **Стиль (Style)** – стиль (штрихова, штрихпунктирна і т.д. – всього там понад 30 варіантів). Кнопка **Редактировать (Edit)** дозволяє редагувати стиль (рис 10.33):

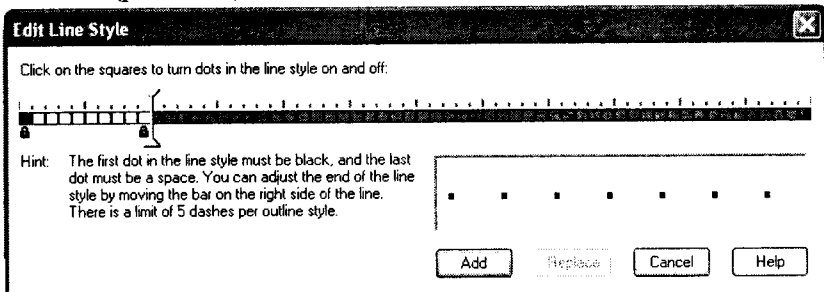


Рис. 10.33

10.3.8. Робота з текстом

У CorelDRAW розрізняють два типи текстових об'єктів: **Фигурный текст (Artistic Text)**, за допомогою якого створюють короткі написи, заголовки, до яких можна застосувати різноманітні ефекти, та **Простой текст (Paragraph Text)**, за допомогою якого працюють із великими текстовими блоками, подібно до текстових процесорів.

Для введення тексту натискають на піктограмі **Текст (Text)** панелі інструментів чи натискають клавішу **F8**. Який спосіб введення тексту буде застосовано, залежить від подальших дій.

Якщо натиснути мишею де-небудь на робочому полі і розпочати набір тексту, це буде першим варіантом оформлення тексту. Якщо ж окреслити мишею прямокутник, то він визначить розміри блоку тексту. Текст, оформлений у цьому блоці, вже буде простим (**Paragraph text**). Типи текстових об'єктів зображено на рис. 10.34.

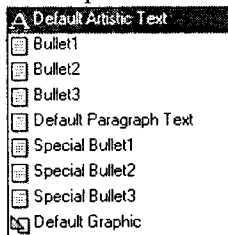


Рис. 10.34

Панель керування параметрами тексту показано на рис. 10.35.



Рис. 10.35

Перша кнопка та список необхідні для встановлення параметрів вводу тексту за замовчуванням (кнопка активна лише тоді, коли не виділено жодного текстового об'єкту), список служить для вибору, редагування та присвоєння стилю. Коли виділений текстовий об'єкт немає, у списку висвітлюється **По умовчанняю фигурный текст (Default Artistic Text)** та решта полів активні. У них можна встановити параметри тексту, що вводиться, причому після кожної зміни висвітлюється запит, для якого саме тексту застосовувати параметри (рис. 10.36).

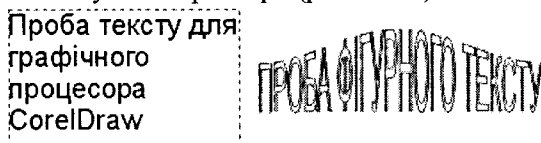
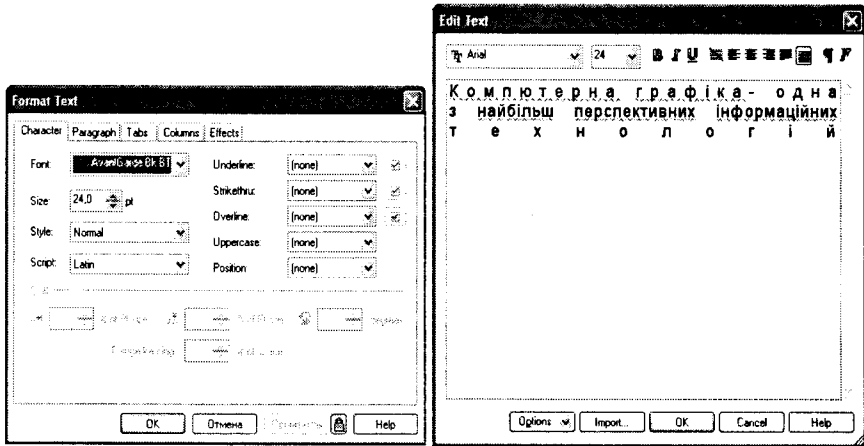


Рис. 10.36

У наступних полях керуючої панелі задаються гарнітура, розмір, стиль накреслення та вирівнювання, відступ, наявність маркованого списку, буквиці та недрукованих символів. Кнопка викликає діалогове вікно **Форматирование текста (Format Text)** (рис. 10.37,а), у якому можна провести більш детальне налаштування параметрів тексту, а кнопка – діалогове вікно **Изменить текст (Edit Text)** (рис. 10.37,б), оформлене у вигляді текстового міні-редактора. У ньому продубльована керуюча панель, є можливість імпорту тексту та декілька додаткових опцій.



а

б

Рис.10.37

Запитання для самоперевірки

1. Що вивчас комп'ютерна графіка?
2. Назвіть основні пристрої вводу-виводу графічних зображень.
3. Які використовуються основні типи графічних зображень?
4. Яким інструментом Adobe Photoshop виділяють області на основі подібності кольорів?
5. Основні види тексту в Adobe Photoshop.
6. Охарактеризуйте панель інструментів CorelDRAW.
7. Принцип роботи інструмента редагування вузлів зображення. Як додати нові вузли до існуючих вершин зображення?
8. Друкування документів у CorelDRAW.
9. Типи текстових об'єктів у CorelDRAW.

11. ОПРАЦЮВАННЯ ТЕКСТОВОЇ ІНФОРМАЦІЇ: РОЗПІЗНАВАННЯ, РЕДАГУВАННЯ, ПЕРЕКЛАД

☰ План викладу матеріалу:

1. Система оптичного розпізнавання тексту FineReader.
2. Системи перевірки правопису та перекладу ProLing Office, Pragma.

↔ Ключові терміни розділу

✓ <i>ABBY FineReader</i>	✓ <i>OCR</i>
✓ <i>Сканування</i>	✓ <i>Перевірка правопису</i>
✓ <i>Сегментування</i>	✓ <i>Машиинний переклад</i>
✓ <i>Розпізнавання</i>	✓ <i>ПУТА</i>
✓ <i>Пакет</i>	✓ <i>Pragma</i>

11.1. Система оптичного розпізнавання тексту FineReader

Серед прикладних офісних застосувань важливе місце посідають програми автоматичного опрацювання текстової інформації. Пояснити це доволі просто, оскільки навіть в наш час стрімкого розвитку інформаційних технологій ще великий обсяг інформації ми отримуємо на паперових носіях. Для подальшого опрацювання необхідно конвертувати цю інформацію в електронний вигляд, для подальшого обміну нею – мати інструментальні засоби перекладу, редагування і т.д. Слід ще наголосити на такому аспекті: в процесі розпізнавання мегабайти графічної інформації перетворюються на кілобайти текстової – очевидна економія вільного місця на жорсткому диску комп'ютера.

Для конвертації текстової інформації в електронний вигляд необхідні сканер та програма оптичного розпізнавання тексту (OCR – Optical Character Recognition). Користувачеві сьогодні пропонують на вибір сотні добротних сканерів, а от серед програмного забезпечення OCR лідер один – це **FineReader** розробки фірми **ABBY**.

11.1.1. Робоче вікно ABBY FineReader

Розглянемо версію програми **ABBY FineReader 8.0 Professional**. Особливостями цієї версії є удосконалення розпізнавання факсимільних

документів та документів, знятих на цифрову камеру, покращена обробка макету документів, захист файлів **PDF**. Додано такі нові можливості, як заплановані операції, розпізнавання скріншотів та менеджер автоматизації для обробки серії завдань. До беззаперечних переваг програми завжди належали пряма робота зі сканерами протоколу **TWAIN32**, розпізнавання таблиць та багатомовних документів, вбудована підсистема перевірки орфографії, передача даних в інші додатки, пересилання електронною поштою. Загальна кількість мов розпізнавання у восьмій версії складає 179, а перевірка орфографії за словниками можлива для 36 мов.

На рис. 11.1 зображене головне вікно програми **FineReader**. Для зручності ознайомлення з функціональними можливостями на рисунку представлені розгортки усіх керуючих кнопок: **Scan&Read**, **Відкрити**, **Читати все**, **Перевірити**, **Зберегти**.

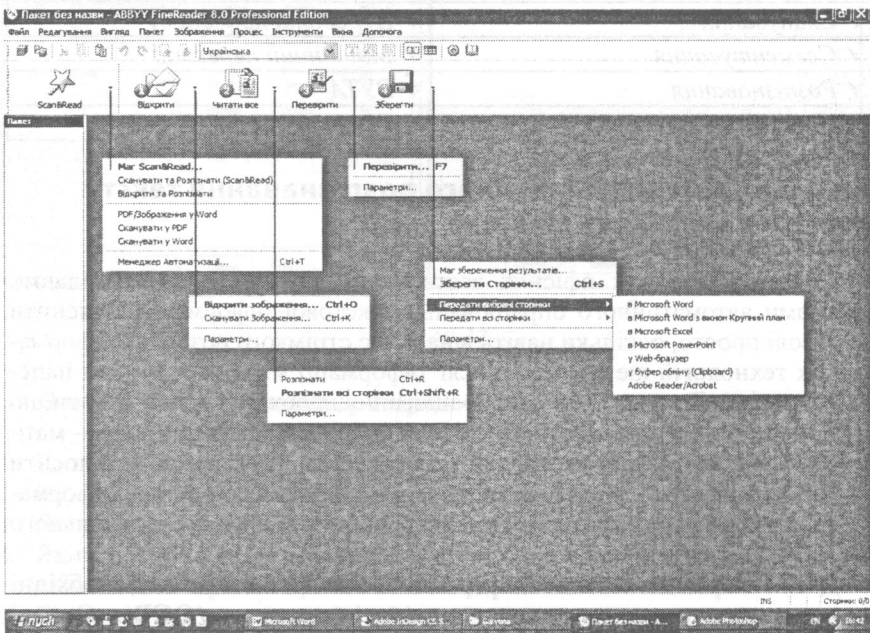


Рис. 11.1

Доступ до основних функцій програми також можна отримати через її головне меню, яке містить дев'ять пунктів: **Файл**, **Редагування**, **Вигляд**, **Пакет**, **Зображення**, **Процес**, **Інструменти**, **Вікна**, **Допомога**.

Меню **Файл** призначене для відкриття чи сканування графічних файлів, друку та збереження результатів розпізнавання. Також у цьому меню можна відкривати та перезаписувати **пакети** – набір одного чи декількох

графічних файлів, об'єднаних для зручності розпізнавання, наприклад, це можуть бути проскановані сторінки однієї книги. Один пакет може містити до 9999 графічних сторінок. Усі вони відображаються у вертикальному вікні **Пакет** ліворуч робочого поля.

Меню **Редагування** передбачає знайомі з інших прикладних програм дії – відмінити/поновити останню дію, копіювати, вирізати, вставити, пошук/автозаміна.

Меню **Вигляд** служить для зміни розташування та пропорцій вікон на робочому полі (наприклад, вищезгадане вікно **Пакет** із зображеннями для розпізнавання можна поставити у зверху, знизу та праворуч робочого поля, модифікувавши при бажанні сам спосіб відображення вікна, рис. 11.2). В цьому ж меню налаштовується і вигляд панелі інструментів.

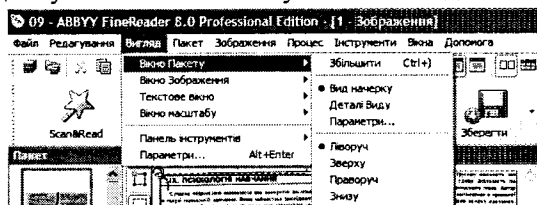


Рис. 11.2

Для роботи безпосередньо з пакетом призначене меню **Пакет**. Воно дозволяє виконувати навігацію по сторінках, перенумерувати сторінки в пакеті, вилучити сторінку або ж сам пакет.

Спрощено весь процес роботи з програмою зводиться до наступних дій: сканування (відкриття) зображення, сегментування, розпізнавання та перевірка тексту, збереження результатів.

Сегментування зображення – це його поділ на частини (блоки) відповідно до інформаційного навантаження: текст, таблиця, рисунок чи штрих-код. Тип блоку задається у меню **Зображення** (рис. 11.3).

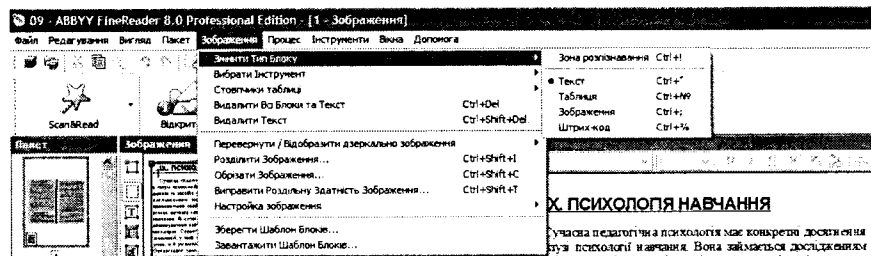


Рис. 11.3

Також меню **Зображення** містить пункти для керування структурою таблиці, вилучення блоків чи тексту, а також власне для обробки відсканованих зображень: поворот, підрізання, очищення тощо.

Меню **Процес** дозволяє задати комплексне завдання (відкрити та розпізнати, сканувати та розпізнати), автоматизоване комплексне завдання, а також дає змогу виконати перевірку розпізнаного тексту та зберегти результати.

У меню **Інструменти**, крім перевірки тексту, також можна переглянути та за необхідності відредагувати вбудовані словники розпізнавання, мови та еталони користувача. Вибір **Інструменти/Параметри** викличе вікно **Параметри** (рис. 11.4), на шести закладках якого можна встановити базові налаштування FineReader.

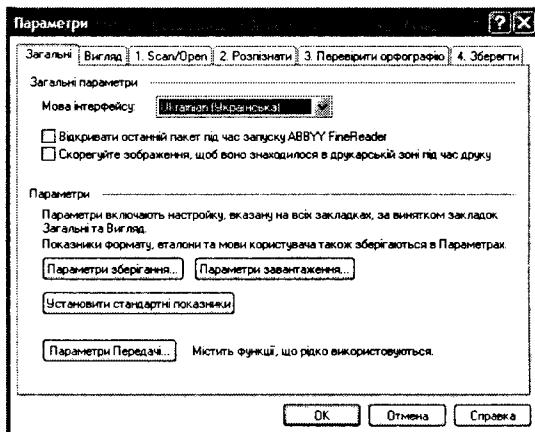


Рис. 11.4

Два останні пункти меню **Вікна** та **Допомога** стандартні для будь-якого Windows-застосування. Слід лише зазначити, що довідкова система у FineReader повністю українізована.

Під рядком меню розташовані керуючі кнопки **Scan&Read** (залежно від останньої дії вона може змінюватись на **Open&Read**), **Відкрити**, **Читати все**, **Перевірити** та **Зберегти**. Вони повністю автоматизують роботу з програмою і в цілому дублюють однойменні команди головного меню.

Тепер розглянемо детальніше усі кроки роботи з програмою: від сканування графічного документа до збереження розпізнаного тексту.

11.1.2. Сканування зображень

Вибір з меню **Процес/Додати зображення/Сканувати зображення**, натискання кнопки **Відкрити/Сканувати зображення** чи клавіатурне скорочення **Ctrl+K** ініціюють процес сканування. Залежно від типу сканера діалогове вікно сканування матиме різний вигляд (наприклад, рис. 11.5).

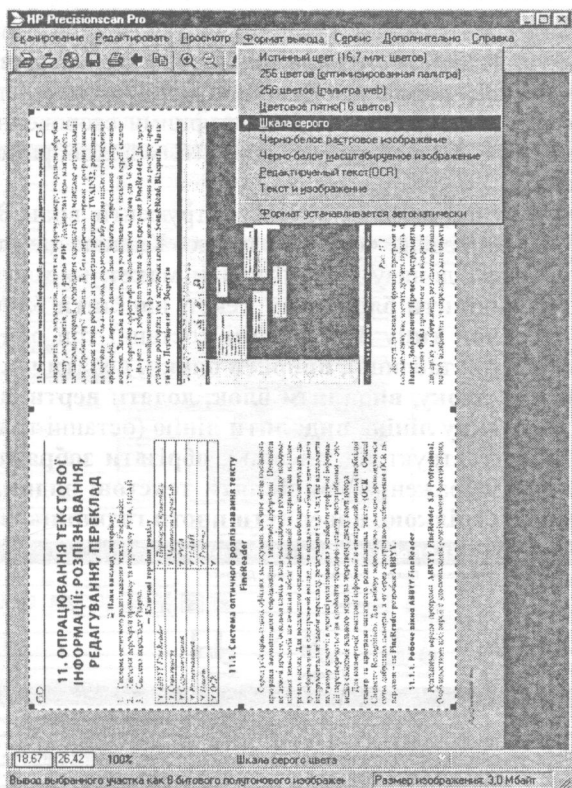


Рис. 11.5

За допомогою миші визначається область сканування, з головного чи контекстного меню вибирається формат виводу, розподільна здатність, при потребі застосовують додаткові параметри, користуючись інструкцією до сканера. У меню **Сканирование** динамічно з'явиться пункт **Вернуть изображение в FineReader**, цю ж функцію виконує і кнопка **←** на панелі інструментів.

