

Эффективное изучение Java

Как быстро вырасти от новичка до профессионала



2011 г.

Оглавление

Для кого эта книга.....2

Немного об авторе.....2

10 основных сложностей, с которыми вы сталкиваетесь при изучении Java3

10 ошибок, которые обязательно приведут вас к проблемам.....7

10 советов начинающим программистам.....11

Резюме.....14

Для кого эта книга



Если вы решили изучить перспективный и развивающийся язык программирования Java, но не знаете с чего начать, как двигаться вперед, какие подводные камни вас ожидают – эта небольшая книжка для вас.

Из нее вы узнаете, как реализовать свой творческий потенциал и вырасти от новичка до профессионала, приложив минимум усилий. Книга расскажет о самых распространенных ошибках при изучении Java и способах их избежать, а также укажет на основные моменты, из-за которых многие новички бросают изучение на полпути.

Немного об авторе

Меня зовут Тимур Батыршинов. Мой стаж программирования на Java – более 6 лет. Будучи студентом без опыта и знаний начал изучать Java и параллельно применять эти знания «в боевых условиях» на своем первом рабочем месте.

За эти годы я прошел путь от новичка, который только научился выводить на экран «Hello world!», до программиста, способного строить большие комплексные системы. В процессе изучения Java и наблюдения за другими программистами постоянно замечал одни и те же ошибки, которые допускают как новички, так и профессионалы. Сам я, конечно же, не стал исключением, и наступил не на одну граблю, но считаю, что именно такой путь сделал меня конкурентоспособным программистом.

В свободное время занимаюсь проведением курсов и семинаров для студентов выпускников, которые решили построить свою карьеру в качестве Java программиста.

Также увлекаюсь духовным и физическим развитием человека, особенно их практическим применением в жизни.



10 основных сложностей, с которыми вы сталкиваетесь при изучении Java

1. Первый вопрос, который вы себе задаёте: **С чего начать?**

Итак, вы решили изучить Java, пытаетесь делать первые шаги, покупаете книгу и через некоторое время понимаете, она написана так, будто гений писал себе шпаргалку - все сложно и коротко. Вроде что-то понятно, есть знакомые слова, но неинтересно и как-то вязко продвигается обучение. Энтузиазм немного спадает.

Вы думаете, книга - не вариант, пойду-ка я в интернете на форумах посижу и прямо так и спрошу - напишите мне, что нужно сделать, чтобы написать свою первую программу. В ответ - куча постов, среди которых в основном ругань модераторов за дублирование темы, издевательства псевдо-гуру и ссылки, по которым вы перешли, но ничего из написанного там не поняли. Желание задавать другие вопросы пропадает напрочь.

Но вы не опускаете руки - пытаетесь написать свою первую программу - но не тут то было, элементарная программа не запускается. Вместо "Привет мир!" на экране - ошибка, которая вам мало о чем говорит.

Сидите Вы и думаете: "Из книги мало что понял, в интернете меня принимают в штыки, как только появляюсь со своими глупыми вопросами, я даже первую программу запустить не могу. Может я не такой умный как все и Java не для меня? Как же другие начинают?". И вы начинаете злиться на самого себя.

2. Ещё одна мысль, которая посещает новичка: **"Java - слишком сложный язык, наверное, не всем дано на нем программировать"**.

После того как вы с трудом запустили свою первую программу вы чувствуете себя самым умным человеком на земле. Но через некоторое время возникает вопрос: Если я с такими трудом запустил элементарную программу, как же я буду писать программы уровня предприятия? Для этого наверно надо иметь мозги не хуже, чем у создателя космического корабля. Мне кажется я не потяну. Чтобы научиться программировать на таком уровне, нужно прожить несколько жизней.

3. Следующая боязнь новичка: **"Новичков никто не любит, им никто не хочет помогать, над ними смеются и издеваются"**

Если Вы начинаете первые шаги в каком-либо направлении, то знаете - что вначале всегда тяжело. Потому что пробиться сквозь первый барьер непонимания, осуждений не так-то просто. Здесь нужны воля и упорство. Не у всех они есть. Мало того, вас часто пытаются осадить более умные коллеги, поставить на место, иногда унижить и за счет этого потешить свое эго. Вам и так то нелегко поначалу, а тут ещё и какие то "псевдо-гуру" пытаются на вас давить и насмехаться над вами.

