http://primat.at.ua/publ/19

http://rybanoff.narod.ru/bibliotek/discret/diskr.htm

http://rybanoff.narod.ru/bibliotek/discret/diskr.htm

http://www.intuit.ru/department/ds/discrmath/

http://www.univertv.ru/

http://prosolupov-univer.narod.ru/

http://discretka.narod.ru/

http://www.intuit.ru/video/

http://otvety.google.ru/otvety/thread?tid=7c3d11c9c891f202

http://booksforstudy.ru/281-skachat-besplatno-knigu-algoritmy-diskretnoj.html

<http://sceptic-ratio.narod.ru/index.htm>

<http://tgspa.ru/info/education/faculties/ffi/impi/dis/dis_inform.htm>

<http://crow.academy.ru/dm/> МГу

Видеолекции

http://www.univertv.ru/video/matematika/d957bd15/?mark=science1

<http://mathhelpplanet.com/viewtopic.php?f=41&t=56>

<http://tgspa.ru/info/education/faculties/ffi/impi/dis/dis_video.htm>

<http://fvl.fizteh.ru/courses/>

<http://ru.gogetpapers.com/Faculty/Computer_Science>

<http://ru.gogetpapers.com/Papers/Undergraduate_%D0%A1%D0%B5%D0%BC%D0%B8%D0%BD%D0%B0%D1%80_%D0%BF%D0%BE_%D0%B4%D0%B8%D1%81%D0%BA%D1%80%D0%B5%D1%82%D0%BD%D0%BE%D0%B9_%D0%BC%D0%B0%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B5>

<http://math.hse.ru/courses_math/09/math_total.html#dis>

Re: Нужность дискретной мАтематики

Для программирования надо **учить** **дискретную** **математику**. Но перед этим надо выучить **математический** анализ, это как азбука для **математика**. В

 программе нужно было считать коэффициент сцепления напыления с подложкой. Спецы дали книгу, ткнули в формулу - пиши. Но программа упорно для любых данных давала коэфф. равный  2/sqrt(3). Дня 2 ошибку искал. Затем внимательней прочел книгу и обнаружил ляпу при выводе той самой формулы. А если выводить правильно, то как раз и получаются те 2/sqrt(3).
А мат. логика, а теория автоматов, а теория информации, а ...
Знание математики ОБЯЗАТЕЛЬНО! Просто некоторые применяют ее даже не задумываясь и не замечая. Конечно некоторые разделы можно было бы давать обзорно, зато другие нужно по полной программе и так, чтоб на всю жизнь запомнилось.

В догонку... Мой сотрудник только что рассказал историю. Однажды ему поручили задачу, которая сводилась к численному решению системы диф.уравнений. После длительного периода "кодирования" и тестирования, когда задача все никак не решалась, он отбросил программирование и посмотрел на это с точки зрения математика (сразу надо было это сделать). Оказалось, что задача даже в теории не имела решения. Он показал доказательство начальнику и отдел начал уточнять постановку задачи. А если бы не математика? Как долго они бы еще решали нерешаемую задачу?

<http://forums.avalon.ru/forum/topic.asp?TOPIC_ID=1919>

Что же касается дискретной математики, то любой уважающий себя программист обязан иметь хотя бы какое-то представление о булевой алгебре, наивной теории множеств и подобных вещах; в противном случае следует признать факт низкой квалификации такого программиста.
По моему глубокому убеждению, будущий математик-программист должен:
а) обязательно иметь первое образование именно техническое, а не гуманитарное;
б) прослушать хотя бы обзорный курс о численных методах как неотъемлимую и важнейшую часть
подготовки по этой специальности.