11.1.3. Сегментування зображень

Процес сегментування полягає у виділенні блоків однотипної структури для розпізнавання. Сегментування можна здійснити вручну – визначити мишею межі кожного блоку. Автоматичне задання розбиття зображення на блоки можна виконати з меню **Процес/Розпізнати/Аналіз макета сторінки (/Аналіз макета для всіх сторінок)** чи натиснути **Ctrl+E** (**Ctrl+Shift+E** для всіх сторінок).

Приклад. Для сканованого зображення з рис. 11.5 застосовано команду **Зображення/Повернути за годинниковою стрілкою** і після того здійснено сегментування. Як бачимо, у макеті присутні чотири текстові блоки, одна таблиця і одне зображення (рис. 11.7).

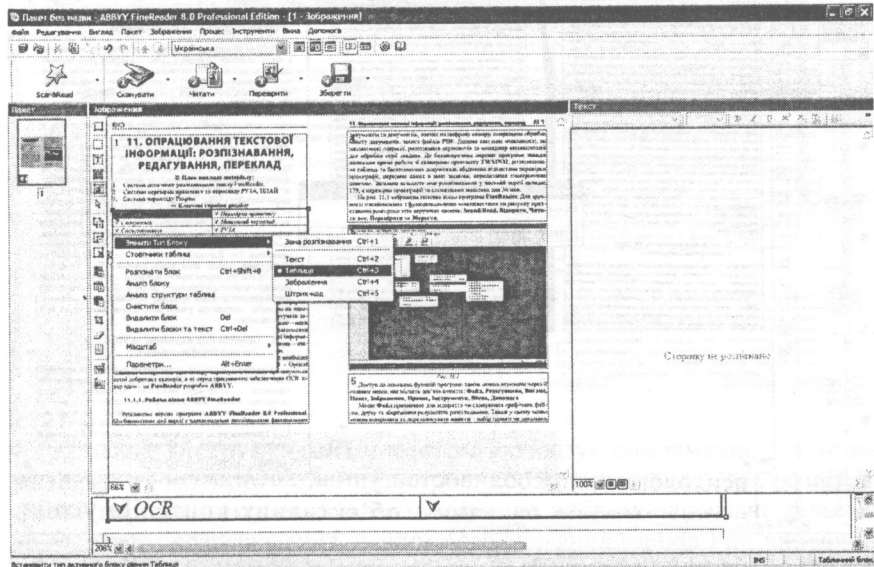


Рис. 11.7

Отримані після сегментування блоки можна буде редагувати: пересувати, змінювати межі, вилучати (щоб вилучити зайвий блок, достатньо натиснути клавішу **Delete**), змінювати тип, додавати свої блоки. Користуючись панеллю інструментів вікна **Зображення**, можна створювати блоки складної геометричної структури, вирізати частину блоку, витирати частину зображення тощо. У меню **Процес** чи в контекстному меню блоку можна вибрати пункти **Аналіз структури блоку** та **Аналіз структури таблиці**.

11.1.4. Процес розпізнавання

Натиснення кнопки **Процес/Розпізнати**, однойменний вибір з меню чи натискання **Ctrl+R** ініціюють процес розпізнавання. Перед тим слід звірити мову тексту, що розпізнається, і в разі виникнення розбіжностей вибрати потрібні зі списку **Українська** панелі інструментів.

Процес зчитування інформації ілюструється заповненням рядків блакитним кольором, потім слова, які не вдалось розпізнати (не знайдено аналога у словниках) виділяються синім кольором. Розпізнаний текст виведеться у вікно (рис. 11.8):

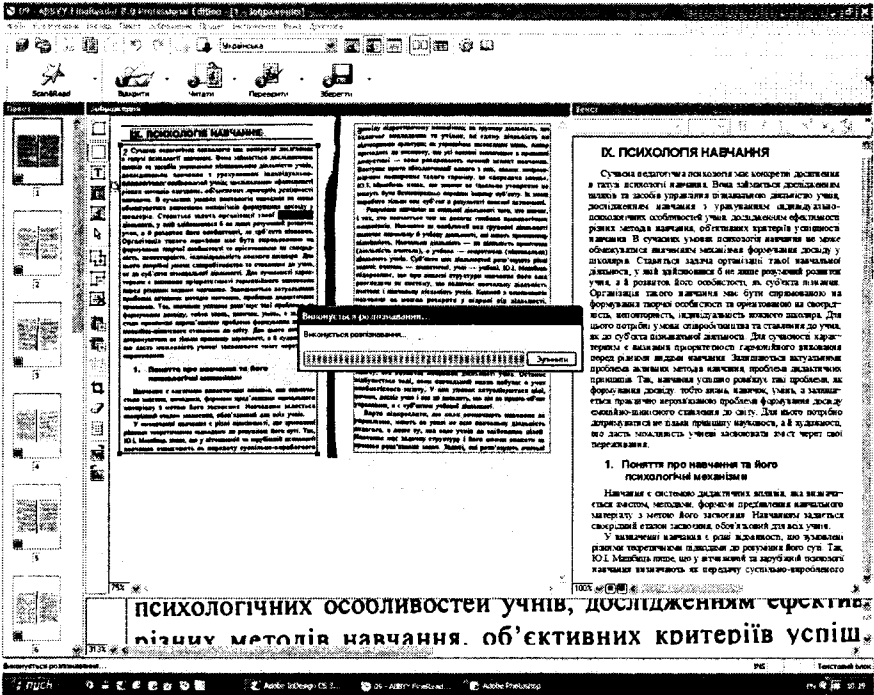


Рис. 11.8

Під час розпізнавання тексту програма може видавати системні підказки: наприклад, поради щодо того, як слід сканувати документ, щоб полегшити програмі (і собі) процес розпізнавання.

Для більшості текстів тип друку визначається автоматично. Для розпізнавання текстів, надрукованих на матричному принтері або на друкарській машинці, можна вказати програмі на цей тип друку. Для цього слід виконати Інструменти/Параметри і на закладці Загальні натиснути кнопку Параметри передачі. Відкриється вікно Параметри передачі, у полі 2.Розпізнавання якого встановити відповідний перемикач Друкарська машинка чи Матричний принтер у полі Тип друку.

11.1.5. Редагування розпізнаного тексту

Передостання кнопка Перевірити виконує команду Перевірити напис. Це ж саме можна зробити, натиснувши F7. Завантажується діалогове вікно Перевірка, зображене на рис. 11.9. У ньому відображається слово, якого не знайдено у словнику, варіанти заміни, кнопки керування: Пропустити, Пропустити всі, Додати, Замінити, Замінити всі.

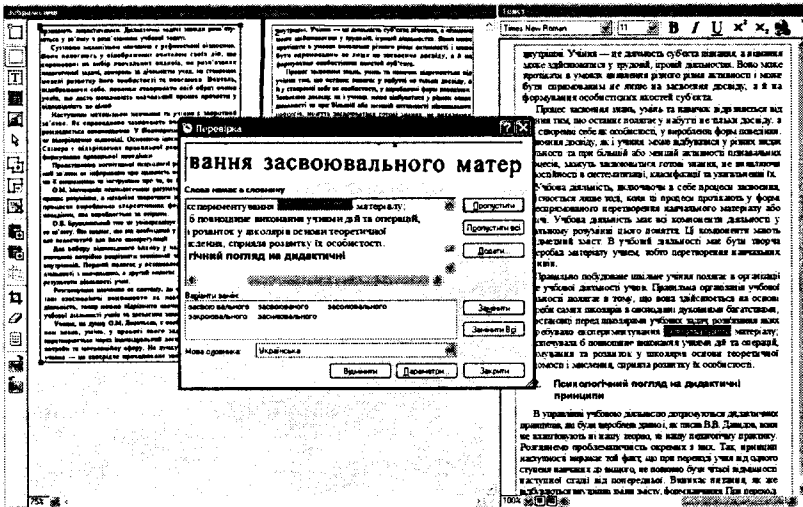


Рис. 11.9

Команда **Інструменти/Параметри** активізує одноімнене діалогове вікно, де на закладці **Розпізнати** (рис. 11.10) задаються: мова розпізнавання (перша опція у списку дає можливість вибрати одночасно декілька мов), режим розпізнавання, режим розпізнавання файлів Pdf, виділення гіперпосилань, а також розпізнавання з навчанням. Останній пункт цікавий тим, що в режимі «навчання» можна вказати програми на недоліки в розпізнаванні. При розпізнаванні з еталоном користувача отримані «уроки» FineReader братиме до уваги.

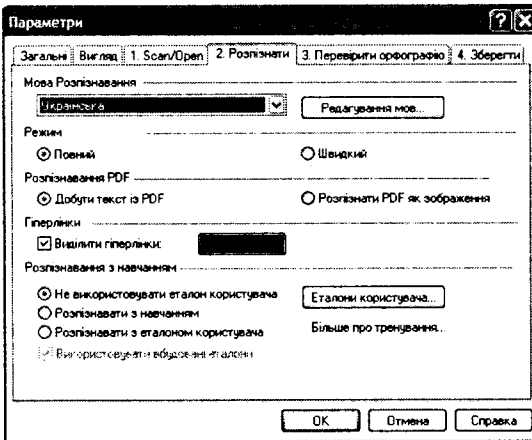


Рис. 11.10

Переглядати та редагувати словники напряму (рис. 11.11) можна, виконавши команду **Сервіс/Переглянути словники**.

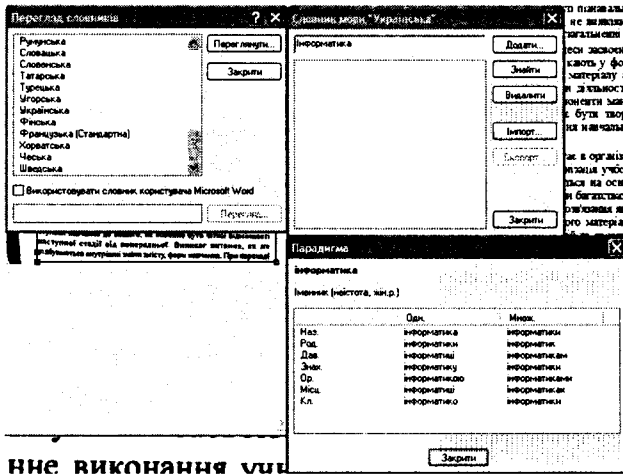


Рис. 11.11

Кнопка **Зберегти**, розгорнутий вигляд якої показано на рис. 11.12, дозволяє передати результати розпізнавання у обрану програму чи записати їх у файл. Зовнішній вигляд іконки змінюється залежно від останнього збереження; підпис **Зберегти** змінюється на назву обраної програми.

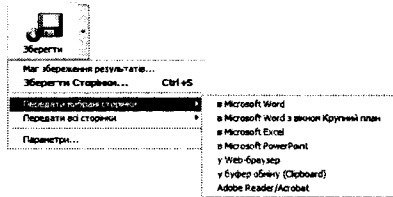


Рис. 11.12

FineReader може зберігати результати розпізнавання в таких форматах:

- Microsoft Word Document (*.DOC);
 - Rich Text Format (*.RTF);
 - Adobe Acrobat Format (*.PDF);
 - HTML;
 - Microsoft PowerPoint Format (*.PPT);
 - Простий текст (*.TXT). FineReader підтримує різні кодові сторінки (Windows, DOS, Mac, ISO) і кодування Unicode;
 - Microsoft Excel Spreadsheet (*.XLS);
 - інші формати: з кома-роздільником, формати баз даних тощо.
- На цьому процес роботи із **FineReader** можна вважати завершеним.

11.2. Системи перевірки правопису та перекладу ProLing Office, Pragma

Значно полегшити роботу з документами можна, користуючись програмним забезпеченням, орієнтованим на автоматичний переклад та автоматичну перевірку правопису. Поширеними програмними продуктами цього напрямку на даний час є **ProLing Office** та **Pragma**.

У **ProLing Office** об'єднано такі широко відомі програми фірми ПроЛінг, як система перевірки правопису українських та російських текстів **РУТА**, програма перекладу текстів з російської мови на українську та навпаки **ПЛАЙ** та російсько-українсько-російський електронний словник **УЛІС**.

За допомогою програми **РУТА** можна:

- перевірити правопис, тобто автоматично знайти та виправити помилки в словах;
- здійснити граматичний, пунктуаційний та стилістичний контроль у реченнях;
- розставити переноси в словах при форматуванні тексту;
- скористатися словником синонімів та близьких за значенням слів (тезаурусом);
- завантажити до **Windows** тексти, що набрані в будь-якій українській кодовій таблиці **DOS**, виправити помилки нестандартного кодування літер української мови, в тому числі при перенесенні текстів із **DOS** у **Windows** і навпаки.

Програма **РУТА**, інтегрована в пакет **Microsoft Office**, дає змогу перевірити правопис усього тексту, його виділеного фрагмента, а також тексту в клітинках таблиці (для **Excel** та **Access**). Перед початком перевірки слід встановити курсор на місце в тексті, з якого бажано почати перевірку, або виділити фрагмент тексту. Виклик функції перевірки правопису здійснюється стандартним способом для кожної із програм (детальніше про перевірку правопису у **Word** див. пп. 6.6).

Програма **ПЛАЙ** призначена для перекладу текстів із російської мови на українську та з української на російську. Перекладати можна весь текст або виділений його фрагмент. Запустити опцію перекладу можна декількома способами: вибором з меню **ПЛАЙ** чи натисканням відповідної кнопки на панелі інструментів.

У процесі перекладу тексту на екрані комп'ютера відображається інформаційне вікно з індикатором процесу і прогнозованим часом завершення операції.

Після перекладу на екран комп'ютера виводиться перекладений текст. За замовчуванням – це новий документ **Microsoft Word**; при цьому вхідний документ з оригінальним текстом залишається без змін. Переміщення

по тексту документа відбувається з синхронізацією вхідного й перекладеного тексту.

Щоб швидко проглянути переклад слова з російської мови на українську і навпаки, доцільно скористатись електронним словником УЛІС. Він входить до комплекту **ProLing Office**.

За його допомогою для введеного слова можна:

- отримати переклад;
- переглянути граматичні характеристики;
- отримати інформацію про схеми синтаксичного керування;
- визначити належність до позначень живих чи неживих істот, можливість вживання лише в однині чи множині (для іменників);
- отримати інформацію про вид (доконаний/недоконаний) та перехідність/неперехідність (для дієслова);
- пояснити значення для найуживаніших і багатозначних слів та приклади вживання в словосполученнях;
- переглянути синоніми слова;
- переглянути парадигму слова.

Вигляд електронного словника зображено на рис. 11.13.

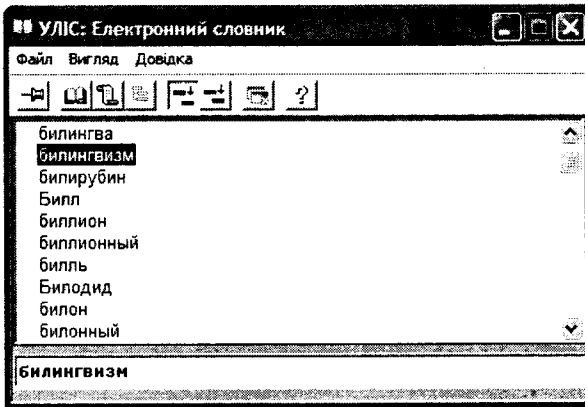


Рис. 11.13


Перша кнопка панелі інструментів задає розташування поверх усіх вікон (за замовчуванням вимкнена). Друга-четверта кнопки дозволяють проглянути відповідно переклад, парадигму та синоніми слова (якщо є). Наступна група кнопок визначає напрямок перекладу: російсько-український чи українсько-російський. Передостання кнопка закриває усі вікна з перекладом (які у програмі названі *картками*), остання викликає довідкову систему.

Інформація про словникову потужність програми наведена у довідковому вікні на рис. 11.14.

