**IT-Колледж “Сириус”**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**ДОКЛАД**

по дисциплине “Введение в специальность”

на тему “Android-разработка”

Выполнила:  
Студент группы

1.9.7.3

Благодарова Алина Игоревна

Принял:

Старший преподаватель  
Тенигин Альберт Андреевич

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

IT-Колледж “Сириус”  
2022

**СОДЕРЖАНИЕ**

[I. Введение 3](#_Toc120971184)

[II. Основная часть 5](#_Toc120971185)

[**1. «Вкусные» названия** 5](#_Toc120971186)

[**2. Android-разработчики и чем они занимаются** 7](#_Toc120971187)

[**3. Основные технологии и инструменты** 10](#_Toc120971188)

[**4. Java или Kotlin** 12](#_Toc120971189)

[**5. Основные компоненты и коллбэки** 16](#_Toc120971190)

[**6. Создание простого приложения в Android Studio** 17](#_Toc120971191)

[III. Заключение 22](#_Toc120971192)

[Список литературы 23](#_Toc120971193)

# I. Введение

История операционной системы Android началась в октябре 2003 года, за несколько лет до того, как Apple анонсировала свой первый iPhone. Android Inc была создана в Пало-Альто, штат Калифорния. В ходе своего выступления в Токио в 2013 году Энди Рубин заявил, что ОС Android изначально предназначалась для цифровых фотоаппаратов, но даже тогда рынок фотокамер стремительно сокращался, и поэтому через несколько месяцев стартап решил переключиться на использование своей ОС в мобильных телефонах.

Следующая большая глава в истории Android началась в 2005 году, когда стартап приобрела корпорация Google. Рубин и другие основатели Android Inc остались, чтобы продолжить разработку ОС уже под крылом Google. Тогда было принято решение использовать Linux в качестве основы для Android. Это означало, что операционную систему можно было предложить сторонним производителям смартфонов совершенно бесплатно. Google и команда Android посчитали, что компания может зарабатывать деньги не на самой ОС, а на других услугах, в том числе приложениях.



*Рисунок 1. Нынешний логотип Android*

Знакомый всем нынешний логотип Android (рис. 1), который выглядит как смесь робота и зелёного жука, был создан Ириной Блок, когда она работала в Google. Блок говорила, что единственное указание, которое ей дала команда дизайнеров Google, — это сделать логотип похожим на робота. Также она заявляла, что окончательный дизайн лого Android был отчасти вдохновлён табличками с изображениями мужчин и женщин на дверях туалета [1].

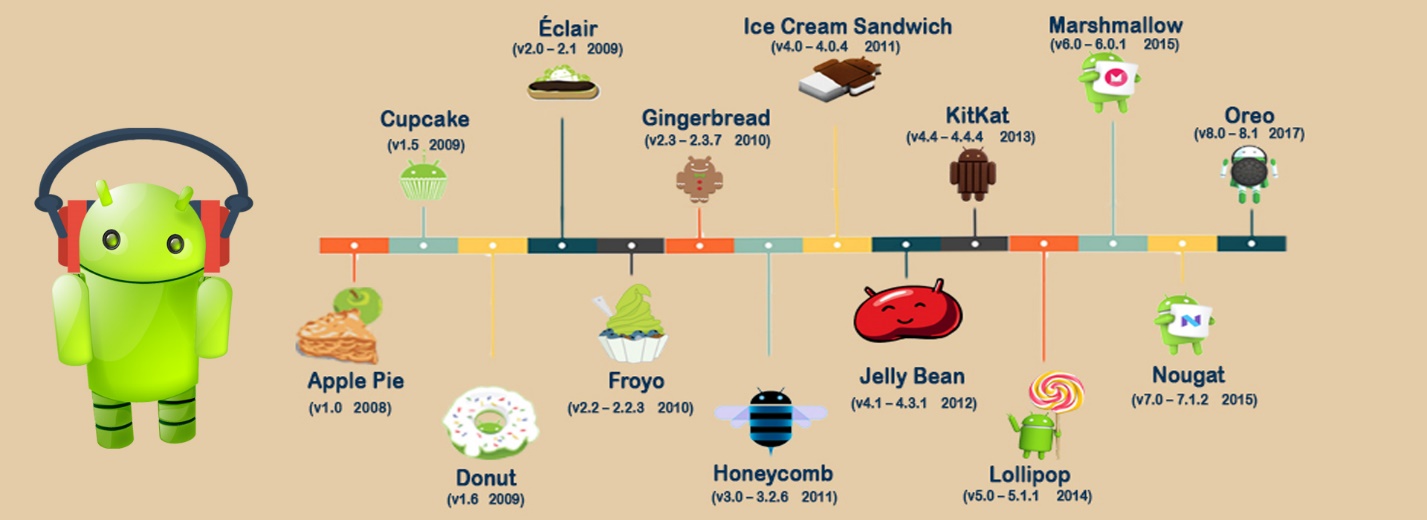
А теперь я хотела бы рассказать о самих Android-разработчиках, о том, чем они занимаются, какие технологии используют в своей работе, а также как они создают приложения.