<http://b-al-u.livejournal.com/16798.html>

Теперь конкретно по вопросу. Зачем нужна дискретка. А затем, что все в программировании вертится вокруг ее базовых принципов. Точнее покоится на них. Некоторые особо мощные языки, известные как языки функционального программирования - не что иное, как реализация математических понятий в чистом виде.
Аргумент "где мои 17 лет" - к сожалению, я тоже не могу отнести себя к этой хорошей категории, что не мешает мне пересматривать, время от времени, картину мира. В результате -  что-то отвергать, что-то пересматривать, дополнять, углублять. И изучать новое. А в программировании, как известно, чтобы стоять на месте нужно бежать из всех сил. Процесс постоянного освоения нового неизбежен. Учиться новому - такая же работа, как и создание кода.
И наконец мои советы. Используйте то, что в своих программах используют лучшие университеты мира, занимающиеся computer  science - MIT, Беркли, Кембридж:
1) По дискретке. Мне очень понравились такие книги:
    **Род. Хаггарти. Дискретная математика для программистов**. Очень хорошая книга. К сожалению перевод имеет тот недостаток, что в нем много опечаток, иногда очень существенных - редактора надо выгнать как бракодела. Но достоинства - основная часть материала очень хорошо изложена и показано реальное применение описанных разделов дискретки на практике.
    **Биркгоф Г., Барти Т. Современная прикладная алгебра.** Книга хоть и старая, но мне очень понравилась.
2) Настоятельно рекомендую **SICP**  - это программистская суперкнига. [**Вот**](http://local.joelonsoftware.com/mediawiki/index.php/%D0%9E%D0%BF%D0%B0%D1%81%D0%BD%D0%BE%D1%81%D1%82%D0%B8_%D0%BE%D0%B1%D1%83%D1%87%D0%B5%D0%BD%D0%B8%D1%8F_%D0%BD%D0%B0_Java), что пишет о ней Джоэл Спольски. А [**вот**](http://vshabanov-ru.blogspot.com/2007/03/sicp.html) В. Шабанов. В сети в свободном доступе есть русский перевод. А на сайте MIT есть и виделекции этого курса и сама книга.
3) **Книга Э. Реймонда "Искусство программирования для Unix"**. Скажу лишь, что это моя самая удачная покупка в 2006 году. Спольски тоже [**отзывался**](http://russian.joelonsoftware.com/Articles/Biculturalism.html) о ней. Автор распространяет текст книги бесплатно.
4) Курса лекций **Introduction to functional programming** Джона Харрисона. ***Предварительный*** перевод доступен  [**здесь**](http://code.google.com/p/funprog-ru/downloads/list).  А [**здесь**](http://code.google.com/p/funprog-ru/) сам сайт по переводу.
5)  Знакомство с культурой Unix - дало мне огромную пользу. И не только мне - не зря это один из основных инструментов для обучения в вышеперчисленных  ВУЗ-ах. Самый простой способ с ней познакомится - поставить какой-то из свободных клонов Unix - Linux, например.

Человек же спрашивал о программировании. Почти любое программирование - это приложение дискретной математики. Правда, человек, пишущий программы, может об этом и не догадываться, хотя бы в силу своего возраста и образования.

Как часто(если) вам в программировании приходилось использовать Дискретную математику?

Достаточно часто при проектировании систем, использующих те или иные математические методы.
Например, при разработке методов AI в игровом движке и т.п.
P.SНа самом деле при эффективной разработке она используется все время. Элементарный пример - в любой системе полно инвариантов, которые надо контролировать (в частности, у любой абстракции есть representation invariants).. А почему ты так выделяешь дискретку из математики и науки в целом

Системное программирование, кодирование аудио.

*университете меня заставили проходить год на лекции по математическому анализу, но я никогда из него ничего не использовал ни на занятиях по информатике, хотя для них матанализ был необходимым условием, ни в реальной жизни)*

При чем здесь матанализ? Он действительно может не понадобиться, если задачи решаются вне той области, где это действительно нужно. А дискретка - вещь нужная любому программисту.

PS. Даже если матанализ не будет никогда испольоваться - его все равно надо учить, чтобы не изобретать велосипеды, если попадется что-то на эту тему. Хотя-бы на уровне "Помню, что-то такое мы проходили ..." - чтобы приблизительно знать в какую чторону копать.

Для быдлокодера может и не нужна. Опять жу у вас психология ущербная: лишнее знание НИКОГДА НЕ ПОВРЕДИТ, а вот незнание может очень серьёзно навредить. Если вам рост в профессиональном смысле не нужен, то можете вообще бросить университет. Для быдлокодера образование вообще вредно....

**Re: Нужность дискретной математики**

а кто такой "быдлокодер"?если лишнее знание НИКОГДА НЕ ПОВРЕДИТ,отчего бы вам на досуге не выучить язык древней цивилизации Майя?

**Re: Нужность дискретной математики**

Запишите меня в быдлокодеры, но Голуб отчасти прав. В программировании от владения языком (человеческим, а не ЯВУ) и четкого логического мышления (такого, которое вырабатывается курсом логики) зависит не меньше, чем от знания математики. Тем более, что математика нужна довольно специфическая. А если учесть, что работа в команде требует некоторых навыков общения, то увеличение доли гуманитарной (языковой) кажется хорошей идеей.

Есть здесь те, кто программирует 15-20 лет и больше? Интересно было бы услышать их мнение.

>а кто такой "быдлокодер"?если лишнее знание НИКОГДА НЕ ПОВРЕДИТ,отчего бы вам на досуге не выучить язык древней цивилизации Майя?

Вы не поверите, но профессиональные программисты учат таки "языки Майа", применительно к своей области, разумеется, такие как: lisp, рефал, smalltalk. Наверное им просто делать нечего, раз они такой ерундой занимаются... И вообще причём тут языки - вопрос был о теории.

**Re: Нужность дискретной математики**

Хотя я программирую активно не более 6 лет, считаю, что все это очень важно.

