



**Как стать программистом,
переехать в США
и получить работу мечты**

Виталий Ампилогов

Виталий Ампилогов
**Как стать программистом,
переехать в США
и получить работу мечты**

*http://www.litres.ru/pages/biblio_book/?art=63697381
ISBN 9785005306425*

Аннотация

В этой книге вы найдете подробное руководство и максимальное количество полезных советов для того, чтобы одним прекрасным солнечным калифорнийским утром сесть в свою Tesla и поехать на любимую работу в Google, Apple, Facebook или Amazon. Книга содержит ответы на многие важные вопросы: что конкретно нужно делать для переезда в США? Как освоить востребованную специальность в сфере разработки ПО? Как получить работу в американской ИТ компании?

Содержание

Об авторе	7
Введение	9
Структура книги	13
Зачем уезжать в США?	15
Что вам понадобится	20
Базовые навыки	21
Английский язык	26
Глава 1. Как стать программистом	34
Избавляемся от стереотипов	36
Выбираем язык программирования	40
Python	41
Swift и Kotlin	42
JavaScript	43
Java	44
Как изучать программирование	47
Видеоуроки и видеокурсы	48
Книги по программированию	52
Онлайн-школы	53
Офлайн-школы	54
Создание своих проектов	57
Идея	58
Дизайн	59
Контроль версий	61

Разработка	62
Публикация	64
Поиск работы	66
Составление резюме	68
Поиск вакансий	74
Разговор с рекрутером	77
Тестовое задание	79
Собеседование	82
Конец ознакомительного фрагмента.	85

Как стать программистом, переехать в США и получить работу мечты

Виталий Ампилогов

© Виталий Ампилогов, 2021

ISBN 978-5-0053-0642-5

Создано в интеллектуальной издательской системе Ridero

В этой книге вы найдете подробное руководство и максимальное количество полезных советов для того, чтобы одним прекрасным солнечным калифорнийским утром сесть в свою Tesla и поехать на любимую работу в Google, Apple, Facebook, Amazon.

Книга содержит ответы на многие важные вопросы: что конкретно нужно делать для переезда в США? Как освоить востребованную специальность в сфере разработки ПО? Как получить работу в американской ИТ-компании? Как проходит процесс собеседования и найма в Google, Apple, Facebook, Amazon и другие компании?

Мне потребовалось много лет для того, чтобы переехать из обычного российского города в Кремниевую долину

и устроиться на работу в Google. Десятки инженеров поделились со мной опытом, как они перебрались в США и получили работу мечты. Используя эту книгу, вы сможете преодолеть такой же путь, но намного быстрее.

Об авторе

Всем привет! Меня зовут Виталий, сейчас я живу в Кремниевой долине и работаю разработчиком мобильных приложений в Google. У меня не было богатых родителей или врожденных способностей, поэтому, чтобы здесь оказаться, мне потребовалось приложить достаточное количество усилий.

Я родился в бедной семье в пригороде Перми, окончил 9 классов в обычной школе. Затем была учеба в техникуме и заочное обучение в университете. В это время меня интересовали мобильные телефоны, а когда стали появляться истории о людях, которые разбогатели, создавая мобильные приложения, я понял, что тоже хочу этим заниматься. Больших денег я не заработал, но программировать научился.

Дальше моя карьера начала стремительно развиваться: сначала я устроился в крупную ИТ-компанию в Перми, а затем получил приглашение на работу в «Тинькофф» и переехал в Москву, где несколько лет занимался разработкой основного приложения «Тинькофф Банка».

У меня всегда была идея уехать из России и устроиться на работу в одну из лучших американских ИТ-компаний, однако переехать в США оказалось совсем не просто. В процессе поиска американского работодателя я отправил более тысячи резюме, что дало мне всего лишь три собеседования.

В результате я получил далеко не лучшую работу и переехал в Нью-Йорк.

После переезда в США мне открылось большое количество новых возможностей для трудоустройства в ведущие американские ИТ-компании. После нескольких месяцев усердной подготовки я начал проходить собеседования в Google, Amazon, Facebook, Apple, Snap, Netflix и другие крупные ИТ-компании. Получив несколько предложений о работе, я выбрал Google и уехал из Нью-Йорка в Кремниевую долину.

Моя история не является каким-то уникальным случаем, я убежден, что каждый может пройти такой же путь. О том, как это сделать, и пойдет речь в этой книге.

Введение

За время жизни в Кремниевой долине я встретил много инженеров из крупнейших мировых ИТ-компаний. У каждого из них был свой путь в США и своя история получения работы мечты. Но практически все сходились во мнении, что несколько лет назад они не могли представить, что будут работать в Google/Apple/Facebook/Amazon.

Первое, что приходит в голову, когда думаешь о работе в Google: «Где я и где Google». Складывается впечатление, что это где-то в другой галактике и недоступно для обычных людей. Такие мысли не имеют ничего общего с реальностью. Хорошие специалисты всегда востребованы, особенно в США, где ценится профессионализм и трудолюбие.

Написать эту книгу меня побудило желание показать, как обычный парень или девушка могут кардинально изменить свою жизнь вне зависимости от того, где они сейчас находятся, какие у них родители и сколько денег на счетах. Приложив достаточное количество усилий, каждый может переехать в США, получить работу своей мечты, хорошо зарабатывать и наслаждаться жизнью.

По пути в США я совершил большое количество ошибок, провалил много собеседований, а также достаточно долго осваивал профессию в ИТ. У меня не было четкого плана или знакомых, которые бы объяснили, что нужно делать

и в каком направлении двигаться. Эта книга могла бы сэкономить мне несколько лет.

Как можно догадаться из названия, книга будет посвящена освоению профессии программиста, иммиграции в США и трудоустройству в мировые технологические компании. Самой главной частью книги является получение работы в американской ИТ-компании, поскольку именно работа программистом позволит вам получать высокий доход и переехать в США по рабочей визе. Нет смысла ехать в США, если вы не сможете там много зарабатывать и обеспечивать себе комфортный уровень жизни.

По данным Министерства труда США, больше половины рабочих виз выдано разработчикам программного обеспечения. В Америке достаточно своих менеджеров, экономистов, специалистов по продажам, юристов, но в ИТ существует сильный кадровый голод. Американским компаниям требуется больше миллиона ИТ-специалистов каждый год. Удовлетворить спрос на квалифицированных разработчиков ПО за счет внутренних кадров невозможно, и это подталкивает американские компании к поиску специалистов за пределами страны.

Программирование дает возможность зарабатывать в несколько раз больше средней зарплаты, что может улучшить качество жизни как в родной стране, так и за границей. Крупные американские компании, такие как Google, Apple, Amazon, Facebook, постоянно нуждаются в высоко-

квалифицированных специалистах и готовы платить очень высокие зарплаты своим сотрудникам. Например, разработчик ПО со стажем два года получает около 8 000 \$ (после вычета налогов), со стажем от пяти лет получает уже около 12 000 \$, а со стажем десять лет – больше 18 000 \$. Неплохая мотивация выучить программирование, не правда ли?

Вероятно, у вас возникает логичный вопрос: почему тогда каждый второй не становится программистом? Ответ достаточно прост: выучить программирование – это тяжелый ежедневный труд, который под силу далеко не всем. Мотивация пропадает достаточно быстро, и основной путь приходится проходить, полагаясь только на свою силу воли и терпение.

Не стоит расстраиваться, если вы не умеете программировать или у вас нет соответствующего образования: освоить программирование может абсолютно каждый вне зависимости от врожденных способностей, возраста или других факторов. В начале книги мы подробно рассмотрим, как быстро и эффективно научиться программировать, а также найти первую работу.

Если вы уже работаете в сфере разработки программного обеспечения, то на страницах этой книги найдете подробную информацию об особенностях трудоустройства в американскую компанию – от подготовки резюме до обсуждения будущей зарплаты.

Впереди вас ждет интересный путь, который начинается с мечты переехать в США и заканчивается интеграцией

в американское общество. Предстоит пройти много препятствий и преодолеть большое количество сложностей, но результат однозначно того стоит.

Структура книги

Книга начинается с вводной части, где будут разобраны ключевые элементы, которые вам понадобятся. Сразу начнем с самого важного: настойчивости, продуктивности и времени. Отдельно разберем общий подход к ежедневной работе, который позволит максимально эффективно использовать время в процессе изучения программирования, подготовки к собеседованию, поиска работы, да и вообще любого дела.

Также поговорим, какую роль играет высшее образование в получении рабочей визы и дальнейшего карьерного роста. Подробно разберем наиболее эффективные способы изучения английского языка, уделяя особое внимание тем навыкам, которые помогут получить работу в США.

В первой главе книги мы поговорим о том, как достаточно быстро освоить профессию программиста, при этом не обладая особым складом ума и не имея специального образования. Далее будут рассмотрены важные темы поиска первой работы и дальнейшего развития в сфере ИТ.

Основная часть книги будет посвящена непосредственному переезду в США и трудоустройству в американскую компанию. Рассмотрим все варианты получения виз, которые позволят легально жить и работать на территории страны. Следующий большой блок коснется получения работы

в США, в нем будут затронуты такие важные темы, как написание резюме, поиск американского работодателя и подготовка к собеседованию.

Кульминацией книги является описание прохождения собеседования в американской компании. Именно к этому моменту ведут все предыдущие главы. Вы узнаете все тонкости и особенности получения работы в США.

В заключительной части мы рассмотрим жизнь в США и социальную адаптацию. Поговорим о том, как быстрее разобраться в местных правилах и не выглядеть дикарем в глазах окружающих.

Каждая глава будет содержать раздел «Избавляемся от стереотипов», в котором будут рассмотрены самые популярные мнения относительно определенной темы. Основная задача этого раздела – избавиться от навязанных нам стереотипов, которые сильно искажают реальность.

В конце книги будут подведены итоги и рассмотрены ключевые моменты иммиграции в США, а также вы найдете ссылку на группу в Telegram, где сможете задать свои вопросы и пообщаться со мной лично.

Зачем уезжать в США?

У всех разные мотивы переезда в США. Многие мои знакомые переехали сюда в поиске карьерных возможностей, лучшего будущего для своих детей и более высокого уровня жизни. Лично у меня изначально не было однозначного понимания, зачем я хочу уехать, но была уверенность, что это того стоит.