Часто бывает так: что то не получилось - а спросить не у кого. Профессионалы зачастую не любят отвлекать себя для того, чтобы помочь новичку, потому что для него это уже пройденный этап, и он

смотрит далеко вперед и не желает оглядываться назад. Если даже он вам ответит пару раз, не думайте что он всю жизнь будет решать все ваши проблемы, потому что у него и своих хватает.

И вы остаетесь один на один со своими проблемами.

4. Четвертая проблема новичка: “Где находить нужную информацию?”

В интернете куча разрозненной информации, не знаешь, за какую братья. Приходится все собирать по крупицам, самому систематизировать, что неудобно и отнимает время, которого и так немного.

Интернет представляет собой большую помойку, и вы в этом каждый раз убеждаетесь, пытаетесь отыскать что то нужное. Прежде, чем вы находите ответ на свой вопрос, вам приходится перебирать много мусора. Вы буквально копаетесь в мусорном контейнере, зная, что кто-то обронил туда золотую монету. Найти актуальную информацию очень непросто.

Все это приводит к тому, что у вас падает интерес к обучению, и дальше вы начинаете продвигаться очень медленно или совсем останавливаетесь.

5. Рассмотрим ещё одну проблему: новичок уверен, что **нужно прочитать не один десяток книг, чтобы научиться программировать.**

У вас не возникало такой мысли: “Учиться по книге могут только гении, ну или приближенные к ним” или “Чтобы дочитать полностью книгу нужно обладать супер дисциплиной ”?

Вы несколько раз пытаетесь покорить какую-нибудь книгу, но обязательно в каком-то месте застреваете и дальше идти уже нет сил и желания. Возникает вопрос: если я не могу осилить одну книгу, что уж тогда говорить о десятке?

Вы злитесь на автора книги, который не удосужился всё объяснить нормальным человеческим языком. Вместо этого он использует какие-то заумные слова, после которых мозг от перегрева начинает плавиться.

6. Ещё одна мысль, которая посещают начинающих: **“Java очень быстро развивается, мне за ним не угнаться, тем более я ещё даже не начал изучение”.**

Java развивается такими темпами, что никто не знает где ему найдут применение через пару лет. У него много направлений, каждый из которых можно изучать до конца жизни, но так и не понять всех его тонкостей.

Сейчас Java - это такой Змей Горыныч, у которого отрубают одну голову (в виде устаревших технологий), а на этом месте вырастает две, которые ещё сильнее чем предыдущая. И так до бесконечности.

Поэтому кажется, что поезд уже ушел, тот кто успел на него запрыгнуть - счастливики, а нам придется продолжать программировать на своем “родных” Паскале и Visual Basic, в глубине души завидуя всем остальным. Про себя вы все время повторяете, что находитесь не то что в начале пути, а вы даже не ступили на этот путь.

7. Следующая проблема новичка: “Хочу устроиться на работу программистом, но без опыта нигде не берут”

Вы изучаете новый язык программирования, чтобы реализовать свой творческий потенциал и зарабатывать на этом деньги. Но проблема в том, что работодатель часто смотрят на ваш опыт, а у вас там только университетская программа, которая в наше время очень сильно хромает и не является показателем. Если вы только конечно не отучились где -нибудь в Оксфорде или Кембридже (тогда не надо искать работу, она сама вас найдет :))

В большинстве случаев реальность такова, что пройдя университетскую программу обучения, вы получаете следующие качества:

- умение договариваться с людьми (преподаватель становится вашим лучшим другом, когда подходит время сессии)
- умение быстро ориентироваться в экстренной ситуации
- умение находить решение своей проблемы за счет других людей (списать, выдать чужое решение за свое, попросить сдать друга экзамен вместо вас)
- неумение планировать время
- отсутствие дисциплины
- отсутствие реальных знаний по программированию.

Даже если вы изучили visual basic или Паскаль, и если вы напишите это в своем резюме, работодатель рассмеется вам в лицо.