Как-то читал на java.sun.com статью, где один товарищ сравнивал написание программ с сочинением стихов и утверждал, что интуиция и умение чувствовать прекрасное - необходимые навыки для настоящего программиста. Сам же я нередко сначала чувствую, что в коде что-то не так, а потом действительно нахожу ошибку. :)

Если серьезно, то можно писать программы без знания математики. Но есть еще другие профессии, такие как архитектор, алгоритмист, и самое сложное - постановщик задачи. Если есть желание быть не только простым программистом-кодером, то математику знать необходимо. Кроме того, занятия математикой формируют особый способ мышления.

**Re: Нужность дискретной математики**

*> интуиция и умение чувствовать прекрасное - необходимые навыки для настоящего программиста.*

Очень хорошо сказано - "навыки". Это именно навыки - постоянно используемые на практике.

*> архитектор, алгоритмист, и самое сложное - постановщик задачи*

Это три \_очень\_ разные роли (хотя это не профессии ни разу). Алгоритмисту, конечно, математику знать необходимо. А вот двум другим - это полезно, но умение четко выражать мысли, общаться с людьми и организовывать работу не менее полезны. Это пограничные роли - там разбираться в людях нужно не меньше, чем в технике дела.

**Re: Нужность дискретной математики**

*>Вы не поверите, но профессиональные программисты учат таки "языки Майа", применительно к своей области, разумеется, такие как: lisp, рефал, smalltalk. Наверное им просто делать нечего, раз они такой ерундой занимаются... И вообще причём тут языки - вопрос был о теории.*

я употреблял "язык Майя" в прямом смысле.учить что-то потому что оно не мешает и потому что знания лишними не бывают - мне кажется довольно сомнительным занятием.

**Re: Нужность дискретной математики**

если кому-то кажется, что дискретку учить не нужно, то пусть не учит. а потом на своей жн румяной заднице почувствует шрамы от больнобьющего математического кнута.

кстати, о кнутах: Кнута читать нужно. "конкретеую математику".

**Re: Нужность дискретной математики**

Опять же, Голуб писал про тех, кто решает тактические задачи. То бишь code monkeys. А для тех, кто решает стратегические задачи (те самые архитекторы, алгоритмизаторы, эксперты по выбору средств реализации и т.д.) знание математики на уровне как минимум технического ВУЗа --- необходимость. Во-первых, потому что ум в нужный порядок приводит и учит правильно думать, и во-вторых, потому что математические знания приходятся к месту очень часто и не в математических задачах.

PS: Будущих code monkeys обычно видно уже во время обучения в университетах/институтах.

Будущих code monkeys обычно видно уже во время обучения в университетах/институтах.

по каким признакам

**Re: Нужность дискретной математики**

*> Опять же, Голуб писал про тех, кто решает тактические задачи. То бишь code monkeys*

По приведенной цитате я этого не заметил

*> для тех, кто решает стратегические задачи (те самые архитекторы, алгоритмизаторы, эксперты по выбору средств реализации и т.д.) знание математики на уровне как минимум технического ВУЗа --- необходимость*

Я не то чтобы спорю с этим утверждением, но - ты привел 3 совсем разные роли. Каков твой опыт работы в любой из этих ролей? На основании чего судишь?

*> потому что ум в нужный порядок приводит и учит правильно думать*

С этим никто не спорит. Но, может быть, не только математика приводит ум в порядок?

*> PS: Будущих code monkeys обычно видно уже во время обучения*

И их признаками является что? Хочу проверить на себе.

>И их признаками является что? Хочу проверить на себе.

1) Излишняя самоуверенность и самодовольство;
2) Бездумный copy/paste (особенно распространено в среде халявщиков);
3) Преобладание местечково-эмпирических методик получения результата над фундаментальным теоретическим обоснованием решения задачи;
4) Нездоровая алергия к конкретным продуктам, технологиям, средствам разработки, языкам программирования (типа "этого X не знаю и не люблю, а потому этот X - отстой");
5) Шаблонность мышления, затруднения при решении нестандартных задач.

В последнее время наблюдается тенденция к появлению новой группы code/PR monkeys aka внештатных агентов мелкомягких.

*> То бишь code monkeys.*

Обычным кодерам тоже ДМ нужна, хоть в каком-то объеме. Взять довольно-таки часто встречающийся пример - разбор регулярного текста (например, разбивка на лексемы). Человек, знающий ДМ построит простой и понятный КА, в то время как незнающий начнет лепить кучу условий и кучу проверок, в результате чего запутается в этом и сдохнет-таки от всех этих взаимосвязей, взаимовлияний и т.п. То же касается графов, множеств и пр. и пр.

**Re: Нужность дискретной математики**

*> 1) Излишняя самоуверенность и самодовольство;*

Свойственно всема молодым

*> 2) Бездумный copy/paste (особенно распространено в среде халявщиков);*

Мачит казлов

*> 3) Преобладание местечково-эмпирических методик получения результата над фундаментальным теоретическим обоснованием решения задачи;*

Хлестко сказано, но в реальности нет времни на "фундаментальное теоретическое обоснование", и эмпирика (местечковая или еще какая-нибудь) рулит. Более того - если ты уперся в задачу, которая требует "фундаментального теоретического обоснования", ты должен обратится к специалисту по фундаментальной теории, которым ты по определению не являешься.

