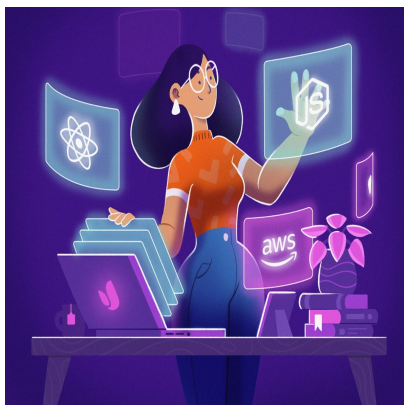




14 Языков программирования для разработки мобильных приложений

Описание

Несколько лет назад существовали только два способа создания мобильного приложения — один код для приложений iOS и один код для Android. Но сегодня у разработчиков значительно больше возможностей для кодирования приложений. Я создал это руководство, чтобы помочь вам разобраться в самых популярных языках программирования и фреймворках для разработки мобильных приложений. Вы можете использовать эту информацию, чтобы решить, какой язык изучать и инвестировать в ваше мобильное приложение.

Для целей этого ресурса я опущу описание стандартных типов разработчиков мобильных приложений. Технически, вам не нужно изучать, как использовать языки программирования code use для такого типа платформы разработки мобильных приложений. Ознакомьтесь с нашим руководством по [пяти способам создания приложений](#) для получения дополнительной информации об этих альтернативных методах.

Типы мобильных приложений

Прежде чем мы продолжим, я просто хочу быстро рассказать о различных типах разработки приложений с точки зрения программирования. Вообще говоря, разработчики мобильных приложений могут создавать приложения в одной из этих трех категорий:

Собственные приложения

Разработка собственных приложений написана на языке, который изначально поддерживается конкретной операционной системой мобильных устройств. (Пример: родное приложение для iOS против родного приложения для Android). Это используется, если вы создаете приложение специально для Apple App Store или Google Play Store.

Это идеальное решение для приложений с высоким уровнем настройки, которые должны использовать собственные компоненты каждого устройства. Он отлично подходит для игровых приложений, приложений виртуальной реальности и приложений с расширенной графикой. Но один код не будет работать на обеих платформах с собственной разработкой.

Гибридные приложения

Гибридные приложения предназначены для кросс-платформенной разработки. Они написаны на одном языке, который может работать на нескольких как на iOS, так и на Android. Это ускоряет сроки разработки, поскольку вам придется кодировать все только один раз, а не дважды (по одному разу для каждой ОС). По сравнению с нативным вы немного потеряете гибкость в плане того, что вы можете делать с гибридными приложениями. Но это нормально для подавляющего большинства разработчиков.

Самая распространенная тема, которую вы услышите при обсуждении разработки мобильных приложений, – это разница между нативными и гибридными приложениями. Это стало большой дискуссией для разработчиков в течение достаточно долгого времени. Как я кратко упоминал выше, собственные приложения создаются для конкретной операционной системы. Итак, если вы хотите разработать приложение для iOS и Android, вам нужно будет создавать отдельно для iOS и отдельно для Android. У этого метода, как и у других, есть свои плюсы и минусы. Мы подробнее рассмотрим преимущества и недостатки этих методов разработки приложений по мере прохождения этого руководства.

Приложения PWA

PWA (progressive web app) – это легкое приложение, которое запускается по URL-

адресу веб-браузера устройства. Это выглядит и ощущается как мобильное приложение, но оно не поставляется на устройстве изначально. Разработчики, имеющие опыт веб-разработки, могут легко создавать PWA. Вы уже должны быть знакомы с языками программирования, используемыми в процессе разработки. Так что кривая обучения не так уж велика. С учетом сказанного, эти приложения будут немного более ограничены с точки зрения возможности использования собственных компонентов устройства.

Лучшие языки программирования для разработки приложений

Давайте подробнее рассмотрим 14 лучших языков программирования для мобильных приложений. Каждый вариант имеет свои преимущества и недостатки в зависимости от вашего уровня квалификации и предполагаемого варианта использования. Независимо от вашего оборудования и программного обеспечения, ниже вы можете найти программистов и языки, соответствующие вашим потребностям.