Один из первых вопросов, возникающих у людей, мечтающих переехать в США: «Стоит ли переезжать в моей конкретной ситуации?» Можно долго размышлять на эту тему и искать причины не переезжать. Но реальность такова, что если вы обладаете востребованной профессией в сфере ИТ, то вам однозначно стоит ехать в США.

Я знаю несколько семей, которые переехали с тремя детьми. Кто-то переехал, имея всего лишь тысячу долларов в кармане. Многие приехали временно поработать по контракту и остались.

Стоит понимать, что все, что показывают про Америку в новостях и тем более в фильмах, имеет мало общего с реальностью. Настоящая Америка, вероятнее всего, окажется не такой, как вы ее представляете. Только прожив здесь продолжительное время, можно делать какие-либо выводы.

Доход

Программистам и так платят высокие зарплаты, но в США эта цифра еще выше. Например, специалисты со стажем пять лет в среднем получают около 12 000 \$ (после налогов). На такую зарплату можно достаточно комфортно жить семье из четырех человек.

Карьерный рост

Возможности профессионального роста практически неограниченны. В США сотрудников оценивают в первую очередь по их результатам, поэтому, приложив достаточно усилий, вы сможете продвинуться по карьерной лестнице все выше и выше. Но даже когда достигнете карьерного потолка в текущей компании, вас с радостью будут ждать в компаниях-конкурентах, где у вас появятся новые возможности и интересные проекты.

Взаимоотношения людей

Мрачные, озлобленные, вечно недовольные люди – так часто описывают свое окружение жители бывшего СССР. В США люди всегда настроены доброжелательно по отношению к другим. Здесь практически нет конфликтов, борьбы за свои права и хамства. Очень часто незнакомые люди делают комплименты, поздравляют с праздниками и просто спрашивают, как дела. В сфере обслуживания права клиента всегда стоят на первом месте.

Природа, животные и экология

В США очень серьезно относятся к охране окружающей среды. За вредительство или плохое отношение к животным – серьезные штрафы. Во многих городах по улицам спокойно ходят олени, индейки и другие животные. Чувствуется, что животные в США живут в гармонии с людьми. Здесь очень много национальных парков, в которых оборудованы места для отдыха, пешеходные маршруты и прочая инфраструктура. Хайкинг – одно из самых популярных занятий американцев в выходные.

Отдельное внимание уделяют чистоте воздуха – все автомобили должны соответствовать высоким стандартам в области выхлопных газов, в результате чего воздух очень чистый. Высокий уровень экономического развития позволяет американцам уделять больше внимания экологии и окружающей среде.

Климат

Доказано, что хорошая погода положительно влияет на настроение и качество жизни в целом. Во многих штатах, особенно в Калифорнии, очень мягкий климат, много солнечных дней в году, свежий воздух, тепло зимой и не слишком жарко летом. В США много крупных городов на всей территории страны, поэтому вы всегда сможете найти климатические условия, которые вам нравятся.

Чистые улицы

Если не рассматривать центры мегаполисов, то можно удивиться тому, насколько чисто в США. На улицах практически нет грязи, луж, мусора и тем более собачьих какашек. После этого становится понятно, почему американцы ходят дома в обуви. Здесь вы не увидите бездомных животных на улицах и дворов, заставленных машинами.

Отдельно стоит отметить, что в США многие стараются жить в пригороде, иметь свой дом и участок земли. Это несколько отличается от Москвы и Санкт-Петербурга, где считается престижным жить в центре города.

Отсутствие социального давления

В США принято уважать интересы и индивидуальность другого человека. Можно заниматься чем захочешь, одеваться как захочешь, и окружающие будут уважать твой выбор. Здесь не важно, сколько тебе лет, из какой страны ты приехал, какой у тебя цвет кожи, – все имеют равные права и возможности. Большинство жителей США – это иммигранты, поэтому у вас не будет ограничений, связанных с вашей национальностью. Например, CEO Google, Microsoft, Uber, Tesla не родились в США, так же как и руководители многих других крупных технологических компаний.

Отсутствие информационной пропаганды

Американцев мало интересует жизнь за пределами стра-

ны. Никто не следит за тем, что происходит в других странах. Вы не найдете новостей о том, что США разработало новое оружие для подавления России, а протесты и беспорядки в других странах практически не освещаются. Такая позиция американцев обусловлена тем, что они сосредоточены на своей жизни и не хотят тратить время на бесполезную информацию.

Именно поэтому большинство СМИ сконцентрировано на событиях, происходящих внутри страны. Основное внимание в области политики направлено на противостояние демократов и республиканцев.

Всегда можно вернуться

Необязательно уезжать в США навсегда. Вы без проблем сможете вернуться на родину. С опытом работы в американских компаниях вас с радостью пригласят на более высокие должности. Также работа в США позволит вам легко купить квартиру практически в любом городе.

Что вам понадобится

Для переезда в США вам потребуется стратегия и активные действия для ее реализации. Мы подробно рассмотрим стратегию переезда в США на страницах этой книги, но сначала стоит обратить внимание на базовые навыки и умения, без которых ничего не получится: настойчивость, продуктивность и умение организовывать свое время. Еще одним необходимым навыком, без которого не удастся переехать в США, является знание английского языка.

Базовые навыки

Настойчивость

Самый важный совет, который я слышал как от американцев, так и от иммигрантов из стран СНГ: если ты умеешь каждый день работать над своими целями, то ты точно добьешься успеха. Изучение английского, поиск работы, подготовка к собеседованию – все это основано на ежедневной работе. Если вы научитесь выделять один час в день на свои цели, то переезд в Америку станет всего лишь вопросом времени.

Люди постоянно ищут «секретный ингредиент» или «тайное знание» для достижения чего-либо. Этим пользуются многие компании в своей рекламе: «Похудей за неделю на 10 килограммов», «Выучи программирование за месяц с нуля», «Разбогатеешь без вложений» и так далее. К сожалению, эти обещания не имеют ничего общего с реальностью. Секретный ингредиент уже всем давно известен – это ежедневная работа над своими целями. Вот только далеко не каждый может им воспользоваться, поскольку на пути встают лень и прокрастинация.

Не стоит рассчитывать на мотивацию или надеяться на внезапный прилив сил. Вашим главным помощником будет сила воли. В любом деле мотивация пропадает через несколько дней, и дальнейший путь приходится пре-

одолевать, полагаясь только на свою силу воли. Сила воли очень похожа на мышцу, которую можно развить и накачать. О том, как это сделать, написано уже много книг, одна из лучших, на мой взгляд, это книга Келли Макгонигал «Сила воли». В ней рассказывается о нескольких практических способах по укреплению своей силы воли и самоконтроля. Наиболее эффективные техники из этой книги – наблюдение за своими мыслями, постепенное увеличение нагрузки и медитация.

В какой-то момент появляется сильное желание все бросить. Вас будет одолевать раздражение и ощущение отчаяния. Это вполне нормальное состояние в любом нелегком деле, и через это проходит каждый. Главное – ежедневно продолжать двигаться к своей цели, неважно с какой скоростью.

Возможно, этот подход покажется вам занудным, но неоспоримый факт заключается в том, что ежедневные действия в правильном направлении гарантированно приведут вас к желаемому результату. Эта книга подскажет вам правильное направление, а вам предстоит проявить самодисциплину и приложить ежедневные усилия для получения результата. В дальнейшем способность повторять одни и те же действия каждый день будет играть важную роль не только на всех этапах переезда в США, но и окажет положительное влияние на всю вашу жизнь.

Я использовал этот принцип на каждом этапе своего пути в США. В процессе изучения программирования – чи-

тал по 20 страниц книги каждый день, когда искал работу в США – отправлял десятки резюме каждое утро, перед собеседованиями в Google, Apple, Facebook, Amazon я на протяжении многих месяцев ежедневно решал задачи на алгоритмы и структуры данных. Этот же подход я использую и сейчас для написания этой книги – каждое утро печатая новый материал или редактируя существующий.

Продуктивность

В современном мире идет большая война за наше внимание. Социальные сети поглощают все больше свободного времени, которое можно было бы инвестировать в куда более полезные вещи. Множество отвлекающих факторов порождают самую большую проблему – постоянное переключение внимания и прокрастинацию. В результате небольшое дело может затянуться на несколько часов.

Существует огромное количество методик и подходов для поднятия продуктивности и концентрации внимания. Также написано множество книг на эту тему. Здорово, если вы уже нашли то, что работает именно для вас. Если нет, то обратите внимание на методику Pomodoro.

Суть этого метода предельно проста: 25 минут концентрированно работаете, 5 минут отдыхаете. Таким образом вы разбиваете свое время на высокоэффективные промежутки и в качестве вознаграждения, помимо результата, получаете пять минут на любые занятия. Это один из самых эффек-

тивных методов для поднятия продуктивности, который используют многие успешные люди по всему миру.

Исследования подтверждают, что использование техники Pomodoro в среднем увеличивает продуктивность в два раза. Неплохо, правда? За час вы можете сделать тот же объем работы, который обычно делаете за два часа. Достаточно установить приложение на ваш телефон или компьютер и начинать использовать.

Управление временем

Очень часто можно услышать про недостаток свободного времени. От этого ощущения очень легко избавиться: запишите все, что вы делали в течение дня, в разрезе каждого часа. Я уверен, что вы удивитесь тому, как много времени у вас есть.

Еще один важный момент заключается в том, что один час вечером не равен одному часу утром в плане продуктивности и самоконтроля. К вечеру у нас накапливается усталость и десятки причин для откладывания важных дел на потом.

Лично я очень долго считал себя совой, вставал как можно позже, а по ночам засиживался до последнего. Но стоило мне поменять свой режим на жаворонка, и я обнаружил, что за один час утром я успевал сделать больше, чем за несколько часов вечером. Объяснение этому эффекту очень простое: запас умственной энергии и силы воли утром в разы выше, чем вечером. Именно поэтому в США рано ложатся

спать и рано просыпаются.

Как бы вам ни было больно от этой мысли, но если вы хотите еще увеличить свою эффективность, то придется перестроиться и начать вставать рано утром и как минимум один час уделять движению к своим целям, будь то изучение программирования, подготовка к собеседованию или поиск работы в США.