*> 4) Нездоровая алергия к конкретным продуктам, технологиям, средствам разработки, языкам программирования (типа "этого X не знаю и не люблю, а потому этот X - отстой");*

аЛЛергия. В остальном - пожалуй, верно.

*> 5) Шаблонность мышления, затруднения при решении нестандартных задач.*

Меру надо знать. Принцип KISS помнить.

****

**Re: Нужность дискретной математики**

s/НEжность/Нужность ;)

**Re: Нужность дискретной математики**

*> Есть здесь те, кто программирует 15-20 лет и больше? Интересно было бы услышать их мнение.*

На самом деле нужно и то и другое. И даже третье.

*>> Будущих code monkeys обычно видно уже во время обучения в университетах/институтах.*

*> по каким признакам*

Обычно очень любят работать и очень не любят учиться.

*>>> Будущих code monkeys обычно видно уже во время обучения в университетах/институтах.*

*>> по каким признакам*

*>Обычно очень любят работать и очень не любят учиться.*

Тогда я, пожалуй, и есть code monkey

**Re: Нужность дискретной математики**

*> Тогда я, пожалуй, и есть code monkey*

Suum quique

Однажды некто, прогуливаясь по лесу, увидел дровосека, который пилил дерево тупой пилой. Прохожий обратил внимание дровосека, что не мешало бы немного заточить пилу. На что дровосек ответил: "Мне некогда заточить пилу. Мне нужно пилить!"

Ну или, в более вульгарном варианте: "А чего тут думать? Трусить надо!" (C)

Несомненно, что худо быть полным недостатков; но еще хуже быть полным их и не желать сознавать в себе, потому что это значит прибавлять к ним еще и недостаток самообмана. ( Блез Паскаль )

Вот вам задача: у вас есть жесткое ограничение памяти. Дан односвязный список. Надо за конечное время (при ответе дать оценку стоимости) определить, является ли список зацикленным.

Попробуйте решить эту крайне типичную в программировании задачу без дискретной математики.

Вторая задача - раз уж вы web-программированием увлекаетесь: есть workflow - набор страниц и передаваемое между ними состояние (для конкретики - допустим, это система бронирования авиабилетов). Требуется доказать, что нет таких переходов, при которых состояние теряется, или же выявить те переходы, в которых целостность нарушена. При этом учесть, что часть состояния хранится в реляционной базе данных.

Решить без дискретной математики, хи хи хи. Решить всё тестированием - не предлагать. Заказчик уже кипятком мочится, и если вы ему бракованный продукт подсунете, то больше у вас заказчиков не будет вообще, по миру пойдёте.

цитата:

*и "программист", не знающих на отл. ДМ - не программист вообще.*
Хм, даже не знаю что сказать. Беседовал с программистами, 7и кто про дискретку не вспоминает, все говрят "да был такой предмет"

Это не программисты. Это кодеры. Они ничего не умеют. Они не нужны индустрии. Труд этих обезьянок стоит очень дёшево, и им никогда не достичь высокооплачиваемых ролей.

 **st83**:
*Интересно, кто и когда умудрился эту мерзость напихать детям в головы? Ведь не сами же они до такого додумались!*
Сами. Уже со второго курса все становится ясно.

А скажите, кто вы такой, чтоб вам было "всё ясно"? Неужели вы считаете себя умным?!?

Разве вы не знаете, что "всё ясно" может быть только тому, кто отлично знает дискретную математику. Да и умным человеком можно быть только зная дискретную математику - как минимум, такой её раздел, как логика.

Без логики "всё ясно" может быть только как в анекдоте про йога: типа, сидел великий йогский гуру 20 лет в позе лотоса перед кирпичной стеной, медитировал. А когда вышел из транса, встал и сказал: "так, ну теперь со стеной всё ясно!".

цитата:

**st83**:
Прикладная Теория Цифровых Автоматов. Блин, на кого я учусь кто нибудь ябьяснит? Шел на программиста но чем дальше тем более рпихожу к сведению что из меня хотят сделать цифоровго схемотехника Но елки-палки есть же соседняя специальность где готовят как раз их, программировать Pic и другие контролееры и строить комплексы с их учатисем там же.

Вы даже не догадываетесь, что web-приложение - это тоже часто автомат. И что без знаний теории автоматов вы не сможете сложные вещи делать просто. Про качество ваших будущих поделок и вовсе скромно промолчим - вы ведь не сможете никогда гарантировать даже целостность. Даже не сможете доказать, что ваше решение покрывает все случаи использования, о которых просил заказчик.

