

# Бот план (Компиляция постов Поступашек)

[https://t.me/postypashki\\_old/584](https://t.me/postypashki_old/584)

[https://t.me/postypashki\\_old/861](https://t.me/postypashki_old/861)

<https://academy.yandex.ru/handbook/python>

## Шаг 0.

*Посвятим его освоению "олимпиадных тем": комбинаторика, теория чисел, принцип Дирихле, графы.*

1. "Ленинградские математические кружки" С. А. Генкин.  
Книжка моего детства, ставшая классикой, давно красуется на гербе "Поступашек" в лапке у пиплапа. Разбираем все.
2. Problems.ru  
(<https://problems.ru/>)К несчастью, ЛМК не покрывают такие темы, как Оценка+Пример, Алгоритм Евклида, что восполняет проблемс своими соответствующими разделами.

## Шаг 1.

### Теория множеств.

1. "Элементы теории множеств" Н. К. Верещагин, А. Шень.  
Разбираем только 1.1-1.6
2. "Рассказы о множествах" Н. Я. Виленкин.

### Математический анализ.

1. Первый семестр курса (<https://teach-in.ru/course/calculus-shaposhnikov-part1?ysclid=I59kjsx8xn293785650>) С. В. Шапошникова.  
Советую внимательно отнестись не только к определениям, но к

доказательствам, ибо конструкции оттуда используется при решении задач. По этой же причине советую время от времени поглядывать за примерами в Зорича, "Контр примеры в анализе".

2. Семинары ([https://vk.com/id19160711?w=wall19160711\\_2022%2Fall](https://vk.com/id19160711?w=wall19160711_2022%2Fall))

первого семестра А. Сукабичевского.

[https://disk.yandex.ru/d/ZViQIMLaJqFUFg/%D0%BC%D0%B0%D1%82%D0%B0%D0%BD\\_1%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80](https://disk.yandex.ru/d/ZViQIMLaJqFUFg/%D0%BC%D0%B0%D1%82%D0%B0%D0%BD_1%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80)

Разобраны все основные задачи стандартного курса.

3. "Математический анализ в задачах и упражнениях" том 1 И. А.

Виноградова и другие.

Разбираем только главы 1-5. Помимо кучи задач, в том числе теоретических (к несчастью, искусственных) есть теор материал с примерами решения основных задач.

4. Листик по монотонности от М. А.

В стандартном курсе анализа редко разбираются задачи на монотонность, хотя они встречаются даже в абитуриентских олимпиадах.

5. "Комбинаторика" Н. Я. Виленкин и другие.

Смотрим рекуррентные соотношения.

## **Алгебра.**

1. Семинары (<https://teach-in.ru/course/algebra1-seminars-kanunnikov?ysclid=I59kpr95rn960649268>) А. Л. Канунников.

Теория по стандартному курсу элементов алгебры вас больше запутает, в ней нет ничего интересного— только справочный формализм. Поэтому разбираем только семы и задачи.

2. "Перестановки: все что вы хотели знать о них, но боялись спросить"

Егор Ясинский

Название говорит само за себя.

3. "Сборник задач по алгебре" А. И. Кострикин

Думаю вам вполне хватит Канунникова, но на всякий случай.

4. "Алгебра и теория чисел для математических школ" Н. Б. Алфурова, А. В. Устинов  
Смотрим раздел многочлены.
5. 3Blue1Brown <https://youtube.com/playlist?list=PLVjLpKXnAGLXPaS7FRBjd5yZeXwJxZil2>

### **Аналитическая геометрия.**

Как таковой в ШАДе нет, а что есть решается проекцией линала на  $R^2$ ,  $R^3$ . Но геометрией полезно осязать линал малых размерностях, да и поможет в следующих курсах.

1. Семинары (<https://teach-in.ru/course/analytic-geometr-seminars-penskoy>) по ангеу А. В. Пенской.  
Разбираем семинары 1-14.
2. "Сборник задач по аналитической геометрии и линейной алгебре" Смирнов Ю. М.  
По нему идет Пенской, решаем соответствующие темы.

