

1. ЯП и их применение

Наиболее популярными направлениями программирования являются в наше время следующие направления:

- *Веб-разработка;*
- *Разработка мобильных приложений;*
- *Разработка компьютерных игр;*
- *Разработка приложений для десктопов.*

Нельзя сказать, что за каждым из них жестко закреплены те или иные языки программирования. Например, на Java, C# можно успешно писать программы любой из перечисленных категорий. В то же время определенная специфика, безусловно, есть.

1.1 Веб-разработка

В последние годы, когда сеть продолжает развиваться, появилось несколько различных потоков веб-разработки:

Frontend (также известная как веб-разработка на стороне клиента) касается части веб-интерфейса, с которой пользователь взаимодействует - часть, с которой знакомы все, кто пользуется Интернетом.

Backend (или веб-разработка на стороне сервера) охватывает все те вещи, которые работают в фоновом режиме, чтобы создавать веб-сайты или веб-приложения, например базы данных и скрипты.

Есть **Fullstack** разработчики, которые могут сделать все вышеперечисленное.

1.1.1 Front

JavaScript, CSS и HTML - вместе они составляют святую троицу разработки веб-интерфейса. HTML (Hyper Text Mark Up Language) - это язык веб-браузеров - с помощью которых сделаны сайты.

CSS (каскадные таблицы стилей) заставляет их выглядеть стильно и со вкусом - гораздо лучше, чем те ужасные сайты, действующие с первых дней работы в Интернете. Для разработчиков веб-интерфейсов очень важно знать эти инструменты от и до. Они также окажутся полезными для backend разработчиков: вы сможете понять, как изменения вашей серверной части влияют на конечного пользователя.

SASS превратил CSS в приличный язык программирования, представленный в виде препроцессингового движка, реализующего в таблицах стилей вложенность, переменные, миксины, расширения (extends) и

логику. Так что вы можете лучше организовать свои CSS-файлы, и вам доступны несколько способов разложения больших кусков CSS-кода по более мелким файлам. В своё время это стало прекрасным нововведением.

JS- Язык интерфейса, используемый для создания и разработки веб-сайтов, настольных приложений и игр. JavaScript работает во всех браузерах и может работать с программами, которые не размещены в Интернете. Он поддерживает как функциональные, так и объектно-ориентированные стили программирования, и в основном, это ваш подход к созданию потрясающих пользовательских интерфейсов и веб-сайтов / приложений / игр, которые выглядят супер круто. Понимание JavaScript важно, даже если вы выберете серверной части. Компоненты, структуры данных и алгоритмы JavaScript применяются практически к любому другому языку.

TypeScript — это расширенная версия JavaScript, главной целью которого является упрощение разработки крупных JS-приложений. Этот язык добавляет много новых принципов — классы, дженерики, интерфейсы, статические типы, что позволяет разработчикам использовать разные инструменты, такие как статический анализатор или рефакторинг кода.

1.1.2 Back

Хотя название порой путают с JavaScript, **Java** — не просто другой язык, а почти противоположность JavaScript: это более консервативный выбор, проверенный годами. Его предпочитают использовать в бэкенде тех проектов, для которых надёжность куда важнее веяний моды, — например, банковских систем. В рейтинге популярности языков TIOBE Java занимает первую строчку, а это значит, что в мире есть миллионы рабочих мест Java-разработчиков.

Python — прогрессивный и популярный

Если сравнивать Python с Java, то он будет выигрывать в объёме кода, но проигрывать в скорости выполнения. Для большого высоконагруженного проекта, где важна производительность, Python будет не лучшим выбором. Язык отлично подойдёт для стартапа: например, чтобы показать инвестору прототип до того, как закончатся все инвестиционные деньги. В 2016 году язык неожиданно получил ещё одно преимущество: на нём удобно писать ботов для мессенджеров.

PHP — популярный и относительно несложный

Популярность PHP бесспорна: его используют Facebook, «ВКонтакте», «Википедия» и, к слову, vc.ru. На нём также основана система WordPress, под управлением которой работает около четверти всех сайтов. PHP лёгок в освоении, однако у него спорная репутация, и две эти вещи связаны. Из-за простоты языка в индустрии появилось много новичков с непреодолимым желанием профессионально заниматься PHP-разработкой, не вникая в

детали. Так появилось много плохого PHP-кода. Именно из-за этого одни люди критикуют PHP, а другие возражают, что в умелых руках он показывает себя только с лучшей стороны.