По окончании учебы невольно приходят мысли: “Что делать? Где набраться опыта? Почему я раньше об этом не думал, вместо этого гулял и развлекался? Нужно было все осваивать самому, а не надеяться на университет”. Чувство потерянного и неправильно спланированного времени не покидает вас.

8. Восьмое препятствие, которое сидит у новичка в голове: “Нужно хорошо владеть английским языком, чтобы научиться программировать на Java”

К сожалению вся вменяемая литература и информация по Java на англ.языке. А вы не то что по-английски, по-русски то не можете понять что хочет донести до вас автор книги. Отсюда возникает следующее мнение: прежде, чем начать изучать Java, нужно овладеть английским языком, а на это уйдет пару лет. Тем более если в английском - полный ноль.

Вы думаете, нет, Java точно не для меня, английский я не знаю, и нет желание его изучать. Почему переводчики не удосужились перевести хотя бы основные книги по Java на русский язык? Да так, чтобы было написано по-человечески.

9. В скором времени все новички приходят к такому вопросу: «Какую среду программирования выбрать? Сколько людей - столько и мнений».

Выбор среды программирования - это как выбор машины. Вы начинаете спрашивать знакомых, какую машину вам купить. Каждый предлагает свои варианты, в зависимости от того, на чем ездят

они. Вам кажется, что вы покупаете машину один раз и на всю жизнь. В итоге покупаете самую дешевую и неприхотливую.

Когда спрашиваете у кого-то, почему нужно выбирать, например, Eclipse, а не Netbeans, причины выбора далеко не так очевидны. Одному нравится дизайн, другому скорость работы, третьему - удобство.

Вы думаете, как было бы хорошо, если все эти плюсы были собраны в одной программе, в которой бы все разрабатывали свои приложения. Как с машиной - как было бы хорошо, если бы все лучшее, что есть в машинах, собрали в одной.

Но такого нет. И вам приходится метаться между выборами, вы пробуете сначала начать программировать в одной среде, потом переключаетесь на другую, потому что кто-то на форуме порекомендовал. И так и скачете между ними, теряя драгоценное время, которое можно было бы потратить на освоение самого языка.



10 ошибок, которые обязательно приведут вас к проблемам.

1. Долгое раздумывание и бездействие

Как и в любом деле, если будете долго думать и размышлять (начинать или не начинать, а что будет потом, а все ли получится, а что скажут другие), то всё так и останется на абстрактном уровне.

Конечно же, бывает, что необходимо предварительное ознакомление (например, изучение инструкции перед разминированием бомбы), но в случае языков программирования - не нужно сильно уделять этому внимание.

Так, Вы топчетесь на месте, оглядываясь по сторонам, чтобы удостовериться, что вы ни один такой, хотя на самом деле вы со всеми вместе идете назад.

2. Внутренняя установка, что у меня ничего не получится. Java - это сложно, он необъятный.

Настоящих лидеров никогда не останавливало слово “сложно”. Если вы будете убегать от этого слова к другим словам (“легко”, “быстро”, “без проблем”) - то можете не начинать изучать Java.

Покажите мне человека, который знает все аспекты языка и умеет применять все его современные технологии на практике - и я оболью его бензином и сожгу, потому что посчитаю его нечистой силой, которая существовать на земле не может.

Если вы думаете, что для изучения Java вам потребуется несколько жизней – вы правы. На самом деле вы будете изучать только несколько интересных вам направлений, и этого вам хватит как говорится “за глаза”. На этом можно построить успешную карьеру, на этом можно зарабатывать хорошие деньги. Этим можно наслаждаться и получать от этого кайф.

Нет в Java ничего такого сложного, чего нельзя было бы понять. Главное - двигаться от простого к сложному, а не от сложного к сложному.

3. Боязнь совершать ошибки.

Поначалу Вы думаете, что все должно работать идеально и с первого раза. Любая ошибка для вас фатальна. Вы боитесь ошибок как огня, думаете, что если вы ошибаетесь - то у вас не хватает мозгов на написание правильного кода и вам нечего делать в программировании.

Конечно же, это не так. Не ошибается только тот, кто ничего не делает. (Не допускает ошибок тот программист, который не программирует).