### **Алгебраические структуры.**

Группы, кольца, поля и т.д. встречаются только на уровне определений.

1. Курс (<https://teach-in.ru/course/algebra-arzhantsev>) лекций И. В. Аржанцева.  
Смотрим соответствующие лекции. На пальцах объяснены смежные классы и теорема Лангранжа
2. "Теорема Абеля в задачах и решениях" В. Б. Алексеев.  
Куча простых задач на теорию групп, решаем одну за другой— и понимание не заставит себя ждать. Все в лучших советский традициях: только определения и задачи. К несчастью, капиталисты в конце поместили решения((
3. "Введение в алгебру. Часть 1" А. И. Кострикин.  
Листаем на уровне определений соответствующую главу про залупы, кольца, поля.  
Также будет полезно посмотреть How to заботать алгебру и аналитическую геометрию ([https://t.me/postypashki\\_old/716](https://t.me/postypashki_old/716)).

# Алгоритмы и структуры Данных

## Python, но лучше учить плюсы, потому что большую часть кода будешь писать на плюсах

1. Хороший [учебник](#) для знакомства с базовыми возможностями языка.
2. [Сосайт](#), где разобраны все возможные вопросы, есть книги и даже приведено руководство по Numpy 🤖🤖

## С++. Компиляция планов по боту [C](#) и С++

### Шаг 0.

Лучше ботать Сишку в её естественной среде обитания, которой является **UNIX** (на нем она и написана). Так как далеко не все версии ников доступны и просты, ставим (сравнительно) несложный дистрибутив **Linux** (Ubuntu, Fedora, Mint, Debian, не Arch и не Gentoo). Ко всему, эти системы распространяются под свободными лицензиями, поэтому можно получить бесплатно (это та самая халява, которую зовут физтехи перед НК). Вообще это профессиональный стандарт, причём не только для разработчиков.

1. Ставим Линукс, привыкаем к нему, любим его;
2. Читаем и практикуем по жизни всю книгу *А. В. Столярова* (под боевым псевдонимом Столяр) по **UNIX** (кроме главы *"Инструментарий программиста"*). Можно вместо этого читать его же *"Программирование: введение в профессию", том 1*, но лишь главы 1.1-1.2. Автор хоть и неисправимый сторонник Паскаля на 1 курсе, но пособия годные, поэтому разобраться на уровне продвинутого пользователя в Линуксе можно;
3. Полезно пописать скриптики. Скорее всего, будет достаточно повседневного использования и заданий *"Командная строка Unix"* из задачника Столяра
4. Освоить **Git** и **GitHub** (это разное!). Для начала хватит связать их по **SSH** и понять, что и как с **push, pull, add, commit, branch, checkout, merge...**

5. Устройство компьютера и общие сведения из **Computer Science**.

В данном случае вполне подойдут книги *Владстона Фило* серии *"Теоретический минимум по Computer Science"* (сначала первая, зелёная, потом, возможно, остальные, пока их две). Также подойдут главы 1.3-1.4 из всё той же книги *Столяра*. Первые книги представляют из себя чуть-чуть теории с примерами на задачах, похожих на реальные. Последняя чуть-чуть перегружена математикой. Начинать читать их стоит не сильно раньше главы 11 в *Прате*;

6. Будет очень хорошей идеей научиться элементарной отладке программ с помощью **gdb**, **valgrind** и **address sanitizer**. Ещё лучше познакомиться с **GUN make** и **Doxygen**. Всё это понадобится как на **Си**, так и в **"Плюсах"**... И много где ещё.

## Шаг 1

### 1\*. Стивен Прата, "Язык программирования C++", 6-е издание.

Как и все книги Праты, прекрасна и хорошо подходит даже для первого знакомства с программированием. Материал изложен очень подробно и понятно. Как всегда, за это надо заплатить чем-то, поэтому объём велик: 1200+ страниц. А о цене даже не спрашивайте — **дорогая, очень**. Читать всю, решать тоже всю. Можно пропустить, если времени совсем уж мало и есть уверенность в своём background;

### 2. Андрей Столяров, "Введение в язык Си++", 5-е издание.

Читать следует аккуратно, чтобы **не заразиться взглядами автора**.