АНГЛИЙСКИЙ ЯЗЫК

Английский язык будет необходим на всех этапах переезда в США. Изучая программирование, потребуется читать много документации на английском, а в процессе трудоустройства в американскую компанию предстоит много разговаривать на английском. Сложно представить процесс иммиграции без знания иностранного языка. Поскольку процесс изучения английского занимает достаточно продолжительное время, стоит начинать его как можно раньше.

Основа английского, как и любого другого иностранного языка, это незнакомые нам слова. Вам предстоит запомнить критическую массу английских слов, которые являются фундаментом для всех основных навыков. Если вы только начинаете изучать английский, то стоит сфокусироваться на пополнении словарного запаса. Можно начать с изучения тысячи самых популярных слов в английском языке. Также вам понадобятся базовые знания грамматики для того, чтобы правильно строить предложения.

К сожалению, в школах и вузах преподают академический английский, в котором делают основной акцент на грамматику и различные правила. Существует большая пропасть между академическим языком и тем, который используют американцы в повседневной жизни.

Для того чтобы комфортно жить в США, вам нужно по-

нимать, что вам говорят, и уметь говорить самому. Можно знать все правила английского языка, но от них не будет никакого толка, если в реальной ситуации вы не сможете сказать ни слова. Необходимо сфокусироваться на аудировании и разговорных навыках, а грамматика, чтение и письмо не требуют особого внимания.

Аудирование

Это самый важный навык. Тяжело вести разговор с собеседником, если не понимаешь, что он говорит. Особую сложность представляет общение с людьми, для которых английский – родной язык. Постарайтесь уделить максимальное внимание этому навыку, поскольку понимать английскую речь вам придется чаще всего.

Для развития навыков аудирования нужно много практиковаться. Самый лучший способ – это просмотр видеороликов и сериалов на английском. Важно просматривать видео без субтитров – только так вы сможете развивать навыки восприятия языка на слух. Если вы только начинаете изучать английский, вам подойдет сериал Extra. Тем, кто чувствует себя более уверенно, я советую посмотреть сериал «Офис»: он наполнен множеством различных диалогов, а речь главных героев очень похожа на речь американцев в повседневной жизни. Также можно пересматривать ваши любимые фильмы в оригинале.

Дополнительно к просмотру сериалов можно подписаться

на англоязычных блогах в Instagram и на YouTube. Большое количество образовательного контента бесплатно доступно на Coursera, Udemy, TED. Выбирайте интересные вам темы, это придаст дополнительную мотивацию к изучению английского языка.

Еще одним хорошим источником для практики аудирования являются аудиокниги и подкасты. Можно слушать их по дороге на работу или учебу. Это помогает не только улучшить восприятие английской речи, но и получить новые знания.

Разговорная речь

Мало понять собеседника, нужно еще донести до него свои мысли. Для практики разговорной речи можно использовать онлайн-школы или занятия в классических языковых школах. Один из самых распространенных вариантов – занятия с преподавателем один на один. Преимущество этого подхода в том, что у вас не будет другого выбора, кроме как разговаривать на английском. Важно не забыть сказать преподавателю, что для вас важно развить навык разговорного английского, иначе вы, вероятнее всего, будете проходить стандартную программу по грамматике.

Еще один способ практики разговорной речи – записывать на камеру короткие монологи на любую тему. Затем просматривать видеозаписи и анализировать ошибки. Этот способ часто используют при подготовке к собеседованию

в американские компании. Вы можете найти список наиболее часто задаваемых вопросов и отрепетировать свои ответы. Просмотр того, как сам разговариваешь на ломаном английском, не доставляет особого удовольствия, но такой подход является одним из самых эффективных методов улучшения навыков разговорного английского и подготовки к собеседованию.

Произношение также играет немаловажную роль. Ваш собеседник может испытывать трудности в понимании из-за неверного произношения и интонации в каком-либо слове. Эффективным способом улучшения качества вашей английской речи является такой подход, как shadowing. Можно найти на YouTube каналы англоговорящих блогеров на интересные вам темы и во время просмотра просто повторять все, что слышите. Данный подход закрепит в вашем подсознании правильное произношение слов.

Самый лучший совет в области общения на английском: говорите проще. Правда в том, что не нужно придумывать сложные конструкции или использовать хитрые обороты, всегда намного лучше сказать просто, понятно и однозначно. И поверьте, ваш собеседник будет вам очень благодарен за то, что ему не нужно будет ломать голову, что же вы на самом деле имели в виду.

Грамматика

К сожалению, в классических учебных заведениях и раз-

личных языковых школах тратят слишком много времени на грамматику. Как бы сильно учитель по английскому языку ни убеждал вас в важности и необходимости грамматики, не верьте ему. От грамматики нет никакого смысла, если вы не можете понять, что вам говорит собеседник и тем более ответить ему.

Не стоит переживать о том, что вы скажете что-то неправильно. Каждый год в Америку приезжают сотни тысяч иммигрантов со всего мира, для которых английский не родной язык. Американцы очень толерантны к другим национальностям и прекрасно понимают, что грамотно говорить на иностранном языке это достаточно сложная задача.

Достаточно знать базовые времена, основные неправильные глаголы и порядок слов в предложении. Этого будет достаточно для того, чтобы комфортно общаться на английском языке.

Чтение

Читая, вы заставляете свой мозг обрабатывать информацию на английском языке. Также это хороший способ узнать новые слова и их употребление в определенном контексте. Важно окружить себя англоязычными источниками информации. Если вы любите читать книги, новости или блоги, то постарайтесь заменить русскоязычные ресурсы на иностранные.

Переключите язык на вашем смартфоне и компьютере.

Это простое действие поможет вашему мозгу проще воспринимать английский текст и привыкнуть к ежедневному использованию иностранных слов.

Письмо

В процессе переезда и поиска работы предстоит отвечать на множество писем по электронной почте и сообщения в чатах. В современном мире ответить в письменной форме на английском языке не составляет труда. На помощь приходят переводчики, автоматическая проверка орфографии и поиск шаблонных фраз.

Практиковать письменный английский можно в различных зарубежных чатах и социальных сетях. Реальная переписка поможет понять, как иностранцы общаются в повседневной жизни. Если вам сложно разговаривать на английском и слова совсем не хотят соединяться в предложение, то попробуйте выражать свои мысли письменно.

Умение вежливо общаться

Основная сложность заключается в том, что в русском языке мы достаточно прямолинейно выражаем свои мысли. Такой тип общения в США считается грубым. Если вы будете общаться с американцами в привычной для вас манере, то это может оттолкнуть людей.

Никогда не говорите в указательном тоне, поскольку это считается очень неприличным. Также важно предоставить

собеседнику возможность сказать «нет». Например, выражение «Дайте ручку» (Give me a pen) звучит достаточно грубо в отличие от «Не могли бы вы дать ручку?» (Could you give me a pen?).

Американцы стараются обращать внимание на положительные моменты и смягчать негативные. Так сложилось исторически: когда иммигранты приплывали в США, то работу получали те, кто был позитивно настроен и не жаловался на жизнь. В общении и повседневной жизни это проявляется следующим образом: например, вы отправили отчет на проверку – в ответ вам в первую очередь скажут о том, что было сделано хорошо, а затем намекнут, какие места стоит исправить. Никогда не критикуйте результаты чужой работы и аккуратно указывайте на ошибки.

Все эти мелкие отличия не играют большой роли при общении на русском языке, но очень важны для англоговорящих собеседников. Несмотря на большое количество иммигрантов, в США по-прежнему сохраняется культура вежливого общения.

Не стоит думать, что когда вы переедете в США, то английский волшебным образом выучится сам. Вам будет совсем не до этого, поскольку нужно будет искать жилье, оформлять документы и решать много мелких бытовых вопросов.

Стоит отметить, что для программистов английский язык не является основным навыком. Работодатель не будет тре-

бовать от вас идеальных знаний английского языка. Кроме того, американцы достаточно толерантно относятся к иммигрантам и многие уже привыкли к относительно слабому знанию языка.

Секрет успеха в изучении английского языка заключается в том, чтобы понемногу внедрять его в ежедневную рутину. Например, утром перед тем, как открыть Instagram, открывайте приложение для изучения новых слов. В дальнейшем понемногу добавляйте ежедневные задания: прослушивание подкастов/аудиокниг по пути на работу, просмотр сериалов на английском по вечерам. Можно пойти дальше и отписаться от всех русскоязычных блогеров, смотреть фильмы и читать книги только на английском языке. Так ваш мозг начнет постепенно адаптироваться к англоязычной среде.

Глава 1. Как стать программистом

Разработчик программного обеспечения – это самая востребованная специальность XXI века. Из-за большой потребности в программистах многие компании готовы платить высокие зарплаты и предлагать максимально комфортные условия работы. Специалисты в области разработки программного обеспечения имеют хорошие карьерные перспективы, а также возможности переезда в США по рабочей визе и трудоустройства в американские ИТ-компании.

Программирование – это то, чем явно стоит заняться. Результат многократно окупит все затраченные усилия. И помните, что самое лучшее время начать изучать программирование – это сегодня. Не повторяйте мою ошибку – не откладывайте изучение программирования на потом. Отбросьте сомнения и начинайте действовать прямо сейчас.

В этой главе мы поговорим о том, как быстро освоить профессию программиста и найти первую работу по этой специальности. Очень важно как можно скорее начать работать в области разработки программного обеспечения – это позволит получить профессиональный опыт и поможет развиваться в сфере ИТ.

Далее будут разобраны основные моменты, связанные с выбором языка программирования, обучением, поиском первой работы и развитием карьеры. Но перед тем как на-

чать, давайте разберем основные страхи и стереотипы, связанные с программированием.

Избавляемся от стереотипов

1. Нужно быть очень умным.

Программирование – это базовый навык, как умение читать, писать и считать. Все могут научиться писать программы, потратив определенное время и усилия. Не нужно обладать каким-либо талантом или предрасположенностью для того, чтобы писать код.

2. Без образования невозможно стать программистом.

Для того чтобы научиться программировать, нет необходимости получать классическое образование. Как правило, классическое образование – это самый неэффективный способ научиться программировать, поскольку требует долгого обучения и дает очень мало практических навыков, востребованных в реальной жизни. Намного эффективнее заниматься самообразованием или посещать специальные курсы. Большинство российских компаний будут готовы взять вас на работу без высшего образования. Однако стоит отметить, что отсутствие высшего образования в ИТ уменьшает количество возможных вариантов переезда в США.