1.1.3 Full

Бэкэнд-код пишется на разных языках программирования. Ruby, PHP, Python, Java и JavaScript — самые распространенные. Бэкэнд, помимо написания кода, на котором сайт работает, часто включает в себя и то, что принято называть задачами «системного администратора», к которым относятся установка серверных инструментов, настройка баз данных и т.д.

Масштабируемая, простая и сверхбыстрая, **Ruby** и **Ruby on Rails** - это дуэт мечты, который предлагает язык full stack, а также рамки для быстрого создания полных программ. Любимый среди предпринимателей и начинающих, Ruby имеет широкий выбор сторонних надстроек, которые могут заставить его делать практически все, что вам нужно. Twitter и Basecamp используют Ruby - неплохой показатель, не так ли?

1.2 Мобильная разработка

рейтинг IEEE включает 55 из 300+ языков, использующихся на GitHub. Выборка была проранжирована по частотности, с которой тот или иной язык встречается на 8 различных площадках и сайтах, например StackOverflow, Google Trends и Reddit. Также к языкам привязаны теги, обозначающие сферу их применения.

Лидерами в области мобильной разработки остаются [Java](#), [C](#) и [C++](#). Google продолжает навязывать разработчикам [Kotlin](#) как ключевой язык Android-разработки, хотя ему вряд ли удастся обогнать Java в ближайшее время, отмечает Dice. Apple вместо [Objective-C](#), до недавнего времени доминирующего языка iOS-разработки, активно продвигает [Swift](#). Поддержка компаний-создателей этих экосистем обеспечивает двум более молодым языкам высокое место в списке.

Android

Java

Официальный и самый популярный язык программирования, который широко используется при мобильной Android-разработке. Впрочем, язык популярен и без Android. Он в топе рейтинга TIOBE, один из первых по числу исходников на GitHub и т. д. и т. п. Именно поэтому, если вы планируете стать Android-разработчиком, изучение Java должно стать одной из приоритетных задач в вашем расписании.

Да, язык непростой, но он стоит потраченных усилий. К тому же, многие утверждают, что сложно добиться каких-либо значимых успехов в Android-разработке, совершенно не зная Java.

Kotlin

Отличный язык программирования, а в связке с «Джава» он ещё лучше. Его популярность растёт, и он тоже является официальным языком для разработки приложений на Android. Мало того, в 2019 году компания Google назначила его приоритетным языком для платформы Android.

Kotlin совместим с Java, имеет больше синтаксического сахара и считается более лёгким для начинающих. Прекрасный язык программирования для тех, кто учится разрабатывать под Android. Но учтите, что некоторые специалисты считают, что знать Java всё же необходимо (статью о Kotlin и Java вы можете почитать [здесь](#)). Как бы там ни было, с помощью Kotlin вы не будете чувствовать особых проблем при создании нативных Android-приложений.

C/C++

Говоря по правде, Google предоставляет разработчикам 2 среды разработки: SDK для Java и [NDK](#), где нативные языки программирования — это C/C++. Да, вы не напишете полноценное приложение, используя лишь эти языки, однако с их помощью сможете создать библиотеку, которую потом подключите при помощи Java к основному телу программы.

К сожалению, большинству Android-разработчиков нет дела до NDK. Однако этот инструмент позволяет получать улучшенную производительность, а это тоже немаловажно, особенно когда мы говорим о хорошей реализации хорошей идеи на Android.

C

Многие скептически относятся к продуктам Microsoft, но C# этого точно не заслуживает. Язык программирования хорош и вобрал в себя лучшее от Java, устранив многие недостатки.

Что касается Android-разработки, то к вашим услугам такие функциональные среды, как [Visual](#) и [Xamarin Studio](#). Кроме того, знание C# станет отличным бонусом, если вы доберётесь до Unity 3D. А вот с этим набором ваши возможности точно станут безграничны.

Python

Да, Android не поддерживает применение Python при создания нативных приложений. Но нет ничего невозможного, ведь существует много инструментов, позволяющих скомпилировать код в нужное состояние. Например, следует отметить фреймворк Kivu, который поможет создать приложение для Play Market.

Lua

Этот язык старше Java, но он не так популярен, хотя имеет динамическую типизацию, простой синтаксис и другие плюсы. Он используется при разработке игр, а также мобильных кроссплатформенных приложений, работающих на Android и не только. Для мобильных приложений существует специальная среда разработки — [Corona SDK](#), где Lua — главный инструмент. Среда распространяется бесплатно, подходит для начинающих разработчиков и имеет много полезной информации и в англоязычном, и в русскоязычном сегментах сети.