Рассчитана на переход с языка C со значительным на нём опытом. Очень хорошо объяснены многие принципы языка, уделено внимание подходам ООП и АТД. Нет рассмотрения стандартов, что очень хорошо, так как язык стал очень уж "спорным" (что бы ни говорили школьники 10 cип'ов из 10 coul'ов). Особенно ценно то, что обращено особенное внимание на некоторые типичные ошибки и заблуждения;

### 3. Бьерн Страуструп, "Язык программирования C++", 2-е издание (1991 год оригинал; 1993 год на русском, Диасофт).

Да, от создателя языка. **Читать ещё аккуратнее предыдущей**, так как выпущена даже до стандартизации самого языка. Последнее подходящее

для новичка издание этой книги, где понятным языком изложено всё, что необходимо, как это работает, зачем оно есть и как пользоваться. Как бы ни была стара эта книга, альтернатив ей нет и не предвидится. Стоит прочитать всю, в том числе предисловия, и решить упражнения. Некоторые источники из указанных в ней также вполне полезных и актуальны в наши дни. Найти проблематично даже б/у, особенно оба тома сразу (она в двух частях).

Кроме знания самого языка и основных практик разработки на нём, нужно не забывать о том, что когда-нибудь, рано или поздно, придётся всё-таки заняться разработкой в контексте профессии, а это тоже надо заботать.

### **1. Брайан Керниган, Роб Пайк, "Практика программирования".**

Если не читали ранее — прочитайте. Есть достаточно много чисто практических рекомендаций. Есть как общие для разработка, так и специфичные для языка моменты. Уделено достаточно много внимания читаемости, оптимизации, отладке, проектированию, масштабируемости, переносимости, сопровождению и т. д. Написано в очень хорошем техническом стиле, всем бы такой;

### **2. Чед Фаулер, "Программист-фанатик".**

Описано в "лирическом" стиле как правильно **работать на дядю**, почему именно **за копейки** и как при этом быть счастливым в общечеловеческом смысле. Своеобразное руководство "Как не продолбать жизнь впустую". Как лёгкое чтение подойдёт прекрасно, причём не только для C++. Содержит как советы, так и мотивационные моменты.

Кроме того, есть хорошие материалы для обучения в формате видеозаписей лекций и конспектов к ним. Автор крайне не советует использовать их без предварительного/последующего чтения текстовых.

Нетрадиционные материалы:

### **1. Константин Владимиров, лекции в МФТИ.**

Уникальный товарищ, который разбирается в C++ на уровне разработки трансляторов (и не только) в крупных организациях;

## 2. Тимофей Хирьянов, лекции в МФТИ.

Очень грамотный лектор, лекции интересные и полезные;

## 3. Илья Мещерин, лекции в МФТИ.

Смотреть и читать конспекты от КТЛ осторожно и не очень спешить с ними;

## 4. Александр Чернов, Андрей Шестимеров, семинары в МГУ.

Полезные практические занятия, есть также замечания по инструментам разработки;

## 5. [cppreference.com](http://cppreference.com).

Отличный электронный справочник по языку и стандартной библиотеке различных стандартов. Советую использовать его во всех целях.

## Toolchain.

Выбор не столь важен до какого-то момента, но сложен, так как правильность познаётся лишь со временем в сравнении с другими. Советы не претендует на полноту и единственность. IDE на этом этапе лучше не использовать из-за перегруженности интерфейса и требовательности к системе при весьма сомнительных преимуществах для обучаемого. Скорее даже недостатках. В качестве ОС подразумевается всё тот же Linux, что позволяет перенести все навыки обращения с ним и его инструментами из прошлого.