3. Я уже слишком стар для изучения программирования.

Это самый распространенный стереотип, навязанный обществом. В России принято считать, что программисты – это молодые парни до 30. Это всего лишь общественное мнение, на которое не стоит обращать внимания. Изучение программирования требует одинаковых усилий в любом возрасте и не зависит от того, сколько вам лет. Возможно, стоит задать себе вопрос, не является ли этот стереотип вашей отговоркой, чтобы ничего не делать?

4. Мне нужны профессиональные знания.

Программирование не требует специальных знаний, достаточно базового понимания математики на уровне школы. Разумеется, знания в области компьютерных наук будут плюсом, но не являются обязательными. Ничего не мешает вам прямо сейчас начать изучать программирование.

5. У меня нет мотивации заниматься программированием.

Не надейтесь на мотивацию, она пропадет достаточно быстро. Вместо мотивации относитесь к изучению программирования как к тренажерному залу, в который вы ходите, чтобы стать лучше. Будут дни, когда придется заставлять себя идти на тренировку, но стоит только начать, как появится сильное желание продолжать. Ключевой момент в изучении программирования – регулярность и усердие, а не мотивация или талант.

6. Я попробовал, у меня ничего не получается.

Предстоит привыкнуть к тому, что в начале у вас не будет ничего получаться и огромное количество вещей будет непонятным. Программирование – это один из тех навыков, освоение которых наиболее сложно на начальных этапах. Вполне нормально, когда у новичков не получаются самые элементарные вещи. Через это проходит каждый программист.

7. Я за месяц научусь программировать.

Не все так просто. Научиться программировать за месяц не получится, потребуется куда более продолжительный промежуток времени, особенно если вы изучаете программирование с нуля.

8. Я легко найду работу с высокой зарплатой.

Когда у вас уже будет несколько лет профессионального опыта, найти высокооплачиваемую работу не составит труда. Но при поиске первой работы программистом не стоит рассчитывать на высокую зарплату и большое количество предложений.

9. Программистов стало слишком много, скоро они будут никому не нужны.

Действительно, количество разработчиков программного обеспечения ежегодно увеличивается, но и количество ра-

боты тоже увеличивается. Все больше сфер нашей жизни проходит через процесс диджитализации, а для этого нужно большое количество специалистов в области программирования. Ожидается, что в ближайшие сотни лет программист по-прежнему будет одной из самых востребованных профессий.

10. Результат не стоит тех усилий, которые необходимо приложить.

Изучение программирования – это лучшая инвестиция, которую вы можете сделать, поскольку это гарантированно позволит вам хорошо зарабатывать и откроет доступ к большим возможностям. Да, вам предстоит потратить определенное время, чтобы освоить эту профессию, но при любом стечении обстоятельств эти старания будут сполна вознаграждены.

Выбираем язык программирования

В настоящее время существует более ста языков программирования, поэтому сделать правильный выбор не так-то просто. Основным критерием для выбора языка программирования должна быть его востребованность и простота изучения. Это существенно сужает область выбора.

Не менее важно выбирать язык программирования в зависимости от области личных интересов: веб-разработка, мобильные приложения, серверные приложения, машинное обучение, разработка игр. Определившись с областью, в которой вы хотите строить карьеру, можно переходить к выбору языка программирования.

Веб-разработка: JavaScript, PHP, Python

Мобильные приложения: Swift, Kotlin

Серверные приложения: Java, Python

Машинное обучение: Python

Разработка игр: C#, C++

Как вы можете заметить, наиболее популярные современные языки программирования: Python, Java, JavaScript, Swift, Kotlin, C#, C, C++, PHP. Не стоит выбирать экзотические языки программирования, это усложнит обучение и поиск работы в будущем.

В далекой перспективе ваш выбор не так важен, поскольку, имея несколько лет опыта работы, вы сможете достаточно легко выучить новый язык программирования. Но на начальном этапе изучения программирования важно сделать выбор и придерживаться этого выбора в дальнейшем. Не стоит переключаться с одного языка программирования на другой, поскольку это негативно скажется на процессе обучения. Далее мы коротко рассмотрим самые популярные и востребованные языки программирования, это должно помочь вам сделать правильный выбор.

Python

Это универсальный язык программирования, который используется во многих сферах, таких как машинное обучение, автоматизация, анализ данных, веб-разработка, бэкенд-разработка. По результатам опросов Stack Overflow, Python удерживает первую строчку в списке самых популярных языков программирования уже несколько лет.

Большая популярность этого языка программирования обусловлена в первую очередь простотой и понятностью синтаксиса. Это один из самых главных принципов Python. Программисты любят Python, поскольку код, написанный на этом языке, легко читать и поддерживать. Также Python-разработчики очень востребованы на рынке труда и легко могут найти работу.

Многие крупные компании, такие как Google, Dropbox, Uber, выбирают Python, поскольку разработка на этом языке занимает меньше времени, чем на других языках программирования. Для Python доступно большое количество библиотек, которые помогают решать самые разные задачи. Также большое сообщество разработчиков позволяет без проблем находить ответы на возникающие вопросы.

Благодаря простоте Python многие школы используют его для обучения детей программированию, а в интернете доступно большое количество обучающих статей и видеоуроков.

Python – это хороший выбор для первого языка программирования. У него простой синтаксис, широкая область применения, большое сообщество разработчиков и много вакансий.

Swift и Kotlin

Мобильная разработка – один из самых популярных трендов последних лет. Если вам нравятся мобильные приложения, то стоит обратить внимание на Swift и Kotlin. Это современные языки программирования для мобильных устройств на базе iOS и Android.

Swift разработан компанией Apple для создания приложений для iPhone, iPad, Mac, Apple Watch и Apple TV. Многие разработчики любят Swift за простоту синтаксиса, большое

комьюнити и высокий спрос на Swift-разработчиков. Стоит отметить, что до появления Swift приложения для устройств Apple писались на Objective-C. В настоящее время уже нет смысла изучать Objective-C, поскольку этот язык стремительно теряет популярность.

Kotlin – это простой и современный язык для написания Android-приложений. Разработкой и развитием Kotlin занимается компания JetBrains. Также Kotlin продвигается Google как основной язык программирования для Android. До появления Kotlin Android-приложения создавались в основном на Java.

Разработка мобильных приложений – это очень интересное и перспективное направление. Вам предстоит работать с самыми современными технологиями и постоянно изучать что-то новое. Также у вас будет возможность всегда видеть результат своей работы в виде приложений на смартфонах пользователей.

JavaScript

Это самый популярный язык программирования для создания веб-приложений и сайтов. Если вас привлекает веб-разработка, то JavaScript – ваш выбор. Свою популярность JavaScript получил благодаря широкой поддержке во всех интернет-браузерах. Помимо веб-разработки, JavaScript можно использовать для написания серверной ча-

сти, используя библиотеку Node. js, а также для написания мобильных приложений, используя фреймворк React Native.

JavaScript достаточно прост в изучении, но имеет свои особенности – например, отсутствие классов и наследования. Вместе с JavaScript обычно изучают HTML для верстки веб-сайтов и CSS – для создания визуальных стилей. Также предстоит освоить один из самых популярных JavaScript-фреймворков – Angular, Vue или React.

Далее можно развиваться в сторону backend-разработки, изучая Node. js. Это трендовая технология, которую используют BMW, Amazon, Apple и другие крупные компании. Так вы расширите область своих знаний JavaScript за пределы управления веб-страницами.

Веб-разработка напрямую связана с JavaScript, поскольку это единственный язык программирования, который работает в интернет-браузерах. Простота и популярность JavaScript делают этот язык программирования идеальным выбором для будущих веб-разработчиков.

Java

Это универсальный язык программирования с более чем 25-летней историей. Сотни тысяч компаний, среди которых Google, Apple, Amazon, Netflix, используют Java для разработки серверных и клиентских приложений. Принято считать, что Java – это язык для разработки больших систем,

в котором основной упор сделан на простоту и читаемость кода. Это позволяет сотням программистов работать над одним большим проектом.

Если вам нравится бэкенд-разработка, то Java может стать для вас хорошим выбором в качестве первого языка программирования. Благодаря простому синтаксису и большому сообществу разработчиков научиться программировать на Java будет не так сложно. Также последние несколько лет Java активно развивается и перенимает все современные подходы из других языков программирования.

Java-разработчики всегда будут востребованы на рынке труда, поскольку нужно как разрабатывать новые системы, так и поддерживать существующие. Помимо разработки бэкенд-приложений Java широко используется в Big Data, устройствах IoT, Android-приложениях, облачных вычислениях и многих других сферах.

Спросите себя, что вам больше всего нравится: мобильные приложения, веб-сайты, серверные приложения, машинное обучение, – и на основании этого сделайте свой выбор между Python, Swift, Kotlin, JavaScript и Java. Все эти языки очень востребованы на рынке труда и никуда не пропадут в ближайшем будущем. В процессе изучения программирования стоит придерживаться сделанного выбора – это позволит сфокусироваться на одном языке программирования и быстрее достичь результатов.

Стоит отметить, что для крупных американских корпо-

раций не имеет большого значения, какой ваш основной язык программирования. Считается, что если вы владеете каким-либо языком программирования, то без проблем сможете переключиться на другой. Например, когда я пришел в Google, то мне пришлось быстро изучить Java и писать часть кода, используя этот язык программирования.

Как изучать программирование

Поскольку классическое образование является не самым лучшим способом изучения программирования, предстоит найти более эффективные методики обучения. В интернете вы без проблем найдете множество способов для изучения любого языка программирования: книги, видеоуроки, онлайн- и офлайн-школы. Проблема заключается в том, что информации слишком много и становится непонятно, с чего начинать.

Перед тем как начать изучать программирование, нужно определить, сколько времени ежедневно вы готовы выделять на обучение. Без четкого плана обучения будет очень сложно достичь результата. Критически важно ежедневно выделять время на изучение программирования – так вы будете непрерывно двигаться к своей цели. Лично я использую этот подход каждый раз, когда изучаю новый язык программирования.

