**IT-Колледж “Сириус”**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
  
  
  
  
ДОКЛАД**

по дисциплине “Введение в специальность”

на тему “Android-разработка”

Выполнила:  
Студентка группы

1.9.7.3

Благодарова Алина Игоревна

Принял:

Старший преподаватель  
Тенигин Альберт Андреевич

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

IT-Колледж “Сириус”  
2022

**СОДЕРЖАНИЕ**

[I. Введение 3](#_Toc121568426)

[II. Основная часть 6](#_Toc121568427)

[**1. История операционной системы Android** 6](#_Toc121568428)

[**2. «Вкусные» названия** 8](#_Toc121568429)

[**3. Основные технологии и инструменты** 10](#_Toc121568430)

[**4. Java или Kotlin** 12](#_Toc121568431)

[**5. Основные компоненты и коллбэки** 16](#_Toc121568432)

[**6. Создание простого приложения в Android Studio** 18](#_Toc121568433)

[III. Заключение 24](#_Toc121568434)

[Список литературы 25](#_Toc121568435)

# Введение

На сегодняшний день на 8 из 10 смартфонов установлена операционная система Android. Её доля составляет 73% от всего рынка, а это более 2,5 миллиардов пользователей. Кроме смартфонов, под управлением Android работают многие планшеты, умные холодильники и бытовые роботы. Благодаря тому, что операционная система Android основана на бесплатном ядре Linux с открытым исходным кодом, для неё создано огромное количество мобильных приложений. На начало 2020 года в онлайн-магазине Google Play насчитывалось 2,5 миллиона мобильных приложений, а уже к сентябрю их число перевалило 3 миллиона. И их количество с каждым годом продолжает только расти. Пользователям нужны новые мессенджеры, навигационные и банковские программы, службы заказа такси и еды на дом, приложения магазинов и коммунальных услуг. Android-приложения применяются также в торговом оборудовании, смарт-устройствах и даже в автомобилях. В настоящее время практически не осталось такой сферы жизни человека, которую он не смог бы облегчить при помощи мобильного приложения. Ну, и конечно, игры. На их долю по статистике приходится около трети всех скачиваний.

Цель Android-разработки заключается в том, чтобы максимально облегчить жизнь и удобство пользования приложениями конечному пользователю. Приложение должно работать стабильно и правильно даже на старых и дешевых устройствах, на любой версии операционной системы, не должно «вылетать». Его интерфейс, который создают дизайнеры и реализуют программисты, должен быть интуитивно понятен и удобен. Именно над этими задачами работают команды разработчиков каждый день. Они не только создают новые приложения, но и модернизируют и поддерживают уже существующие. Эта профессия очень востребована сейчас. Работы хватает не только опытным профессионалам, но и новичкам.

В компаниях android-разработчиков могут объединять в команды, чтобы за каждым человеком закреплялся узкий набор задач: одни являются «бэкендерами» и обеспечивают правильную работу приложений на серверах, другие работают в «фронтэнде» - разрабатывают интерфейс и дизайн приложений. Кто-то пишет код, другие его тестируют, третьи отвечают за обновления и так далее. Но есть и «мобильщики» - самые опытные и профессиональные программисты, которые создают приложения целиком.

Мобильный разработчик — это очень стрессовая и нагруженная работа, связанная с постоянным общением. Необходимо обладать крепкими нервами и быть уверенным в своих знаниях, а также быть готовым к критике своей работы, как со стороны заказчика, так и со стороны пользователей. Поэтому будущий мобильный разработчик должен обладать несколькими важными качествами:

* Он должен прежде всего понимать пользователя, разбираться в принципах проектирования интерфейсов, интуитивно понимать UI/UX.
* Командная работа — важный залог создания полезного и удобного приложения: придётся взаимодействовать с тестировщиками, коммерческой службой, разработчиками веба и десктопа, службой поддержки и т.д. Эта работа скорее для экстравертов.
* Аналитический склад ума — мобильный разработчик должен обязательно анализировать множество факторов, учитывать их в разработке.
* Внимание к деталям и высокая работоспособность обеспечат отсутствие багов в приложении и высокую скорость разработки, которая в мобильных приложениях крайне важна из-за интенсивных изменений предпочтений в устройствах, обновлений операционных систем и т.д.
* Готовность к непрерывному обучению — гайдлайны постоянно меняются, Google и Apple проводят интересные и полезные мероприятия, поэтому важно успевать «схватить» всю информацию на лету и успеть втащить её в продакшен [1].

Если вы обладаете этими качествами, добро пожаловать в Android-разработку! Эта работа не только очень интересна, но и хорошо оплачивается.

В 2022 году средняя зарплата Android-разработчика 97,5 тысяч рублей (рис. 1) — это примерно столько же, что и год назад. Больше всего вакансий в Московской области и в республике Крым, а самый высокий средний доход в Курганской области — здесь работодатели готовы платить 150 тысяч рублей.

*Рис.1 - Изменение уровня средней зарплата Android-разработчиков*

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 [2].