## 1. Текстовый редактор.

Код будет набираться именно в нём, поэтому выберите любой удобный. Это не обязательно **vim** или **emacs**, хотя это круто. Нужно немного: моноширинный шрифт и подсветка синтаксиса. Сердца пользователей в этом вопросе украл **Sublime Text**;

## 2. Компилятор.

Любой популярный, так как это упростит решение возникающих проблем при помощи гуглинга и снижает количество ошибок в самой системе трансляции. Скорее всего, это будет **g++** или **clang++**;

### 3. Система автоматизированной сборки.

Здесь стоит оставить систему GNU make. CMake реально не упрощает и не ускоряет подготовку Makefile. Сразу же научитесь писать хорошие мейкфайлы;

### 4. Отладчик.

Очевидно, **gdb** или **lldb**;

### 5. Профилирование.

Всё, что полезно, используйте. Лучше уметь пользоваться **valgrind**, **gprof**, **strace**, **ltrace**, **perf**;

### 6. Система контроля версий.

Любая. Но ничего изобретать не стоит, подойдёт **git** в связке с каким-нибудь **GitLab** (в отличие от GitHub, он является свободным и его даже можно поднять "для своих" хоть дома).

Что читать не надо. В топку. Есть такие книги и ресурсы, которыми в пору печку топить, а не учиться по ним. Кратко пробежимся по ним.

#### 1. Каналы на YouTube.

Да, Гоша Дударь не идеален, как и Хауди Хо. А вы думало, они из любви к человеку делают контент? Нет;

#### 2. Stepik.

Кто-то скажет, что на Физтехе, в ВШЭ, МГУ или какой-то другой помойке его используют. Но там это делают только безумцы;

#### 3. Приложения в стиле "учимся программировать самостоятельно".

Как правило, содержат краткие текстовые введения или видеоуроки и сразу после них бесполезные задания на 10-20 строк. Более обоснований не требуется;

#### 4. Книги Шилдта, все.

Автор делает огромное количество заявлений, несоответствующих стандарту. Этим всё сказано;



**5. "Выучи самостоятельно", "за 21 день", "как написать свистелку", "весь язык за 3 минуты", "для чайников".** Вряд ли кто-то в здравом уме будет рассматривать их всерьёз, но некоторые даже учатся по ним;

**6. Буквально любая книга по [ссылке](#).** Хотя пара из них, по моему мнению автора, не заслужили такой участи.

## **Практика.**

Лучше всего близко к завершению или после завершения чтения текстового (не только по языку) материала устроиться в стажёром или джуном в любую занимающуюся разработкой организацию. Даже если это будет **работа на дядю за копейки**. Главное, о чём нужно позаботиться, — свой рост как разработчика. Большую роль в этом играет наставник или просто старший коллега за соседним столом.

Если же вы лох или девственник и поэтому не смогли найти достойного мужчину, держите идеи для реализации. Отнеситесь со всем пониманием, как к пет-проекту, который покажете на собеседовании. Тем более в некоторых компаниях, не очень серьёзно относящихся к набору сотрудников, это сработает. Используйте все полученные умения и знания, чтобы сделать настолько конфетку, чтобы всю жизнь было не стыдно пользоваться.

## **1. Библиотека математических объектов.**

Это так любимые преподавателями комплексные числа и операции с ними, рациональные числа в дробном представлении, геометрические векторы, кватернионы, матрицы... Проявите воображение или найдите в гугле интересное. Хорошо подумайте, где использовать структуры, а где классы. Важно понять отличие концепции ООП от АТД;

## **2. Абстрактные типы данных.**

Скорее всего, это уже делалось на языке Си. Это будет своеобразным методом осознания удобства нового языка и самой концепции АТД;

## **3. Простые игры.**

Желательно с вынесением необходимого в библиотеку движка. Для начала лучше написать на базе какой-либо графической библиотеки вроде OpenGL. Например, "пятнашки", а потом на её движке "тринашки";

#### 4. Игры на готовых движках.

Есть простенькие движки, на которых можно написать нужную игрушку. Сапёр ошибается один раз, но в ваших играх можно это исправить;

#### 5. База данных с поддержкой SQL.

И добавить нечего, пиши да оптимизируй. Можно упростить требования, это нормально;

#### 6. Текстовый редактор с графическим интерфейсом.

Хорошо подойдёт графическая библиотека FLTK. Можно и на Qt, вообще на чём душа пожелает. Главное понять, при чём здесь C++;

**7. Приложение электронных таблиц.** Аналогично предыдущему, но имеет более выраженное отношение к ООП.

#### Шаг 0.

Для тех, кто ни разу не слышал про алгосы.