Вероятно, у вас появился вопрос, что же конкретно нужно делать для того, чтобы выучить программирование. Обучение состоит из двух основных частей: освоение основ языка программирования и практика. Именно практика является главным элементом обучения, на нее предстоит потратить основную часть времени. Нет смысла просто читать книги по программированию или просматривать обучающие ви-

део, если вы не выполняете практические задания, поскольку научиться программировать можно только на практике.

Видеоуроки и видеокурсы

Видео – это самый эффективный способ восприятия информации. В настоящее время доступно огромное количество бесплатного видеоконтента в области самообучения. Созданы тысячи видеоуроков по всевозможным языкам программирования.

Изучение программирования по видеоурокам достаточно эффективный и быстрый способ получить базовые знания и освоить синтаксис языка. Не забывайте выполнять практические задания и упражнения. Нет особого смысла смотреть видеоуроки как сериал, в этом случае вы просто потратите время.

Первое, что необходимо сделать, – это найти подходящий набор видеоуроков. Поиски можно начать в Google по запросу [*название языка программирования*] + *видеоуроки*. Если у вас нет опыта программирования, то выбирайте русскоязычные видеоуроки. Это сильно упростит восприятие материала.

Вероятно, вам будут попадаться видеоролики формата «Выучи Python за час». Возможно, такой вариант может послужить хорошим стартом, но для обучения вам понадобится более подробный курс. Много хороших бесплатных ви-

деоуроков и практических заданий по программированию для начинающих можно найти на YouTube и образовательной платформе Stepik. Также если у вас есть знакомые программисты, можно спросить у них совет по выбору видеокурса.

Ниже я собрал список бесплатных видеокурсов для изучения основ самых популярных языков программирования, которые вы легко сможете найти в интернете. Помните, что ежегодно выходят новые видеокурсы, поэтому будет нелишним самостоятельно поискать свежие материалы.

Python

ADV – IT – «Python для Начинающих» (YouTube) – основы языка

Гоша Дударь – «Уроки для начинающих» (YouTube) – основы языка

«CS центр» – «Python для решения практических задач» (Stepik) – практика

Coding Medved – «Джанго с нуля на примере создания простого интернет-магазина» (YouTube) – практика

Swift

Alex Skutarengo – «Swift Марафон» (YouTube) – основы языка

Loftblog – «iOS приложение ToDo List на Swift» (YouTube) – практика

The Swift Developers – «UIKit с нуля» (YouTube) – практика

Kotlin

Devcolibri – «Базовый курс по Kotlin» (YouTube) – основы языка

Brain Softer LLC – «Видео-уроки Kotlin» (YouTube) – основы языка

JetBrains – Kotlin (Stepik) – практика

Start Android – «Kotlin. Уроки по основам разработки android-приложений» (YouTube) – практика

JavaScript

DKA-DEVELOP – «JavaScript с нуля» (YouTube) – основы языка

Sorax – «Основы JavaScript» (YouTube) – основы языка

Web Developer Blog – «Уроки JavaScript» (YouTube) – основы языка и практика

Java

Devcolibri – «Java для начинающих» (YouTube) – основы языка

Гоша Дударь – «Уроки Java для начинающих» (YouTube) – основы языка

Mail.ru Group – «Разработка веб-сервиса на Java» (Stepik) – практика

Многие мои знакомые выучили свой первый язык программирования, используя эти видеоуроки. Лично я выучил основы Swift на канале Алексея Скутаренко «Swift Марафон». Алексей создал бесплатный и простой для понимания курс с практическими заданиями. Курс не требует знаний программирования и идеально подходит для новичков. К слову, моя жена выучила Swift по этому курсу, не имея никаких знаний в области программирования.

Постарайтесь не тратить много времени на выбор курса, поскольку сильных различий между ними нет. Будет большим заблуждением полагать, что есть какой-то секретный суперкурс, который позволяет выучить язык программирования лучше и быстрее.

Помимо более эффективного способа восприятия информации, видеокурсы позволяют получить обратную связь в виде комментариев к урокам. Вполне вероятно, что на возникший у вас вопрос уже есть ответ в обсуждении под видео.

После выбора курса следует самый важный этап – ежедневное самообучение. Предстоит проявить усердие и не сдаваться через несколько недель. Нужно понимать, что изучение программирования – это тяжелый труд, требующий концентрации и силы воли. Некоторые вещи будут абсолютно непонятными, первое время ничего не будет получаться, а мотивация быстро иссякнет. Важно не останавливаться и продолжать обучение – это гарантированно прине-

сет результат.

Книги по программированию

Несмотря на то что я сам выучил свой первый язык программирования с помощью книги, я бы не советовал этот способ самообучения. Проблема заключается в том, что большинство книг быстро устаревает и содержит неактуальную информацию. Также многие книги не подходят на начальных этапах изучения программирования, поскольку написаны слишком сложно.

Чтение не самый эффективный и увлекательный способ усвоения информации. Однако книга может послужить хорошим сопутствующим источником информации для получения более глубоких знаний.

Вероятно, вам встретятся рекомендации таких популярных книг по программированию, как «Алгоритмы: построение и анализ» (Томас Кормен, Чарльз Лейзерсон), «Совершенный код» (Стив Макконнелл), «Искусство программирования» (Дональд Кнут). Это хорошие книги, но они не подходят для получения первоначальных знаний в области программирования. Вы получите максимальную пользу от этих книг, когда уже какое-то время поработаете по специальности.

Онлайн-школы

Онлайн-школы – достаточно популярный и эффективный способ изучения программирования. Этот формат очень похож на видеоуроки, но подразумевает более высокий уровень обратной связи. Еще одним плюсом является проверка ваших практических заданий.

Учиться бесплатно в онлайн-школе не получится, цены начинаются от 10 000 рублей за месяц обучения. Платное обучение может добавить вам мотивации не пропускать занятия и выполнять домашнюю работу.

Наиболее известны онлайн-школы от «Яндекса» («Яндекс. Практикум») и Mail.ru (GeekBrains), но в последнее время появляется все больше небольших онлайн-школ. Вероятно, вы часто будете видеть рекламу с обещаниями научиться программировать за три месяца и устроиться на работу с зарплатой от 100 000 рублей. Реальность, разумеется, отличается от рекламных слоганов. За три месяца возможно выучить основы языка, но для получения прикладных навыков требуется куда больше времени.

Если вам позволяют финансовые возможности, то онлайн-школа может стать для вас хорошим вариантом для изучения программирования. Если отнестись серьезно к занятиям и домашним заданиям, то можно достаточно быстро освоить базовые навыки.

Офлайн-школы

Спрос на программистов настолько высок, что некоторые ИТ-компании готовы бесплатно обучать программированию и готовить специалистов. Офлайн-школы – наиболее эффективный способ усвоения информации, а получить ответы на возникающие вопросы можно при живом общении. Еще одним большим плюсом офлайн-школ является возможность трудоустройства после окончания.

Наиболее популярные школы на момент написания книги: «Тинькофф Финтех», школа программистов HeadHunter и «Школа 21» – проект Сбербанка, основанный на методике, используемой в инновационной французской школе программирования Ecole 42. Ежегодно появляются новые школы, поэтому стоит самостоятельно проверять информацию в поисковике.

Когда я работал в «Тинькофф», мы нанимали очень много ребят из нашей финтех-школы, поскольку выпускники имели хорошие базовые и практические навыки для решения повседневных задач. Сейчас обучение программированию проходит в Москве, Санкт-Петербурге, Новосибирске, Екатеринбурге и Рязани. Доступно множество направлений, среди которых разработка приложений для iOS и Android, программирование на Python, Scala, JavaScript.

HeadHunter в своей школе обучает программированию

на Java и JavaScript. Занятия проходят по вечерам в московском офисе компании. Продолжительность курса – полгода: три месяца теории и три месяца практики. По завершении обучения у вас будут хорошие шансы устроиться в компанию.

«Школа 21» отличается от классических офлайн-школ программирования. В этой школе нет преподавателей, а задания проверяют сами студенты. Также оказывается помощь в поиске оплачиваемой практики в ИТ-компаниях. Обучение в школе бесплатное и проходит в Москве и Казани. Перед началом обучения в течение месяца проходит жесткий отбор, после которого начинается более глубокое изучение программирования. Доступны все популярные языки программирования: C, C++, Swift, PHP, Python, Ruby, JavaScript и т. д.

Обучение в офлайн-школе подойдет не всем, поскольку большинство школ находится в Москве и крупных городах. Но если у вас есть возможность поступить и посещать офлайн-школу, то постарайтесь использовать этот шанс, поскольку это откроет для вас больше перспектив.

Я помню, как один парень ездил из Казани в Москву на каждое занятие в «Тинькофф Финтех» и в результате успешно закончил обучение, а затем получил работу.

В изучении программирования определяющую роль играет не выбранный способ обучения, а постоянство и настойчивость. Не теряйте много времени на поиски наилучшего

варианта. Если сомневаетесь, начните с обучения по видеоролкам – это универсальный и бесплатный способ начать осваивать программирование.

Будьте готовы к тому, что вначале у вас ничего не будет получаться, а мысли о том, чтобы бросить обучение, будут приходить очень часто. Ваш мозг будет искать тысячу причин для пропуска занятий или переноса на потом. Все это стандартная реакция нашего организма на получение полезной, но неинтересной и сложной информации. Просто нужно продолжать двигаться вперед.

Изучение основ языка программирования и приемов прикладной разработки займет несколько месяцев. Время обучения будет сильно зависеть от того, сколько времени вы сможете выделять и насколько эффективно работать. Но недостаточно просто пройти определенный материал и выполнить практические задания – важно овладеть самостоятельными навыками создания проектов. Это важный этап на пути к поиску первой работы, о котором мы поговорим в следующей главе.

Создание своих проектов

После того как вы освоили основы программирования и получили базовые навыки, самое время начинать свой проект. Создание своих проектов – один из самых важных этапов в процессе изучения программирования и развития в сфере ИТ. Это позволит вам получить реальный опыт прикладной разработки программного обеспечения, а также создать первые проекты для своего резюме.

Вам не нужно знать все тонкости выбранного языка программирования для написания своего первого проекта – пробелы в знаниях заполняются в процессе работы. Вам потребуется применить на практике все то, что вы изучили ранее, а также самостоятельно найти ответы на сотни вопросов, возникающих в процессе разработки.