21062009

by [Jack](http://next2nothing.ru/author/Jack/)

[математика](http://next2nothing.ru/tag/%D0%BC%D0%B0%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0/), [экзамен](http://next2nothing.ru/tag/%D1%8D%D0%BA%D0%B7%D0%B0%D0%BC%D0%B5%D0%BD/)

# Математика для сачка



Знаете, давным-давно, после того как я получил первую двойку по теории вероятности я написал статью про то, как сдавать математические дисциплины. Эта статья была опубликована на сайте [www.sachok.ru](http://next@nothing.ru/go.php?http://www.sachok.ru) и в одноименном бумажном журнале (в сокращенном варианте, конечно).

Недавно, перебирай свой электронный хлам на диске, накнулся на эту статью 2003 года и сразу вспомнил тот первый опыт авторства, те первые 400 рублей гонорара за статью и три номера этого журнала с моей статьей…

Решил ее тут опубликовать, так как не удалось найти ее на “сачковском” сайте. Вдруг кому пригодится мнение о процедуре сдачи высшей математики из 2003 года..

Итак, полный текст статьи читаем ниже

### Математика для сачка

У тебя еще не болит голова от словосочетания высшая математика? Тебе не снятся кошмары от предвкушения того, что скоро экзамен по алгебре? Ты усердно учился весь семестр, поэтому интегралы Лебега-Стильтьеса тебя не пугают? На практических занятиях ты успешно решал задачи по теории вероятности, в результате чего получил автомат по этому “приятному” предмету? Тогда можешь расслабиться и не читать все, что написано ниже – у тебя проблем с математикой не будет. Особых проблем. А вот если ты не знаешь, как выглядит лектор, у тебя нет ни одной лекции по предмету, ты не имеешь понятия, как решать комбинаторные задачи (несмотря на то, что экзамен уже дышит в спину), то внимательно читай, что я тут тебе скажу :)
Итак, большинство проблем с математикой возникает именно из-за того, что нам просто лень сидеть и учить. Иногда настолько лень, что о том, что надо сдавать экзамен, мы вспоминаем непосредственно за 3 дня, отведенных на подготовку к нему. Вот тут-то и появляется проблема: как его сдать. Тем более тройку получать не очень и хочется…

Что нужно для того, чтобы не завалить высшую математику (алгебру, теорию вероятности, геометрию, дискретную математику и остальные разделы этой науки)? Что нужно для того, чтобы за 3 дня овладеть годовым курсом? И что нужно знать, чтобы получить максимальную оценку, при минимуме знаний? То есть: “неудовлетворительные знания – на хорошо и отлично!”
Поехали.

#### Что нам понадобится?

1. Лекции препода. Желательно того, которому будем сдавать.
2. Учебная литература. Учебники, методички, сборники задач, лекции других курсов, других преподов – одним словом все, что можно добыть у дружелюбных товарищей и в библиотеке. У кого с этим возникнут проблемы – поищите в нете: порой много интересного можно накопать по теме. Хотя по математике маловато лекций…
3. Время для подготовки. От 1 до 3 полных дней. Если времени больше трех дней, то материал учится еще лучше…
4. Список вопросов к экзамену.
5. Желание получить оценку, отличающуюся от “неудовлетворительно”. Это самое главное.

Ну вот. Все, что написано выше, у нас есть. Теперь общая методика “обмана” препода.

#### Как будем учить – методика.

Как правило, экзамен по математике состоит в билете, в котором содержится 2 теоретических вопроса. Иногда в билет пихают задачку, которая, как правило, решается на практических занятиях. Поэтому вывод: тетрадку с ПЗ все-таки желательно иметь.
Теоретические вопросы – это или доказательство теоремы, определения понятий, свойства чего-либо. Здесь нужно заметить важную особенность. Есть определения, формулировки теорем, свойств. А есть доказательства теорем и всего остального. Ввиду отсутствия времени, про выучивание доказательств придется забыть.
Про задачи и семинары. Решать задачи не научишься за три дня – как бы ты не старался… Поэтому прорешивать все задачи – бессмысленно. Тем более что время поджимает. Мы пойдем другим путем! :)

#### Что НУЖНО выучить наизусть:

1. Определения
2. Формулировки теорем
3. Свойства и различные понятия
4. (на “5”) Примеры и типовые задачки.

#### Что учить необязательно:

1. Доказательства теорем.
2. Задачи с номера n по номер m, где m>>n. Или запоминать, как решать конкретные задачи из задачника.

#### Как выучить определения?

Система такая: 2 тетрадки (или куча листочков) – в одной пишем все определения и формулировки, вместе со свойствами; в другой – номер билета и список определений, формулировок, которые нужно знать к этому билету. Тетрадку с определениями выучиваем – нужно неоднократно написать каждое определение и теорему на черновике, чтобы запомнить до мелочей! От этого зависит твоя оценка на экзамене. Желательно выучить все это за день. За первый день. Или раньше.