# II. Основная часть

## **1. «Вкусные» названия**

Но перед тем, как начать говорить о самой разработке, я расскажу об одной интересной особенности версий Android, а именно об их «вкусных» наименованиях. У некоторых версий имеется своё название, связанное с каким-либо десертом или сладостью (рис. 2). Все версии Android с «сладкими» названиями перечислены ниже:

1. Android 1 (Apple Pie)
2. Android 1.5 (Cupcake)
3. Android 1.6 (Donut)
4. Android 2.0 / 2.1 (Éclair)
5. Android 2.2 (Froyo)
6. Android 2.3 (Gingerbread)
7. Android 3 (Honeycomb)
8. Android 4.0 (Ice Cream Sandwich)
9. Android 4.1 / 4.3 (Jelly Bean)
10. Android 4.4 (KitKat)
11. Android 5 (Lollipop)
12. Android 6 (Marshmallow)
13. Android 7 (Nougat)
14. Android 8 (Oreo)
15. Android 9 (Pie)
16. Android 10 (Queen Cake / Quince Tart)
17. Android 11 (Red Velvet Cake)
18. Android 12 / 12.1 (Snow Cone)
19. Android 13 (Tiramisu)



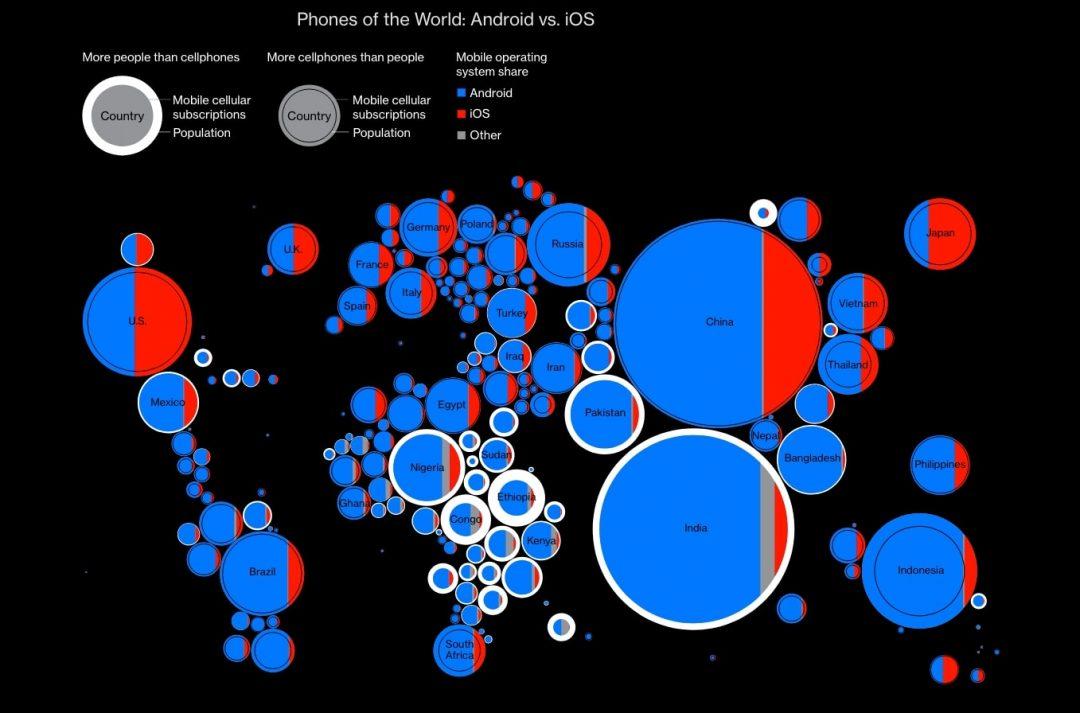
*Рисунок 2. «Вкусные» версии Android*

Изначально в Google планировали отказаться от «вкусных» наименований с выходом юбилейной десятой версии Android, оставив лишь цифру в названии. Но традиции одержали победу и десятка, наравне со своими предшественниками, получила кодовое имя. И даже не одно, а несколько. Одно из названий Android 10, Quince Tart, что в переводе означает «пирог с айвой», обозначалось разработчиками как внутреннее. Второе же, Queen Cake («королевский торт» или «торт королевы»), стало основным, ушедшим в «мир». Во всяком случае, оба варианта имеют место быть, а какой из десертов ближе именно вам — вам, собственно, и решать.