Об ошибках и неудачах уже так много сказано различными гуру личностного роста, что мне добавить нечего. Все, что они говорят - полностью подходит и для роста в изучении Java.

Не бойтесь ошибок! Это ваши помощники! Это ваш опыт! По началу ошибки будут сами вас находить. Главное - это ваше отношение к ним.

Томас Уотсон, один из основателей ИВМ, сказал примерно следующее: “Кто совершит в этом месяце больше всех ошибок, тому увеличиваю зарплату”.

4. Забегание вперед без изучения предыдущих основ

Возможно вы где-то услышали, что сейчас J2EE используется на всех крупных проектах, а на J2SE программируют только древние люди. И кинулись изучать J2EE в то время, когда еще плохо ознакомились с J2SE. Вам кажется J2EE намного круче, а J2SE никому не нужен, поэтому можно через него перепрыгнуть.

Многие новички обожгли на этой свои руки (в том числе и я) пытаюсь изучить Java нахрапом. В-первых, при таком подходе вы в скором времени понимаете, что нужно вернуться назад, в исходную точку и практически начать все сначала, потому что нет базы, фундамента. Но уже более обдуманно, а не как до этого. Так теряется драгоценное время.

Если вы не знаете, для чего нужны интерфейсы и абстрактные классы, но уже хотите программировать еjb компоненты, то это мало к чему приведет. Будет больше каши в голове и как следствие - разочарование и отсутствие энтузиазма.

Бессмысленное копирование рабочего кода без понимания как он работает - верный путь к пробелам в знаниях и как причина - непрофпригодности, которая выявляется на первом же серьёзном собеседовании.

5. Программирование не объектно-ориентированном стиле.

Если в институтской программе вы познакомились с visual basic или delphi, то наверно помните как приятно на них программировать - кинул кнопку, написал для нее код и готово. Вы помните как быстро делали все институтские задачки - покидал кнопки, навешал на каждую из них нужные функции - автомат на экзамен обеспечен.

В Java такой подход оборачиваются неприятностями, потому что неумение в целом организовать архитектуру программы за счет использования принципов объектно-ориентированного программирования - ещё одна дорога для вашей программы на кладбище похороненных программ. Есть такие программисты - их называют “программисты на уровне кнопок”. Так вот, работодатели боятся их больше всего, потому что они быстрее всех могут запутать большую систему, что неизменно приведет к лишним затратам.

Попробуйте написать систему, состоящую из более 10 000 строчек кода, в “студенческом стиле” - просто навешиванием действий на кнопки. Вас стошнит. А потом чтобы изменить какой-нибудь функционал - стошнит ещё раз.

Другое дело если использованы принципы ООП. Вы знаете, что и как с чем связано, кто за какую работу отвечает. Нужно что-то изменить - берете и меняете, попутно зная, как это повлияет на другие

объекты.

Для меня это была одна из самых сложных проблем при переходе от университетских задач - к жизненным. Приучиться сначала думать, как все это будет работать между собой, какие объекты нужно создать, куда разместить и пр. Потом уже начинать писать код.

Если бы новичков каждый раз били по рукам, когда они пытаются программировать не в ООП стиле - все новички остались бы преподавать в университете, потому что там за такое глядят по головке и ставят наивысшие оценки (работает - и ладно).

6. Отсутствие практики

Есть такой жизненный закон - что не развивается, то отмирает. Есть либо развитие, либо деградация. Золотой середины вы здесь не найдете. Это вам опять же скажет любой гуру личностного роста. Но не все принимают этот факт в силу своих убеждений и принципов.

Ходите заниматься в спортзал - мышцы развиваются. Не ходите - дрябнут. Бегаете по утрам - укрепляете здоровье, не бегаете - теряете здоровье.

Так же и в программировании: нет практики, начинают отмирать и дряхлеть ваши программистские качества, теряются наработанные навыки.

Даже если вы тоннами изучаете теорию, но не пытаетесь практиковать прочитанное - это ни к чему не приведет. Есть много людей, которые в теории знают Java наверно даже лучше Джеймса Гослинга, создателя этого языка. Но не могут написать ни одной мало-мальской программы. (У нас были такие преподаватели в университете)

7. Скачивание и накопление информации “про запас”.

