

УДК 004

ОБЗОР ЯЗЫКОВ ПРОГРАММИРОВАНИЯ ПОД ANDROID И IOS ДЛЯ НАЧИНАЮЩЕГО РАЗРАБОТЧИКА

Ткачев Е.В.

студент,

ФГБОУ ВО «Калужский государственный университет

им. К.Э. Циолковского»

Калуга, Россия

Белаш В.Ю.

к.пед.н., доцент,

ФГБОУ ВО «Калужский государственный университет

им. К.Э. Циолковского»

Калуга, Россия

Аннотация: В данной статье были рассмотрены языки программирования: Java, Kotlin и Swift для смартфонов, построенных на базе операционных систем Android и iOS. Для каждого языка приведены основные возможности, достоинства и недостатки.

Ключевые слова: android, kotlin, java, программирование.

OVERVIEW OF PROGRAMMING LANGUAGES FOR ANDROID AND IOS FOR A NOVICE DEVELOPER

Tkachev E.V.

student,

Kaluga State University named after K. E. Tsiolkovsky

Kaluga, Russia

Belash V.Yu.

Ph.D., Associate Professor,

Kaluga State University named after K. E. Tsiolkovsky

Kaluga, Russia

Abstract: In this article, programming languages were considered: Java, Kotlin and Swift for smartphones built on the basis of Android and iOS operating systems. The main features, advantages and disadvantages are given for each language.

Keywords: android, kotlin, java, programming.

Перед любым начинающим мобильным разработчиком всегда встаёт вопрос: «А на каком языке программирования начать писать приложение?». В этой статье постараемся получить ответ, разобрав главные особенности популярных языков для мобильной разработки под Android и iOS: Java, Kotlin и Swift.

Из всего разнообразия операционных систем на заре появления смартфонов особую популярность и признание получили лишь две операционные системы: Android и iOS. Для разработки программного обеспечения под устройства, которые основаны на системе iOS, Apple официально рекомендует использовать собственный язык программирования Swift, специально разработанный для данной операционной системы. Для создания программного обеспечения для устройств, которые находятся под управлением операционной системы Android, как правило разработчики используют Java или Kotlin. Однако, «если вы запускаете новый проект, вы должны писать его на Kotlin. Код на Kotlin сэкономит ваше время – вам придется меньше печатать, меньше тестировать и меньше поддерживать [приложение]» [1] – с таким обращением Google, создавший Android, обратился к разработчикам, объявляя Kotlin главным языком разработки под систему Android.

Java – разработан в 1995 году компанией Sun Microsystems (на момент написания статьи права принадлежат корпорации Oracle) и вдохновлён языком Objective-C, благодаря чему имеет схожесть с языками из семейства C (C, C++, C#, Objective-C). Отличительной особенностью языка является то, что приложения написанные на Java транслируются в байт-код Java (специальный

Дневник науки | www.dnevniknauki.ru | СМИ Эл № ФС 77-68405 ISSN 2541-8327

набор инструкций), который исполняется специальной виртуальной машиной Java (JVM). Эта особенность позволяет запускать любые приложения написанные на этом языке программирования на любом оборудовании, у которого есть поддержка JVM. Виртуальная машина Java по своей сути является посредником между вашим кодом и устройством, на котором он запущен. Принцип работы с JVM представлен на рисунке 1.

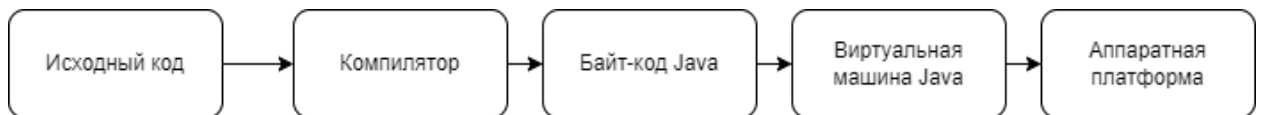


Рис. 1. Принцип работы с JVM

Ко всем виртуальным машинам принято относить низкую производительность, так как использование посредника всегда замедляет работу на конечном устройстве. Однако, оптимизация и обновление языка, использование нативного кода в библиотеках и специальные аппаратные средства, позволяющие ускорить обработку байт-кода на устройстве, дают значительное повышение производительности JVM на некоторых системах.