Начиная с Android 10, сладкие названия предполагалось опускать на второй план, ограничиваясь лишь порядковой цифрой. Но время расставило всё по местам, и добрая традиция продолжилась. Даже сейчас, в преддверии выхода Android 13 название системы всё так же вызывает неподдельный аппетит, а теперь вы ещё и увидели, как все эти вкусности, оставшиеся в истории названий версий операционных систем Android выглядят [2].

## **2. Android-разработчики и чем они занимаются**

Android установлен в 8 из 10 смартфонов в мире (рис. 3). Это 2,5 миллиарда пользователей, а также крупнейший рынок мобильных приложений. Всю эту вселенную обслуживают android-разработчики.



*Рисунок 3. Какая операционная система установлена на смартфонах пользователей мира*

Android-разработчик создаёт приложения и поддерживает их работу. При этом он не только пишет код, но может и заниматься другими вещами:

* продумывать интерфейс и логику (если для этого нет дизайнеров),
* защищать данные и устранять уязвимости,
* отвечать за производительность приложений,
* изучать пользовательские пожелания и делать обновления.

В компаниях android-разработчиков могут объединять в команды, чтобы за каждым человеком закреплялся узкий набор задач: одни пишут код, вторые его тестируют, третьи отвечают за обновления и так далее. А могут и свалить всё на одного умельца, дай бог ему здоровья.

В 2020 году средняя зарплата android-разработчика 103 тысячи рублей (рис. 4, 5) — это примерно столько же, что и год назад. Больше всего вакансий в Московской области и в Крыму, а самый высокий средний доход в Курганской области — здесь работодатели готовы платить 150 тысяч рублей [3].

Если же брать разные уровни разработчиков, то зарплата Junior разработчика находится в промежутке от 25 000 до 125 000 рублей. Middle разработчики получают же от 60 000 до 230 000 рублей. Senior специалисты получают от 80 000 до 460 000 рублей [4].