#### Что потребуется дополнительно:

1. Шпоры. Самая важная часть экзамена по математике. В них должны содержаться основные определения, формулировки и ДОКАЗАТЕЛЬСТВА ТЕОРЕМ!!!
2. Как альтернативу шпорам – бомбы (заранее заготовленные листы формата А4, на которых подробно изложен ответ на заданный вопрос билета.) Соответственно – номера на бомбах проставляются карандашиком, чтоб потом стереть. Обратите внимание на то, что бомбы пишутся ручкой одного цвета, и желательно одинаковым почерком. Были случаи, когда препод увидел бомбу на столе готовящегося, нашел там ошибку и попросил ее исправить (видимо, пытаясь помочь бедному бомбильщику). А несчастный написал на бомбе ручкой другого цвета, за что и был выгнан с бананом. :(
Я предпочитаю использовать шпоры. Хотя бы потому, что листочки, которые дают для подготовки, могут содержать роспись препода, подделывать которую я считаю грязным делом – зачем злить человека? Он свою подпись узнает и на тебя “обидится”… И не простит никогда… Разве только после того, как спросит у тебя доказательство Великой Теоремы Ферма. И после того, как ты ему ее не докажешь – с улыбкой попросит тебя приходить в следующий раз (скажу по секрету, если ты попадешь на пересдаче к этому преподу, то вероятность того, что снова получишь двойку – 60-90%, в зависимости от характера препода.)
3. Удача на экзамене. Она зависит от многих причин: счастливые вопросы, халявный препод, удачное списывание, удачные наводящие и дополнительные вопросы…

#### Итак, поехали…

Пишем шпоры, понимаем доказательства.
Потом усиленно читаем шпоры (бомбы) и понимаем все доказательства в них. Если шпор нет и бомб нет, то их нужно написать в срочном порядке. Хоть ночью сидите! Практика показывает: за час можно написать около 4-6 бомб, и около 3-4 шпор. Это при условии, что процесс написания совпадает с осознанием и пониманием написанного. Писать шпоры автоматически нежелательно – можно потом не разобраться на экзамене. Но обычно наша группа пишет шпоры на А4, уменьшает их до разумного размера, путем сканирования с соответствующим разрешением и распечатывает на лазерном принтере. Поэтому у желающих всегда есть шпоры заранее.. После того, как выучены определения и готовы шпоры, которые поняты и осознаны – наступает шлифовальная работа: открываешь билет и смотришь, что там нужно говорить. Все это называешь – переходишь к следующему. И вразнобой. На все это, при условии быстрого вспоминания определений и формулировок, уходит не более 3 часов (чтоб несколько раз прогнать). Это нужно для защиты от атак препода вида: “расскажите мне теоремку из этого вопроса. А какие там еще теоремы есть?” Когда ты ему начинаешь по этому вопросу впаривать заученные определения, то он невольно начинает думать, что тебе действительно не слабо было выучить весь материал, и что шпорами ты не пользовался. Так что обрати внимание! Остается не забыть запомнить типовые методы решения задач из методички или сборника задач. И попытаться почитать лекции или учебник. Подготовка закончена.

#### Стратегия сдачи самого экзамена

Как проходит экзамен? Ты получаешь билет, тихо офигеваешь от вопросов в нем, идешь готовиться. Достаешь шпору или бомбу. Скатываешь или подменяешь ответ на вопрос. Точнее, как обычно, на два вопроса. Потом идешь отвечать преподу. Втираешь ему скатанное. Отвечаешь на дополнительные вопросы и решаешь предложенные задачки. И протягиваешь ему зачетку, чтобы он нарисовал там пятерочку. Он рисует и ты, счастливый, уходишь.
Теперь подробнее. После получения вопроса – ты садишься туда, где легче списать или легче достать бомбу. О том, как списывать, учить не буду – слишком большая тема, да и сам уже умеешь. Хотя….

#### Как правильно списывать

Интересный способ списывания со шпор: за 3 года обучения ни разу не запалили. Шпоры кладутся в носовой платок, как бы по разделам: десять шпор в один раздел сложенного носового платка. Иногда требуется 2 платка. В этом случае влезает около 60-70 шпор в оба кармана. При доставании шпоры делаешь вид, что сморкаешься, тем временем ища нужный вопрос. Хватаешь его кончиком пальцев. В то время, как убираешь носовой платок обратно – вытаскиваешь шпору и, пряча ее в ладони, кладешь руки на стол… Это нехитрое действие абсолютно естественно (попробуй высморкаться и обратить внимание на последовательность действий при этом процессе) Соответственно – эффективно.

Если ты спишешь, то можно сказать, что приведенная здесь система даст тебе шанс получить пять. Если не спишешь, то все будет зависеть от случая и от препода. Многие не особо смотрят на основной вопрос, а гоняют по определениям и задачам. А некоторые, увидев отсутствие доказательств – ставят 2.