Языки программирования iOS

Платформа iOS была создана Apple. Если вы разрабатываете приложение для iOS, оно будет работать в экосистеме Apple, как iPhone и iPad. Приложения, созданные с использованием языка программирования iOS, могут быть доступны в Apple App Store для загрузки пользователями. Для создания приложения для iOS вам необходимо иметь учетную запись разработчика Apple, чтобы начать работу. Вам также потребуется IDE Xcode, установленная на компьютере Mac (вы не можете правильно создавать и отлаживать на компьютере с Windows). В Xcode есть все необходимое для создания приложений для всех устройств Apple. В этом наборе инструментов для разработки есть редактор кода, симуляторы, отладчик и пакеты SDK. Для разработки iOS существует два родных языка программирования — [Objective-C и Swift](#).

Давайте подробнее рассмотрим каждый из этих языков программирования Apple ниже.

Objective-C

Objective-C был первым языком программирования Apple для поддержки мобильных приложений на своей платформе. Это ОО (объектно-ориентированный язык программирования), который использует синтаксис из C и объектно-ориентированные аспекты SmallTalk. Язык не очень удобен для разработчиков. Одним из недостатков является то, что синтаксис кажется неуклюжим, а квадратные скобки могут быть сложными для отладки.

Swift

Swift был представлен в 2014 году как язык программирования Apple. В конечном итоге он был доступен для разработки в Xcode в следующем году. Этот язык быстро стал предпочтительным языком разработчиков при создании приложения для iOS. Если вы хотите разрабатывать приложения для iOS, это один из лучших языков программирования для рассмотрения. Использование Objective-C сократилось с момента появления Swift. Для любых современных приложений, созданных на базе Apple, настоятельно рекомендуется использовать Swift. По сравнению с Objective-C и другими языками программирования Swift проще и компактнее. У любого разработчика Apple, который уже знает, как создавать с помощью Objective-C, не должно возникнуть проблем с переключением на Swift.

Языки программирования Android

Android – это платформа разработки программного обеспечения с открытым исходным кодом, которой управляет Google. В то время как у Google есть свои собственные варианты мобильных устройств для телефонов и планшетов, другие производители, такие как Samsung, Huawei, Microsoft и другие, также выпускают телефоны и планшеты, работающие на ОС Android.

Чтобы создать приложение для Android, вам необходимо получить инструментарий разработки Android, который содержит отладчики, эмуляторы и необходимый SDK. Лучшей интегрированной средой разработки (IDE) для разработки приложений Android является Android Studio. Это упрощает разработку мобильных приложений для Android. Существуют и другие доступные варианты, включая кроссплатформенные инструменты разработки приложений (для создания приложений для Android и iOS, а также мобильных приложений для других операционных систем), но Android Studio, безусловно, является самой популярной для разработки приложений для Android.

IDE Android обычно можно запускать на любой ОС, включая Windows, Mac и Linux.

Давайте подробнее рассмотрим два языка программирования Android — [Java и Kotlin](#).

Java

С момента официального запуска Android в 2008 году Java стала языком разработки по умолчанию для написания приложений для Android. Этот объектно-ориентированный язык был первоначально создан еще в 1995 году. Хотя Java имеет свою изрядную долю недостатков, она по-прежнему остается самым популярным языком для разработки под Android, поскольку работает на виртуальной машине. В качестве объектно-ориентированного варианта мобильной разработки Java обычно используется для разработки приложений для Android.

Большинство других языков Android считаются версией Java или разновидностью Java.

Kotlin

Google объявила, что начнет поддерживать язык программирования Kotlin в 2017 году. Это язык, альтернативный традиционному Java для разработки под Android, и он работает на виртуальной машине Java. Даже будучи новым языком, он очень популярен. Kotlin и Java совместимы, что означает, что они могут использовать одну и ту же информацию. Доступ ко всем вашим библиотекам Java можно получить с помощью Kotlin. С точки зрения выполнения язык Kotlin соответствует байт-коду Java. В целом, это считается более аккуратной и чистой версией Java.

Языки программирования для собственной разработки

Как я уже говорил, все языки программирования предназначены для нативных мобильных приложений. есть свои плюсы и минусы. Используете ли вы Objective-C или Swift для Apple или используете Java или какую-либо другую версию Java (например, Kotlin) для Android, это преимущества и недостатки.