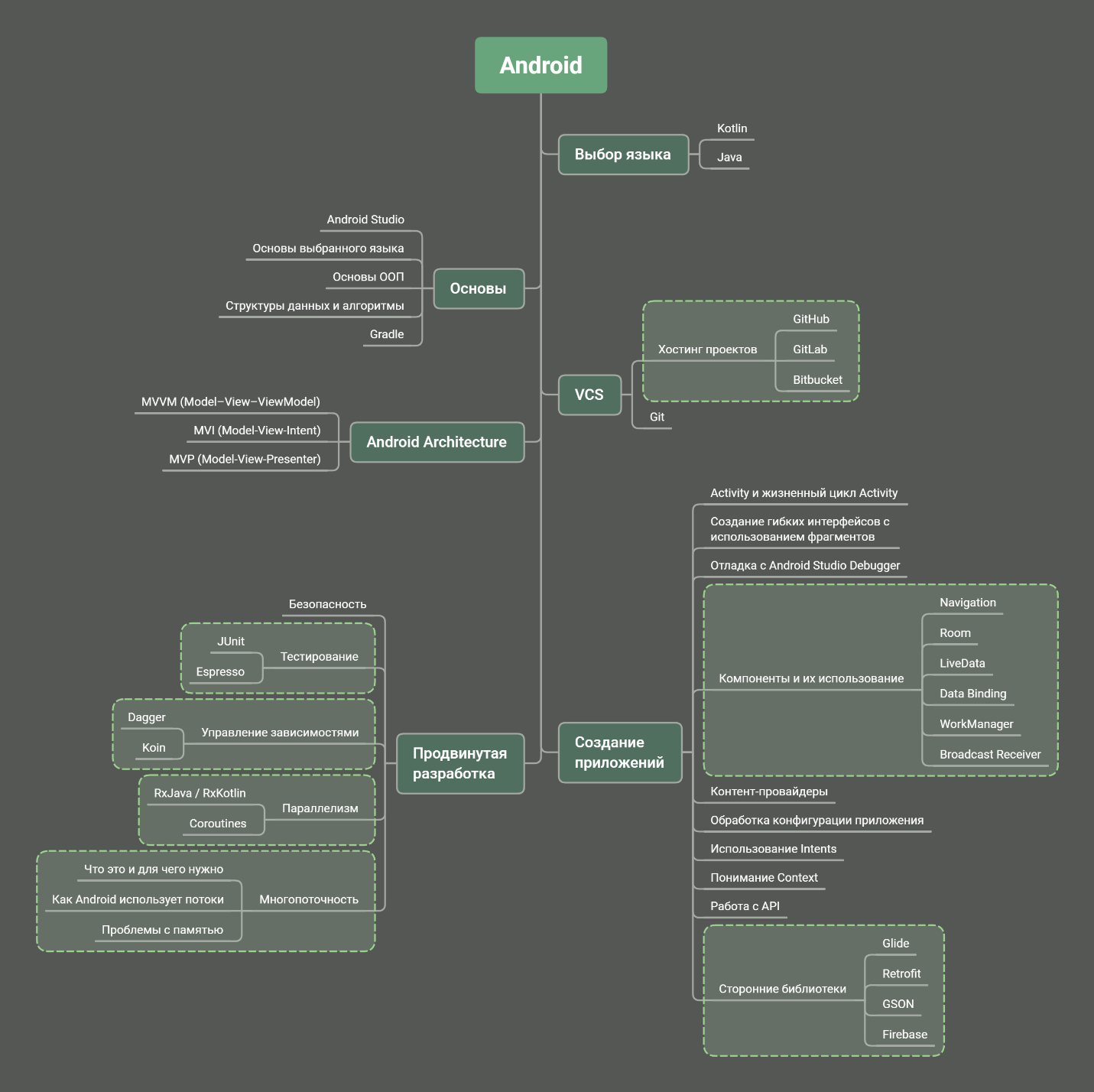
Зарплата андроид-разработчика 


Зарплата андроид-разработчика 


*Рисунки 4, 5. Зарплата Android-разработчиков*

## **3. Основные технологии и инструменты**

Основные технологии и инструменты можно увидеть на дорожной карте (рис. 6) Android-разработчика.



*Рисунок 6. Дорожная карта Android-разработчика*

Основным инструментом является программа Android Studio. Она постоянно обновляется, совместима с любой операционной системой, доступна для бесплатного скачивания и находится на сайте [developer.android.com](https://developer.android.com/).

Android Studio позволяет написать приложение с нуля или начать с макета:

* есть шаблоны под смартфоны и планшеты на Android OS,
* под приставки и телевизоры на Android TV,
* под умные часы и переносные устройства на OS Android Wear,
* под водительские гаджеты на Android Auto и другую технику.

В Android Studio встроен редактор макетов и есть функции для командной работы, тестирования и рефакторинга кода. Поддерживаются языки Kotlin и Java [3].

Не менее важным является система контроля версий Git и хостинги, такие как Github и Gitlab. В любых проектах это незаменимый инструмент, так как помогает следить за версиями проекта, переключаться между ними. Особенно помогает Git в командных проектах, где используются его «ветки». С их помощью можно разделять проект на несколько частей между разработчиками, а потом просто совместить всё.

## **4. Java или Kotlin**

Писать приложение на Android можно как на Java, так и на Kotlin. И возникает вполне резонный вопрос: «Какой же язык всё-таки использовать»? Когда в мае 2019 года компания Google объявила Kotlin предпочтительным языком для Android-разработки, споры по выбору языка разгорелись с новой силой. С одной стороны, всё просто: писать нужно на том языке, на котором удобно лично вам. Но нельзя отрицать и ряд факторов, которые способны склонить чашу весов в пользу Java или Kotlin. Поэтому давайте сравним эти 2 языка по 6 пунктам.

1. *Возраст языка*

У Java большое комьюнити, много библиотек, решений и готовых модулей. Грубо говоря, если разработчик сталкивается с какой-то проблемой, он может быстро получить ответ на свой вопрос.

А вот в случае с Kotlin всё наоборот. Молодость языка и связанных библиотек, таких как Kotlinx.Serialization или Exposed, заставляет потратить немало времени на поиск решений. К тому же, документация языка часто сводится к объяснениям в духе «это сделано как на Java, только тут нужно немного поменять».

1. *Количество кода*

Хоть эти языки внешне очень похожи, но вместе с тем и сильно отличаются друг от друга. Да, синтаксис Java предполагает, что код будет более громоздким (рис. 7), чем в Kotlin, а потому написание займёт больше времени.

Kotlin позволяет написать то же самое, что и Java, используя меньшее количество строк (Extensions, Null Safety, Data Classes). Отдельно стоит отметить работу с UI: благодаря Android Extensions больше не нужно заниматься бойлерплейтом с findViewById.



*Рисунок 7. Сравнение кода на Java и Kotlin*

1. *Что выбрать на старте Android-разработки*

Сегодня для разработки под Android лучше всего подойдёт именно Kotlin. Но если вы начинающий программист, то на старте лучше выбрать Java: этот язык более старый, и можно легко найти информацию о том, как решить ту или иную проблему. Не стоит также забывать, что Kotlin — это обёртка над Java, и в процессе компиляции он конвертируется в байт-код Java.

Сейчас трудно представить Android без Kotlin, но документация Kotlin подразумевает, что вы знаете Java, поэтому, если вы только начали свой путь в Android-разработчики, лучше начать изучение именно с Java, а после переходить на Kotlin.

1. *Среда разработки*

Разработка приложений для Android давно и тесно связана с Android Studio. Данная среда изначально была заточена под работу с Java, а потому код можно писать буквально по одной букве — IDE самостоятельно подтянет всё необходимое.