Не стоит недооценивать умение искать информацию в Google. Этот навык поможет вам быстро находить ответы на возникающие вопросы. Поскольку вы программируете на одном из популярных языков программирования, то, вероятнее всего, решение вашей проблемы уже опубликовано на Stackoverflow или другом форуме. Стоит понимать, что большинство информации доступно только на английском языке, следовательно, вам предстоит искать ответы на английском.

Обладать какими-либо знаниями и уметь применять их

на практике – далеко не одно и то же. Именно поэтому стоит как можно быстрее начинать работу над своим проектом. В следующих главах мы подробно рассмотрим каждый этап работы над ним.

В процессе работы над проектом вы приобретете много полезных навыков, главный из которых – решение неочевидных, на первый взгляд, проблем. Самостоятельный поиск решений – это один из самых главных навыков программистов.

Перед тем как получить свою первую работу программистом, я написал два приложения, которые могли продемонстрировать моему будущему работодателю наличие у меня необходимых практических навыков. Множество моих друзей и знакомых нашли свою первую работу таким же образом.

Идея

Первое, что вам стоит сделать, – определиться с основной идеей продукта. Для этого выпишите на листок список приложений, сервисов или сайтов, которые были бы вам интересны. Идея не обязательно должна быть уникальной, вполне нормально написать приложение для списка дел, трекер выпитой воды или прогноз погоды. Еще больше идей можно найти в Google по запросу *side project ideas*. На этом этапе важно определиться с выбором и двигаться дальше.

Для первого проекта не стоит выбирать идею со слишком

сложной реализацией. Например, новичкам будет непросто написать приложение по обработке видео или графический редактор.

На этапе выбора идеи можно определить, какой минимальный набор функций должен содержать ваш продукт. Не старайтесь сразу сделать большой продукт, выделите базовую функциональность и сконцентрируйтесь на ее реализации, в дальнейшем вы легко сможете добавить новые функции. Например, когда я писал приложение для составления списка фильмов, в первой версии я реализовал только самые важные функции: поиск фильмов и добавление их в избранное. А в дальнейшем добавил синхронизацию между устройствами, рейтинги и расширенную информацию.

Дизайн

После того как вы определились с идеей, стоит подумать о дизайне и UX. Можно начать с набросков интерфейса и основных функций в блокноте. Затем перенести этот дизайн в графический редактор – тут хорошо подойдет Sketch или Figma. Если вам не хочется разбираться с графическими редакторами и самостоятельно рисовать интерфейс, то можно попробовать найти готовый дизайн.

В интернете множество сайтов, где вы можете скачать бесплатные шаблоны для сайтов и приложений. Можно целиком использовать готовый дизайн или взять только основ-

ные элементы. Также хорошим ресурсом для поиска шаблонов дизайна является Pinterest. На этом сервисе вы без особых проблем найдете множество хороших примеров, которые сможете использовать в своем проекте. Для первых проектов вам не нужен уникальный дизайн, можно взять готовые шаблоны для экономии времени.

Например, если вы хотите создать веб-сервис, связанный с фильмами, то достаточно ввести в Google-картинках *movies web ui* – и вы получите десятки вариантов дизайна для вашего приложения.

Не стоит пытаться сделать сложный пользовательский интерфейс с большим количеством нестандартных элементов. Правило «чем проще, тем лучше» хорошо работает в области разработки программного обеспечения.

В результате у вас должны получиться экраны и элементы вашего приложения в виде графических материалов. Перед тем как начать программировать, очень важно определиться с набором основных экранов в приложении и сценарием взаимодействия с ними.

Вероятно, у вас появится желание сначала написать код, а только потом приступить к дизайну. Это одна из наиболее часто совершаемых ошибок среди новичков. В результате такого подхода многое нужно будет переписывать, поскольку в процессе дизайна выявляется большое количество подробностей и требований, которые невозможно предусмотреть на этапе разработки. Соблюдайте баланс и не тратьте очень

много времени на создание идеального дизайна, но и не переходите к разработке без готовых макетов.

Контроль версий

Представьте, что вы что-то изменили в коде и ваш проект перестал работать – нужно срочно вернуть все обратно, но вы уже не помните, что было изменено. Для решения таких проблем и существуют системы контроля версий, которые хранят все сделанные изменения, а также облегчают командную разработку.

Git – это одна из самых известных систем контроля версий с открытым исходным кодом, на которую полагаются миллионы разработчиков по всему миру. Git хранит информацию о том, кем какие изменения были сделаны, и позволяет очень легко ими управлять. Рано или поздно вам придется разобраться с основами Git, поскольку эта система версий является неотъемлемой частью ежедневной работы большинства программистов.

На первых этапах нет необходимости погружаться во все тонкости Git, достаточно будет освоить базовые функции. Для того чтобы начать работу с Git, нужно будет создать учетную запись и скачать одноименное приложение с официального сайта github.com. Далее, используя это приложение, необходимо создать репозиторий, а в дальнейшем добавлять новые изменения. Если у вас возникнут сложности,

то в интернете вы без проблем найдете множество статей и курсов, подробно рассказывающих о принципах работы с Git.

В результате вся история изменений будет храниться на сервере GitHub. Это позволит вам не потерять важные изменения, а также при необходимости вернуть проект в предыдущее состояние. Еще одним плюсом является возможность показать свой код другим программистам для получения обратной связи и проверки ошибок.

Разработка

Настало время применить имеющиеся навыки программирования на практике. Вероятно, то, что вы изучили ранее, не будет получаться на практике, а некоторые проблемы будут казаться нерешаемыми. Вам потребуется самостоятельно справиться с этими проблемами и продолжать превращать дизайн в реальное приложение.

Важным моментом в процессе создания первого приложения является обратная связь. Будет здорово, если у вас есть друг-программист, который сможет посмотреть ваш код и указать на ошибки. Если у вас нет знакомых, то можно поискать конференции программистов в своем городе или найти группы в социальных сетях. Еще один вариант поиска – написать личные сообщения программистам в LinkedIn с просьбой посмотреть ваш код.

Обратная связь очень важна для того, чтобы узнать лучшие практики и избежать распространенных ошибок, которые совершают большинство новичков. Необходимость показывать свой код другим людям может вызвать дискомфорт, особенно при написании первого приложения. Это нормальное чувство, с которым сталкиваются начинающие разработчики. Важно помнить, что советы опытных программистов могут сэкономить вам много времени в процессе развития навыков программирования.

Следующим важным этапом в процессе разработки является декомпозиция задач. Полезно разбить проект на мелкие подзадачи – это позволяет лучше контролировать процесс разработки и примерно понимать, какой объем работы еще предстоит сделать. Также выполнение подзадач создает ощущение прогресса и добавляет мотивации в процессе создания проекта. Для управления подзадачами можно использовать Trello – в этом приложении вы сможете разбить ваш проект на небольшие подзадачи и двигать их по статусам. Таким образом процесс разработки будет более прозрачным и эффективным.

За несколько месяцев работы вы сможете создать свое первое приложение. Важно довести приложение до какого-либо финального состояния. Возможно, придется отказаться от каких-либо функций и пойти на компромиссы в дизайне, но важно получить финальный результат и опубликовать свой продукт.

В процессе работы над своими проектами вы приобрете много полезных навыков, которые пригодятся вам на будущей работе. Перед тем как получить свою первую работу программистом, я написал два приложения. Это позволило получить минимально необходимый опыт для того, чтобы пройти собеседование и в дальнейшем справляться с рабочими обязанностями.

Публикация

Не стоит стесняться опубликовать свой проект для большой аудитории. Если вы разработали веб-сервис или сайт, то купить хостинг и домен не потребует больших финансовых вложений. Для разработчиков мобильных приложений необходимо заплатить небольшую сумму за возможность публикации в магазине приложений.

Созданные вами продукты будут сильной составляющей вашего резюме. Для работодателя это будет означать, что вы можете работать самостоятельно, разбираться в сложных вопросах и доводить дела до конца. Также ваши проекты могут принести вам дополнительный доход.

Еще один способ увеличить свои шансы при поиске работы – опубликовать свой проект в открытый доступ на GitHub. Так вы сможете показать свое умение писать качественный код. Перед публикацией в открытый доступ обязательно следует потратить время на то, чтобы привести код в нормаль-

ное состояние. Проверьте структуру проекта, напишите тесты, удалите лишние комментарии, избавьтесь от магических переменных и замысловатых конструкций. Нужно понимать, что вашему потенциальному работодателю не важно, что делает ваше приложение, ему куда интереснее увидеть, как оно написано. Во многих компаниях кандидаты, которые присылали ссылки на проекты, всегда имеют больше шансов получить приглашение на собеседование.

Поиск работы

Термины:

Рекрутер – специалист по поиску кандидатов

Интервью – собеседование

Интервьюер – человек, проводящий собеседование

Оффер – письменное предложение о работе с указанием заработной платы

Нужно как можно раньше устраиваться на работу программистом для того, чтобы начать накапливать реальный опыт в области разработки программного обеспечения. Также не стоит ограничивать поиски работы только родным городом, обязательно обратите внимание на Москву и Санкт-Петербург.

Реальность такова, что зарплаты программистов в Москве и Санкт-Петербурге в 2—3 раза больше, чем в регионах. Если вы сейчас за пределами этих городов, то стоит серьезно задуматься о поиске работы и переезде в столицу. Нет большой разницы, в каком городе искать работу, в плане сложности прохождения собеседования. Помимо высокой зарплаты, переезд в Москву и Санкт-Петербург откроет много карьерных перспектив и поможет получить новые связи. Также это отличный вариант оценить все сложности смены места жительства и подготовиться к последующему переезду в дру-

гую страну.

В процессе поиска работы важно поставить цель добиться лучшего результата. Например, устроиться на работу в московский офис «Яндекса», Mail.ru, «Тинькофф», Avito, Vadu или Revolut. Возможно, на первый взгляд это может показаться малореалистичным, особенно для программиста без опыта работы, но в реальности у вас есть все шансы. В следующих главах мы подробно поговорим о том, что нужно делать для получения работы мечты.