Я думаю этой проблемой страдают более 90% людей, изучающих какую либо информацию. Это касается не только новичков и не только программирования. Очень трудно избавиться от этой проблемы, на уровне инстинкта мы качаем и качаем нужную и ненужную информацию с целью изучения ее “потом, когда будет побольше времени”.

Со временем вы начинаете замечать, что не только на жестком диске куда-то девается свободное место, но и в вашей голове - вроде столько информации скачал, а в голове пусто, или наоборот, все свободное место занято и в голову больше не лезет. Наступает своеобразный ступор.

8. Желание доходить до всего самостоятельно.

Кто-то скажет, что это только в плюс. Но бывает так, что уходит слишком много времени. Либо находите решение не такое хорошее, хотя вам могли подсказать идеальное, самое лучшее именно для вашей проблемы и впоследствии вы уже схожую проблему решали бы сами, быстро и качественно.

Когда вы некоторое время не можете решить проблему, полностью изматываетесь, уже начинаете думать, что смысл вашей жизни - в решении этой проблемы - вот тогда наступает “тупик”.

Настроение в это время скверное, ни с кем не охота разговаривать, веселые и жизнерадостные лица вызывают в вас раздражение. Кажется - весь мир над вами издевается. Так оно и есть. Потому что весь мир - это вы и есть, вы издеваетесь над собой, пытаетесь решить проблему самостоятельно, какой бы сложной она не была, ваше внутренне эго полностью контролирует ваше сознание.

9. В противоположность предыдущему пункту - желание все всегда спрашивать у других

Если Вы не хотите думать самостоятельно, постоянно по любому поводу обращаетесь к другим людям с просьбой о помощи, если, увидев вас в коридоре, коллеги стремятся поскорее свернуть, чтобы не наткнуться на ваши расспросы - вы долго не протянете в мире программирования, поскольку около 80% задач в нем приходится решать самостоятельно.

Конечно поначалу у вас будет куча вопросов, разгрести которую вам скорее всего помогут ваши друзья или коллеги. Но не думайте, что они будут вести вас за ручку до конца жизни и решать все ваши проблемы - иждивенцев никто не любит.

10. Неиспользование всего спектра доступной информации и обучения.

Когда вы начинаете изучать Java, вам, скорее всего, понадобятся видео инструкции, как запустить первую программу, как установить jvm и т.д. На них вы быстрее обучитесь основам, чем если бы вы читали текст без картинок и по словам пытались определить действия автора.

Также часто бывает, когда обычные текстовые примеры лучше любого видео покажут и расскажут, как можно решить проблему.

Т.е. в зависимости от цели и аудитории тот или иной вид подачи материала способствует его лучшему усвоению.

Таким образом, изучая материалы, которые вы не можете понять из-за отсутствия в них наглядности - вы очень скоро теряете интерес к изучению вообще.

10 советов начинающим программистам.



1. Начните действовать.

Тут все просто и всем это известно. Не нужно сидеть на месте и крутить в голове мысли, начинать или не начинать. Всё приходит в действие, начнете действовать - появится результат (положительный или отрицательный - не важно). Не будете действовать - так и останетесь в роли отстающего.

2. Изучайте код других программистов

Общайтесь с другими программистами, пробуйте и читайте их программы. Зачастую это важнее, чем чтение книги в одиночку.

С друзьями или коллективом, объединенным общими целями, можно быстрее достигнуть желаемого. Если вы все вместе будете изучать программирование, делиться друг с другом мыслями - вы ускорите свое изучение во много раз.

Изучайте программы других программистов, находите в них ошибки или учитесь. Думайте как их можно усовершенствовать.

Например, вы решили изучить код, написанный опытным, по вашему мнению, программистом. Поначалу вы можете ничего не понять. Через некоторое время вы уже понимаете, как этот код работает, ещё немного - уже находите в нем ошибки, ещё через некоторое время - совершенствуете его. Так происходит переход от новичка к профессионалу. 90% программистов повышают свои знания за счет чтения кода более опытных программистов.