1. Для нуля подойдет известный [курс на степике](#) от **CS** и [его продолжение](#). Объем небольшой, что дает надежду не умереть на половине курса, но знакомит со всем основным, дает скромное представление о чем эта (((наука))) все вообще.
2. "Грокаем Алгоритмы" *Адितья Бхаргава* и "Фундаментальные алгоритмы на C++" *Роберт Седжвик*. Два брата акробата: один попроще, другой посложнее. **Своеобразные Тик-Токи** того времени: все в картинках и не особо строго. Самое то, если школьная математика успешно забыта. В эти же ворота всякие книжники для школьников по олимпиадному программированию в духе той же "Олимпиадное программирование" *Анти Лаксонен* или все, где есть в заглавии "Для начинающих" в духе "Алгоритмы для

начинающих" *Панос Лурида*. Сюда же А. Шень "Программирование теоремы и задачи" \_.

3. Если же мама курила, употребляла алкоголь во время беременности и понимание всего написанного не пришло, то возможно попробовать **учить алгосы по Тик Токам**. Смотреть их визуализацию на графах, квадратах, табличках: [сайт\\_1](#), [сайт\\_2](#), [сайт\\_3](#), [сайт\\_4](#)\*\*. Упражняемся в построении структур, глядим на них в работе, создаем свои анимации, становимся ТикТокером— пробуем себя в профессии будущего

## Шаг 1.

Дальше хорошо было бы структурировать всю эту теорию и привести ее к чему-то строгому и осмысленному, не оставаться кривозубым самоучкой. Здесь подойдут академические курсы с ИТМО, ФКН, ФПМИ, которые, конечно же, в большинстве просто копипаст *"Алгоритмы: Построение и Анализ"* Томаса Кормена. Но его из-за большого объема освоить будет тяжело и это больше справочник, чем учебное пособие. Поэтому могу посоветовать взять за основу какой-либо из следующих курсов, а к Кормену обращаться лишь для расшара и повторения. В дополнение непонятым главам/ темам может идти *"Алгоритмы"* Дасгупта С., Пападимитриу Х., Вазирани У., где немало полезных заданий для понимания материала.

1) [Лекции](#) ИТМО и [конспект](#), [листочки](#) 2) [Сайт](#) ФКН

3) [Конспект](#) и [лекции](#) основного потока ФПМИ

Если что-то непонятно в одном курсе, бежим смотреть в другой, не закливаем на одном лекторе. Для этого в комментариях оставляю кучу ссылок на другие лекции.

## Шаг 2.

На предыдущем шаге, как только пробел по теме был устранен, ее срочно нужно зарешать на задачах. Помимо листков из **Шаг-2**, конечно же, топаем на

1) [Leetcode](#) 2) [Codeforces](#) 3) [hackerrank](#) И не такие известные

4) [eolymp](#) 5) [CSES](#) 6) [Timus Online Judge](#) Также есть замечательный сайт со всеми возможными темами и алгоритмами [MAXimal](#).

<https://neetcode.io/roadmap>

## Шаг 2.

После матриц и пределов самое время браться за остальное.  
Многомерный анализ и интегралы.