Надеюсь, все списано и лежит перед твоим носом. Отлично. Теперь заметай следы (прячь шпоры обратно) и запомни, что ты ничего не списывал! Нужна уверенность в своих знаниях! Препод сам знает, что он все знает, а ты ничего не знаешь… Поэтому будем строить из себя человека, который не ЗНАЕТ, а ПЫТАЕТСЯ все знать, старается в этом вопросе, добросовестно учит. Отсюда – НИКОГДА не говорите преподу, что вы все знаете, а он не прав. Вы учили, все помните, но вылетело из головы… :)

Вот сидишь ты перед написанным и ни фига не понимаешь. Плохо. Совет: лучше не написать ничего, чем написать то, чего объяснить не сможешь. Поэтому внимательно врубись в написанное (т.е. в списанное) и перепиши (в крайнем случае) то, что ты понимаешь. Маленькое замечание: за три дня подготовки – ты должен ПОНЯТЬ все, что написано на шпорах. Лучший способ – на черновике написать основные доказательства и понять, что и откуда следует. Но все это после того, как ты выучишь определения и формулировки (хотя можно и параллельно учить определения и понимать доказательства). Потому что ты быстрее врубишься, когда при ссылке в доказательстве увидишь название какой-нибудь теоремы. Согласись, что поиск теоремы отнимает больше времени, чем поиск у себя в мозге… Заодно это и проверка твоих знаний определений и формулировок.. Если при подготовке ты понял ход доказательства, понял, откуда что взялось – то можешь его забывать на время и переходить к следующему. Если не понял, то можно или забить, или попытаться почитать другую литературу на данную тему. Выбор за тобой. Чем больше ты поймешь, тем выше вероятность пятерки на экзамене.

#### Как убедить препода, что ты не списывал.

Допустим, что ты списал и понял все, что написано. Программа минимум – выполнена. Можно, в крайнем случае, идти сдавать (напомню, ты сидишь на экзамене). Но если есть время, то нужно еще кое-чем заняться. Преподы – такой народ, что знают, когда ты списал, а когда нет. Почти на 100%. И если препод хороший, то он закрывает на это глаза… А если плохой (непереводимый поток матных слов и выражений в адрес нехороших преподов), то … просит тебя написать все это еще раз. Перевернув твой листочек. Приемы борьбы с этими редисками: после того, как ты понял то, что у тебя написано – надо выучить доказательства. Определения и формулировки ты уже знаешь. Так что минут за 20 можно запомнить с точностью до буковки все то, что написано у тебя на листочке. Теперь ты готов. Теперь либо ищи халявного препода и иди к нему, либо сиди и вспоминай какие-нибудь красивые примеры, задачи на доказанные теоремы и используемые свойства, чтобы потом продемонстрировать свою эрудицию.

#### Как вести себя с преподом

Тебя вызвали и ты у препода. Ответ должен быть спокойным, уверенным (у тебя все написано и ты все запомнил), не наглым (не надо лишний раз убеждать препода в том, что ты не списывал, он ведь может и стать плохим преподом, а нам этого не надо). Аккуратно втираешь преподу написанное, следя за его реакцией. Психологический момент: если препод видит знакомые буковки, он соглашается и кивает головой. Если ты замечаешь его согласие с написанным тобой, то можно повторить это еще один раз другими словами (в гуманитарных науках это называется “лить воду”). Если препод хмурится, то проскакивай лишние слова и прорывайся дальше, к следующей теореме или к следующему вопросу.
Если грамотно втирать материал (лектору желательно втирать его собственные лекции), то препод не будет задавать дополнительных вопросов и поставит 5. Если он начал задавать вопросы (в моей практике никто из моих знакомых не оставался без дополнительных вопросов), то они могут быть 2-х видов: по билету и по курсу.