Інформація про словники		
Дата останнього оновлення: 20.04.2004		
Для української мови		
Словник	Додаток	Обсяг
Загальний	ПЛАЙ, РУТА, УЛІС	179642 слів, 2,9 млн. словоформ
Перекладів	ПЛАЙ	106550 слів
Перекладів	УЛІС	104366 словникових статей
Синонімічний	ПЛАЙ, РУТА, УЛІС	12706 груп, 35580 слів
Словосполучень	УЛІС	4324
Для російської мови		
Словник	Додаток	Обсяг
Загальний	ПЛАЙ, РУТА, УЛІС	134735 слів, 2,4 млн. словоформ
Перекладів	ПЛАЙ	101875 слів
Перекладів	УЛІС	99853 словникових статей
Синонімічний	ПЛАЙ, РУТА, УЛІС	8086 груп, 19833 слів
Словосполучень	УЛІС	4646
OK		

Рис. 11.14

Більше мов для перекладу підтримує програмний засіб **Pragma**. Мова вхідного документа визначається автоматично і переклад здійснюється у всіх напрямках між російською, українською, англійською, німецькою і французькою мовами. Програмне забезпечення **Pragma** включає два незалежні застосування – **Монітор** і **Коректор**. Вони дають змогу перекладати текстові документи з однієї мови на іншу та створювати і коригувати словники користувача.

Після інсталяції **Pragma Монітор** постійно запущений в операційній системі і його іконка  знаходиться на панелі завдань біля годинника.

Переклад тексту здійснюється так: слід скопіювати потрібний фрагмент у буфер обміну і натиснути на іконку. **Pragma** визначить мову вихідного тексту і запропонує варіанти перекладу. З'явиться діалогове вікно (рис. 11.15). У полі **З/На** визначається напрямок перекладу. У полі **Тематика** можна вибрати одну з понад сотні тематик вихідного тексту (за замовчуванням значення **Загальна**). Перекладений текст виведеться у вікно **Pragma Fast Translation**.

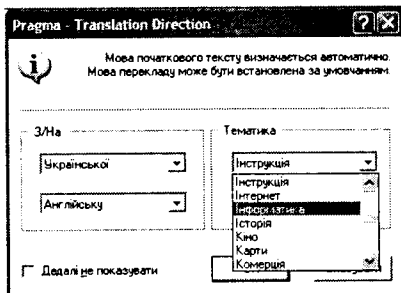


Рис. 11.15

Залежно від версії засіб **Коректор** може функціонувати тільки в режимі перегляду словників або в повному обсязі (перегляд і коригування словників, альтернатив і неперекладених слів). Загальний вигляд цього програмного за- собу з інформацією про словникову потужність наведено на рис. 11.16.

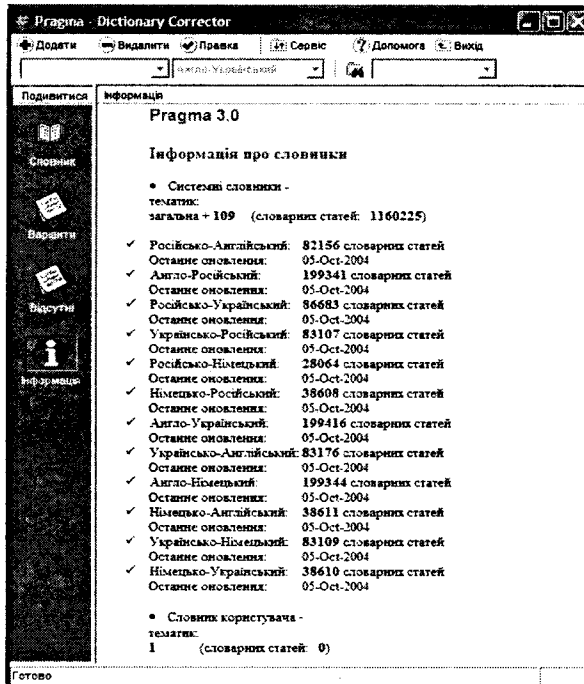


Рис. 11.16

Запитання для самоперевірки

1. Які засоби необхідні для конвертації інформації з паперових носіїв в електронний вигляд?
2. Які основні правила сканування документів?
3. У чому полягає процес сегментації зображення?
4. Які параметри розпізнавання тексту можна задати у FineReader?
5. Як виконується перевірка правопису в ABBYY FineReader?
6. У які програми можна передати текст після розпізнавання в ABBYY FineReader?
7. Який інтерфейс програм РУТА, ПЛАЙ?
8. Основні принципи роботи з електронним словником УЛІС.
9. Принцип перекладу тексту засобом Pragma Монітор.
10. Для чого призначений програмний засіб Pragma Коректор?

12. ОСНОВИ КОМП'ЮТЕРНИХ МЕРЕЖ

📄 План викладу матеріалу

1. Мережні сервіси
2. Середовища передавання
3. Мережні топології
4. Методи доступу до середовища передавання
5. Методи передавання даних
6. Принципи взаємодії відкритих систем. Модель OSI
7. Мережні операційні системи

↪ Ключові терміни розділу

✓ <i>Телекомунікації</i>	✓ <i>Модель OSI</i>
✓ <i>Локальна мережа</i>	✓ <i>Протокол</i>
✓ <i>Глобальна мережа</i>	✓ <i>Мережна архітектура</i>
✓ <i>Середовище передавання</i>	✓ <i>Віддалений термінал</i>
✓ <i>Метод доступу до середовища передавання</i>	✓ <i>Файл-сервер</i>
✓ <i>Метод передавання</i>	✓ <i>Принт-сервер</i>
✓ <i>Мережна топологія</i>	✓ <i>Сервер застосувань</i>

Потреба в передачі інформації на відстань (а саме це і визначає термін *телекомунікація*) виникла на початках цивілізації. У древніх ці функції виконували спеціальні гінці, торговці та мандрівники. Першою організованою системою гарантованої доставки повідомлень стала поштова служба. У двадцятому столітті, з відкриттям радіозв'язку та телефонії, стала можлива передача інформації на великі відстані у режимі реального часу. Телефонний зв'язок став першою телекомунікаційною мережею, що пройшла шлях від локальних мереж (з'єднання кількох абонентів або ж невеликої кількості абонентів однієї телефонної станції) до глобальної мережі, яка сьогодні об'єднує найвіддаленіші куточки земної кулі.

Виникнення та розвиток комп'ютерних мереж (КМ) пов'язаний з появою у 1960-х роках великих обчислювальних машин (мейнфреймів), що представляли собою складний комплекс електронних та електромеханічних пристроїв, потребували спеціальних умов експлуатації та великого штату обслуговуючого персоналу. Зрозуміло, що ефективним могло бути тільки колективне використання потужностей таких обчислювальних машин. Так з'явилися перші системи *телеобробки* (обробки на відстані) завдань. Такі системи надавали можливість багатьом користувачам під'єднуватись до мейнфрейма за допомогою термінальних станцій

(найпростіші термінали складались з монітора та клавіатури) з квантуванням часу роботи центрального процесора між терміналами. Однак справжній вибух у розвитку КМ пов'язаний з появою у 1980-х роках персональних комп'ютерів. Дуже швидко стало очевидним, що без об'єднання персональних комп'ютерів одного відділу, підприємства чи установи, неможливо організувати ефективне використання великих масивів даних, сумісне розв'язування задач та використання спільних ресурсів (вільного простору на магнітних дисках, пристроїв зчитування з оптичних дисків, друкуючих пристроїв, факс-модемів та ін.). Такі мережі стали називати локальними (LAN – Local Area Network). Невдовзі з'явилась потреба об'єднання комп'ютерних систем не лише у межах одного офісу, великого підприємства, але й у масштабах регіону, країни та всього світу. На відміну від локальної глобальна мережа (WAN – Wide Area Network) не обмежена територіальними рамками та об'єднує різномірні канали передавання даних.

12.1. Мережні сервіси

Комп'ютери, що входять у мережу, діляться на два типи: робочі станції, призначені для користувачів, і сервери, як правило, більш потужні комп'ютери, що надають свої ресурси всім користувачам мережі. Набір послуг (сервісів), що надаються користувачам, залежить від призначення та реалізації мережі. Коротко охарактеризуємо основні *мережні сервіси*.

Файл-сервер дає змогу користувачам доступатись до файлів, розміщених на носіях інформації серверу. В повному обсязі сервісу частина логічного дискового простору робочої станції є відображенням частини дискового простору файл-серверу, що дає змогу працювати з цією областю диска файл-серверу так, як з локальним диском робочої станції. Завданням сервера є забезпечення заданого рівня множинного доступу робочих станцій до файлів, розв'язання колізій у випадку одночасного звернення кількох станцій до одного набору даних, розмежування прав доступу тощо. Спрощений варіант – файловий обмін, у процесі якого вузли мережі (робочі станції або сервери) можуть тільки пересилати один одному файли.

Принт-сервер забезпечує обслуговування користувачів мережі друкуючими пристроями (принтерами). При цьому сервер забезпечує прийом та постановку завдань на друк у чергу, виведення їх на принтери з урахуванням замовлених послуг друку та встановлених пріоритетів. Звичайно, до принт-сервера підключаються принтери, здатні забезпечити широкий спектр послуг друку (швидкісні алфавітно-цифрові; лазерні для якісного чорно-білого, струменеві для кольорового друку тощо).

Факс-сервер забезпечує колективне використання користувачами мережі факс-модема та телефонної лінії як пристрою виведення (на зразок принтера).

Віддалений термінал (алфавітно-цифровий або графічний) забезпечує доступ робочої станції до обчислювальних ресурсів потужного комп'ютера в режимі терміналу.

Сервер застосувань є одним з варіантів технології "клієнт-сервер", в якому основна обробка та пошук інформації для користувачів одного застосування здійснюється на сервері. Функції клієнтської частини застосування, встановленої на машині користувача, можуть бути зведені до введення та відображення результатів. Такий підхід порівняно зі звичним колективним доступом до даних дає змогу суттєво зменшити завантаження мережі (*мережний трафік*).

Електронна пошта (E-mail) забезпечує обмін повідомленнями (файлами даних) між користувачами, незалежно від ступеня їх віддаленості один від одного. Електронна пошта не вимагає присутності адресата за комп'ютером у момент надсилання йому повідомлення. Аналогічно до звичайної пошти, електронний лист, що має адреси відправника та отримувача, через систему поштових серверів доставляється в особисту "скриньку" (спеціально виділену дискову область) поштового сервера, на якому зареєстрована E-mail-адреса отримувача. До електронного листа можна приєднувати файли даних (тексти, малюнки, звукові повідомлення тощо).

Діалог (Chat) дає змогу двом користувачам (або групі користувачів) мережі обмінюватись повідомленнями в реальному часі. При наявності відповідних мультимедійних технічних засобів можна здійснювати аудіо- або відеодіалог. Мультимедійне спілкування групи користувачів у реальному часі забезпечує можливість проведення аудіо- або відеоконференцій між віддаленими користувачами.

Перелічені базові мережні послуги можуть спільно (чи в певних комбінаціях) використовуватись у складних системах розподіленої обробки інформації. Наприклад, Веб-сервер та Веб-клієнт (броузер) забезпечують за текстовими та графічними вказівниками віддалений доступ до текстової, графічної, аудіо- та відеоінформації, розсіяної на сайтах мережі Інтернет.

12.2. Середовища передавання

У будь-якій комп'ютерній мережі передавання даних здійснюється за допомогою електромагнітних сигналів. Середовище, у якому поширюються електромагнітні хвилі, може бути обмеженим (фізичний провідник сигналу – кабель) або ж необмеженим (передавання мікрохвильових, інфрачервоних або радіосигналів через відкритий ефір). Найбільш поширеними у КМ є такі середовища передавання:

- *телефонні кабелі*, що найчастіше використовуються для під'єднання домашнього або офісного комп'ютера за допомогою *модема* до глобальної мережі;

- *коаксіальні кабелі*, що за своєю будовою та зовнішнім виглядом нагадують кабелі телевізійних антен, однак відмінні від них за електричними характеристиками;
- *виті пари*, що представляють собою набір пар скручених між собою на зразок спіралі ізольованих провідників (у спільну ізолюючу оболонку поміщають, як правило, чотири таких витих пари);
- *волоконно-оптичні кабелі*, що виготовляються зі скла або світлопровідних пластикових волокон і проводять світлові імпульси, які генеруються лазером або світлодіодом та приймаються фотодетектором.

Кожен з фізичних провідників має свої обмеження щодо швидкості затухання сигналу, стійкості до впливів сторонніх електромагнітних завад тощо. В оцінці якості каналів зв'язку використовують величину – *пропускну здатність*, яка характеризує максимальну кількість інформації, що може бути передана від відправника до отримувача за 1 секунду (Кілобіт/с, Мегабіт/с, Гігабіт/с або у скороченому вигляді – Кб/с, Мб/с, Гб/с). Телефонні кабелі здатні забезпечувати пропускну здатність у кілька десятків Кб/с і вважаються "повільними" каналами. Середньою пропускну здатністю володіють канали на коаксіальних кабелях – до 10 Мб/с та витих парах – від 10 до 1000 Мб/с. Високу пропускну здатність у кілька Гб/с забезпечують волоконно-оптичні кабелі.

12.3. Мережні топології

За допомогою кабелів кожен комп'ютер у КМ з'єднується з іншими комп'ютерами. Для під'єднання кабелю до комп'ютера використовуються мережні адаптери (мережні карти). Мережні адаптери підбирають залежно від типу середовища передавання, яке використовується. Часто мережні адаптери є вбудованими (інтегрованими) у материнську плату комп'ютера.

Розрізняють кілька способів фізичного з'єднання вузлів між собою, які називають *мережною топологією*. У одних випадках єдиний кабель проходить від вузла до вузла, послідовно з'єднуючи усі вузли. Така конфігурація називається *шинною* топологією або просто шиною (рис. 12.1).

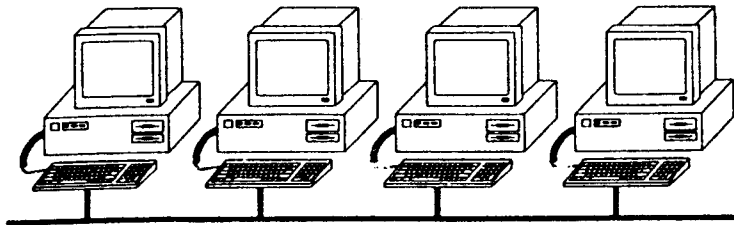


Рис. 12.1. Шинна топологія

Якщо кінці такої шини з'єднати між собою, то утвориться інша – *кільцева топологія*.

Однією з найбільш популярних сьогодні є *зіркова топологія*. У такій мережі до кожного вузла підходить власний кабель від центрального комп'ютера або ж пункту комутації, який називають *кабельним центром* або *хабом* (Hub) (рис. 12.2).

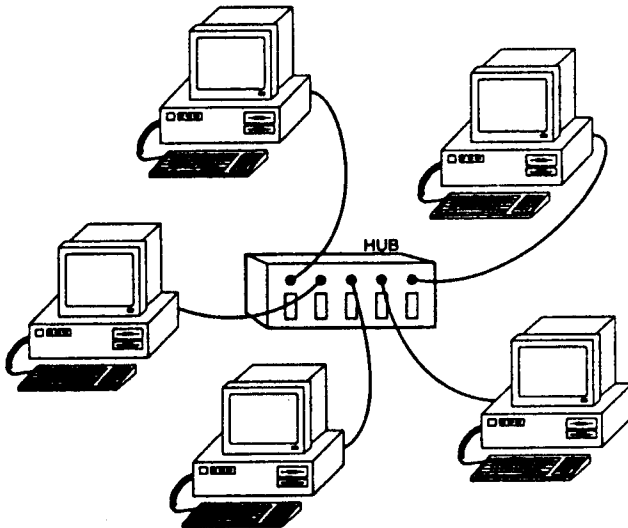


Рис. 12.2. Зіркова топологія

Зазвичай кабельні центри наділені власним мікропроцесором, який дозволяє за потребою імітувати шинну або кільцеву топологію. Завдяки цьому розрізняють *фізичну* та *логічну* топології мережі.

Кожен тип кабелю має обмеження на максимальну довжину одного неперервного відрізка (для різних типів коаксіального кабелю це 200 або 500 м, для витой пари – 100 м) та кількість вузлів, що можуть бути об'єднані в локальну мережу. У зв'язку з цим для побудови корпоративних мереж, які є об'єднанням багатьох локальних, використовують спеціальні з'єднувальні елементи та відповідне програмне забезпечення. До з'єднувальних елементів відносять мости, маршрутизатори та шлюзи.

12.4. Методи доступу до середовища передавання

Важливим у КМ є також метод доступу до середовища передавання.

Усі методи розділяються на два класи – *конкурентні* та *детерміновані*. У логічній шині інформація, що передається у середовище передавання, одночасно доступна всім вузлам. Однак зчитує та обробляє її

лише той вузол, чия адреса співпадає з адресою отримувача. Інші вузли таку інформацію ігнорують. При потребі передавання інформації мережний адаптер вузла відправника прослуховує середовище передавання на предмет відсутності у ньому інших сигналів. Якщо середовище передавання вільне – розпочинається сеанс передачі. Таким чином діє кожен вузол мережі, а отже можлива ситуація, коли передачу інформації одночасно розпочнуть два чи більше вузлів КМ. Такий випадок називають *колізією*, інформація при цьому руйнується і процес передачі доводиться відкласти на певну (випадкову) кількість квантів часу. Це – конкурентний метод доступу до середовища передавання. Такий метод може бути реалізований лише для логічної топології шини. Якщо вузлів у мережі багато та кожен з них активно використовує мережні ресурси, то колізії виникають часто, а середовище передавання є перевантаженим. У цьому випадку реальна пропускна здатність мережі може бути нижчою від максимальної допустимої для даного типу кабелю.