IOS

Разработка приложений под iOS может быть кросс-платформенной или нативной. Первый способ позволяет использовать код на Java, C++, Python и других языках — просто компилируете приложение под iOS, и оно будет работать.

Второй — писать на Objective-C или Swift — языках, которые лучше всего подходят для создания приложения под операционные системы Apple.

Нативная разработка для iOS

Нативные приложения отлично работают без всяких костылей, потому что используются языки, которые подходят для конкретной платформы. В случае с iOS — Objective-C и Swift.

Objective-C

В 1980 году Брэд Кокс в своей компании Stepstone создал Objective-C — позже он стал основным языком разработки для устройств Apple. Objective-C — это компилируемый объектно-ориентированный язык программирования (ООП), основанный на C.

С помощью объектов решается проблема повторяемости кода. Но делается это не так топорно, как при структурном программировании, в котором увеличение объема и сложности программы снижает качество.

Основное отличие Objective-C от C++ (тоже поддерживает ООП и тоже был разработан в восьмидесятых годах) в том, что вызов метода в нем интерпретируется не как функция, а как сообщение — как в языке Smalltalk. Такие сообщения могут содержать что угодно, а отправить их можно куда потребуется.

Также объект может переслать часть полученного сообщения другому объекту (даже если он находится не на этом компьютере), что позволяет делегировать задачи — по такому же принципу работают нейросети.

Objective-C — компилируемый язык. Однако сообщения привязываются к функциям исключительно во время выполнения программы, что позволяет создавать динамичные приложения. Еще в языке поддерживаются протоколы и метаянформация.

Swift

Крис Латтнер, главный автор LLVM, вдохновившись Objective-C, Rust, Ruby, C#, Haskell и другими языками, в 2010 году начал разработку Swift, а в 2014-м уже выпустил первую версию.

Swift изначально заточен под операционные системы Apple, поэтому он производительнее и функциональнее, чем Objective-C, для работы с которым использовалась подключаемая runtime-библиотека. В 2015 году вышла версия 2.0, которая была еще быстрее, получила API для работы с ошибками и проверку доступности функций для целевых ОС.

Синтаксис изменился, и теперь программы больше похожи на код C-подобных языков — с той лишь разницей, что после операций не ставится символ «;»

1.3 Разработка ИГР

Разработка игр - творческая работа, требующая серьезной технической подготовки и компьютерной компетентности. Стоимость создания одной игры может достигать нескольких миллионов долларов. Доверить такое дело можно лишь высококвалифицированным и опытным программистам.

В зависимости от требований к игре, разработчик нуждается в том или ином языке программирования для работы.

Даже самые увлеченные разработчики игр порой сталкиваются с трудностями при выборе языка программирования для создания новой игры. Несмотря на то что почти все языки программирования наподобие C++ и Python подходят для этой цели, некоторые являются более производительными в сравнении с остальными.

Кроме того, выбирая язык, нужно всегда учитывать цель своей разработки и то, на какую аудиторию рассчитана игра.

Выбранный язык повлияет и на саму структуру игры, и на игровой движок.

Все вышеперечисленное усложняет выбор. По этой причине к каждому языку программирования, упомянутому в статье, будет приложена информация о его сильных сторонах, которыми следует воспользоваться при разработке.

1 - C#



Один из самых популярных языков среди разработчиков игр в наше время. Используется с различными игровыми движками. Новичку будет

не так легко сразу освоить его, при разработке есть шанс столкнуться с трудностями. В таком случае можно использовать справку по домашнему использованию C#.

Язык использует XNA - набор инструментов для разработки игр от Microsoft. По этой причине C# лучше всего подходит для игр на Xbox или Windows.

2 - C++

Является объектно-ориентированным языком. Разработчики регулярно прибегают к C++ при создании игр, невзирая на его относительную сложность. Язык поддерживает более автоматизированный контроль над аппаратным обеспечением компьютера. Он - основа известнейших игровых движков, поскольку предлагает широкий диапазон функций в управлении всеми параметрами компьютера, включая память. Эти функции улучшают производительность игр, делают взаимодействие пользователя с игрой более удобным.