Поиск работы – это увлекательный процесс, который начинается с составления резюме и заканчивается подписанием приглашения на работу. Со временем приходит осознание того, что для успешного прохождения собеседования вам необходимо понять, чего ожидает от вас потенциальный работодатель. Для этого вам предстоит продемонстрировать хорошие технические знания и доказать, что вы без проблем справитесь с поставленными задачами. Не менее важно произвести хорошее впечатление и показать, что с вами будет приятно работать. Я провел около 50 собеседований во время работы в «Тинькофф Банке» и могу с уверенностью сказать, что каждый из этих пунктов играет важную роль.

Процесс поиска работы, как правило, состоит из следующих этапов:

1. Составление резюме.
2. Поиск вакансий.

3. Выполнение тестового задания.
4. Техническое собеседование.
5. Обсуждение и подписание приглашения на работу.

Важно правильно подойти к каждому этапу, и тогда вы сможете получить достойное предложение о работе в хорошей компании. Далее мы подробно разберем каждый из этих этапов.

Составление резюме

Хорошо составленное резюме многократно увеличивает ваши шансы на получение работы. Стоит потратить время и силы на этом этапе, для того чтобы в дальнейшем получить больше возможностей для достойного предложения.

Если не вдаваться в подробности, задача рекрутера состоит в том, чтобы найти кандидата, который соответствует описанию вакансии. Это значит, что данные в вашем резюме должны дать рекрутеру всю необходимую информацию для принятия решения о том, что вы подходите на имеющуюся вакансию. Еще одна задача рекрутера – проверить, нет ли в вашем резюме неправды. Я настоятельно рекомендую не врать в процессе поиска работы.

Далее я приведу пример резюме, которое вы сможете использовать в качестве шаблона.

Иванов Сергей

Junior iOS Developer (Swift)

Дата рождения: 17.09.1990

Город: Пермь (готов к переезду)

Телефон: +7 (123) 456 78 90

Email: sergey.ivanov@gmail.com

GitHub: github.com/sergey.ivanov

ОПЫТ РАБОТЫ

iOS-разработчик (март 2020 – настоящее время)

Полный цикл разработки приложений, от идеи до продвижения в App Store.

Movie List – простое и удобное приложение для составления списка фильмов.

- Реализовал клиент-серверное взаимодействие с The Movie Database API

- Разработал более 10 экранов с нестандартным дизайном, используя UIKit

- Реализовал синхронизацию данных с использованием CloudKit

- Использовал CoreData для кеширования пользовательских данных

- Локализовал приложение на 4 языка и опубликовал в App Store

- Написал 30 юнит-тестов и 10 UI-тестов
- Приложение было скачано более 7 000 раз за 6 месяцев
- Travel Guide – офлайн-путеводитель по популярным городам мира.
- Реализовал поддержку офлайн-карт на базе фреймворка MapBox
- Использовал Core Data для доступа к данным без интернета
- Добавил монетизацию приложения с использованием StoreKit
- Реализовал 12 экранов с нестандартными UI-компонентами, используя UIKit
- Опубликовал и продвигал приложение в App Store

КЛЮЧЕВЫЕ НАВЫКИ

- Умение писать чистый и понятный код
- Навыки разработки и дизайна современного UI
- Опыт проектирования и разработки клиент-серверных приложений
- Умение заканчивать работу в срок

ТЕХНОЛОГИИ

Swift, UIKit, Foundation, CoreLocation, Autolayout, CocoaPods, Git, JSON, Client-Server API, GCD/Operation,

MVP, Core Data, StoreKit, iCloud Sync

КУРСЫ

Видеокурс Swift – Алексей Скутаренко

Разработка iOS приложений на Swift – Geekbrains

ОБРАЗОВАНИЕ

Высшее образование: Пермский национальный исследовательский технический университет – «прикладная информатика» (2015—2020)

Первое, на что стоит обратить внимание, – это оформление вашего резюме. Для того чтобы сделать его более привлекательным для рекрутера, стоит придерживаться нескольких правил. Во-первых, резюме начинающего разработчика не может занимать больше одной страницы. Во-вторых, не нужно использовать сложные шаблоны, самый лучший вариант оформления – это просто черный текст. Также важно соблюдать одинаковые отступы между блоками и абзацами. В качестве инструментов оформления можно использовать размер шрифта, отступы, выделение жирным и курсивом.

Давайте рассмотрим подробнее наиболее важные элементы резюме. Первое, на что обращает внимание рекрутер, – это название должности. Важно указать конкретную пози-

цию, например Junior iOS Developer (Swift). Таким образом вы не только привлечете внимание рекрутера, но и сэкономите свое время, исключив предложения о нерелевантных вакансиях. Ошибкой будет написать в должности просто «Программист», поскольку это вносит большую долю неопределенности о кандидате.

Добавьте к вашему резюме фото: люди хотят видеть, с кем им предстоит работать. Ваша фотография должна быть сделана в бизнес-стиле, не стоит выбирать снимок с отдыха или вечеринки. Будет здорово, если у вас получится произвести позитивное и дружелюбное впечатление, используя фотографию.

Не забудьте добавить контактные данные. Если вам неудобно отвечать по телефону в течение дня, то добавьте комментарий о том, что предпочтительный способ связи – email. В этом же разделе следует указать ссылку на свой GitHub-профиль. Не забудьте предварительно привести свои репозитории в порядок и добавить описание к каждому проекту. Хороший GitHub-профиль обеспечит вам приглашение на собеседование, а в некоторых случаях может послужить заменой тестового задания. Подробно об оформлении GitHub-профиля мы поговорим во второй части книги.

Следующий важный момент – это опыт работы. Основной упор должен быть сделан на релевантном опыте. Даже если вы успешно проработали последние несколько лет в индустрии, не связанной с ИТ, не добавляйте это в резюме.

Ничего страшного, если у вас будет всего лишь одна запись о самостоятельной разработке приложений. Коротко опишите проект, укажите ваши обязанности и технологии, с которыми вы работали. Будет здорово, если вы добавите конкретные цифры и результаты, например процент покрытия тестами, количество скачиваний, или коротко опишете сложные технические решения.

Хорошее впечатление также можно создать, добавив личные качества. В последнее время этому разделу резюме начинают уделять все больше внимания. Подумайте, что у вас действительно хорошо получается: возможно, это умение доводить начатое до конца или способность сфокусированно работать.

Не забудьте указать список технологий и фреймворков, с которыми работали. Рекрутеру важно увидеть в вашем резюме определенный список ключевых слов, которые содержатся в вакансии. Для понимания востребованности технологий вы можете поискать вакансии на HeadHunter и проанализировать, какие есть требования к кандидатам в рамках определенного языка программирования. Посмотрев вакансии, вы найдете пробелы в своем опыте и поймете, какие технологии наиболее востребованы.

Наиболее часто встречаемая ошибка в составлении резюме – это добавление ненужной информации. Никому не интересно, какие у вас хобби или опыт работы в McDonald's. Не добавляйте информацию о наличии автомобиля, семей-

ном положении и спортивных достижениях. Компании нужен хороший программист, а не водитель или спортсмен.

Несмотря на то что образование не играет решающей роли при найме кандидата, указание высшего образования или специализированных курсов может добавить веса вашему резюме.

Как вы можете заметить, разработав самостоятельно всего два приложения, можно составить неплохое резюме. На рынке множество вакансий с опытом работы от года, что вполне соответствует времени обучения и создания своих проектов. Поэтому важно как можно раньше начинать работу над своими проектами, которые станут хорошей основой для вашего резюме.

Поиск вакансий

Первый этап поиска работы – это поиск вакансий и установка контакта с рекрутером компании. Найти вакансии можно на различных онлайн-площадках, таких как HeadHunter, SuperJob, «Хабр Карьера», Rabota.ru. Здесь нет ничего сложного: регистрируетесь на сайте, заполняете свои данные и отправляете отклики. Для кандидатов эти сервисы всегда бесплатные.

Последнее время активно развивается поиск работы в Telegram-каналах. Самые популярные из них: DevJobs (@devjobs), Job Geeks (@jobgeeks), Tproger

(@tproger_official), MyJobIT (@myjobit), GetIT Russia (@headzio), Logic (@logic), ForDev (@fordev), Mobile Dev Jobs (@mobile_jobs), DevOps_Jobs (@devops_jobs_feed), JavaScript Jobs (@javascript_jobs_feed). Можно подписаться на эти каналы и откликаться на вакансии, которые соответствуют вашим навыкам и опыту.

Еще один вариант – поиск вакансий на сайтах компаний. Не все ИТ-компании публикуют объявления на сторонних ресурсах. Также если вы хотите попасть в определенную компанию, но в настоящее время у них нет подходящей вакансии, вы всегда можете отправить свое резюме и поинтересоваться, нет ли у них подходящей для вас позиции.

Следующий способ поиска работы – социальная сеть LinkedIn. Я настоятельно рекомендую вам зарегистрироваться в LinkedIn, заполнить там профиль и развивать свою сеть контактов, несмотря на то что этот сервис, вероятно, по-прежнему заблокирован на территории России. В дальнейшем LinkedIn будет играть важную роль в поиске работы не только в России, но и в США. Не забудьте указать, что вы в активном поиске работы, и ежедневно добавляйте новые контакты вне зависимости от личного знакомства. Широкая сеть контактов откроет вам доступ к рекрутерам крупных ИТ-компаний по всему миру.

Вероятно, в LinkedIn вас найдут независимые рекрутеры, которые работают сразу с несколькими компаниями. Не переживайте, вам ничего не придется платить, поскольку аген-

ты получают комиссию с компании, если вы пройдете собеседование и примете оффер.

Москва и Санкт-Петербург предлагают большое количество вакансий, получить приглашение на собеседование будет достаточно просто. В других крупных городах выбор вакансий будет меньше, но по-прежнему получить отклик потенциального работодателя не должно составить большого труда.

Если у вас есть компания мечты, в которой вы очень хотите работать, то не стоит отправлять туда резюме в первую очередь. Для начала потренируйтесь на других вакансиях. Это позволит вам лучше понять структуру собеседования и привыкнуть к стрессу.

В процессе поиска работы обязательно попробуйте устроиться в крупные российские ИТ-компании, такие как «Яндекс», Mail.ru, Avito, Сбербанк, «Тинькофф Банк». Если у вас получится попасть в одну из этих компаний, то вы быстрее наберетесь опыта и получите возможность поработать над большими и интересными проектами. Также можно попробовать устроиться в аутсорс-компании (занимаются разработкой под заказ), которые имеют офисы в США. Это позволит в дальнейшем перевестись в американский офис компании. Одна из самых известных компаний в этой сфере – EPAM.