На відміну від конкурентного методу доступу до середовища передавання детерміновані методи не допускають виникнення колізій. Є два основних методи детермінованого доступу: *опитування* та *передавання маркера*. У методі опитування один з вузлів отримує повноваження первинного вузла. Цей вузол у визначеному порядку опитує інші (вторинні) вузли стосовно наявності у них інформації, готової до передавання. Метод опитування може використовуватись для різних мережних топологій. Однак найбільш природною для нього є топологія зірки, в якій центральний вузол відіграє роль первинного вузла. Метод передавання маркера подібний на метод опитування, який працює без первинного вузла. Первинним за чергою стає кожен з вузлів, що отримує спеціальний об'єкт – маркер. Передавання маркера розподіляє управління доступом між усіма вузлами мережі. Кожен вузол знає, від кого отримано і кому слід передати маркер. Правила визначають кожному вузлу максимальний час управління маркером. Метод реалізується для обох логічних топологій – кільця та шини. Детерміновані методи викликають певну надлишковість у використанні каналу, вимагають додаткового часу та зменшують можливості передавання для кожного з вторинних вузлів. Перевагами цих методів є повна відсутність колізій, визначений час проходження сигналу, що мало залежить від трафіку, та можливість забезпечення найбільш активним вузлам пріоритетного використання каналу.

12.5. Методи передавання даних

Окрім середовища передавання та методів доступу до нього важливим є спосіб організації передавання даних, або ж *метод передавання*. Розрізняють три основні методи передавання:

- комутація каналів;
- комутація повідомлень;
- комутація пакетів.

Метод комутації (перемикання, з'єднання) каналів полягає у попередньому встановленні фізичного з'єднання між відправником та отримувачем. Комутований канал являє собою з'єднання через пункти комутації окремих ділянок кабелів на час передавання даних. Недоліком такого методу є затримки на встановлення з'єднань та очікування вивільнення потрібних кабельних ділянок, а також неможливість одночасного використання окремих ділянок для передавання даних інших відправників та отримувачів. Прикладом мережі з комутацією каналів є телефонна мережа (АТС – пункти комутації кабельних ділянок).

У випадку комутації повідомлень інформація між відправником та отримувачем проходить шляхом запам'ятовування у проміжних пунктах комутації. Оскільки при такому способі передавання у будь-який момент часу зайнято є лише одна кабельна ділянка, що з'єднує сусідні пункти комутації, то решта ділянок можуть використовуватись для передавання інших повідомлень.

Метод комутації пакетів є вдосконаленим різновидом комутації повідомлень, у якому кожне повідомлення поділяється на маленькі частинки (пакети) фіксованого розміру. Кожен пакет містить інформацію про адреси відправника та отримувача, а також свій порядковий номер у повідомленні. Пакети передаються у мережі незалежно один від одного і можуть надходити отримувачу різними маршрутами (у кожному пункті комутації щоразу обирається найменш завантажений канал). У пункті призначення з використанням нумерації пакетів формується вихідне повідомлення. Метод комутації пакетів є одним з найпоширеніших у КМ, оскільки дає змогу досягнути найефективнішого використання пропускну здатності каналів.

Об'єднання визначених топології мережі, методу доступу до середовища передавання, типу кодування сигналів, швидкості передавання, структури фреймів називають *мережною архітектурою*. Кожній з мережних архітектур відповідає свій набір компонент – кабелі, роз'язки, мережні адаптери, кабельні центри тощо.

12.6. Принципи взаємодії відкритих систем. Модель OSI

Під *відкритою системою* розуміють таку систему, яка у випадку дотримання певних вимог (правил відкритості) може бути без будь-яких доповнень та змін під'єднана до іншої відкритої системи. Вимоги до відкритих систем визначені стандартом Open System Interconnection (OSI), що був введений у 1977 році Міжнародною організацією стандартів (ISO) та отримав код ISO7498.

Розглянемо дві системи А та В, які обмінюються даними через деяке середовище передавання (рис. 12.3). Цей процес можна розділити на фізичний (яким чином дані передаються в середовище передавання системою, що їх надсилає, та передаються із середовища до системи, що їх приймає) та логічний обмін (яким чином система, що передає дані, формує їх та забезпечує, щоб система, яка їх приймає, розуміла ці дані). Отже, фізично кожна система взаємодіє із середовищем передавання; логічно кожна система взаємодіє з іншою системою.

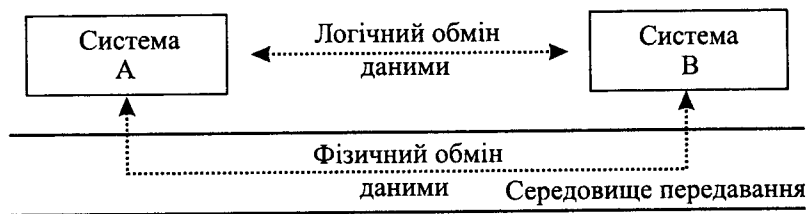


Рис. 12.3. Логічна та фізична взаємодія систем

При спрощеному розгляді вважають, що мережа містить три основні компоненти:

- 1) з'єднання – складаються з фізичних елементів апаратури, необхідних для під'єднання комп'ютера до мережі, а також із середовища передавання інформації між вузлами;
- 2) зв'язок – визначає правила, за якими вузли з'єднуються один з одним для обміну інформацією;
- 3) послуги – визначають функції, які вузли можуть виконувати в мережі. Для того, щоб мережа працювала, з'єднання, зв'язок та послуги мають бути об'єднані з дотриманням таких вимог:

- з'єднання забезпечують магістраль, по якій сигнали переміщуються між вузлами;
- зв'язок працює так, щоб один вузол міг надсилати повідомлення, а інший – приймати та розуміти його;
- вузли уміють працювати спільно так, щоб користувач одного комп'ютера міг використовувати послуги, що їх надають інші комп'ютери чи периферійні пристрої.

Отже, ми розділили процес комунікації двох користувачів комп'ютерів у мережі на три рівні. Модель OSI визначає сім рівнів та завдання кожного з них. Усередині кожної системи взаємодія відбувається між рівнями по вертикалі. Міжсистемна взаємодія логічно відбувається по горизонталі – між відповідними рівнями. Реально ж, за відсутності безпосередніх горизонтальних зв'язків, відбувається спуск до нижнього рівня в системі, що відправляє повідомлення, зв'язок через середовище передавання та підйом до відповідного рівня в системі, що приймає повідомлення.

Перелічимо основні принципи архітектури відкритих систем, при побудові технології яких за основу взято поділ комунікаційного програмного забезпечення на певну кількість рівнів:

- потрібно створити стільки рівнів, щоб процес передавання та обробки інформації можна було розбити на частини, достатні для легкого розуміння (пакели, кадри, фрейми);
- рівнів має бути небагато, аби уникнути труднощів їх інтеграції та опису;
- межу між рівнями слід провести в таких точках, де опис сервісу може бути коротким, а кількість взаємодій через цю межу мінімальною;
- аналогічні функції мають зосереджуватись в одному рівні;
- поділ слід зробити так, щоб кожний рівень був незалежним і давав змогу модернізувати та замінити його, не змінюючи інших рівнів.

Перш ніж розглянути основні рівні моделі взаємодії відкритих систем OSI, зауважимо, що еталонна модель має на меті визначити абстрактні рівні, всередині яких можуть розроблятися свої стандарти. Модель OSI визначає сім рівнів, кожний з яких є достатньо автономним і виконує чіткий набір своїх завдань. Сукупність правил та форматів, які визначають взаємодію об'єктів на n-рівні, називають n-протоколом. Важливо також усвідомлювати різницю між абстрактною моделлю та конкретною реалізацією, в якій декілька еталонних рівнів нерідко об'єднуються спільною реалізацією – *протоколом (стеком (набором) протоколів)*.

№	Система А	N-протоколи	Система В	Функції рівня
---	-----------	-------------	-----------	---------------

7	Прикладний (Application)	↔	Прикладний (Application)	Забезпечує інтерфейс прикладної програми із середовищем OSI
6	Подання даних (Presentation)	↔	Подання даних (Presentation)	Узгоджує різницю у поданні даних між користувачами
5	Сеансовий (Session)	↔	Сеансовий (Session)	Налагоджує логічний зв'язок між користувачами OSI
4	Транспортний (Transport)	↔	Транспортний (Transport)	Забезпечує наскрізні (прозорі) з'єднання та цілісність даних
3	Мережний (Network)	↔	Мережний (Network)	Забезпечує міжмережну взаємодію та маршрутизацію
2	Канальний (Data Link)	↔	Канальний (Data Link)	Забезпечує формування фреймів, виправл. помилок фізичного рівня
1	Фізичний (Physical)	↔	Фізичний (Physical)	Забезпечує фізичне кодування біт фрейму в електричні (оптичні) сигнали та передавання їх через середовище передавання
Середовище передавання				

Рис. 12.4. Модель взаємодії відкритих систем OSI

7. Прикладний рівень

Це найвищий рівень моделі, який (єдиний з рівнів) забезпечує прикладним програмам (процесом) доступ до середовища OSI. Приклади завдань цього рівня – передавання файлів, електронна пошта, управління мережею. До протоколів прикладного рівня відносять, зокрема, FTP (File Transfer Protocol) – протокол пересилання файлів.

6. Рівень подання даних

Цей рівень відповідає за сумісність подання даних між прикладними процесами, що взаємодіють (перетворення форматів даних, кодових таблиць, стискання та розпаковування даних).

5. Сеансовий рівень

Сеансовий рівень забезпечує взаємодію та підтримку діалогу між процесами певного типу (такий логічний діалог називають сеансом). Можуть передбачатись кілька різних сеансових рівнів (і, відповідно, кілька протоколів) для процесів різних типів. Для взаємодії двох або більше процесів різних типів мають бути визначені сеанси взаємодії цих процесів. Отже, до переліку завдань сеансового рівня належить синхронізація та коректне передавання файлів під час діалогу, а також надійність з'єднання до закінчення сеансу.

До протоколів сеансового рівня належить, зокрема, NetBIOS (Network Basic Input/Output System), який реалізує завдання 5, 6 та 7 рівнів.

4. Транспортний рівень

Цей рівень відповідає за прозоре передавання інформації між об'єктами сеансового рівня з визначеним рівнем якості (швидкість, економічна доцільність, рівень вірогідності). На цьому рівні дані розділяються на частини, що поміщаються в нумеровані пакети та передаються на нижні рівні. У процесі прийому даних аналізуються номери прийнятих пакетів, а їх вміст у потрібному порядку збирається та передається на вищі рівні. Транспортний рівень є проміжним та зв'язуючим між верхніми рівнями, що залежні від прикладних процесів, та нижніми, що прив'язані до конкретної мережі.

Нижні рівні можуть забезпечувати або не забезпечувати надійне передавання, у процесі якого отримувачу надходить безпомилковий пакет або повідомлення про неможливість передавання. Сервіс нижніх рівнів може бути зорієнтованим на налагодження з'єднання – при цьому передавання здійснюється без нумерації пакетів, оскільки кожен з них слідує за попередником тим самим шляхом. Після закінчення сеансу передавання з'єднання розривається. Зв'язок без налагодження з'єднань вимагає нумерації пакетів, оскільки вони можуть губитися, повторюватися, надходити не за порядком.

Приклади протоколів транспортного рівня:

– TCP (Transmission Control Protocol) – протокол UNIX та Internet-мереж з налагодженням з'єднань;

– UDP (User Datagram Protocol) – протокол UNIX-мереж без налагодження з'єднань;

– SPX (Sequenced Packet Exchange) – протокол Novell NetWare з налагодженням з'єднань.

3. Мережний рівень

Мережний (його ще називають пакетним) рівень реалізує додаткові функції маршрутизації для того, щоб кадри (фрейми) каналного рівня були доступні (прозорі) для різноманітного мережного обладнання, засобів передавання та виду доступу. Завданнями рівня є забезпечення міжмережної взаємодії, пошук шляху від відправника до отримувача (або між двома проміжними вузлами), налагодження та обслуговування логічного зв'язку між вузлами.

Приклади протоколів мережного рівня:

– IP (Internet Protocol) – протокол Інтернет-мереж;

– IPX (Internetwork Packet Exchange) – базовий протокол Novell NetWare, що відповідає за адресацію та маршрутизацію пакетів та забезпечує сервіс для SPX.

2. Канальний рівень

Канальний рівень забезпечує формування кадрів (фреймів), що передаються через фізичний рівень отримувачу. На цьому рівні здійснюється також управління доступом до середовища передавання, яке використовується кількома вузлами мережі.

1. Фізичний рівень

Найнижчий рівень моделі OSI забезпечує фізичне кодування біт фрейма в електричні (оптичні) сигнали та передавання їх через середовище передавання. Фізичний рівень визначає типи кабелів, роз'єднання, призначення контактів та формат фізичних сигналів.

Мережна архітектура визначається реалізацією фізичного та каналного рівнів моделі OSI.

12.7. Мережні операційні системи

Мережна операційна система (ОС) – це пакет програм, що забезпечує реалізацію та управління мережею, дає змогу клієнтам користуватися мережними сервісами. Основними завданнями мережної ОС є забезпечення сумісного використання та розподілу ресурсів мережі; надання клієнтам доступу до мережних сервісів; адміністрування мережі; обмін повідомленнями між вузлами мережі; взаємодія процесів у мережі; надійне зберігання даних та інших завдань, пов'язаних з функціонуванням мережі. Важливою функцією мережної ОС є забезпечення системи захисту – конфіденційності зберігання даних, обмеження прав доступу до ресурсів, парольний захист, виявлення спроб

несанкціонованого доступу, контроль за діями користувачів, ведення журналів системних подій тощо.

Програмне забезпечення клієнтської частини перетворює запити прикладної програми на використання мережних ресурсів у відповідні мережні формати, забезпечує їх пересилання через середовище передавання та здійснює зворотні перетворення. Клієнтська частина залежить від ОС, що встановлена на робочій станції (DOS, Windows, Unix, Macintosh, OS/2), та типів мереж.

У мережах з централізованим управлінням (виділеним сервером) мережна ОС є головною (або єдиною) системою, що управляє ресурсами сервера. Такі системи зазвичай мають високу продуктивність та функціональні можливості, використовують власні дискові та файлові системи, що оптимізовані для роботи в мережі (NetWare, Windows Server).

Однорангові мережі дають змогу кожному вузлу мережі одночасно виступати в ролі сервера та клієнта (Personal NetWare, Windows 98'XP). Клієнтська частина реалізується у вигляді оболонки, що обслуговує мережні запити та працює під управлінням вихідної ОС вузла (DOS, Unix, OS/2), або є органічною частиною ОС (Windows). Звичайно, клієнт може мати одночасний доступ до ресурсів різних мереж, що використовують спільне середовище передавання.

Запитання для самоперевірки

1. Поясніть термін "телекомунікації". Наведіть приклади телекомунікаційних систем.
2. Перелічіть та опишіть основні послуги (сервіси), доступні користувачам у комп'ютерних мережах.
3. Наведіть основні типи середовищ передавання та їхні характеристики.
4. Які Ви знаєте мережні топології? Поясніть, чому розрізняють фізичну та логічну топології мережі.
5. опишіть основні методи доступу до середовища передавання. Вкажіть переваги та недоліки кожного з методів.
6. опишіть основні методи передавання даних. Вкажіть переваги та недоліки кожного з методів.
7. Як відбувається взаємодія між двома відкритими системами в мережі? Що таке логічний та фізичний обмін даними?
8. Скільки та які рівні включає модель взаємодії відкритих систем OSI?
9. Що таке n-протокол? Які мережні протоколи Ви знаєте? Що називають стеком протоколів?
10. Які завдання виконує мережна ОС? Чим відрізняються однорангові мережі від мереж з централізованим управлінням?

13. ГЛОБАЛЬНА МЕРЕЖА INTERNET

План викладу матеріалу

1. Сервіси Internet
2. Доступ до віддаленого комп'ютера в Інтернет. Протокол *Telnet*
3. Передача файлів через Інтернет. Протокол *FTP*
4. Електронна пошта в Інтернет
5. Служба новин в Інтернет
6. Служба *Web*
7. Інтерактивні служби Інтернет

← Ключові терміни розділу

✓ <i>Віддалений термінал</i>	✓ <i>Список розсилки</i>
✓ <i>Telnet</i>	✓ <i>Конференції UseNet</i>
✓ <i>Протокол FTP</i>	✓ <i>World Wide Web (WWW)</i>
✓ <i>Електронна пошта</i>	✓ <i>Web-сервер</i>
✓ <i>Поштовий сервер</i>	✓ <i>Web-браузер</i>
✓ <i>Сервер новин</i>	✓ <i>Internet Explorer</i>
✓ <i>Пошуковий сервер</i>	✓ <i>IP-телефонія</i>

13.1. Сервіси Internet

Користувачів Інтернет цікавлять, перш за все, служби (*сервіси*), які надає ця мережа. Вони здійснюють і контролюють ті функції, які дозволили зробити Інтернет зручним, дружнім і багатофункціональним засобом спілкування.