Плюсы нативного программирования:

- Максимальный контроль над устройством
- Низкоуровневое кодирование для передовых технологий, добавляемых к устройству
- Быстрый доступ к новейшим и лучшим функциям на вашем языке
- Самый быстрый результат в исполнении

Недостатки нативного программирования:

- Самый медленный для развития
- Самый дорогостоящий метод разработки
- Для создания мобильных приложений для iOS и Android требуются высококвалифицированные и специализированные разработчики мобильных приложений
- Высокий барьер для входа

В то время как родные языки программирования дают вам наибольший контроль над вашим приложением, они сложны в освоении и требуют много времени для разработки. Если вы не создаете узкоспециализированное приложение, вам, вероятно, не нужно будет идти по собственному пути.

Гибридные языки программирования

[Гибридные приложения](#) разрабатываются один раз, но пишутся с использованием языка программирования, который работает для нескольких платформ. Чаще всего одна разработка будет работать как для iOS, так и для Android. Хотя некоторые гибридные языки расширяют свою функциональность на другие платформы, такие как progressive web apps (PWA) или мобильные веб-приложения. Это приятно иметь

для тех из вас, кто переходит в более дружелюбную к интернету среду. Когда вы создаете гибридное приложение или хотите разрабатывать кроссплатформенные приложения, вы обычно имеете дело с каким-то языком, фреймворком или инструментарием на основе JavaScript. Гибридные приложения работают на разных мобильных устройствах. Давайте подробнее рассмотрим некоторые из этих вариантов ниже.

C#

Разработанный Microsoft, C # (произносится как C sharp) является еще одним объектно-ориентированным языком программирования. Это популярный язык программирования для разработки игр и написания сценариев командной строки для операционных систем Android. Другие альтернативы с низким уровнем кода, такие как OutSystems и Kony, имеют SDK, который можно использовать с разными языками, а не только с одним. Существуют и другие языки для разработки мобильных приложений, которые используют язык системного программирования с синтаксисом, похожим на C #. Используя IDE для гибридной разработки, код C # кросс-компилируется для запуска на устройствах iOS и Android.

Xamarin

В конечном итоге Microsoft приобрела платформу Xamarin, которая позволяет разработчикам приложений программировать с использованием C # в сравнении с другими платформами. Технически Xamarin не является языком. Это платформа разработки с открытым исходным кодом для приложений iOS, Android и Windows. Это платформа .NET, которая использует C # в качестве основного языка.

React Native

Reactive Native использует JavaScript для взаимодействия с готовой функциональностью, которая является родной для их фреймворка. Это позволяет вам управлять пользовательским интерфейсом, собирать данные и извлекать данные, чтобы вы могли представить их пользователю. По сути, это означает, что вы в значительной степени полагаетесь на JavaScript для управления собственными компонентами. Программирование мобильного приложения таким способом имеет свои плюсы и минусы.

Ускоритель приложений

Как и Reactive Native, Appcelerator также использует JS для взаимодействия с функциями в собственной среде. Appcelerator Titanium позволяет создавать собственные приложения для iOS, Android и Windows.

Преимущества Appcelerator:

- Использование Java для манипулирования чем-то родным
- Доступ к встроенным функциям непосредственно из JavaScript

Минусы Ускорителя приложений:

- Не использует такие вещи, как HTML5 или CSS, которые являются технологиями, которые обычно используются кем-либо, разрабатывающим на JavaScript, для свободного управления собственным пользовательским интерфейсом

Барьер для входа в изучение этого метода программирования находится примерно на среднем уровне.

Кордова / PhoneGap

Фреймворки Cordova / PhoneGap и Ionic type на самом деле просто построены поверх языка программирования Apache Cordova. Разработка гибридных мобильных приложений с использованием этого языка программирования строится путем переноса веб-интерфейса в собственный интерфейс.

Что это значит?

Этот метод позволяет вам создавать точно так же, как вы делали бы это для веб-сайта. Так что, если вы веб-разработчик, вы будете чувствовать себя здесь как дома. Он использует Javascript, HTML и CSS. Эта веб-среда изначально перенесена на iOS и Android. Окончательное программное обеспечение будет работать на нескольких типах мобильных устройств.