Составьте список компаний, где бы вы хотели работать, но не спешите начинать свои поиски работы с этих компа-

ний. Сначала нужно потренироваться в прохождении собеседований, для этого отправляйте резюме и отклики на все подходящие вакансии. После нескольких собеседований вы будете чувствовать себя намного уверенней и будете знать, чего примерно ожидать в следующий раз.

В процессе поиска первой работы вам предстоит отправлять десятки резюме и получить много отказов. Но благодаря большому количеству вакансий у вас будет возможность увеличить свои шансы, совершая много попыток.

Разговор с рекрутером

Если ваше резюме привлекло внимание рекрутера, то вскоре вы получите email с предложением пообщаться. На этом этапе основная задача рекрутера – отсеивать неадекватных кандидатов. Также он выясняет, соответствуете ли вы имеющейся вакансии и насколько релевантен ваш опыт работы.

Как правило, общение с рекрутером начинается со знакомства. Он коротко расскажет о компании и вакансии, на которую вас рассматривают. Далее вам предстоит рассказать о себе. Важно построить свой рассказ таким образом, чтобы за одну минуту успеть рассказать обо всех своих самых значимых успехах в области разработки программного обеспечения. Не забудьте упомянуть свои опубликованные проекты и различные сложности, которые вы преодолели.

Далее могут последовать еще несколько вопросов о вашем предыдущем опыте и практических навыках. Иногда в процессе телефонного разговора вам могут задать простые технические вопросы, а также спросить о зарплатных ожиданиях. В конце разговора у вас будет возможность задать свои вопросы.

Если вы договорились о телефонном звонке, то постарайтесь найти тихое место, где вам не будут мешать. Старайтесь разговаривать вежливо вне зависимости от того, как разговаривают с вами. Для того чтобы увеличить свои шансы на прохождение на следующий этап, упомяните, что вы готовы выполнить тестовое задание. В заключение разговора поблагодарите за звонок и уделенное вам время.

Как правило, телефонное интервью длится 15—25 минут. Если все пройдет хорошо, то рекрутер покажет ваше резюме техническим специалистам, и они примут решение о приглашении на собеседование. Также перед техническим собеседованием вас могут попросить выполнить тестовое задание.

Одна из самых распространенных ошибок на этом этапе — слишком много говорить о деньгах. Для рекрутера это является признаком того, что долго вы в компании не проработаете. Отсутствие интереса к вакансии тоже может послужить причиной к отказу.

Тестовое задание

Тестовое задание помогает потенциальному работодателю понять, как вы пишете код. Это очень важный этап, и стоит отнестись к нему максимально серьезно. Хорошо выполненное тестовое задание даст вам большое преимущество перед другими кандидатами.

Первое и самое важное, что стоит сделать, – это понять задание. Внимательно прочитайте его: возможно, некоторые требования будут для вас непонятными. В этом случае составьте список вопросов и отправьте их рекрутеру. Иногда компании специально добавляют неопределенность в задание для того, чтобы увидеть способность кандидата уточнять требования. На начальном этапе необходимо полностью определить требования, иначе в дальнейшем придется многое переделывать.

Вас обязательно спросят о том, сколько времени вам потребуется для выполнения задания. Не старайтесь произвести впечатление оптимистичными сроками. Трезво оцените свои силы и добавьте дополнительные 30% на случай непредвиденных обстоятельств. В ответе рекрутеру постарайтесь обосновать, почему вам потребуется именно столько времени. Нет ничего плохого в том, чтобы сказать, что в будни вы заняты на текущей работе, а в выходные можете выделить только ограниченный интервал времени. Лучше

назвать реалистичный срок выполнения и сделать задание раньше, чем сорвать дедлайн.

Процесс выполнения тестового задания очень похож на упрощенный процесс создания своего проекта. Начните с составления to-do-листа. Постарайтесь разбить задание на много мелких подзадач. Не забудьте добавить в этот список тестирование, проверку чистоты кода, написание комментариев, юнит-тесты.

Тестовое задание для программистов не предполагает изысков в дизайне, поэтому используйте стандартные компоненты. Придерживайтесь простого и аккуратного визуального стиля в вашем проекте. Вполне вероятно, что программист, который будет проверять ваше задание, даже не запустит ваш проект, а будет смотреть только код.

Не поддавайтесь желанию как можно раньше начать программировать. Перед тем как начать писать код, будет полезно составить примерный список основных элементов вашего проекта: визуальные компоненты, сервисы, хелперы, утилиты и т. д. Если ваш проект содержит пользовательский интерфейс, то набросайте прототип на листе бумаги – это поможет понять, какие компоненты вам понадобятся. Таким образом, потратив немного времени на планирование, вы сэкономите много времени в процессе разработки.

Теперь пришло время применить свои основные навыки в области программирования. В процессе написания тестового проекта уделяйте большое внимание чистоте и понят-

ности кода. На практике это подразумевает хорошие названия для функций и переменных, добавление комментариев, написание понятных протоколов (интерфейсов). Избегайте запутанных решений и различных хаков. Также не оставляйте бесполезные комментарии и закоментированный код.

После окончания разработки стоит еще раз просмотреть все файлы и проверить код. Будет здорово, если у вас есть знакомый программист, который сможет посмотреть ваш проект. Дополнительно стоит потратить время на написание юнит- и UI-тестов, поскольку в процессе проверки тестового задания на это обращают большое внимание.

Заключительный этап – это публикация и добавление описания. Выложите свой тестовый проект в свой репозиторий на GitHub. Человек, который будет проверять ваше тестовое задание, будет за это очень благодарен. Проверять задания на GitHub на порядок удобнее, чем задания, присланные в виде архива. Не забудьте добавить в ваш репозиторий подробное описание проекта и дополнительные сведения. Дополнительно Также можно добавить скриншоты приложения или видео, демонстрирующее его работу.

Наличие тестовых проектов в вашей репозитории может очень пригодиться в дальнейших поисках работы. Иногда компаниям достаточно посмотреть имеющийся у программиста код на GitHub вместо выполнения нового тестового задания.

Собеседование

Собеседование с техническим специалистом компании — это главный этап в процессе поиска работы. В основе собеседования лежит проверка ваших профессиональных навыков, способности писать качественный код и показывать хорошие результаты. Собеседование состоит из нескольких частей и, как правило, включает в себя следующие этапы: знакомство и обсуждение опыта работы, проверка знаний языка программирования, проверка прикладных знаний, решение задач. Этапы могут отличаться в различных компаниях. Где-то вас могут спросить только про прикладные навыки, а где-то заставят решать задачи на алгоритмы и структуры данных.

Постарайтесь быть вежливым и не забывайте улыбаться, даже если вам хамят. Важно, чтобы о вас сложилось хорошее впечатление вне зависимости от результатов собеседования. Такой подход добавит вам дополнительные баллы при принятии решения о вашем найме, поскольку мало кому хочется работать с токсичными людьми.

Знакомство и предыдущий опыт

Первая часть собеседования самая простая: вам предстоит коротко рассказать о своем опыте и проектах, над которыми вы работали. Заранее подготовьте небольшую самопрезентацию на 2—3 минуты. Особое внимание стоит уделить

наиболее актуальному опыту, полученному в последний год. Постарайтесь построить свой рассказ на технических данных и вашем вкладе в развитие проекта. Интервьюеру интересно услышать о том, какие сложные технические проблемы вы решали. Не стоит много рассказывать о продуктовых функциях приложения и прочих деталях, не связанных с вашей основной деятельностью.

Технические вопросы

Далее вам предстоит ответить на вопросы, связанные с языком программирования и наиболее популярными фреймворками. Для подготовки к этому этапу поищите в поисковике информацию по запросу [*название вашего языка программирования*] + *вопросы на собеседовании*. Предстоит подготовить ответы на эти вопросы, чтобы увеличить свои шансы на получение работы. Сходив на несколько собеседований, вы поймете, какие темы в основном интересуют интервьюеров. Я встречал парней, которые ходили на десятки собеседований и уже знали ответы на практически все вопросы. Однако опытный интервьюер в такой ситуации начинает задавать более глубокие вопросы. Поэтому просто зазубрить ответы без общего понимания и практического использования не получится.

Следующая часть собеседования – это архитектура, паттерны и общие принципы разработки. Вас обязательно спросят о том, какую архитектуру вы использовали в своих при-

ложениях, какие есть плюсы и минусы этой архитектуры. Также будьте готовы рассказать про принципы SOLID и привести примеры из реальной жизни.

Достаточно часто в процессе собеседования просят решить какую-нибудь несложную задачу. Задачи бывают двух типов: алгоритмические и прикладные. Алгоритмические задачи направлены на проверку знаний в области компьютерных наук. Например, вас могут попросить развернуть связанный список или определить, является ли строка палиндромом. Для получения базовых знаний в этой области можно прочитать книгу «Грокаем алгоритмы». В ней просто и понятно донесены основные идеи, связанные с алгоритмами и структурами данных. В дальнейшем эти знания пригодятся вам для поиска работы в США. В российских компаниях решение алгоритмических задач не является основным показателем кандидата в отличие от собеседований в США. Если вы хотите попрактиковаться в решении задач, то можете бесплатно это сделать на сайте leetcode.com.

В конце технического собеседования у вас будет возможность задать свои вопросы интервьюеру. Стоит подготовить вопросы заранее – это поможет вам понять, как устроена компания изнутри. Как правило, полезно спросить, какие технологии используются, как устроен процесс разработки и работа в команде. Задавая вопросы, вы покажете свою заинтересованность в работе именно в этой компании.

Конец ознакомительного фрагмента.

Текст предоставлен ООО «ЛитРес».

Прочитайте эту книгу целиком, [купив полную легальную версию](#) на ЛитРес.

Безопасно оплатить книгу можно банковской картой Visa, MasterCard, Maestro, со счета мобильного телефона, с платежного терминала, в салоне МТС или Связной, через PayPal, WebMoney, Яндекс.Деньги, QIWI Кошелек, бонусными картами или другим удобным Вам способом.