Найбільш поширеними службами Інтернет є:

- віддалений термінал;
- передача файлів;
- електронна пошта;
- служба новин;
- всесвітня павутина – *World Wide Web (WWW)* або просто *Web*.

Всі ці служби у процесі функціонування використовують технологію «клієнт-сервер». При використанні цієї технології деякі комп'ютери мережі – *сервери*, обслуговують запити інших комп'ютерів мережі – *клієнтів*.

Нижче розглядаються найбільш поширені програмні засоби, які використовують ці служби.

13.2. Доступ до віддаленого комп'ютера в Інтернет. Протокол *Telnet*

Протокол віддаленого терміналу *Telnet* дозволяє отримати доступ до віддаленого комп'ютера і працювати на ньому так само, як з дисплея локального комп'ютера, тобто вводити команди і запускати програми.

Серверна програма *Telnet*, або *сервер Telnet*, забезпечує доступ локальних або віддалених користувачів до комп'ютера по протоколу *Telnet*. Після запуску сервер *Telnet* переходить в режим очікування. Коли клієнтська програма *Telnet* підключається до сервера, він видає запит на введення імені і пароля. Якщо ім'я і пароль введені правильно, на екран виводиться підказка і клієнтська програма отримує доступ до операційної системи в режимі командного рядка, тобто можна вводити команди операційної системи і отримувати виведення цих команд, ніби безпосередньо на тому комп'ютері, на якому знаходиться сервер *Telnet*.

Управління роботою сервера *Telnet* зазвичай виконується за допомогою команд, які забезпечують наступні можливості:

- запуск, зупинку, переривання і відновлення роботи сервера;
- встановлення порту для роботи сервера (за замовчуванням порт 23);
- стеження за роботою активних користувачів, переривання роботи клієнтських програм і посилання повідомлень користувачам;
- задання максимальної кількості одночасно підключених користувачів;
- задання максимальної кількості спроб підключення користувача (спроб ввести правильне ім'я і пароль);
- задання даних, що записуються у файл стеження за підключенням користувачів.

Клієнтські програми Telnet входять до складу практично всіх мережових операційних систем. Ці програми дозволяють під'єднатися до іншого комп'ютера мережі таким чином, неначебто користувач підключився до віддаленого комп'ютера з локального дисплея.

Клієнтська програма *Telnet* функціонує в двох режимах: режимі клієнта і режимі команд. Перехід з режиму клієнта в режим команд проводиться при натисненні певних клавіш (зазвичай **Ctrl+J**). Перехід з режиму команд в режим клієнта відбувається при натисненні клавіші **Enter**.

У режимі клієнта користувач може вводити команди на виконання на віддаленому комп'ютері (наприклад, команду *dir* для виведення змісту поточного каталога (папки) і отримувати виведення даної команди на екрані комп'ютера.

У режимі команд користувач може вводити команди, які виконують наступні основні функції:

- встановлення з'єднання за протоколом *Telnet* із заданим комп'ютером по заданому номеру порту;

- виведення і настройка параметрів клієнтської програми;
- посилання деяких команд протоколу *Telnet* серверу *Telnet*;
- розривання зв'язку з сервером *Telnet*;
- вихід з програми.

Серверна програма *Telnet* реалізована в операційних системах **Unix** і **Windows 2000/XP**. Клієнтські програми *Telnet* в **Unix** і **Windows 2000/XP** реалізовані з текстовим інтерфейсом, а у **Windows 95/98** – з графічним інтерфейсом.

13.3. Передача файлів через Інтернет. Протокол *FTP*

Передача файлів – це один з найперших і все ще широкоживаних сервісів Інтернет.

Протокол передачі файлів FTP (File Transfer Protocol) призначений для:

- розділення доступу до файлів на віддалених комп'ютерах;
- використання ресурсів віддалених комп'ютерів;
- забезпечення незалежності клієнта від файлових систем віддалених комп'ютерів;
- ефективної й надійної передачі даних.

FTP – це протокол прикладного рівня за технологією сервер-клієнт. Він може використовуватися як в програмах користувачів, так і у вигляді спеціальної утиліти операційної системи.

Для передачі файлів, як і для віддаленого терміналу, існує два типи програм: програми для сервера *FTP* і програми для клієнта *FTP*.

Існує два режими доступу до файлів на сервері *FTP*:

- 1) анонімний доступ, або анонімний *FTP (anonymous ftp)*;
- 2) доступ для зареєстрованих користувачів.

У першому випадку в якості імені й пароля можна ввести довільну інформацію (зазвичай ім'я вводиться як *anonymous* і як пароль – адреса електронної пошти). У другому випадку користувач повинен бути заздалегідь зареєстрований на віддаленому комп'ютері і у відповідь на відповідні запити ввести своє ім'я і пароль.

Для кожної з мережевих операційних систем існує декілька програмних реалізацій серверів *FTP*. Не дивлячись на відмінності в інтерфейсі, ці програми багато в чому схожі і забезпечують такі функції:

- управління доступом до сервера (задання порту *TCP* – за замовчуванням порт 21, час неактивності підключеного користувача в секундах, при перевищенні якого він відключається від сервера, максимальна кількість одночасно підключених до сервера користувачів, режими доступу анонімний і/чи з вказанням ідентифікатора і пароля);
- задання і редагування текстів повідомлень, що посилаються користувачам сервера;

- додавання, видалення і редагування каталогів, доступних для користувача на сервері, задання правил доступу до них;
- задання місцезнаходження і форматів (текстовий файл або база даних) для звітів про роботу сервера;
- задання обмежень на доступ до сервера;
- стеження за роботою сервера (можна вивести список користувачів даного сервера, активних у даний момент, а також відключити окремих користувачів від сервера).

Клієнтські програми FTP дозволяють під'єднуватися до серверів *FTP* і обмінюватися файлами між серверами *FTP* і комп'ютерами клієнтів. Команди або програми, що реалізують сервіс для клієнта *FTP* і працюють в режимі командного рядка, підтримують стандарт протоколу *FTP* і визначають основний набір команд, які повинні підтримуватися всіма варіантами реалізації сервісу *FTP*:

- команди управління сеансом (встановлення з'єднання з сервером *FTP*, зміни імені користувача, закриття сеансу з'єднання з сервером *FTP*);
- команди навігації на локальному і віддаленому комп'ютері (переміщення по каталогах і виведення імені і вмісту поточного каталога на віддаленому комп'ютері);
- команди створення і видалення каталогів на віддаленому комп'ютері;
- команди перейменування і видалення файлів на віддаленому комп'ютері;
- команди установки режимів прийому-передачі файлів (задання типу файла, що передається/приймається, інформації про хід прийому або передачі);
- команди прийому і передачі файлів;
- довідкові команди.

Крім клієнтських програм *FTP* в режимі командного рядка, існує безліч інших програм, що працюють в різних операційних системах і надають користувачам різний сервіс. Ці програми виконують ті самі функції, але надають для користувача зручніший інтерфейс (повноекранний або графічний).

Зокрема, програма **Far Manager**, що є варіантом популярної програми **Norton Commander** для Windows, містить вбудовану (*plug-in*) команду для *FTP-клієнта*. Ця команда працює в текстовому повноекранному режимі і після підключення до сервера *FTP* дозволяє виконувати допустимі для даного користувача операції над файлами або каталогами на сервері так, ніби вони знаходилися на локальному комп'ютері.

Іншим прикладом є програма для Windows з графічним інтерфейсом – **CuteFTP**. Засоби настройки цієї програми дозволяють задавати параметри підключення до конкретного сервера *FTP* (адреса або ім'я сервера *FTP*, ім'я і пароль для підключення, віддалений і локальний каталог для

прийому і передачі файлів, тип передаваного файлу – текстовий або двійковий тощо). Цю програму зручно використовувати в тих випадках, коли доводиться часто приймати або передавати файли на різні сервери *FTP*. Надалі для підключення до того чи іншого сервера *FTP* достатньо вибрати його ім'я. Істотною перевагою даної програми також є можливість відновлення прийому або передачі файлу при ненормальному завершенні сеансу (наприклад, через збої в каналі зв'язку) з точки розриву, а не з початку файлу (проте такий режим можна використовувати лише в тому випадку, якщо його підтримує відповідний сервер *FTP*). Такий режим дуже зручний при прийомі/передачі великих файлів по поганих каналах зв'язку.

У якості клієнтських програм *FTP* можуть працювати і *Web-браузери*.

13.4. Електронна пошта в Інтернет

Як і звичайний лист, повідомлення містять адреси відправників і отримувачів повідомлень. Адреса електронної пошти має наступний формат:

ім'я_користувача@ім'я_поштового_сервера

наприклад:

petrenko@mail.com.ua

Всі програмні засоби електронної пошти функціонують за технологією «клієнт-сервер». Функції зберігання електронної пошти та обробки транзитної пошти, що проходить через вузол, виконують програми, що звуться *поштовими серверами*. Програми користувачів (клієнтські програми) зв'язуються з поштовими серверами і відправляють пошту від користувача, а також приймають пошту, адресовану даному користувачу.

Поштові сервери розраховані на роботу під управлінням однієї з мережевих операційних систем. Незалежно від програмної реалізації, основними функціями поштового сервера є:

- реєстрація користувачів, підключених до даного поштового сервера;
- прийом повідомлень електронної пошти від користувачів та інших поштових серверів;
- тимчасове зберігання повідомлень електронної пошти (отриманих повідомлень користувачів даного поштового сервера, а також транзитних повідомлень);
- передача отриманих повідомлень електронної пошти зареєстрованим користувачам або іншим поштовим серверам;
- обробка повідомлень електронної пошти, в яких ім'я користувача даного сервера задане невірно.
- Управління функціонуванням поштового сервера виконується за допомогою засобів настройки (конфігурації) і стеження (моніторингу).

При конфігурації поштового сервера виконуються такі основні дії:

- задання адреси і порту сервера;

- управління користувачами (задання користувачів і груп користувачів даного сервера, обмеження для окремих користувачів і груп користувачів);
- управління доступом (анонімний доступ або доступ за іменем і паролем, використання засобів шифрування повідомлень, обмеження загального і транзитного доступу для окремих комп'ютерів і користувачів);
- управління поштовими скриньками (папками) повідомлень (склад поштових скриньок і їх структура);
- управління повідомленнями (максимальні допустимі об'єми повідомлень і максимальна кількість повідомлень за один сеанс для користувачів, способи обробки недоставлених повідомлень);
- управління доставкою повідомлень (режими установки з'єднання з іншими поштовими серверами – за розкладом або на вимогу, режими маршрутизації, час зберігання повідомлень на сервері);
- управління конфігурацією (визначення користувачів і/або груп користувачів, які можуть виконувати зміну конфігурації даного поштового сервера).

Засоби моніторингу поштових серверів зазвичай надають такі можливості:

- виведення активних користувачів (тих користувачів, які в даний момент часу приймають або передають повідомлення);
- відомості про повідомлення користувачів даного поштового сервера і транзитні повідомлення на інші поштові сервери;
- відомості про повідомлення для користувачів даного вузла, не отримані ними;
- відомості про повідомлення з невірними адресами, отримані даним поштовим сервером;
- відомості про з'єднання даного поштового сервера з іншими серверами (ім'я сервера, дата і час).

Існує багато *клієнтських програм електронної пошти* для різних операційних систем: від простої програми **mail** в операційній системі **Unix** з інтерфейсом командного рядка до великих і складних програм з графічним інтерфейсом типу **Microsoft Outlook**. Проте незалежно від реалізації клієнтська програма електронної пошти повинна забезпечувати такі основні можливості:

- прийом повідомлень з поштового сервера;
- переглядання отриманих повідомлень;
- введення і редагування повідомлень;
- відправлення готових повідомлень на поштовий сервер.

Додаткові можливості, що надаються багатьма клієнтськими програмами електронної пошти, включають:

- засоби конфігурації (задання адреси поштового сервера, режиму підключення: у локальній мережі чи по модему, а також настройка різних параметрів і режимів функціонування програми);
- засоби створення і супроводу поштових скриньок, а також автоматичного і/або ручного сортування повідомлень, що приймаються і відправляються, по папках (наприклад, за адресою відправника або отримувача, за датою тощо);
- перегляд і редагування заголовків і вмісту поштових скриньок користувача;
- пошук повідомлень в поштових скриньках за заданим критерієм;
- видалення повідомлень з поштових скриньок;
- вставка фрагментів *MIME-munis* в повідомлення;
- проглядання закодованих повідомлень і повідомлень, що містять фрагменти *MIME-munis*, за допомогою відповідних програм (наприклад, **Microsoft Word** або **Adobe Acrobat**);
- збереження повідомлень або їх фрагментів у файли;
- виведення на друк повідомлень або їх фрагментів;
- режим формування відповіді на отримане повідомлення (в цьому випадку відправник отриманого повідомлення автоматично стає отримувачем повідомлення у відповідь);
- формування декількох адрес електронної пошти для користувача з вибором адреси відправника при створенні повідомлення;
- створення і супровід адресної книги, що містить адреси електронної пошти отримувачів повідомлень даного користувача, а також додаткові відомості (наприклад, ім'я, прізвище, назва організації, адреса, телефон тощо);
- додавання електронного підпису, що засвідчує особу відправника;
- засоби шифрування і розшифрування повідомлень;
- засоби перевірки орфографії для повідомлень, що відправляються.

Клієнтські програми електронної пошти можуть функціонувати як окремі програми (наприклад, програма **Pegasus Mail**) або входити до складу програмних комплексів (наприклад, **Microsoft Outlook** у складі **Microsoft Office** або пакету **Eudora** у складі **Netscape Navigator**). Крім того, деякі програми електронної пошти є багатофункціональними (так, наприклад, **Microsoft Outlook** містить також засоби доступу і роботи з групами новин).

13.5. Служба новин в Інтернет

13.5.1. Способи розповсюдження новин в Інтернет

Електронна пошта зазвичай використовується для обміну інформацією між двома користувачами мережі Інтернет. Проте часто виникає потреба

в обміні інформацією між декількома користувачами (групою користувачів), що, наприклад, займаються однією й тією ж проблемою або мають спільне коло інтересів. В цьому випадку слід, щоб повідомлення одного з учасників групи потрапило до решти учасників даної групи.

Для тиражування повідомлень в мережі Інтернет широко використовуються два основні методи:

- відправка статей електронною поштою за списками розсилки учасників конференцій;
- використання розподіленої бази даних і серверів новин системи *UseNet*.

Тиражування новин *по електронній пошті за списками розсилки* здійснюється таким чином. Певні хости, наприклад, комп'ютери користувачів або поштові хости, що виконують сервіс тиражування, зберігають у себе списки певних груп. Ці списки використовуються для відправки копії тиражованої інформації – статей груп. Статті передаються як звичайна пошта, і кожний з хостів повинен переправити всю інформацію конференцій, що поступила, на хости, які зафіксовані в його списку розсилки тощо.

Цей спосіб добре працює, коли список розсилки містить не більше декількох десятків адрес. При зростанні кількості учасників конференції дана схема вимагатиме значних мережевих, процесорних і дискових ресурсів. При додаванні або видаленні користувачів виникає проблема синхронного оновлення списку учасників групи на серверах тиражування. Крім того, такий спосіб тиражування висуває додаткові вимоги до надійності роботи серверів, на яких зберігаються списки розсилки.

Використання *системи серверів новин і розподіленої бази даних для зберігання інформаційних повідомлень в UseNet* дозволяє об'єднати роботу безлічі різних мереж з різноманітним апаратним і програмним забезпеченням.

Всі статті новин даної підмережі зберігаються в одному місці, наприклад, на головному хості локальної мережі або зовнішньому сервері з високошвидкісним доступом, звідки користувачі системи, розташовані на даній підмережі, за допомогою свого локального програмного забезпечення самостійно вибирають статті для читання. Дані технологія називається *телеконференціями* або *групами новин (newsgroups)*.

Телеконференції організовані в певному порядку, заснованому на розподілі дискусій за темами, наприклад, відпочинок, спорт, новини, інформація, релігія тощо. У середині кожної з цих груп можуть знаходитися підгрупи, які, у свою чергу, теж можуть мати ієрархічну структуру.

Конференції *UseNet* поділені на сім категорій (верхній рівень ієрархії):

- 1) comp – комп'ютерна техніка і суміжні області;
- 2) news – питання, що стосуються системи телеконференцій і відповідного програмного забезпечення;

- 3) rec – хобі, мистецтво і розваги;
- 4) sci – науково-дослідницька діяльність і прикладні дослідження;
- 5) soc – соціальні питання;
- 6) talk – дискусії зі спірних питань;
- 7) misc – всі питання, які не потрапляють у вказані вище категорії або потрапляють в декілька категорій.