3 - Java

Java так же, как и C++, использует объектно-ориентированное программирование, однако предоставляет для него больше возможностей. Коды языка работают на Java Virtual Machine, которая исполняет байт-код Java. Следовательно, разработчики имеют возможность создавать игры для любых систем. Язык принадлежит к числу лучших языков программирования для создания качественных игр.

4 - JavaScript

Лучше всего подходит для интерактивных онлайн-игр, поскольку дает возможность интегрировать код в других языках программирования, например в HTML и CSS. Такая межплатформенность приводит к созданию все большего числа мобильных игр.

5 - HTML 5

Наиболее распространенный язык для веб-программирования, который используется во многих мобильных играх. Благодаря интеграции JavaScript, HTML5 может помочь создать замечательную игру. Очень прост в изучении, так как не требует специальных навыков программирования.

6 - Python

Python имеет собственный набор модулей Pygame, что позволяет разработчиками моделировать свои будущие игры.

7 - Rust

Считается одним из преемников языков C и C++. Представляет из себя язык системного программирования. Был разработан сообществом Mozilla и в данный момент спонсируется фондом Mozilla Foundation. В Rust используются приемы объектно-ориентированного программирования.

8 - UnrealScript

Очередной объектно-ориентированный язык. Множественное наследование не поддерживает. Был создан специально под известный движок Unreal engine от Epic Games. Поддерживает все основные игровые платформы, включая Microsoft Windows, macOS, Linux, SteamOS, Android, и PlayStation VR.

9 - Lua

Lua - доминирующий язык в игровой индустрии. Он имеет несложную языковую структуру и представляет собой мультиплатформенный сценарный язык.

Многие новейшие игровые движки используют Lua в качестве основного языка программирования для разработки игр.

Лучший язык для внутреннего уровня любой базы данных. Позволяет игроку авторизоваться и совершать другие операции на сервере. Язык использует все лучшие достижения других языков программирования, включая AR и VR, графику, физику и игровой процесс.

1.4 Разработка приложений для десктопов.

Дело в том, что на рынке доступно огромное количество операционных систем: от Windows до macOS на настольных компьютерах. Разработчикам становится все труднее и труднее не только оптимизировать, но и кодировать приложения для этих платформ в первую очередь.

Программировать приложения можно практически на любом языке. Зависит от архитектуры, функций и требований к приложению. Например, C++ подойдет для сложных задач, но писать на нем долго, а внесение даже небольших изменений займет много времени, больше подходит для низкоуровневых операций например для работы с памятью. Java проще, но требует больше ресурсов от железа, но значительно упрощает процесс портирования на другую платформу, плюс многопоточные приложения в некоторых случаях могут работать эффективней. Еще один востребованный вариант – Python. Относительно прост и подходит для небольших и кроссплатформенных приложений. Поэтому, нужно взвешивать, что важнее: скорость реализации проекта или итоговая производительность. А еще нужно учитывать для какой платформы ведется разработка. Например, если это macOS, то лучше использовать Swift и Objective-C. Конечно, Desktop программист умеет работать со средой разработки, базами данных, баг-трекерами и системами контроля версий.

Дополнительно

Data Science разработка

Чем занимается программист data science?

Программирует прогнозные модели, учит машину искать закономерности, анализировать данные и делать предположения на основе этого. Другая сторона работы заключается в программировании приложений, которые будут собирать эти данные и передавать на сервер. Примеры реализации этих задач: алгоритмы поиска; системы рекомендаций; прогнозы погоды; предсказание изменения курса; сбор и анализ данных сна.

Какие языки и технологии использует?

Data Scientist понимает принципы математической статистики, знаком с методами машинного обучения, deep learning. Понимает, как работают базы данных, умеет записывать и извлекать информацию. Языки, используемые в анализе Big Data – Python, R, SAS. R – язык, который часто используется в науке для статистического анализа и визуализации данных исследований. В Python есть библиотеки для машинного

обучения Scikit-learn и анализа данных Pandas. SAS менее производителен, чем R, плюс не востребован в РФ, но хорош для аналитических и прогностических задач.

Embedded-разработка

Чем занимается?

Embedded-разработка лежит на стыке инженерии и программирования, то есть предполагает работу и с кодом, и с железом. Embedded Developer пишет ПО, которое помогает работать телевизорам, квадрокоптерам, предметам умного дома, медицинскому оборудованию и даже IQOS. Для этого он: продумывает логику работы готового продукта; выбирает алгоритмы решения задач в соответствии с возможностями аппаратной платформы и функциями, которые должен в итоге выполнять программируемый агрегат; программирует чипы и микроконтроллеры, по сути «оживляет» железо; тестирует написанный код и ответ на него программируемого устройства, ищет ошибки.