Так как Android Studio базируется на Intellij Idea, а Intellij Idea и Kotlin созданы и развиваются компанией JetBrains, то в итоге разработчики получают актуальную среду разработки, тесно связанную с языком программирования, что очень упрощает разработку приложения.

1. *Для каких целей выбирается язык*

Большинство существующих Android-приложений написаны на Java, и надеяться, что их перепишут на Kotlin, не стоит. А поскольку UI операционной системы Android разработан на Java, у этого языка есть SDK и много готовых библиотек, так что ряд компаний всё так же отдаёт предпочтение Java.

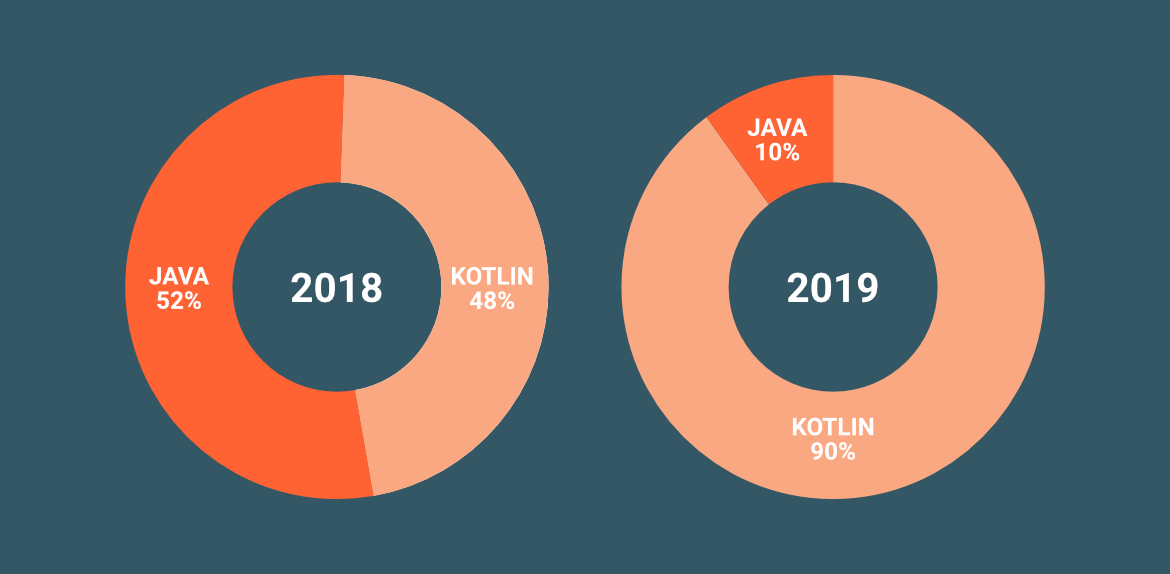
Если приложение имеет долгий срок жизни (например, мобильный банк), и вопрос о кроссплатформенной разработке даже не стоит, то мы считаем оптимальным вариантом Kotlin. Он полностью совместим с Java, то есть в рамках одного проекта можно использовать оба языка. Если сравнить с Java старых версий (7 и старше), Kotlin имеет кучу новых фишек, которые делают написание кода проще. При этом Kotlin не зависит от версии Android так, как Java 8.

1. *Перспективы*

Современные компании всё чаще разрабатывают мобильные приложения на Kotlin. Но также можно встретить специфическую библиотеку, которая работает только с Java, и, если вы не знаете этот язык — вам придётся нелегко. Кроме того, Kotlin — молодой язык, и неизвестно, что будет дальше, в то время как Java характеризуется кроссплатформенностью: на ней не только мобильная разработка держится, но и бэкенд с десктопом.

Сейчас Kotlin развивается достаточно предсказуемо и ориентирован именно на Android-разработку. Его всё чаще выбирают молодые компании, которые берут старт в создании мобильных приложений, и сейчас наблюдается нехватка Kotlin-разработчиков, а потому язык весьма востребован. Вряд ли в обозримом будущем он заменит Java, но вполне может сосуществовать, постепенно набирая популярность.

Так что выбрать — Java или Kotlin? В 2019 году резко взлетела популярность языка Kotlin (рис. 8). По техническим аспектам разница между языками также существенная, но, если вы планируете заниматься Android-разработкой, учите оба языка. Большинство популярных библиотек поддерживают обратную совместимость с Kotlin, код Java можно использовать в Kotlin и наоборот, а вот чтобы решить возникшую проблему или банально понять все нюансы документации, стоит знать Java. Освоив оба языка программирования, вы станете востребованным специалистом и обеспечите себе хорошее будущее в разработке приложений под Android [6].



*Рисунок 8. Использование языков Kotlin и Java*

## **5. Основные компоненты и коллбэки**

Android-приложение состоит из четырёх компонентов. Каждый компонент — это точка входа, через которую система или пользователь может получить доступ.