Наступні рівні ієрархії більш конкретизують теми телеконференцій, наприклад, телеконференція

rec.music.folk

включає дискусії про народну музику.

13.5.2. Програмні засоби супроводу списків розсилки

Хоча в протоколах електронної пошти можна задавати багатоадресні листи, все ж електронна пошта погано підходить для групового обміну, оскільки кожний з учасників повинен сам супроводжувати список розсилки для решти учасників своєї конференції. До того ж, якщо до конференції додаються нові учасники або хтось припиняє участь в конференції, кожний з учасників повинен бути оповіщений про зміну списку розсилки.

Тому першим кроком у створенні конференцій став *централізований супровід списку розсилки* одним з її учасників, найчастіше системним адміністратором одного з вузлів, який добровільно брав на себе ініціативу коригування списку розсилки і розповсюдження статей відповідно до повідомлень, отриманих від користувачів електронною поштою.

Наступним кроком стала *автоматизація супроводу списку розсилки* за допомогою спеціальних програмних засобів. Першим таким програмним засобом стала *utilima listserv*. Потім з'явилися інші утиліти, найбільш поширеними з них є *maillerv*, *majordomo* і *almanac*. Оскільки ці утиліти користуються для коригування списку розсилки практично однаковими командами, їх функціонування можна розглядати на прикладі однієї – *listserv*.

Для включення в список розсилки необхідно послати повідомлення в спеціальному форматі за спеціальною адресою на комп'ютер, який управляє даним списком розсилки. Адреси, комп'ютерів, що ведуть ті або інші конференції, та імена конференцій можна дізнатися або за списками телеконференцій на якому-небудь з вузлів мережі, або у провайдера Інтернет.

Як адреса повідомлення електронної пошти указуються утиліти та ім'я комп'ютера, наприклад:

↳ majordomo@polytech.kiev.ua

Повідомлення не повинне містити поля **Subject:** і містити тільки один рядок в наступному форматі:

subscribe ім'я_списку ім'я-користувача

де *ім'я_списку* – ім'я списку розсилки, *ім'я-користувача* – дані про користувача в довільному форматі, які він вважає за потрібне повідомити (у деяких утилітах цей параметр є необов'язковим).

У полі **From:** повинна бути вказана адреса електронної пошти, на яку висилатимуться повідомлення конференції.

Першим повідомленням, яке буде отримане після заявки на участь в конференції, є стандартний лист, що описує список розсилки і процедуру відміни.

Відмовитися від участі в конференції можна за допомогою повідомлення:

signoff *ім'я_списку*

(в утилітах **majordomo** і **almanac** замість **signoff** використовується ключове слово **unsubscribe**).

Додаткову інформацію про можливості утиліт ведення списків розсилки можна отримати, відправивши повідомлення

help

будь-якому серверу списків, який у відповідь вишле довідкову інформацію.

Для того, щоб зменшити потік даних, пов'язаний з необхідністю розсилки статей всім учасникам конференцій, створюються локальні сервери списків. Такий сервер, отримавши черговий пакет статей, сам розсилає статтю для учасників конференції в своєму домені.

13.5.3. Програмні засоби телеконференцій *UseNet*

Програмне забезпечення хоста, який надає користувачам можливість роботи зі статтями, що зберігаються на ньому, і який управляє доступом до новин і оновленням статей, називається *сервером новин*. Використовуючи *транспортний протокол новин в мережі* – *NNTP (Network News Transport Protocol)*, сервери новин обмінюються між собою статтями новин. Механізм роботи протоколу *NNTP* багато в чому схожий на механізм роботи протоколу *SMTP*. Протокол підтримує канал обміну новинами і простий інтерфейс роботи з розподіленою базою даних новин. При використанні транспортного протоколу *TCP* протокол *NNTP* працює за замовчуванням з портом 119. Алгоритм роботи з системою серверів полягає в наступному. Клієнт, що ініціює відправку, перевіряє, чи існує на сервері група новин, до якої належить дана стаття, після чого вона відправляється. Потім клієнт запрошує список нових статей, що поступили на сервер, на підставі якого він може запитати нові статті. Насамкінець клієнт повідомляє сервер про ті статті, які він вже має, для того, щоб сервер не відправляв ці статті клієнту як нові, якщо вони поступлять ще раз.

Сервер новин *UseNet* використовує протокол *NNTP* для взаємодії з іншими серверами.

Для невеликої кількості користувачів достатньо одного сервера новин. У системах з великою кількістю клієнтів необхідно використовувати так звані *проміжні сервери новин*. Такий проміжний сервер працює в кожному домені мережі і відповідає за забезпечення посередницьких дій між своїми клієнтами і головним сервером новин. Для роботи з *UseNet* клієнт такої мережі спочатку з'єднується зі "своїм" проміжним сервером і читає розміщені на ньому статті новин. Якщо сервер виявляється непрацездатним або у нього немає необхідних статей, програмне забезпечення може звернутися на вищий рівень ієрархії сервера новин.

Так само, як і для інших служб Інтернет, для доступу користувача до телеконференцій існують десятки програм, що працюють як в текстовому, так і графічному режимах.

Клієнтська програма доступу до телеконференцій опитує сервер новин, який може збирати статті не лише з системи *UseNet*, але й з інших джерел, наприклад, списків розсилки або рекламних оголошень. Ці статті зберігаються на сервері протягом деякого встановленого часу, що визначається адміністратором сервера, після чого вилучаються.

Незалежно від конкретної реалізації всі програми доступу до телеконференцій під час сеансу з сервером *UseNet* виконують такі функції:

- вибір телеконференцій для розсилки;
- вибір статей телеконференції для перегляду;
- проглядання статей вибраних телеконференцій;
- запам'ятовування відібраних статей у файлах і виведення на друк;
- посилання статті на телеконференцію;
- відмова від участі в телеконференції.

Програми доступу до телеконференцій дозволяють отримати список *UseNet* телеконференцій, що є на сервері, і вибрати ті з них, які представляють інтерес для користувача. Слід зазначити, що доступ до деяких телеконференцій є платним, тобто за участь в конференції користувач або його провайдер отримують рахунок від власника сервера *UseNet*. Як правило, при доступі до платних конференцій необхідна реєстрація користувача, яка виконується або сервером *UseNet*, або системним адміністратором сервера.

Багато програм доступу до телеконференцій дозволяють видавати не весь список статей (він може бути дуже великим), а відібрати статті за певним критерієм (наприклад, ключовим словам в назві або змісті статті, за прізвищем автора, за часом надходження).

В операційній системі **Unix** для читання телеконференцій найширше використовуються програми **nn** і **tin**. Програма **nn** є однією з найпростіших програм доступу і є практично у всіх версіях Unix. Вона функціонує в режимі командного рядка і при роботі вимагає знання численних опцій, параметрів і команд програми, а також попереднього заповнення

за допомогою текстового редактора списку вибраних телеконференцій. Програма **tin** працює в повноекранному режимі, а керування проводиться за допомогою меню.

Будовані програми доступу до телеконференцій є в *Internet Explorer* і *Netscape Navigator*.

13.6. Служба Web

Як і будь-які програмні засоби за технологією “клієнт-сервер”, програмні засоби *Web* поділяються на серверні програми – *Web-сервери* і клієнтські програми – *Web-браузери*.

Крім програми *Web-сервера*, на комп'ютері розміщуються ті дані, які користувачі отримують по запитах до *Web-сервера* (*Web-сторінки*, зображення, мультимедійні документи, сценарії, аплети *Java*, компоненти *ActiveX* тощо). Зазвичай один каталог на *Web-сервері* при настройці призначається каталогом за замовчуванням, або домашнім каталогом. Доступ до цього каталога виконується, коли в *URL* вказано тільки ім'я вузла та ім'я файлу, наприклад:

<http://www.microsoft.com/main.html>

При настройці вказуються імена файлів в домашньому каталозі, до яких здійснюватиметься доступ, якщо при виклику задається тільки ім'я вузла або (для інших каталогів) тільки ім'я каталога, наприклад:

<http://www.microsoft.com/>

Зазвичай такі файли мають імена *index.htm(html)* і/або *default.htm(html)*.

Для доступу до даних з інших каталогів використовуються так звані *віртуальні каталоги*. При настройці сервера вказуються імена віртуальних каталогів і відповідні кожному з них імена реальних каталогів. Так, якщо при настройці заданий віртуальний каталог *htmldocs* і відповідний йому реальний каталог на сервері *d:\www\docs\htmldocs* (у *Windows*), то для *URL*

<http://www.company.com/htmldocs/main.html>

виконуватиметься передача файлу *main.html* у каталозі *d:\www\docs\htmldocs* *Web-сервера*.

Незалежно від реалізації, *Web-сервер* повинен виконувати такі основні функції:

- управління доступом до сервера (задання адреси *IP* і номера порту, час неактивності підключеного користувача в секундах, при перевищенні якого він відключається від сервера, максимально допустиме число одночасно підключених до сервера користувачів, режими доступу: анонімний і/або з вказівкою ідентифікатора і пароля та інші характеристики);
- визначення домашнього каталога сервера і інших каталогів, доступних для користувачів сервера;

- визначення імен файлів, доступ до яких виконується при заданні лише імені вузла (для домашнього каталога) або імені каталога (для решти каталогів);
- задання місцезнаходження і форматів (текстовий файл або база даних) для звітів про роботу сервера;
- задання обмежень на доступ до сервера;
- стеження за роботою сервера (наприклад, виведення списку користувачів даного сервера, активних в даний момент часу, а також відключення окремих користувачів від сервера).

Крім вказаних функцій, *Web-сервер* може забезпечувати також:

- функції *проху-сервера*;
- передачу управління і початкових даних для програм, написаних з використанням інтерфейсу *CGI* і передачу виведення цих програм користувачам;
- виконання вбудованих у *Web-сторінки* команд, що зветься включеннями на стороні сервера – *SSI (Server Side Includes)*. Зазвичай такі файли мають розширення *.shtml* або *.stm*;
- використання вбудованих технологій обробки *Web-сторінок* на серверах: серверних сценаріїв на *JavaScript* або *VBScript* – технологія активних серверних сторінок *ASP*, сценаріїв на мові *PHP* – технологія *PHP* і вбудованих програм на мові *Java* – технологія *JSP*. Зазвичай такі файли мають розширення відповідно *.asp*, *.php* або *.jsp*.

Існує багато *Web-серверів* для різних комп'ютерних платформ (деякі з них можуть працювати на декількох платформах), причому як безкоштовні, так і досить дорогі. Одне з перших місць займає сервер *Apache*. Ця популярність зумовлена тим, що, не дивлячись на труднощі в установленні й настройці, сервер *Apache* є могутнім і швидким сервером (його реалізації є як для *Unix*, так і для *Windows*) і до того ж розповсюджується безкоштовно. Для нього написані розширення, що дозволяють використовувати всі вищепераховані технології обробки даних на сервері.

У середовищі *Windows* найбільш поширеним сервером є інформаційний сервер Інтернет – *IIS (Internet Information Server)* з вбудованим в нього *Web-сервером*. Сервер *IIS* входить до складу операційних систем *Windows 2000/XP*. Технологія *ASP* безпосередньо включена в *IIS* і *PWS*, і, крім того, в *IIS* можна використовувати *CGI*, а також розроблені розширення для технологій *PHP*. Для технології *JSP* необхідно використовувати спеціалізований *Web-сервер*, наприклад, *Sun Java System Application Server*.

Основними функціями клієнтської програми – *Web-браузера* є:

- передача запиту на ресурси, зробленого за допомогою *URL*, на сервер, і прийом відповіді від сервера;
- формування і переглядання *Web-сторінки* в текстовому і/або графічному режимі.

Практично всі *Web-браузери* забезпечують безліч додаткових можливостей, основними з яких є:

- виведення зображень і мультимедійних вставок (звуку і відео) у *Web-сторінці*;
- паралельне і одночасне виведення декількох *Web-сторінок* (зазвичай в різних вікнах) і навігація (переміщення) по сторінках;
- проглядання коду *Web-сторінки*;
- копіювання документа або його частини на комп'ютер користувача;
- друкування документа або його частини;
- пошук у *Web-сторінці* (це особливо корисно при роботі з великими документами, якщо треба швидко знайти необхідну інформацію);
- операції із закладками (вставка, перегляд, пошук і видалення);
- видача довідкової інформації про *Web-сторінку*;
- установка параметрів браузера для роботи в конкретному середовищі (наприклад, шрифт, мова, кодування, рівень безпеки).

Багато *Web-браузерів* мають також вбудовані можливості, що реалізуються у вигляді додаткових встановлюваних модулів (*plug-ins*), по виконанню програм на мовах сценаріїв, аплетів *Java* і елементів *ActiveX*, а також інших компонентів (наприклад, фільтрів або технологій *Flash*). Деякі *Web-браузери* забезпечують можливість роботи з іншими технологіями *Internet* (електронною поштою, передачею файлів і системою *UseNet*).

Найбільш поширеним *Web-браузером* сьогодні є **Internet Explorer**. В першу чергу це пов'язано з тим, що **Internet Explorer** включений до складу найбільш поширеної операційної системи **Windows** і тісно інтегрований з цією операційною системою (є також версії цього браузера для комп'ютерів **Macintosh** фірми **Apple**).

Рештою *Web-браузерів*, за різними оцінками, користуються лише декілька відсотків тих, хто працює в Інтернет. З них найбільшою популярністю користуються безкоштовні *Web-браузери* **Firefox**, **Netscape Navigator** і **Opera** (для **Opera** існує і комерційна версія). Ці *Web-браузери* мають ряд переваг перед **Internet Explorer**, оскільки весь час з'являються нові версії цих продуктів, в яких враховуються всі останні стандарти Інтернет. **Internet Explorer** давно не оновлювався, крім того, в його системі безпеки є істотні недоліки, що дозволяють проникнення з мережі на комп'ютер користувача, перегляд даних на ньому і навіть їх видалення.

Служба *Web* не містить в своєму складі засобів пошуку, тому інформація по *Web-сторінках* шукається за допомогою спеціальних *пошукових серверів*. Пошукові сервери в Інтернет можна розділити на дві основні категорії: машини пошуку (*search engines*) і каталоги (*directories*).

Машини пошуку є складними інформаційними системами, що використовують в своїй роботі три взаємозв'язані технології:

- збір інформації у визначеній області за допомогою мережевих роботів;

- індексація знайдених *Web-сторінок*;
- опрацювання запитів користувачів у накопиченій базі даних.

Користувач, звертаючись до пошукової системи зазвичай за допомогою *Web-браузера*, наприклад:

<http://www.altavista.com>

за допомогою ключових слів і логічних операторів (NOT, AND, OR) формує запит на пошук потрібної інформації. На деяких пошукових машинах можна обмежувати діапазон пошуку за допомогою додаткових критеріїв (дати створення *Web-сторінки*, географічний район пошуку тощо). Результатом пошуку є список *Web-сторінок*, що задовольняє заданим критеріям.

Каталоги, на відміну від машин пошуку, містять списки адрес *Web-серверів*, що також отримуються за допомогою мережевих роботів, і короткий опис *Web-сторінок* на цих серверах. Каталоги забезпечують для користувача ієрархічний і наочно-орієнтований пошук за заданими темами. Прикладами пошукових систем-каталогів є **Yahoo!**, **Google** і **Yandex**:

<http://www.yahoo.com>

<http://www.google.com> або <http://www.google.com.ua>

<http://www.yandex.com>

Окрему групу пошукових серверів становлять *метапозукові сервери*, в яких для пошуку використовуються декілька інших пошукових серверів. Одним з таких серверів є **MetaCrawler**:

<http://www.metacrawler.com>

Цей сервер використовує в роботі такі сервери, як **Google**, **MSN Search**, **Yahoo!** і **Ask**. Запит на пошук оформляється відповідно до синтаксису відповідного сервера.

Пошук за запитом користувача здійснюється по спеціальній базі даних, розташованій на пошуковому сервері. Для створення і модифікації цієї бази використовуються, як вже указувалося, мережеві роботи – спеціальні програми, які періодично проглядають відомі їм *Web-сторінки* і шукають за гіпертекстовими посиланнями нові *Web-сторінки*.

Використання мережевих роботів часто призводить до виникнення несподіваних ситуацій на *Web-серверах*. Типова ситуація: різке падіння швидкодії сервера, підключеного по низькошвидкісній лінії. Провиною цьому – мережевий робот. В цьому випадку говорять, що сервер потрапив під «швидкий вогонь» (*rapid fire*), тобто один з роботів звертається до сервера набагато частіше, ніж це дозволяє швидкодія лінії. І у адміністратора такого сервера виникає необхідність заборонити звертання до деяких каталогів (або до всього сервера).