Плюсы:

- Низкий барьер для входа

- Любой, у кого есть опыт веб-разработки, может легко начать программировать таким образом
- Кривая обучения проста

Минусы:

- Отказ от небольшой частоты кадров

Если вы разрабатываете мобильные игры, собственное приложение для машинного обучения или собственное приложение дополненной реальности (AR), это решение, вероятно, не лучший язык программирования для вас. Оба из них требуют более высокой частоты кадров.

Языки программирования PWA

Прогрессивные веб-приложения предлагают функции, похожие на приложения, с мобильного сайта. Это быстрая и надежная альтернатива традиционным мобильным веб-сайтам. В отличие от веб-сайта, веб-приложения могут работать в автономном режиме и получать доступ к собственным возможностям устройства (таким как камера, GPS и т. Д.). Вот некоторые из языков программирования, которые вы можете использовать для PWA:

Ruby

Ruby – это язык программирования общего назначения, который можно использовать для широкого спектра вариантов использования, включая PWA. Многие разработчики полагаются на Ruby для веб-приложений из-за его простоты. Этот язык существует с 1990 года и по сей день остается любимым в сообществе разработчиков. Хотя доставка кода с помощью Ruby проста, найти ошибки и отладки не всегда так просто. Так что просто имейте это в виду, если вы идете этим путем.

Питон

Python – еще один универсальный язык программирования. Это объектно-ориентированный язык, который дает разработчикам гибкость как для небольших, так и для крупномасштабных развертываний. В дополнение к PWA, Python обычно используется для анализа данных, визуализации данных, веб-сайтов,

автоматизации задач и других типов программного обеспечения. Большинство разработчиков согласятся с тем, что Python немного сложнее, чем Java. Он имеет более крутую кривую обучения по сравнению с другими языками этой категории.

CSS

CSS (каскадные таблицы стилей) – это язык, основанный на правилах. Это помогает описать представление кода, написанного на языках разметки, таких как HTML. Вы не можете создать PWA только с помощью CSS. Но вы, скорее всего, будете использовать его как дополнение к своему JS или HTML-коду.

JavaScript

Если у вас есть опыт веб-разработки, использование JavaScript для создания PWA может быть лучшим вариантом для вас. Вы можете использовать JS поверх HTML и наряду с CSS для создания своего веб-приложения с нуля. По сравнению с другими вариантами PWA, этот имеет более низкий барьер для входа для тех из вас, кто обладает некоторыми базовыми техническими знаниями и опытом программирования. Но он не такой гибкий, как Ruby или Python для PWA.

PHP

PHP – это общий язык сценариев, который был впервые представлен в 1994 году. Хотя у него есть несколько потенциальных вариантов использования, PHP также может быть использован для оказания помощи PWA. С учетом сказанного, это не обязательно лучший вариант. Вы можете использовать PHP в бэкенде для приложения, но вы все равно захотите использовать HTML, CSS и JS во внешнем интерфейсе. Использование PHP для простого создания веб-сайта не будет иметь такого же внешнего вида, ощущения и возможностей, как традиционный PWA. Короче говоря, сам по себе PHP не обеспечит создание прогрессивного веб-приложения. Но большая часть вашей работы все равно может быть написана на PHP.

Выбор правильного языка программирования

Как видите, у каждого языка программирования есть свои плюсы и минусы. Вы не можете однозначно сказать, что одно лучше другого. Все зависит от типа

приложения, бюджета, сроков и технических знаний. Нативная разработка, как правило, самая дорогая и сложная в освоении. Но это необходимо для определенных типов приложений, таких как игровые приложения. Гибридные языки легче изучать пользователям, обладающим некоторыми техническими знаниями и опытом веб-разработки. С помощью этого метода вы можете быстро вывести свое приложение на рынок, поскольку вам не придется использовать два развертывания.

Когда использовать приложение для программирования

Приложения для программирования, такие как Grasshopper, Sololearn и Mimo, являются отличными вариантами для начинающих. Так что, если вы только начинаете свой путь в качестве разработчика, вы определенно найдете эти инструменты полезными. Кроме того, вы можете использовать конструктор приложений для создания приложения, не написав ни одной строки кода. Создатели приложений без кода, такие как BuildFire, позволяют любому человеку, независимо от технических знаний, создавать приложения для iOS и Android одновременно.