Команда JetBrains начала разработку Kotlin в 2010 году, а в 2016 была представлена первая предварительная версия языка и всего через год Google заявил, что язык программирования будет официально добавлен в новой версии Android Studio 3.0. Kotlin был создан для работы с виртуальной машиной Java работающим поверх JVM, что означает полную совместимость с языком Java, это позволяет сравнительно легко перейти Java-разработчикам на этот язык программирования. Главная цель при разработке Kotlin была создать более простой и безопасный язык, чем Java и C++.

Основные достоинства и недостатки языка Kotlin представлены в таблице 1.

Таблица 1. Язык Kotlin. Достоинства и недостатки

Достоинства	Недостатки
Лаконичность кода	Непредсказуемая скорость компиляции кода

Простая конвертация кода из Kotlin в Java и наоборот	Небольшое сообщество
Открытый исходный код	Малое количество библиотек для Kotlin
Доступ ко всем библиотекам Java	

Несмотря на заявленную полную совместимость Kotlin с Java, переход существующего проекта может сопровождаться определёнными трудностями.

Swift – это надёжный и интуитивно понятный язык программирования от Apple, при помощи которого можно создавать приложения для iOS [2]. Он был показан в свет в 2014 году и, также как Java и Kotlin, корнями уходит в языки семейства C. Однако, не смотря на то, что он что-то позаимствовал у этих языков, Swift имеет некоторые ключевые различия с ними.

- В Swift нет необходимости в двух отдельных файлах, содержащих заголовок и реализацию, вместо этого всё объединено и помещено в один с расширением `.swift`.

- Использование динамических библиотек вместо статических, что позволяет сократить размер загружаемого приложения.

Также Swift обладает преимуществами молодого языка программирования, а именно: высокой производительностью относительно Objective-C, простым и понятным написанием кода, хорошей поддержкой и постоянными обновлениями (актуальная версия на момент написания статьи – Swift 5.7), открытый исходный код.

Подводя итог, хочется отметить, что разработать приложение можно на любом популярном языке программирования, фреймворки и библиотеки уже созданы для большинства необходимых задач. Но если необходимо использовать все возможности, прежде всего следует определиться с системой, под которую планируется написать приложение: для Android лучше всего использовать Kotlin, так как все новые функции и фишки системы будут доступны на нем в первую очередь. А если вы хотите создать качественное

приложение под iOS – определённо стоит использовать Swift. Даже если эти языки вам кажутся поначалу сложными и непонятным – их всё равно стоит освоить, для того чтобы создавать функциональные, красивые и оптимизированные приложения.

Библиографический список

1. Google по-русски: почему компания перевела Android на язык от российских разработчиков // Forbes [электронный ресурс]. URL: <https://www.forbes.ru/tehnologii/376507-google-po-russki-pochemu-kompaniya-perevela-android-na-yazyk-ot-rossiyskih> (дата обращения: 20.03.2023).
2. Swift. Язык программирования с открытым кодом. Мощь, простота и потрясающие приложения // Apple [электронный ресурс]. URL: <https://www.apple.com/ru/swift/> (дата обращения: 20.03.2023).
3. Турушев, Т.К. Многофункциональные высокоуровневые языки программирования Java и Kotlin для мобильных приложений / Т.К. Турушев // Теоретические и практические аспекты развития современной науки: теория, методология, практика : Сборник научных статей по материалам VI Международной научно-практической конференции, Уфа, 16 ноября 2021 года. – Уфа: Общество с ограниченной ответственностью "Научно-издательский центр "Вестник науки", 2021. – С. 55-59.
4. Свищев, А.В. Переход на Kotlin при разработке Android приложений / А.В. Свищев, И.В. Беликов // Инновации. Наука. Образование. – 2021. – № 46. – С. 882-885.
5. Копылова, О.Ю. Перспективы использования языка программирования Swift / О.Ю. Копылова, С.В. Мещанов // Современные инновации: достижения и перспективы III тысячелетия : Сборник статей по материалам VII Международной научно-практической конференции,

Москва, 19–20 декабря 2017 года. – Москва: Проблемы науки, 2017. – С.
33-35.

Оригинальность 94%