Практичним кроком у цьому напрямку було прийняття в 1994 р. угоди, що отримала назву **Стандарт Виключення Роботів (Standard for Robot Exclusion)**. Сенс угоди зводиться до наступного: адміністратор сервера,

очочий обмежити діяльність роботів на своєму сервері, повинен описати ці обмеження у файлі з ім'ям **robots.txt**, розташованому в кореновому каталозі. Обмеження записуються у вигляді інструкцій, що забороняють одному або декільком роботам відвідувати сервер або його окремі розділи.

Не дивлячись на те, що **Стандарт Виключення Роботів** не є офіційним стандартом, багато розробників використовують його при програмуванні роботів.

13.7. Інтерактивні служби Інтернет

13.7.1. Обмін повідомленнями

Наявність електронної адреси дозволяє організувати не тільки обмін електронними листами, але двосторонній обмін текстовою інформацією (діалог) в синхронному режимі (у Інтернет для цього використовується програма **Talk**, при виклику якої як параметр задається адреса електронної пошти). Для такого обміну необхідно, щоб другий користувач був активний, тобто підключений до мережі.

Технологія **IRC** (*Internet Relay Chat* – комутована розмова в Інтернет) є розрахованим на багато користувачів розвитком програми **Talk**. При використанні **IRC** в обміні текстовою інформацією можуть одночасно брати участь декілька користувачів. При запуску програма з'єднується з **IRC-сервером**, який в свою чергу сполучений з іншими активними **IRC-серверами**. Після реєстрації користувач може проглянути список активних груп користувачів (в **IRC** зветься каналами) і список активних користувачів на кожному каналі. Потім користувач може включитися в обмін на одному з каналів або навіть на декількох каналах одночасно. Розширення **IRC – DCC** (*Direct Client-to-Client* – прямиий обмін “клієнт-клієнт”) дозволяє користувачам під час розмови обмінюватися файлами. При роботі в середовищі багатозадачної операційної системи (наприклад, **Windows**) користувач може, не припиняючи розмови, запустити програму проглядання отриманого файлу (який може містити будь-яку форму подання інформації) і оперативно відреагувати на його зміст.

13.7.2. IP-телефонія

Використовуючи протоколи **SLIP** і **PPP**, можна передавати пакети по протоколу **IP** по телефонних мережах. У пакетах **IP** можуть міститися будь-які дані, зокрема перетворені в цифрову форму голосові дані з мікрофону комп'ютера. Проте такі пакети, потрапивши на найближчий сервер, повинні достатньо швидко передаватися далі по мережі, ніде не затримуючись. Це можливо тільки за наявності достатньо могутнього мережевого устаткування і швидкодіючих ліній зв'язку.

Поява технології *IP-телефонії* дозволяє об'єднати телефонні мережі загального користування з можливостями мережі Інтернет з передачі даних за допомогою *шлюзу (gateway)*. Шлюз є пристроєм, в який з одного боку включаються телефонні лінії, а з іншого боку – мережа Інтернет.

Телефонний зв'язок через мережу Інтернет з використанням *IP-телефонії* може здійснюватися наступними способами (рис. 13.1):

- 1) комп'ютер-комп'ютер;
- 2) телефон-телефон;
- 3) комп'ютер-телефон;
- 4) *Web-телефон*.

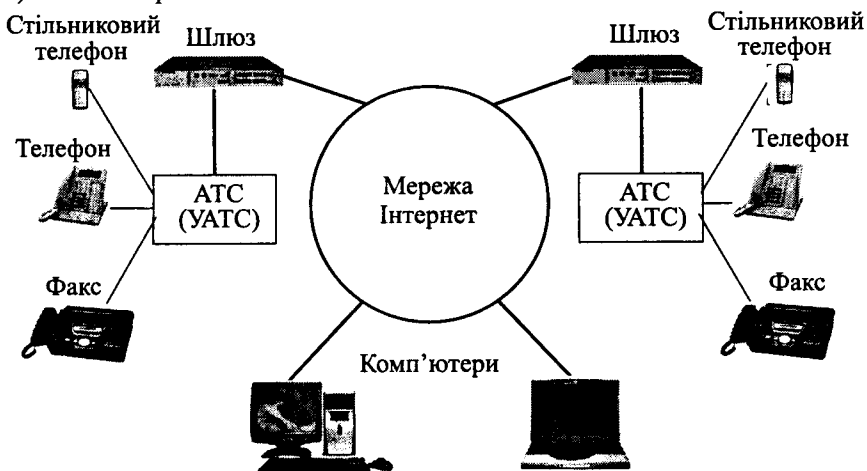


Рис. 13.1. Підключення пристроїв до Інтернет для *IP-телефонії*

Для реалізації *варіанту комп'ютер-комп'ютер* необхідне підключення комп'ютера до Інтернет, на комп'ютері повинні бути звукова карта, мікрофон і звукові колонки, а також відповідне програмне забезпечення. Це програмне забезпечення забезпечує при передачі формування послідовності пакетів *IP*, що містять оцифровані голосові дані, що поступають з мікрофона, а при прийомі перетворює голосові дані послідовності пакетів *IP* в аналогову форму для виведення на колонки. Для цього варіанту бажано високошвидкісне підключення до Інтернет.

При використанні *варіанту телефон-телефон* вхідний дзвінок і сигнальна інформація поступає на *автоматичну телефонну станцію (АТС)* або *установчу АТС (УАТС)*, а потім на шлюз, який перенаправляє сигнальну інформацію іншому шлюзу, що знаходиться на приймаючій стороні мережі Інтернет. Приймаючий шлюз забезпечує передачу сигнальної інформації на телефон через АТС або УАТС, утворюючи наскрізне з'єднання. Після встановлення з'єднання голос оцифровується в шлюзі,

розділяється на пакети *IP* і передається через мережу Інтернет. Пакети *IP*, що приходять на приймаючий шлюз, перетворюються назад в телефонний сигнал і спрямовуються приймаючому абоненту.

Варіант комп'ютер-телефон може використовуватися, якщо на комп'ютері є або програмне забезпечення для першого варіанту, або програма управління підключенням до модема телефоном. Ця програма може бути розміщена як в комп'ютері, так і в УАТС. З боку телефону з'єднання здійснюється аналогічно до попереднього варіанту.

При використанні *варіанту Web-телефон* встановлення телефонного з'єднання відбувається при натисненні курсором на посилання *Web-сторінки*, наприклад, назву компанії, ім'я абонента, що викликається, тощо.

Оскільки при телефонному дзвінку через мережу Інтернет тарифи на передачу даних значно нижчі, ніж тарифи на міжміські і міжнародні телефонні розмови, і до того ж практично не залежать від відстані, вартість розмови через Інтернет значно менша за вартість традиційного телефонного з'єднання.

Запитання для самоперевірки

1. Які основні сервіси надає мережа Інтернет?
2. Які можливості для роботи на віддаленому комп'ютері надає протокол *Telnet*?
3. Які можливості з передачі і прийому файлів забезпечують клієнтські програми *FTP*?
4. Яка структура адреси в електронній пошті?
5. Які функції виконують поштові сервери в мережі Інтернет?
6. Які можливості з передачі і прийому повідомлень забезпечують клієнтські програми електронної пошти?
7. Які методи використовуються для доставки новин в Інтернет?
8. Як задається адреса телеконференції в службі новин?
9. Як організується доступ до новин у клієнтських програмах телеконференцій?
10. Які основні можливості забезпечують *Web-браузери*?
11. Які основні реалізації *Web-браузерів* використовуються у мережі Інтернет?
12. Які категорії пошукових серверів використовуються у мережі Інтернет та які особливості пошуку даних для кожної із цих категорій?

14. ПРОЕКТУВАННЯ WEB-СТОРИНОК.

Введення в HTML

☰ План викладу матеріалу:

1. Мови розмітки документів
2. Технологія і засоби проектування Web-сторінок
3. Введення в HTML (XHTML)

↔ Ключові терміни розділу

✓ <i>Мова розмітки документів</i>	✓ <i>Зв'язування</i>
✓ <i>Проектування Web-сторінок</i>	✓ <i>Гіперпосилання</i>
✓ <i>Тег</i>	✓ <i>Карта-зображення</i>
✓ <i>Контейнер</i>	✓ <i>Фрейм</i>
✓ <i>Атрибут</i>	✓ <i>Провайдер</i>
✓ <i>Посилання</i>	✓ <i>Хостинг</i>
✓ <i>Стилі виведення</i>	✓ <i>Колокація</i>

14.1. Мови розмітки документів

Будь-який документ має три складові:

- зміст;
- структуру;
- стиль.

Зміст документа на папері може бути суто текстовим, а також містити зображення. Якщо документ представлений в електронному вигляді, він може містити і мультимедійні дані, а також гіперпосилання на інші документи. Хоча зміст різних документів різний, їх можна класифікувати за типом, наприклад, книга або залізничний квиток.

Зазвичай зміст документа подається не в довільному порядку, а має певну *структуру*. Структура – це склад і порядок розміщення частин (блоків) документа.

Стиль документа визначає форму виведення його змісту на той або інший пристрій (наприклад, принтер або дисплей). У поняття стиль входять характеристики шрифту (найменування, розмір, колір) всього документа, що виводиться, або окремих його блоків, порядок розбиття на сторінки, розташування блоків на сторінках та інші параметри. Один і той же документ може виводитися різними стилями як на різні носії, так і на один і той самий носій.

Мови розмітки документів є штучними мовами, призначеними для опису структури документа відносин між різними об'єктами структури. Дані розмітки називаються також *метаданими*.

Першою мовою розмітки є мова *GML (Generalized Markup Language* – узагальнена мова розмітки), розроблена співробітниками фірми *IBM* ще в 60-х роках минулого століття. Її безпосередньою спадкоємицею стала мова *SGML (Standard Generalized Markup Language* – стандартна узагальнена мова розмітки), що визначає правила запису елементів розмітки документа. Документ, що відповідає правилам мови, називається *документом SGML*.

Мова *SGML* широко застосовується у сфері технічної документації, де необхідне управління великими складними документами на базі різних платформ: в автомобільній промисловості, в області охорони здоров'я, у багатьох галузях телекомунікаційної індустрії і майже скрізь, де потрібні великі обсяги структурованого тексту.

Перша версія мови *гіпертекстової розмітки – HTML (HyperText Markup Language)*, так само, як і сама технологія *Web*, була розроблена Тімом Бернерсом Лі в 1991 р. Мова *HTML* є реалізацією правил мови *SGML* для типу документів, названих *документами HTML*. Програми обробки документів *HTML* називаються *Web-броузерами*. Результатом обробки документа є *Web-сторінка*, що виводиться на екран дисплея.

У 1994 р. група підтримки Інтернет – *IETF (Internet Engineering Task Force)* розробила специфікацію *HTML 2.0*, з якої почалося широке розповсюдження мови *HTML* в мережі Інтернет. У тому ж році був створений консорціум *W3C (World Wide Web Corporation)*, що об'єднав 165 комерційних і академічних організацій, розробників і користувачів (з моменту створення і по сьогодні цю організацію очолює Т.Б. Лі). Остання версія специфікації *HTML – HTML 4.01* була прийнята консорціумом у грудні 1999 р.

Для стильового оформлення *Web-сторінок* консорціумом *W3C* були розроблені *каскадні таблиці стилів – CSS (Cascading Style Sheets)*. Остання версія цієї мови – *CSS 2.0* була прийнята в травні 1998 р. У даний час в процесі розробки знаходяться версії *CSS 2.1* і *CSS 3*. Таблиці стилів можуть міститися безпосередньо в документі *HTML* або в зовнішніх файлах. В останньому випадку вони запрошуються *Web-броузером* в процесі формування *Web-сторінки*.

Крім того, документ *HTML* може містити фрагменти програм, написані на одній з мов сценаріїв. Найбільш популярною мовою сценаріїв є мова *JavaScript* (її варіант для *Internet Explorer* називається *JScript*). Фрагменти програм, так само, як і таблиці стилів, можуть задаватися і в зовнішньому файлі.

Мова сценаріїв спільно з розробленою консорціумом *W3C* об'єктною моделлю документа – *DOM (Document Object Model)* дозволяє

Web-браузеру керувати виведенням *Web-сторінки* на дисплей. Засоби *JavaScript* і *DOM* називаються інтерактивними засобами *HTML*, оскільки вони забезпечують взаємодію *Web-сторінки* з користувачем, а об'єднання засобів *HTML*, таблиць стилів, *JavaScript* і *DOM* називають іноді *динамічним HTML*, а *Web-сторінки*, сформовані за допомогою цих засобів, – *динамічними Web-сторінками*.

Зі зростанням популярності все більш стали виявлятися недоліки мови *HTML*, перш за все, відсутність засобів розширення, тобто можливості введення нових тегів, атрибутів і суті (наприклад, для відображення у *Web-сторінці* математичних формул). Крім того, у міру розвитку мови і засобів опису стилів деякі теги мови стали зайвими.

Тому стало необхідним створення нових типів документів для використання в мережі, які б доповнювали або розширювали документи *HTML*. Спочатку нові типи документів передбачалося створювати за допомогою мови *SGML*, але через складність цієї мови (специфікація містить понад 500 сторінок) було ухвалене рішення для використання в Інтернет розробити нову мову – *XML (Extensible Markup Language* – розширена мова розмітки). Ця мова є підмножиною мови *SGML*, повністю сумісною з нею. Остання версія специфікації мови *XML* – *XML 1.1* була прийнята в квітні 2004 р.

На основі мови *XML* консерн *W3C* розробив подальший розвиток мови *HTML* – мова *XHTML (eXtended HTML* – розширений *HTML*). Перша версія цієї мови – *XHTML 1.0* була прийнята в січні 2000 р. Ця версія фактично є переформулюванням *HTML 4* як застосування *XML 1.0*. Передбачається, що подальший розвиток мови *HTML* здійснюватиметься відповідно до специфікацій *XHTML*.

Нова версія *XHTML* – *XHTML 1.1* була прийнята консорціумом *W3C* у травні 2001 р. Вона визначає новий тип документа – *XHTML* на основі модулів. Кожен модуль *XHTML 1.1* містить один або декілька елементів і/або атрибутів мови *HTML*.

Введення модульності в *XHTML* пов'язане з просуванням служб *Web* на нові апаратні платформи: мобільні пристрої (наручні комп'ютери, телефони), телевізійні пристрої (цифрові телевізори, *Web-браузери* на базі телевізорів) і прилади (пристрої з фіксованими функціями). Кожний з цих пристроїв має свої особливі вимоги і обмеження.

Модульність *XHTML* надає можливість виробникам вибрати ті модулі *XHTML*, які відповідають вимогам і обмеженням пристроїв, що розробляються, а також створювати свої власні модулі, використовуючи мову *XML*.

В даний час консорціумом *W3C* розробляється нова версія *XHTML* – *XHTML 2.0*, в якій вводяться нові теги і атрибути, а також змінюється дія деяких старих тегів.

14.2. Технологія і засоби проектування Web-сторінок

Процес створення документа на мові *HTML* або *XHTML* часто називають *проектуванням* (або *дизайном*) *Web-сторінок*.

Проектування *Web-сторінок* можна проводити як в текстовому, так і в графічному режимі. У першому випадку операції з редагування проводяться над документом *HTML* (*XHTML*), в другому випадку – безпосередньо над *Web-сторінкою*, тобто обробленим *Web-броузером* документом *HTML* (*XHTML*). Другий режим називають також графічним *режимом* або режимом *WYSIWIG* (*What You See Is What You Get* – що ви бачите, то і отримаєте).

Документ *HTML* або *XHTML* зберігається як файл з розширенням *.htm* або *.html* (у **Windows** і **Unix**).

Об'єкти, що вставляються в документ *HTML*, наприклад, зображення (файли з розширенням *.gif*, *.jpg* або *.png*), зовнішні таблиці стилів (файли з розширенням *.css*) або сценарії (файли з розширенням *.js*), зазвичай розташовуються в тому ж каталозі, що й самі документи, або в підкаталогах цих каталогів, наприклад, підкаталогах *images*, *styles* і *scripts*.

Оскільки документ *HTML* (*XHTML*) є звичайним текстовим файлом, його можна створювати і редагувати за допомогою звичайного *текстового редактора*, наприклад, редактора **Блокнот** у **Windows**. Створення документа за допомогою звичайних текстових редакторів вимагає хорошого знання мови *HTML* (*XHTML*) і особливостей його реалізації на різних *Web-броузерах*. Тестування створеного за допомогою текстового редактора документа виконується за допомогою будь-якого *Web-броузера*.

Спеціалізовані текстові редактори HTML (XHTML), разом з функціями звичайних текстових редакторів, надають наступні можливості:

- автоматична вставка і виділення тегів в тексті документа *HTML* (*XHTML*) (зазвичай іншим кольором);
- синтаксичний контроль тегів *HTML* (*XHTML*) та інших компонентів (таблиць стилів і мов сценаріїв);
- імпорт документів з інших текстових редакторів;
- вбудовані довідкові і навчальні засоби.

Тестування *Web-сторінок* виконується за допомогою вбудованих в програму або зовнішніх *Web-броузерів*.