#### Дополнительные вопросы и как с ними бороться

1. По твоему билету. Поздравляю – ты залетел. Сморозил глупость и сейчас будет умственный секс. Препод всегда прав, и спрашивающий тебя препод оказался садистом – он сначала выслушал, а теперь начинает издеваться над тобой, пытаясь показать тебе ошибки. Обычно в этом случае все впадают в ступор и молчат. Кстати – один из неплохих выходов, чтобы не сказать еще большую глупость, чем ты написал. Помни, что редкий препод будет говорить неверное, чтобы ввести тебя в заблуждение (никто не станет убеждать тебя в том, что в теореме Пифагора вместо равенства – неравенство, т.е. сумма квадрата гипотенузы НЕ РАВНА суме квадратов катетов). Его вопросы будут направлены на понимание тобой того, что ты написал. Поэтому попытайся не показать того, что ты плохо знаешь то, что ты написал. Если ты действительно понимаешь то, что написано, то, думаю, таких проблем не возникнет. В любом случае, скорее всего, твой ответ не будет ниже тройки в этом случае.
2. Вопросы по программе. ПОЗДРАВЛЯЮ! Препод согласился с тем, что свой вопрос ты знаешь хорошо. Теперь о подобных дополнительных вопросах. 99% подобных вопросов – формулировки теорем, определения и основные свойства. Это ты выучил! Поэтому смело отвечай на любой вопрос. Если забыл – то с умным видом скажи преподу “Что-то вылетело из головы, сейчас я вспомню”. И начинай на листочке рисовать символы, относящиеся к этой области. Если забыл надолго, то жди, когда преподу надоест ждать и он спросит: “Ну что?”. Тут с виноватым видом отвечай, что вылетело из головы, наверное, из-за волнения и т.д. и т.п. Маловероятно, что тебя попросят что-то доказать. Если попросят, то должны дать время на раздумье. Тут или списывай снова (что опасно), или пытайся придумать доказательство или вывернуться как-нибудь… Преподы – тоже люди. Они также прекрасно знают, что никто из студентов не знает математику лучше их :) Иногда бывают случаи, когда препод сам объясняет тебе то, чего ты не написал или “не помнишь” (точнее, не знаешь). Сделай вид, будто ты это вспоминаешь по ходу его объяснения. Можно даже пытаться поддакивать по ходу :) Но осторожно: лучше молчать, чем сказать не в тему…
3. Отдельный прием некоторых злобных преподов. По данным студенческой разведки – используется на мехмате. Но преподы взаимообучаемые, и общение с преподами мехмата на других преподов действует не всегда положительно для студентов. Принцип вопросов такой: задаются вопросы почти по всему курсу. Сначала легкие, потом сложнее и сложнее… Как только студент начинает долго думать над ответом, препод психологически давит на незнание этого ответа и задает следующих. Студент привыкает, что вопросы из этой сложной группы он не знает. Вот тут препод и задает очень легкий вопрос, на который студент часто отвечает неверно, почувствовав знакомые символы и слова и поторопившись дать ответ. Препод нагло и довольно улыбается и говорит, что никакой пятерки он получится. Методы борьбы. Ты танк. Большой и тяжелый. Пусть немного тугодум, но дающий правильные ответы на все вопросы (не тормоз, а просто медленный газ). Когда ты над каждым вопросом начинаешь думать, препод либо будет тебя торопить, либо плюнет на эту методику и поставит тебе пять! Если будет пытаться торопить – ничего страшного. Не надо вслух произносить определения и формулировки – напиши их лучше не листочке. Пишешь медленнее – есть время подумать. А придраться к плохому знанию – сложно: ты же пишешь все-таки, и все правильно, ошибок нет… Так что удачи!
4. Задачи. Вот ты и попал. Получив задачу, не надо паниковать. Вспомни все теоремы (формулировки), которые ты учил и которые относятся к задаче. Вспомни те примеры типовых задач, которые ты можешь вспомнить (при подготовке обязательно просмотри задачник и методичку. Обычно перед началом каждой главы или темы есть общие способы решения задач. Их тоже НЕОБХОДИМО ВЫУЧИТЬ!). И решай задачу. Если перед преподом решаешь, то полезно показывать твое знание формулировок теорем, можно их выписать. Реализовать видимость умственного напряжения и т.п. Порой преподы сами помогают решить задачу, а ты опять же соглашаешься с методом решения. Только тут, в отличие от объяснения преподом тебе твоего вопроса, не надо делать все время согласное лицо, сделай вид, что усиленно пытаешься понять, как решить эту задачу по методу препа. Можно даже аккуратно задавать вопросы. Если ты задашь вопрос вида: “а можем ли мы так делать? Ведь по теореме такой-то у нас должно быть это и это. А у нас ВРОДЕ не выполняется этого и этого”. Если попадешь в десятку, то уйдешь с 5 или 4. В любом случае, ты почти закончил экзамен, но не расслабляйся!

#### Как повысить оценку

После того, как препод окончательно тебя домучает, он поставит тебе оценку и отпустит. Если ты видишь, что препод ставит не ту оценку, что ты хотел, можно попытаться поднять ее на балл, если ты считаешь свой ответ не совсем ужасным (осторожно! У всех разное самомнение! Оценивайте себя критично!). Типичные фразы “А на 4 (5) нельзя никак?”, “А может, Вы зададите мне еще какой-нибудь вопрос, чтобы зачетку не портить” и т.д. Мои знакомые таким образом добивались троек в клиническом случае – почти пустой лист бумаги, полное незнание формулировок и определений! Действовали по принципу: матан (или что другое) – большой, что-нибудь да знаю! В результате, все будет зависеть от твоего знания определений. Преподы спрашивают только их и задачи. Хотя я сдавая алгебру, получил задачку – доказать теоремку, которой не было в курсе, но она выводилась из других. 4 раза предлагал преподу доказательство, по ходу пришлось доказать теорему из билетов, а он каждый раз находил какую-нибудь мелкую ошибку и отправлял меня искать новый вариант. За 4 часа доказательства мозги вскипели, но алгебру я сдал :)

И напоследок. Есть такое понятие в математике, как “схватить клина”. То есть на простые вопросы ты не можешь ничего ответить. Клинит на экзамене. Поэтому не рекомендую заниматься после 9 часов вечера перед экзаменом. Ложитесь спать и не забудте съесть утром шоколадку для активации мозговой деятельности. И все будет отлично! Удачи!
Если возникнут вопросы по статье и иного рода вопросы – постараюсь помочь.