1. Второй семестр курса (<https://teach-in.ru/course/calculus-shaposhnikov-part2?ysclid=l59m8kd9jz929956587>) С. В. Шапошникова.  
Лучший курс по многомерному анализу.
2. Семинары второго ([https://drive.google.com/drive/folders/1aslZqspfnpNE5TrnY8eFH2T\\_RpPEEt-a](https://drive.google.com/drive/folders/1aslZqspfnpNE5TrnY8eFH2T_RpPEEt-a)) и третьего (<https://drive.google.com/drive/folders/1Qu1YA84CeJFTcbrZpBMkgY72NiGJvQXC>) семестра А. Сукабичевского.  
Оставим сходимость интегралов, рядов, и криволинейные интегралы на будущие шаги.
3. "Математический анализ в задачах и упражнениях" том 1 И. А. Виноградова и другие.  
Разбираем только главы 7-8.  
Линейная алгебра.  
Может быть полезной подборка How to заботать линейную алгебру ([https://t.me/postypashki\\_old/728](https://t.me/postypashki_old/728)).
4. Курс (<https://teach-in.ru/course/linear-algebra-manuilov/material>) лекций В. М. Мануйлов.  
Проще не найдете, геометрический взгляд на линал: без алгебраического треша. Разбираем все, кроме тензоров.
5. Семинары (<https://teach-in.ru/course/linear-algebra-seminars-smirnov>) С. В. Смирнова.  
Смотрим все, кроме тензоров
6. Задачи по линейной алгебре и геометрии" Гайфулин и другие.  
Разобраны все возможные алгоритмические задачи, которые постоянно встречаются на тестовом этапе в ШАД.
7. "Сборник задач по аналитической геометрии и линейной алгебре" С. В. Смирнова.  
Как не странно, С. В. Смирнов ведет по нему свои семинары.

## 8. "Сборник Задач по алгебре" Кострикина.

Интересен именно теоретическими задачами.

## Шаг 3.

Сходимость рядов и интегралов, криволинейные и кратные интегралы.

К несчастью, теория нас больше запутает: там ничего интересного— тока формализм. Поэтому смотрим только семинары и решаем задачки. Также может полезной подборка How to заботать математический анализ ([https://t.me/postypashki\\_old/518](https://t.me/postypashki_old/518)).

### 1. Семинары второго

([https://drive.google.com/drive/folders/1aslZqspfnpNE5TrnY8eFH2T\\_RpPEEt-a](https://drive.google.com/drive/folders/1aslZqspfnpNE5TrnY8eFH2T_RpPEEt-a)) и третьего

(<https://drive.google.com/drive/folders/1Qu1YA84CeJFTcbrZpBMkgY72NiGJvQXC>) семестра А. Сукабичевского.

### 2. "Математический анализ в задачах и упражнениях" том 2 и том 3 И. А. Виноградова и другие.

Подходим к материалу выборочно: избегаем треша с функана.

### 3. "Дискретная математика" С. Б. Гашков

Смотрим только раздел про производящие функции.

## Шаг 4.

### 1. "Теория вероятностей и математическая статистика" Л. Н. Фадеева, А. В. Лебедев.

Книжка для экономистов. Доступно разобрана базовая теория, много примеров решения задачек и также предлагается много типовых must have задачек.

2. **Семинары Шкляева.** Хороший мехматовский преподаватель, котором не пофигу. На teach-in есть и конспекты, и видосы, еще прилагаю его листочки с задачами. Академический треш терпится.  
<https://teach-in.ru/course/probability-theory-seminar..>

### 3. "Сборник задач по теория вероятностей" Б. А. Севастьянов.

Куча крутых примерчиков, море крутых задачек, к которым есть

решения. Задачник топ.

4. "Вероятность в теоремах и задачах" А. Н. Ширяев.

Зав кафедры теории вероятности мехмата (87 лет)/ Помню он как-то рассказывал первакам (на выдаче студаков) историю и забыл, чем она кончилась.

В книжке же very много прикольных задач, но начинающий этого не оценит. Также как и его "Вероятность" советую только продвинутым.

5. "Задачи с зачётов по теории вероятностей".

Сборник составлен как подарок будущим поколениям от студентов легендарного А. Е. Кондратенко. Задачи в большинстве непростые, но баянистые.