Прикладом спеціалізованого текстового редактора є редактор **CuteHTML** для **Windows**. Для проектування *Web-сторінок* в цьому редакторі використовуються спадні меню і панелі інструментів, що забезпечують підтримку всіх загальноприйнятих тегів *HTML*, включаючи форми і таблиці. У складі цього редактора є досить детальні довідкові системи з елементами навчання.

Разом з безперечними перевагами спеціалізованих текстових редакторів порівняно з текстовими редакторами загального призначення даний тип редакторів має один істотний недолік (властивий також графічним редакторам *HTML*) – відсутність підтримки функціонування інтерактивних засобів.

Вимоги *графічних редакторів HTML* до програмних і апаратних засобів комп'ютера значно вищі, ніж у текстових редакторів. Проте працювати з графічними редакторами значно зручніше, ніж з текстовими, оскільки редагований вміст сторінки виглядає так само або практично так само, як воно виглядало б при перегляді *Web-броузером*. Крім того, для виконання багатьох операцій в графічному редакторі не вимагається знання мови *HTML (XHTML)*.

Редактори *Web-сторінок* можуть бути реалізовані або як окремі програмні продукти (наприклад, *Microsoft FrontPage*), або як складені компоненти інших програмних продуктів. Так, редактор *Microsoft Script editor* є складовою частиною пакету *Microsoft Office*, а редактор *Netscape Composer* разом з *Web-броузером Netscape Navigator* є одним з компонентів пакету *Netscape Communicator*.

Крім графічного режиму, багато редакторів забезпечують також режим редагування початкового документа *HTML (XHTML)*, тобто текстовий режим. Ця можливість використовується, в основному, для внесення в документ таких елементів, які даний редактор не підтримує (наприклад, сценаріїв). Крім того, частина редакторів реалізує (за допомогою вбудованих або зовнішніх *Web-броузерів*) режим тестування створеної або модифікованої за допомогою даного редактора *Web-сторінки*. Істотно полегшує створення *Web-сторінок* також наявність в редакторі бібліотеки стандартних графічних елементів (кнопок, стрілок, різних піктограм тощо).

Найпоширенішими текстовими редакторами є редактор *DreamWeaver* фірми *Macromedia*, для якого розроблені різні модулі розширення, а також редактор *FrontPage* фірми *Microsoft*, що входить до складу *Microsoft Office*.

14.3. Введення в HTML (XHTML)

Для вставки розмітки в документ *HTML* і *XHTML* використовуються *теги*. Теги розміщаються в кутових дужках "<" та ">" і містять тільки текстові символи. Теги в *HTML* можна записувати як рядковими, так і прописними буквами, проте відповідно до специфікацій *XHTML* імена тегів повинні записуватися тільки рядковими буквами (для сумісності з *XML*).

Існує два типи тегів: контейнерні і одинарні теги. *Контейнер* – це пара тегів, тобто початковий і кінцевий теги, що мають відповідно вигляд: *<ім'я атрибуту>* і *</ім'я>*, де *ім'я* – це ім'я тега, а *атрибуту* – параметри тега. Початковий і кінцевий теги контейнера разом з їхнім вмістом

є елементом документа *HTML* (*XHTML*). Наприклад, елемент, що є відформатованим текстом, знаходиться між дескрипторами `<pre>` і `</pre>`:

```
<pre>Це фрагмент відформатованого тексту</pre>
```

Для деяких контейнерів відповідно до специфікації *HTML 4.01* кінцевий тег не є обов'язковим, та все ж рекомендується завжди завершувати контейнер кінцевим тегом. До того ж, за специфікацією *XHTML* *кінцевий дескриптор є обов'язковим для всіх контейнерів*.

Одинарний тег HTML відрізняється від контейнера тим, що це фактично тільки початковий тег. Цей дескриптор не пов'язаний з діями над текстом. Наприклад, тег `<hr>` створює горизонтальну лінію. Відповідно до специфікації *XML* кожен тег (навіть одинарний) повинен бути закритий. Для цього слід або відразу після початкового тега задати закриваючий тег, наприклад `<hr></hr>`, або в кінці тега (перед символом ">") поставити символ `" / "`, наприклад `<hr />`. Для того, щоб старі Web-браузери не сприйняли символ `" / "` як частину імені тега (в цьому випадку тег буде проігнорований як неправильний), рекомендується перед символом `" / "` поставити пропуск: `<hr />`.

Для кожного тега в специфікації *HTML* або *XHTML* визначається набір можливих *атрибутів*. Специфікація атрибуту складається з елементів, розташованих в такому порядку:

- імені атрибута, наприклад `width`;
- знак рівності (=);
- значення атрибута, яке задається рядком символів, наприклад, 80.

Атрибути діляться на дві групи: обов'язкові і необов'язкові. *Обов'язковий атрибут* повинен бути заданий в тезі (інакше ігнорується весь тег). *Необов'язковий атрибут* може бути пропущений, в цьому випадку передбачається, що атрибут має деяке значення за замовчуванням. Деякі атрибути можуть мати тільки одне фіксоване значення, що співпадає з іменем атрибута. В цьому випадку в *HTML* прийнято задавати тільки ім'я атрибута без знака рівності і значення атрибута (у *XHTML* значення повинно указуватися завжди).

Атрибути, як і теги, можна записувати як рядковими, так і прописними буквами, проте для сумісності з *XML* імена атрибутів також повинні задаватися тільки рядковими буквами. Роздільниками між атрибутами слугують один або декілька пропусків, символи табуляції, а також символи повернення каретки і переходу на новий рядок, тобто тег в документі може записуватися в декількох рядках.

Якщо значення атрибутів містять пропуски, то ці значення повинні бути розміщені в одинарні (') або подвійні (") лапки. У решті випадків лапки є необов'язковими, проте в *XML* (і відповідно в *XHTML*) значення *всіх атрибутів* слід брати в лапки.

Деякі атрибути використовуються в різних тегах (наприклад, атрибут `color` для задання кольору використовується і в тезі ``, і в тезі `<hr>`).

Проте слід мати на увазі, що допустимі значення одного й того самого атрибута в різних тегах можуть відрізнятися. Так атрибут `align`, що задає вирівнювання вмісту тега, для тегів `<caption>` і `<legend>` має допустимі значення `top`, `bottom`, `left`, `center` і `right`, а для тегів `<fieldset>`, `<hr>` і `<table>` – значення `left`, `center` і `right`.

У документах *HTML (XHTML)* з однобайтовим кодуванням можуть бути представлені тільки символи відповідної кодової таблиці, наприклад *Windows-1251*. Для представлення інших символів (наприклад, грецьких букв) можна використовувати посилання на символи – незалежний від кодування механізм введення будь-яких символів.

Посилання на символи в *HTML (XHTML)* можуть набувати двох форм:

- числові посилання на символи (десяткові або шістнадцяткові).
- посилання на комбінації символів.

Числові посилання на символи указують код символу в наборі символів документа. Числові посилання на символи також можуть набувати двох форм:

- синтаксис `&#D;`, де `D` – десяткове число, указує символ *Unicode* з десятковим кодом `D`.
- синтаксис `&#xH;` або `&#XH;`, де `H` – число, указує на символ *Unicode* з шістнадцятковим кодом `H`.

Наприклад, посилання `π` або `π` у документі *HTML (XHTML)* виведе на екран *Web-броузера* грецьку букву π .

Друга форма має синтаксис `&im'я;`, де *ім'я* є символічним позначенням посилання, наприклад, для виведення букви π можна використовувати посилання `π`.

Повний список символічних і числових позначень часто використовуваних символів приведений в специфікації *HTML 4.01*. У цей список входять наступні групи символів:

- спеціальні знаки, що не входять до стандартного набору символів клавіатури (наприклад, знак фунта £, знак авторського права © або знак ±);
- латинські букви з надрядковими і підрядковими знаками (наприклад, букви \hat{a} і \check{c} французького алфавіту);
- грецькі букви і математичні символи (наприклад, α або ∞);
- технічні символи (наприклад, символ градуса $^{\circ}$).

Наступні посилання використовуються найчастіше для представлення спеціальних символів:

- `<` (символ «менше ніж», чи ліва кутова дужка) – `<`;
- `>` (символ «більш ніж», чи права кутова дужка) – `>`;
- `&` (амперсант) – `&`;
- `"` (подвійні лапки) – `"`;
- «жорсткий» або «нерозривний» пропуск – ` `;

Оскільки символи "<" і ">" використовуються для позначення початку і закінчення дескриптора, їх поява у вмісті документа призводить до виведення помилки *Web-сторінки*. Нехай задано абзац з наступним вмістом:

```
<p>Тег <b> відзначає початок виведення тексту жирним шрифтом, а тег <i> – початок виведення курсиву.</p>
```

Web-браузер сприйме послідовності символів як теги і виведе у Web-сторінку вищенаведений текст як

```
<p>Тег відзначає початок виведення тексту жирним шрифтом, а тег – початок виведення курсиву.</p>
```

Правильний запис абзацу матиме наступний вигляд.

```
<p>Тег &lt;b&gt; відзначає початок виведення тексту жирним шрифтом, а тег &lt;i&gt; – початок виведення курсиву.</p>
```

Символ & може бути сприйнятий у деяких випадках як початок вмісту, а символ " – як закінчення значення атрибута (у випадку, якщо не заданий кінцевий апостроф в атрибуті).

«Жорсткий» пропуск використовується тоді, коли необхідно задати певну кількість пропусків, оскільки при вирівнюванні вмісту *Web-сторінки Web-браузер* сам регулює кількість пропусків між словами і перенесення рядків, незалежно від кількості пропусків і перенесень рядків, заданих в початковому документі *HTML (XHTML)*.

У будь-якому місці документа *HTML* і *XHTML* можуть бути вставлені коментарі, які так само, як і в текстах програм, служать для пояснення виконуваних дій і не виводяться на екран *Web-браузера*. Коментарі записуються в наступному вигляді:

```
<!--Текст коментаря -->
```

Текст коментаря може записаний в будь-якому допустимому кодуванні і займати декілька рядків.

Запитання для самоперевірки

1. Які три компоненти містить будь-який документ? Дайте коротку характеристику кожного компонента.
2. Як визначаються та де використовуються мови розмітки документів?
3. Які мови та специфікації використовуються при створенні Web-сторінок?
4. Які особливості використання мови XHTML у Web-сторінках?
5. Які типи редакторів використовуються при створенні та модифікації Web-сторінок? Які основні особливості кожного із них?
6. Опишіть синтаксис тегів та атрибутів у мовах.
7. Які основні види посилань на символи визначені в мовах HTML й XHTML?

СПИСОК ЛІТЕРАТУРИ

1. Антонов В. Сучасні комп'ютерні мережі. – К.: МК-Прес, 2005. – 480 с.
2. Архангельский А.Я. Object Pascal в Delphi. – М.: Бином, 2002. – 480 с.
3. Баженов В.А., Венгерський П.С. та ін. Інформатика, комп'ютерна техніка, комп'ютерні технології / За ред. Г.А. Шинкаренка, О.В. Шишова. – 3-тє вид. – К.: Каравела, 2011. – 592 с.
4. Бакушевич Я.М. Інформатика і комп'ютерна техніка. – Львів: Магнолія-2006, 2011. – 286 с.
5. Блаттнер П. Использование Microsoft Office Excel 2003. – М.: Вильямс, 2004. – 864 с.
6. Буров Є.В. Комп'ютерні мережі. – Львів: Магнолія-2006, 2010. – 262 с.
7. Дженнингс Р. Использование Access 2003. М.: Вильямс, 2005. – 1312 с.
8. Дудзяний І.М. Програмування мовою Object Pascal. – Львів: Видавничий центр ЛНУ імені Івана Франка, 2004.
9. Жадаєв А.Г. Самоучитель HTML 4. – К.: Юніор, 2001. – 296 с.
10. Злобін Г.Г., Рикалюк Р.Є. Архітектура та апаратне забезпечення ПЕОМ. – К.: Каравела, 2012. – 304 с.
11. Кравчук С.О., Шонін В.О. Основи комп'ютерної техніки: Компоненти, системи, мережі. – К.: Каравела, 2012. – 344 с.
12. Мак-Клеланд Д. Библия пользователя Photoshop CS 2. – М.–СПб.–К.: Вильямс, 2006. – 938 с.
13. Меженный О.А. Microsoft Office 2003. Самоучитель. – М.: Диалектика, 2005. – 480 с.
14. Олифер В.Г., Олифер Н.А. Компьютерные сети. Принципы, технологии, протоколы. – СПб.: Питер, 2006. – 960 с.
15. Пасічник В.В. Веб-дизайн. – Львів: Магнолія-2006, 2012. – 530 с.
16. Пасічник В.В. Інтелектуальні системи. – Львів: Новий Світ – 2000, 2011. – 408 с.
17. Пасічник В.В., Верес О.М. Системи баз даних та знань. – Львів: Магнолія-2006, 2012. – 456 с.
18. Порев В., Блінова Т. Комп'ютерна графіка. – К.: Юніор, 2004. – 456 с.
19. Послед Б.С. ACCESS 2002. Приложения баз даних: Лекции и упражнения. – СПб.: ДиАСофтЮП, 2002.
20. Смолина М.А. CorelDRAW 11. Самоучитель. – М.: Диалектика, 2004. – 400 с.
21. Уокенбах Д. Microsoft Office Excel 2003. Библия пользователя. – М.: Диалектика, 2004. – 768 с.
22. Форкун Ю.В., Длугунович Н.А. Інформатика. – Львів: Новий Світ – 2000, 2012. – 460 с.
23. Хислор Б., Энжелл Д., Кент П. Word 2003. Библия пользователя. – М.: Диалектика, 2004. – 784 с.

Навчальне видання

Авторський колектив:

**БАЖЕНОВ Віктор Андрійович,
ВЕНГЕРСЬКИЙ Петро Сергійович,
ГАРВОНА Віктор Степанович,
ГОРЛАЧ Віталій Михайлович,
ДУДЗЯНИЙ Ігор Михайлович,
КОРКУНА Михайло Дмитрович,
КРАВЧУК Сергій Олександрович,
ЛЕВЧЕНКО Олександр Миколайович,
ЛІЗУНОВ Петро Петрович,
РЕЗНІКОВ Антон Сергійович,
ШОНІН Володимир Олександрович**

Наукова редакція:

**ШИНКАРЕНКО Георгій Андрійович,
ШИШОВ Олег Володимирович**

ІНФОРМАТИКА.

Комп'ютерна техніка.

Комп'ютерні технології

Здано на складання 01.06.2012 р. Підписано до друку 16.07.2012 р.
Формат 60x84/16. Папір офсетний. Друк офсетний. Гарнітура Таймс.
Ум. друк. арк. 29,5. Обл.-вид. арк 30,55.

Видавництво «Каравела»,
просп. Рокосовського, 8а, м. Київ, 04201, Україна.

Тел. (044)592-39-36, (050) 355-77-75, (068)345-09-51
E-mail: caravela@ukr.net. www.caravela.kiev.ua

Свідоцтво про внесення суб'єкта видавничої справи до Державного
реєстру видавців, виготівників і розповсюджувачів видавничої продукції:
ДК № 2035 від 16.12.2004 р.

Віддруковано з готових діапозитивів
в друкарні ТОВ "Друкарня "Рута"
(свід. Серія ДК №4060 від 29.04.2011 р.)
м. Кам'янець-Подільський, вул. Пархоменка, 1
тел. 0 38 494 22 50, drukruta@ukr.net
Замовлення № 245

ІНФОРМАТИКА

Комп'ютерна техніка

Комп'ютерні технології

Підручник містить основи концепції та методів інформатики, які реалізовані в сучасному апаратному та програмному забезпеченні комп'ютерів. Розглянуто роботу операційних систем Windows XP та Linux. В доступній формі описано принципи роботи з програмами сімейства Microsoft Office: Word, Excel, Access, PowerPoint. Описано роботу з програмами оптичного розпізнавання, перекладу та перевірки правопису тексту. Окремий розділ присвячено питанням комп'ютерної графіки. Значну увагу приділено програмуванню в середовищі Object Pascal. Також розглянуто основи мережних технологій, роботу в глобальній мережі Internet та створення Web-сторінок.

Для студентів вищих навчальних закладів, аспірантів та викладачів, а також усіх тих, хто вирішив зробити досягнення інформаційних технологій повноправним елементом свого арсеналу професійних інтересів.

Стислий зміст

1. Основи інформатики
2. Основи комп'ютерної техніки
3. Операційна система Windows XP
4. Операційна система Linux
5. Основи програмування мовою Object Pascal
6. Текстовий процесор Microsoft Word
7. Табличний процесор Microsoft Excel
8. Робота з базами даних. Microsoft Access
9. Створення презентацій. Microsoft PowerPoint
10. Комп'ютерна графіка
11. Опрацювання текстової інформації: розпізнавання, редагування, переклад
12. Основи комп'ютерних мереж
13. Глобальна мережа Internet
14. Проектування Web-сторінок



ВИДАВНИЦТВО
"КАРАВЕЛА"