3. Не перегружайте себя информацией.

Нереализованная информация давит на вас. У меня в свое время было 30 гб скаченной информации по Java, из которой я изучил ок.1%. Когда я удалил все - на душе стало так хорошо, как будто груз сбросил с плеч.

Если вы изучаете рассылки авторов, не нужно подписываться на все рассылки подряд - большинство из них будут сразу удаляться в корзину. Желательно сначала подписаться на несколько рассылок, но именно по вашей теме, которые вы изучаете в данный момент. Прозапас не нужно.

Держите на своем компьютере только то, что изучаете в данный момент - не более. От остального хлама избавляйтесь.

Наверно замечали, как хорошо работается, когда очистишь свой рабочий стол (виртуальный или реальный) от мусора, разложишь все по своим местам, оставишь только одну книгу или документ перед собой - тогда все ваши мысли будут заняты только этим документом.

Также во время написания кода или изучения материалов отключайте все свои мессенджеры (ICQ, magent и пр.), т.к. они невероятно сильно отвлекают. Бывает, после ответа на вопрос «Как дела? Что делаешь?» не можешь вспомнить на чем остановился и мысль потеряна.

4. Почаще ходите на собеседования.

Если вы боитесь пойти на собеседование только из-за того, что у вас как вы думаете недостаточно знаний - вы никогда не устроитесь на работу. Потому что всегда найдется вопрос на собеседовании, на который вы не сможете ответить. Так ведь это наоборот хорошо - вы узнали свое слабое место и следующий раз уже не допустите промажку в этом вопросе. Так, через 10-20 собеседований у вас не останется тем, которые вы не изучили хотя бы на начальном уровне, достаточном для прохождения собеседования. А дальше, устроясь на работу, вы будете оттачивать свои навыки на практике.

Предварительно можно в интернете найти список наиболее частых вопросов по Java. Найти на них ответы, изучить темы по этим вопросам.

А если новые знания немного подпереть практикой - работа сама вас найдет, нужно будет только выбрать место получше.

5. Изучайте Java разумно

Вам необходимо найти золотую середину между самостоятельным “копанием” и консультацией с более опытными специалистами, и не впадать в крайности.

Начинайте свое обучение с видео материалов, например на youtube. Новичками видео воспринимается лучше всего, потому что нет возможности что-то сделать неправильно, ошибиться, а это важно на начальном этапе. Ищите видео, например, по тегу «Java для начинающих», сортируйте видео по рейтингу или кол-ву просмотров.

Если чувствуете, что не понимаете какой-либо материал - не нужно лбом пробивать стену и пытаться его понять во что бы то ни стало. Не нужно застревать на одном месте. Попробуйте поискать его эквивалент в видео уроках, презентациях, готовых программных проектах, т.е. изучите его в других типах мультимедиа.

Также обратите внимание на скорость печатания на клавиатуре.

Многие не придают этому навыку большое значение. И напрасно. Учитесь печатать несмотря, отличная программа “соло на клавиатуре”. Скорость в изучении увеличивается. Когда печатаешь автоматически, голова не отвлекается, руки сами все делают, а голова не прерывает процесс мышления.

6. Помогайте людям, которым можете помочь

Для повышения знаний очень полезно пытаться отвечать на вопросы участников на форумах. Пытаясь помочь решить чью-то проблему - вы во-первых повышаете свой опыт, во вторых запускаете ментальный закон - чем больше отдашь, тем больше получишь (Чем больше поможешь другим людям - тем больше помогут тебе в будущем).

Ещё способ вырасти – берете любые задачки из интернета (можно поискать соответствующие ресурсы) решаете их по книгам, сами, с чьей-то помощью, потом выставляете на форумах и ждете, что люди скажут. Наиболее опытные программисты укажут на ваши недочеты, наименее опытные - будут учиться на вашем коде.

7. Заведите свой багтрекер.

Багтрекер - это система отслеживания ошибок, наличие которой является правилом хорошего тона при разработке большинства систем. Одна из функций - она позволяет копировать опыт решенных проблем с целью их неповторения в будущем.