6. Листки с ПМИ ВШЭ по теории вероятностей.

Скорее развлечение на вечер, но думаю начинающим пригодиться. К тому же, некоторые из них встречались на интервью в ШАД.

После этих шагов вам ничего не мешает ботать Анализ данных: смотрим подборку How to заботать машинное обучение.

([https://t.me/postypashki\\_old/712](https://t.me/postypashki_old/712))

Перед тем, как разбираться в работе нейронных сетей, стоит изучить алгоритмы классического машинного обучения. На стажеров в Data Science/ ШАД etc. требуют в основном их знание и умение применять. На большей части курсов рассказывают про Numpy, pandas и визуализацию на уровне, необходимом для работы, так что не будем на этом останавливаться.

Материалы можно разделить на два типа: больше практические и больше теоретические.

Из хороших практических курсов можно отметить следующие:

1. [Mlcourse.ai](https://mlcourse.ai)

англоязычная версия Открытого курса по машинному обучению от ODS. Он достаточно полно рассказывает об основных моделях машинного обучения и как их применять на практике, и, в отличие от множества других курсов, уделяет отдельное внимание разведывательному анализу данных и feature engineering, что мало где освещается и помогает получать хорошие результаты в контекстах и отборочных по АД.

## 2. [Введение в Data Science и машинное обучение](#)

Очень много практических примеров и заданий, основное внимание уделяется линейным моделям и деревьям - основным моделям, используемым в табличном машинном обучении. Отточенный навык их применения также сильно улучшит ваши результаты в контестах и задачах классического ML.

## 3. [Машинное обучение и анализ данных](#)

Специализация от МФТИ и Яшки, 6-тилетней давности. Очень много практики и различных приемов, показанных на разных задачах, в том числе на нетабличных данных вроде временных рядов.

Большая часть практических курсов повторяют друг-друга, и редко приносят что-то новое. Если изучить 2 из 3 представленных выше курсов, то вы сможете показать хорошие результаты в любом отборочном контексте по машинному обучению и анализу данных, что позволит пройти без труда пройти в ШАД или на стажировку, магистратуру.

Однако, порой на собеседованиях начинают больше спрашивать про теорию машинного обучения, которую на практических курсах объясняют на пальцах. Чтобы больше в неё погрузиться, можно воспользоваться следующими материалами:

### 1. Курс Воронцова на физтехе

[Лекции](#), есть и [семинары](#). Но основная его ценность заключается в полноте представленной информации и глубокому теоретическому обоснованию всего происходящего. Для тех, кто любит математику.

### 2. [Учебник Яндекса](#)

Очень глубоко рассказывает как про математические основы алгоритмов ML, так и про то, как их эффективно написать. Никаких практических примеров нет, но по нему легко самостоятельно написать эти алгоритмы, что даст вам хороший прирост в понимании и позволит впечатлить HR.

### 3. [Курс ФКН по Машинному обучению](#)

Есть как теория, так и много практических заданий. Наиболее

разносторонний курс из представленных. Очень советуем ознакомиться при наличии времени и желания.

## Шаг 5.

Последний шаг возможно совмещаться со всеми другими.

1. "Задачи со студентческих олимпиад" А. Ю. Эвнин и другие.  
Куча интересных задач с решением. Также советую обратить внимание на "Литература" в конце, где можно найти не менее интересные задачи. Помимо этого чекаем в тырнете варики олимпиад япрофессионал, высшая лига, олимпиады с мехмата, физтеха и т.д. А также и другие сборники задач со студентческих олимпиад.
2. Задачи ([https://t.me/postypashki\\_old/169](https://t.me/postypashki_old/169)) с собеса в ШАД.
3. Задачи на нашем сайте  
(<https://postypashki.ru/%D1%88%D0%B0%D0%B4/>).
4. [#Задача\\_дня](#)
5. Материалы ([https://t.me/postypashki\\_old/590](https://t.me/postypashki_old/590)) онлайн ПТУ