* Активность (activity) — элементы интерактивного пользовательского интерфейса. Одна активность задействует другую и передаёт информацию о том, что намерен делать пользователь, через класс Intent (намерения). Активности подобны веб-страницам, а намерения — ссылкам между ними. Запуск приложения — это активность Main.
* Сервис (service) — универсальная точка входа для поддержания работы приложения в фоновом режиме. Этот компонент выполняет длительные операции или работу для удалённых процессов без визуального интерфейса.
* Широковещательный приемник (broadcast receiver) транслирует нескольким участникам намерения из приложения.
* Поставщик содержимого (content provider) управляет общим набором данных приложения из файловой системы, базы данных SQLite, интернета или другого хранилища [7].

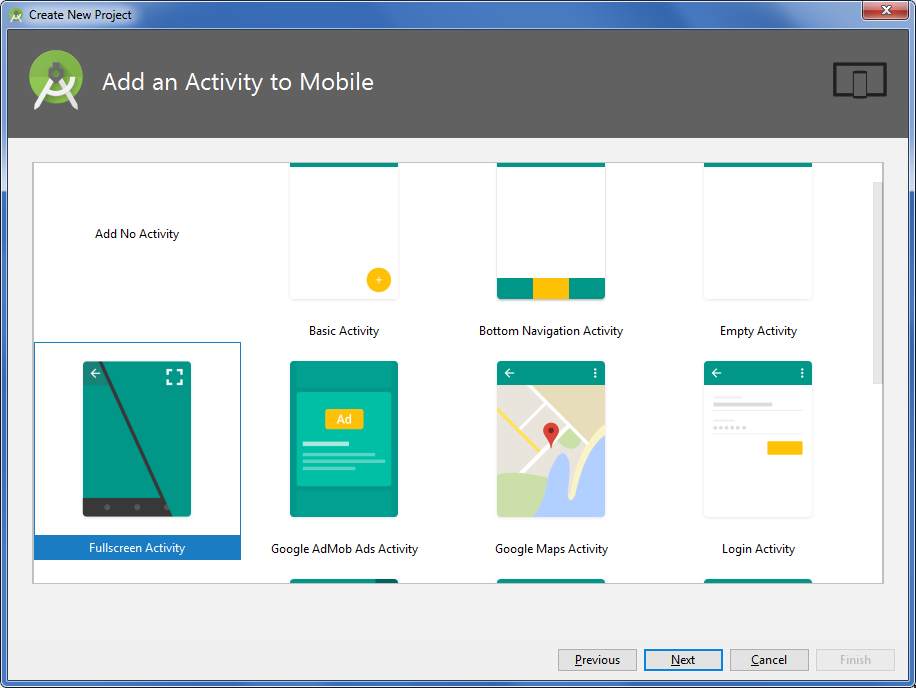
Activity в Android — это по сути окно приложения. Простые приложения состоят из одного окна, сложные — из двух и более. Жизненный цикл Activity включает в себя шесть коллбэков:

* onCreate() — создание активити;
* onStart() — подготовку к отображению на экране;
* onResume() — отображение на экране;
* onPause() — приостановку процессов;
* onStop() — состояние невидимости;
* onDestroy() — завершение работы.

## **6. Создание простого приложения в Android Studio**

Сейчас я расскажу о том, как создать просто приложение в Android Studio

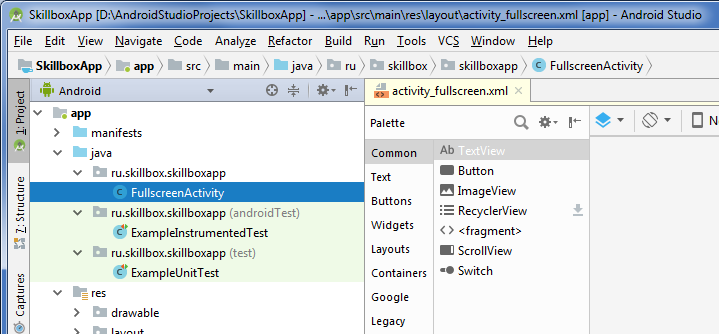
1. Выбираем название приложения, домен компании, путь к проекту и название пакета. Указываем, включить ли поддержку опциональных языков программирования C++ и Kotlin.
2. Задаём одну или несколько целевых платформ для сборки. Для этого используется SDK и AVD, менеджер виртуальных устройств Android. Инструмент позволяет устанавливать в SDK пакеты, которые поддерживают несколько версий ОС Android и несколько уровней API (интерфейсов программирования приложений)
3. Выбираем основную активность (рис. 9), которая будет запускаться при нажатии на иконку приложения, и даём ей имя.