Какие языки программирования и технологии использует? Embedded Developer хорошо знает физику и электронику, программирует на C или C++. Инженер встроенного ПО знает язык ассемблера и разбирается в электронных схемах, микроконтроллерах, микропроцессорах. Он знаком с методами встроенного программирования, цифровой безопасности, моделях передачи данных между устройствами. Умеет управлять конфигурацией программного обеспечения на основе систем контроля версий. Иногда не лишним будет умение паять и создавать компоненты железа самостоятельно

Какие языки программирования и технологии использует? Embedded Developer хорошо знает физику и электронику, программирует на C или C++. Инженер встроенного ПО знает язык ассемблера и разбирается в электронных схемах, микроконтроллерах, микропроцессорах. Он знаком с методами встроенного программирования, цифровой безопасности, моделях передачи данных между устройствами. Умеет управлять конфигурацией программного обеспечения на основе систем контроля версий. Иногда не лишним будет умение паять и создавать компоненты железа самостоятельно.

Automation QA engineer

Чем занимается такой разработчик?

Профессия на стыке программирования и тестирования. Automation QA engineer продумывает логику автоматизированных тестов для ПО и посредством программирования реализовывает ее на практике. То есть,

пишет программы, которые будут искать баги в других программах, сайтах, выполнять нагрузочные, интеграционные тесты и т.п. Для этого программист разрабатывает сценарии, описывает тест-кейсы, создает фреймворки, обеспечивает интеграцию тестов с сервисами автоматического запуска. Automation QA инженеры

Какие языки программирования и технологии использует Automotion QA?

Java – один из самых популярных языков для Automation QA. Но используется и Python, JavaScript и другие языки. ООП Специалисту, который планирует работать с вебом, важно учить язык в связке с инструментами для автоматизации действий браузера, таких как Selenium WebDriver, например. Знает несколько unit frameworks для быстрого создания модульных тестов, умеет работать с базами данных. Но прежде всего, Automation QA engineer понимает теорию тестирования, умеет проводить его вручную.

DevOps engineer

Чем занимается программист в DevOps?

DevOps engineer – это специалист, который объединяет работу программистов, создающих продукт, тестировщиков, команды интеграции продукта и защищает от перебрасывания ответственности за работающий готовый продукт между ними. DevOps в целом – это методология, которая позволяет сделать разработку и внедрение ПО максимально быстрой, эффективной и безопасной. Для этого DevOps engineer: разворачивает программу на сервере; организует среду разработки и тестирования ПО; стандартизирует окружение разработки; контролирует работоспособность инструментов разработки; автоматизирует процессы и создает своеобразный конвейер продакшена.

Какие языки программирования и технологии использует?

В качестве языка программирования чаще используется Python. Он довольно простой в изучении и подходит для требований автоматизации и управления инфраструктурой. Если принципиально он вам не подходит, можно изучить Go или Ruby. Также DevOps engineer знает: операционную систему, чаще Linux; один или несколько инструментов для управления конфигурацией например Puppet, Chef, Ansible; Docker + облачный сервис для оркестрации контейнеров, например, Amazon Web Services или Kubernetes; системы контроля версий и управления конфигурациями.

Программист 1с

Чем занимается?

1С программист занимается автоматизацией бизнес-процессов. На практике это включает в себя: установку и обслуживание программы семейства 1С; доработку ее под нужды конкретного бизнеса, например, сделать новый аналитический отчет или печатную форму; обучение пользоваться 1С бухгалтеров, кладовщиков и т.п; разработку логики и программную реализацию конкретной бизнес-функции, например, организовать рассылку постоянным клиентам или запланировать закупку на основании аналитики продаж предыдущих периодов.

Какие языки программирования и технологии использует?

На территории бывшего СНГ 1С является самым распространенным пакетом программ для бухгалтерского, налогового и управленческого учета. У этих программ собственный язык и среда разработки. Кроме языка программирования программист 1С знает умеет работать с базами данных, владеет основами настройки работы принтеров, компьютеров. Обязательно разбираться в азах бухучета и бизнес процессов. Сам 1С написан на C++. Компанию часто ругают за внутренний проприетарный язык. Но отметим, что зарубежный конкурент 1С, SAP, также реализован на внутреннем языке ABAP/4.