Если вы не поленитесь и заведете у себя на локальном компьютере свой багтрекер, куда будет вносить все свои ошибки и их решения - вы быстро начнете прогрессировать. Когда вы описываете свою проблему и заносите ее куда-либо вместе с решением - в будущем мала вероятность, что вы совершите эту ошибку ещё раз. Либо вы с ней быстро справитесь.

Следующий раз, когда у вас возникнет ошибка, когда-то решенная вами – вы найдете ее в своем багтрекере и не будете заново вспоминать и тратить драгоценное время, как же вы тот раз решили эту проблему.

8. Изучайте Java последовательно

Не нужно забегать вперед и пытаться начинать изучать, например, потоки в Java, когда вы даже не можете вручную скомпилировать класс и запустить его. Все надо делать последовательно.

Вы наверно заметили, почти все книги по Java имеют схожее содержание книги. Т.е. краткое название и последовательность глав книги. Это сделано неспроста, потому что авторы стараются обучать от простого к сложному. Поэтому не надо изобретать велосипед и думать, что вы умнее всех и можно изучать с середины или конца.

Содержание - ваш ориентир в последовательном изучении. Не следует переходить на другие темы или уроки, если вы не освоили предыдущий материал.

Например, многие книги предлагают сначала теоретически понять, что такое объект, и только потом приступают к изучению самого языка, к его синтаксису и пр. На мой взгляд - отличная методика, потому что сразу начиная программировать на уровне объектов - вы закладываете прочный фундамент для развития своих навыков.

9. Научитесь правильно находить информацию

Не нужно по любому поводу бежать на форум и задавать вопрос. Сначала надо поискать самому. Ищите через поисковики google.com или yandex.ru. В большинстве случаев ваша проблема уже кем-то решена, нужно найти тех, кто ее решил.

Пользуйтесь расширенными возможностями поиска google. Правильно построенный поисковый запрос существенно увеличит скорость нахождения решения проблемы. (подробнее о расширенном поиске вы можете узнать, если например введете запрос “скрытые функции поиска google” в поисковике google и пройдетесь по найденным ссылкам)

Также можно искать на английском языке, у кого с ним хорошо. Правило - чем больше сил вложил в поиск проблемы, тем лучше ты запомнишь решение этой проблемы.

Если вы часто сидите на форумах, то наверно заметили, что более опытные программисты не любят отвечать на вопросы прямо. Они скорее укажут вам направление, чтобы дальше вы сами разобрались. Поэтому не нужно просить решить вашу проблему или написать за вас какой-либо код. Просите указать направление, куда вам копать, вам его с радостью покажут. Часто конкретный ответ на вопрос требует много времени, которого у профессионалов немного, а просто показать пальцем, куда нужно идти - это для него легко.

10. Не уходите с головой в изучение

Учите Java быстро и успевайте делать остальные свои дела. Не надо однобоко подходить к изучению, важны и другие сферы жизни (как и в любом деле). Подтягивайте другие сферы жизни, и тогда карьера тоже подтянется, т.к. все находится в гармонии.

Поэтому не надо бросаться в крайность: окунаться с головой в изучение Java, забыв всех и вся. Действуйте легко и непринужденно, не забывая о других своих обязанностях, но в то же время дисциплинированно и обдуманно. Постепенно, шаг за шагом вы будете увеличивать свои знания и повышать свой уровень.

Если развивать только программирование - это как качать только одну ногу. Представьте себя хилым с одной здоровой ногой, которая наоборот теперь мешает ходить и вообще выглядите вы как урод какой-то. А если вы гармонично качаете все мышцы (развиваетесь в разных сферах жизни) вы будете полноценным качком.

Резюме

Когда вы начнете следовать советам, изложенным в этой книге, ваша скорость в изучении Java возрастет, и вы сможете быстро перейти от новичка до профессионала, пока другие будут топтаться на месте.

Держите все перечисленные советы перед глазами. Скоро вы начнете помнить их на уровне инстинкта, и в вашей голове каждый раз автоматически будет загораться красная лампочка, если вы что-то делаете неправильно.

Если вам интересна более углубленная информация по данной теме, ее можно найти на моем сайте www.javabegin.ru

Удачи вам в нелегком, но интересном пути!