*Рисунок 9. Выбор активности в Android Studio*

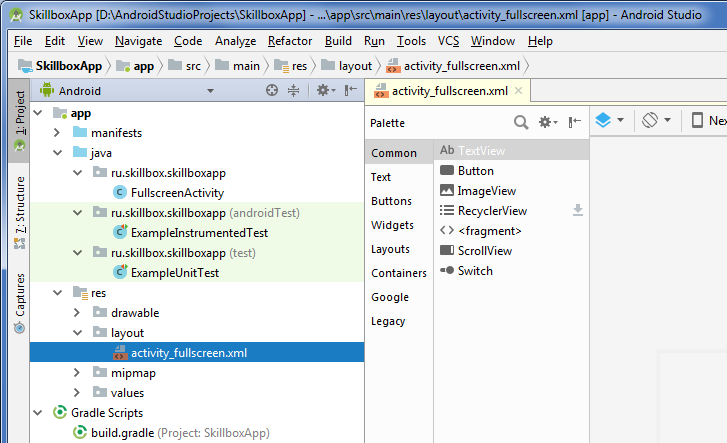
1. После нескольких минут сборки Android Studio открывает интерфейс IDE. Здесь три основных момента.

Если выбрать в выпадающем меню вид Android, то вы увидите файлы проекта. Например, наша основная активность называется app > java > ru.skillbox.skillboxapp > FullscreenActivity (рис. 10). При создании проекта мы указали вместо активности Main полноэкранную активность.



*Рисунок 10. Файлы проекта в Android Studio (Fullscreen Activity)*

Далее можно посмотреть файл app > res > layout > activity\_fullscreen.xml (рис. 11). Это XML-файл с макетом для UI нашей основной активности.



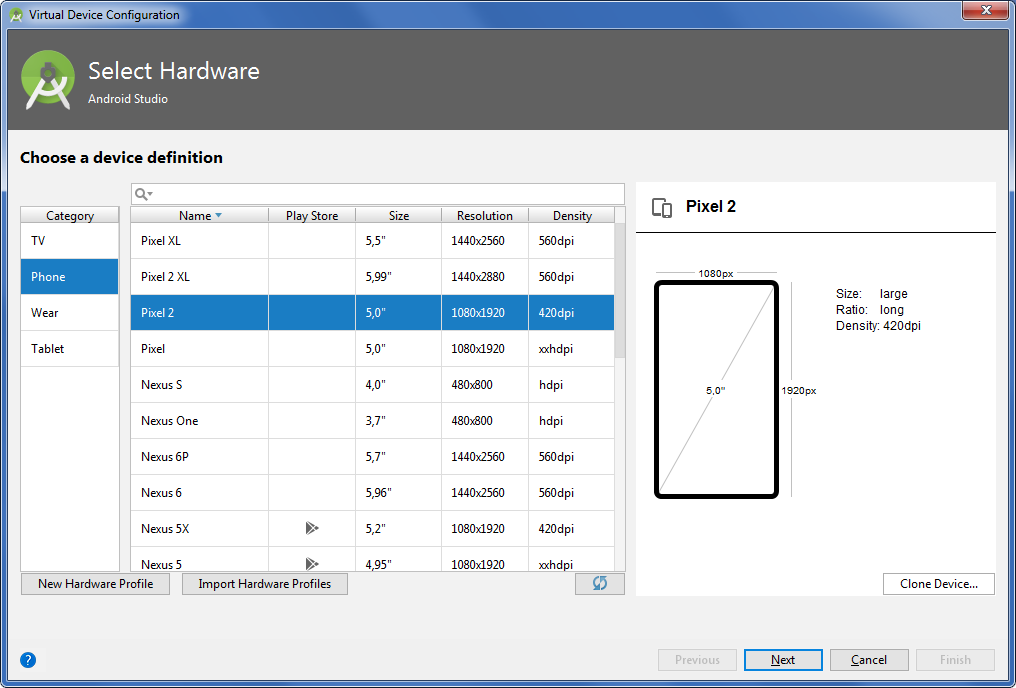
*Рисунок 11. Файлы проекта в Android Studio (activity\_fullscreen.xml)*

Наконец, третий важный файл app > manifests > AndroidManifest.xml описывает фундаментальные характеристики приложения и определяет все его компоненты.

**Запуск приложения**

Созданное нами приложение — это одна активность, которая запускается в полноэкранном режиме и не имеет графических элементов. Запускать наше приложение можно 2 способами:

1. Смартфон или планшет для этого подключаем в режиме USB-отладки, которая активируется в Настройках разработчика в меню Настройки.
2. Для запуска в эмуляторе нажимаем в Android Studio кнопку Run в меню Run (Shift+F10). Выбираем подходящее устройство (рис. 12) и версию ОС, портретную или ландшафтную (альбомную) ориентацию.