Если вы не хотите стать разработчиком, использование app Creator – лучший вариант. Время, которое вам потребуется для изучения разных языков программирования, вероятно, не стоит вложений, если вам просто нужно создать одно приложение для вашего бизнеса. Даже если вы научитесь программировать, ваше первое приложение, скорее всего, не будет готово для реальных пользователей.

MBaaS (Мобильный сервер как услуга)

У всех гибридных фреймворков и родных языков программирования для создания мобильных приложений есть две общие черты — все они должны быть созданы с нуля, и во всех им не хватает основного компонента. Любой, кто в прошлом разрабатывал мобильное приложение, понимает, что само приложение – это лишь часть всей среды и общего решения. Вам также понадобится мощный мобильный сервер как услуга, более известная как MBaaS.

Для чего вам нужен MBaaS? Вот несколько примеров:

- Размещайте свои данные

- Профили пользователей хостинга
- Компиляция аналитики
- Отправка push-уведомлений

Этот список можно продолжать и продолжать. Все это серверы, работающие в облаке, которые вам также необходимо разработать для поддержки вашего приложения. Если вы не создаете простое приложение, например калькулятор, вам обычно требуется какой-то тип аутентификации пользователя, база данных, CMS и т.д.

BuildFire JS

Вот тут-то и вступает в игру BuildFire JS. Фреймворк BuildFire JS позволяет вам создавать точно так же, как вы делали бы это на гибридной платформе Cordova. Вы можете использовать веб-технологии, такие как JavaScript, HTML и CSS. Но этот фреймворк не заставляет вас создавать все с нуля. Такие вещи, как аутентификация и push-уведомления, создаются поверх существующей платформы. Эта платформа обладает всеми типичными функциями, необходимыми большинству приложений, такими как логины пользователей, функции сброса пароля, доступ к базам данных, доступ к платформам CMS и так далее.

С помощью BuildFire JS вам нужно создавать только то, что уникально для вашего конкретного приложения. Серверы аналитики, базы данных, серверы push-уведомлений, API-шлюзы и многое другое – все это часть огромного MBaaS, предоставляемого BuildFire. Все это объединено в среду с открытым исходным кодом, которая позволяет людям постоянно добавлять новые функции в платформу. Вы можете интегрировать эти функции в свое приложение, не беспокоясь о проблемах безопасности или лицензировании. После того, как все сказано и сделано, и вы разработали свое приложение с помощью BuildFire, появится внутренняя панель управления, которая позволит вам администрировать приложение по воздуху, не сталкиваясь с препятствиями, связанными с публикацией и обновлениями.

Поскольку ваше приложение построено на платформе с MBaaS, вам не придется беспокоиться о каких-либо новых политиках, правилах, проблемах соответствия требованиям, функциях и многом другом на iOS и Android. BuildFire гарантирует, что ваше приложение остается совместимым.

Заключительные мысли: лучшие языки программирования

Какой язык программирования лучше всего подходит для разработки мобильных приложений?

Нет правильного или неправильного ответа на вопрос, какой язык программирования вам следует изучить или в какой фреймворк вам следует инвестировать. Лучшие языки программирования для меня и моих мобильных приложений могут оказаться не самыми лучшими для вас и вашего сценария разработки приложений. Все варианты, перечисленные в этом руководстве, являются хорошим и обоснованным выбором для рассмотрения. У каждого из них есть свои плюсы и минусы. Существуют даже дополнительные языки программирования для разработки мобильных приложений, такие как Python для программирования на стороне сервера и другие.

Вам просто нужно выяснить, что лучше для вас, вашего бизнеса и ваших целей. Какой тип приложения вы создаете? Что нужно приложению? Куда вы хотите приложить больше всего усилий? Вы хотите разработать его один раз или несколько раз? На каком мобильном устройстве будет находиться конечный пользователь? Вы создаете для нескольких платформ? Как насчет контроля доступа к данным, статистически типизированного языка программирования или интерпретируемого языка программирования?

Вот некоторые из вопросов, которые вам нужно задать себе, чтобы определить, где лучше всего использовать ваше время, усилия и ресурсы. В конце концов, просто убедитесь, что вы можете быстро выйти на рынок с самым лучшим приложением.

Дата Создания

23.01.2023