*Рисунок 12. Выбор подходящего устройста для запуска в эмуляторе*

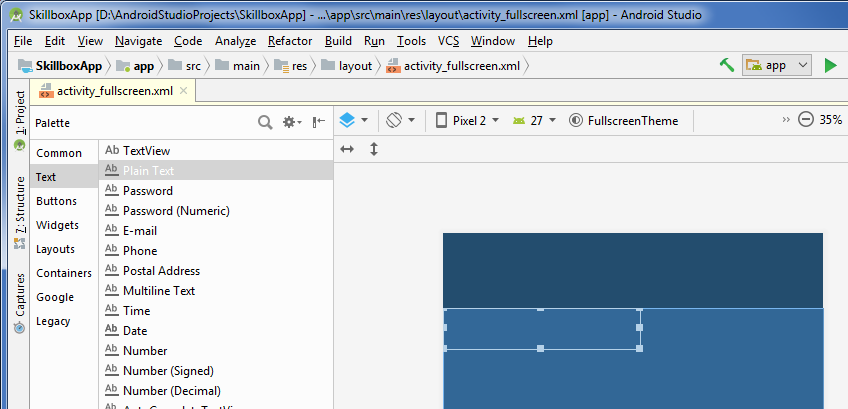
Android Studio установит эмулятор и запустит его.

**Создаём простой пользовательский интерфейс**

Пользовательский интерфейс Android-приложения создаётся через иерархию макетов (layouts, объекты ViewGroup) и виджетов (объекты View). Макеты управляют расположением дочерних виджетов на экране. Сами виджеты — это непосредственно компоненты UI: кнопки, текстовые поля на экране и т.п.

Интерфейс активностей создаётся в Android Studio в редакторе макетов (Layout Editor) и хранится по большей части в XML-файлах.

1. Открываем файл app > res > layout > activity\_fullscreen.xml.
2. Добавляем на экран из палитры (Palette) виджеты перетаскиванием мышью. Например, берём текстовое поле (PlainText). Это виджет EditText, куда пользователь может вводить текст.
3. Добавляем кнопки и другие нужные элементы (рис. 13). Также на экран можно перетащить кнопки и другие элементы.



*Рисунок 13. Создание UI в Android Studio*

1. Добавляем действия, активности и навигацию.

Предположим, мы создали активность с текстовым полем и кнопкой «Отправить». После этого нужно написать, что конкретно будет происходить при нажатии кнопки «Отправить».

1. Заходим в код app > java > FullscreenActivity.
2. Добавляем метод SendMessage() в класс FullscreenActivity, чтобы при нажатии на кнопку вызывался этот метод.
3. Создаём намерения (класс Intent) для перехода от одной активности к другой, новые активности, навигацию и всё остальное, что необходимо для приложения [7].

# III. Заключение

В моём докладе я рассмотрела профессию «Android-разработчика», чем они занимаются и какие технологии используют.

Так как же стать Android-разработчиком? Сначала нужно определиться с языком программирования (но чаще всего выбор тут падает на Kotlin), научиться работать с Android Studio, Gradle и веб-сервисом для хостинга проектов.

Предстоит разобраться в основных паттернах и их отличиях, изучить создание гибких интерфейсов, а также подключаться к другим сервисам через API. Продвинутая разработка предполагает тестирование, управление зависимостями, работу с параллелизмом и многопоточностью [5].

Сейчас данное направление очень популярно, так как мобильные телефоны есть практически у каждого, даже у школьников и некоторых дошкольников есть свои телефоны. Специалисты android-разработки в данный момент очень востребованы на рынке, поэтому найти работу довольно легко.

# Список литературы

1. История Android: эволюция самой популярной операционной системы в мире // Trashbox URL: https://trashbox.ru/link/history-of-android (дата обращения: 26.10.2022).
2. Все названия версий Android от 1 до 13 // China mobile URL: https://china-mobile.ru/vse-nazvaniya-versij-operatsionnyh-sistem-android/ (дата обращения: 17.11.2022).
3. Чем занимаются андроид-разработчики // Код. Журнал Яндекс Практикума URL: https://thecode.media/android-developer/ (дата обращения: 20.11.2022).
4. Сколько зарабатывают Андроид-разработчики на приложениях и играх в 2022 году // Учись Онлайн URL: https://uchis-online.ru/blog/professii/skolko-zarabativayut-android-razrabotchiki (дата обращения: 03.11.2022).
5. Дорожная карта по Android-разработке с нуля // Tproger URL: https://tproger.ru/articles/kak-stat-android-razrabotchikom-s-nulja-dorozhnaja-karta/ (дата обращения: 01.12.2022).
6. Java vs Kotlin для Android-разработки: ответы «за» и «против» // Tproger URL: https://tproger.ru/articles/java-vs-kotlin/ (дата обращения: 23.11.2022).
7. Как сделать приложение для Android самостоятельно // Skillbox URL: https://skillbox.ru/media/code/kak\_sdelat\_prilozhenie\_dlya\_android/ (дата обращения: 11.11.2022).