# Основная часть

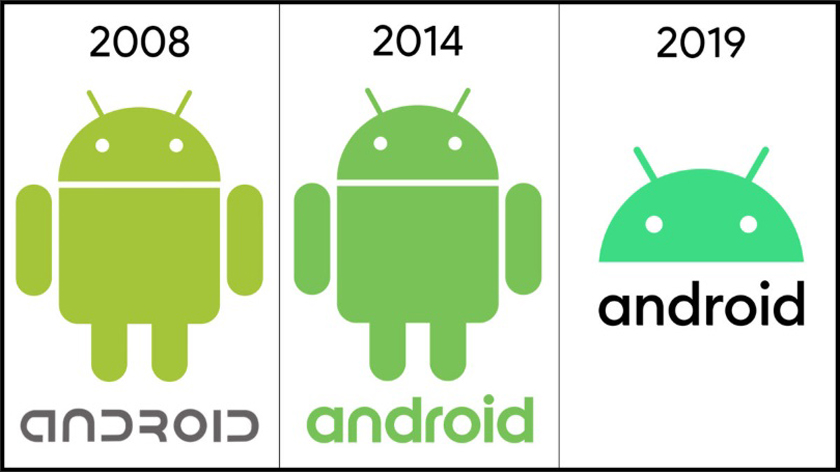
## **1. История операционной системы Android**

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 уже тогда рынок фотокамер стремительно сокращался, и поэтому через несколько месяцев стартап решил переключиться на использование своей ОС в мобильных телефонах.

Когда стартап в 2005 году приобрела корпорация Google, в истории Android началась следующая большая глава. Рубин и другие основатели Android Inc остались, чтобы продолжить разработку ОС уже под управлением Google. Именно в этот период было принято решение использовать Linux в качестве основы для Android. Это привело к тому, что операционную систему Android стало возможно предложить сторонним производителям смартфонов совершенно бесплатно. Было принято решение, что компания Google станет зарабатывать деньги не на продаже самой ОС, а на других услугах, в том числе приложениях.

Началом массовой разработки под Android считается 2008 год, когда появился магазин приложений Android Market (в 2012 его переименовали в Google Play).

Знакомый всем логотип Android (рис. 2), 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 [3].



*Рис.2 - Логотипы Android*

## **2. «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. 3). Все версии Android с «сладкими» названиями перечислены ниже:

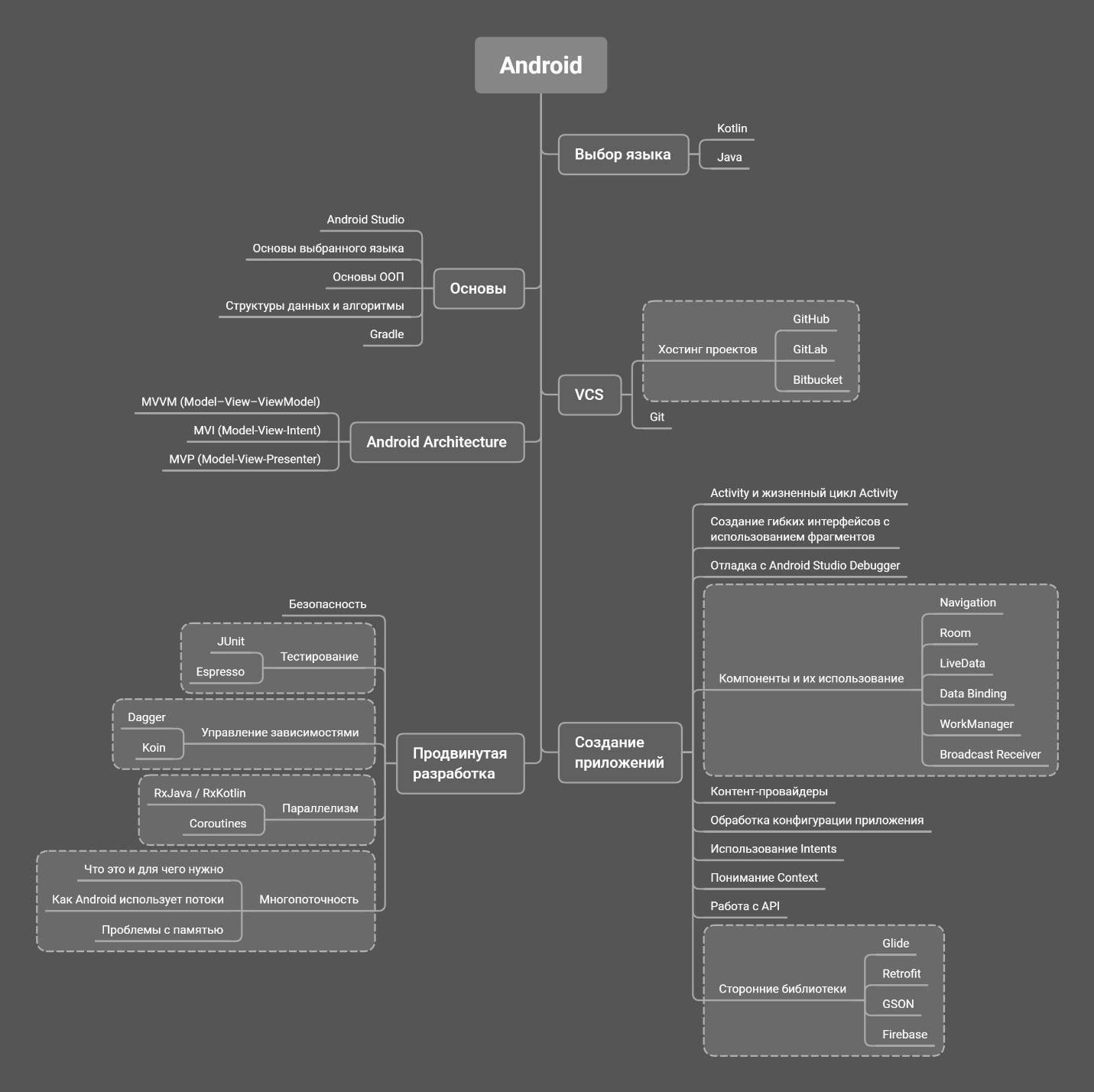
*Рис.3 - «Вкусные» версии Android*

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. Так появились Android 11 (Red Velvet Cake) и Android 12 / 12.1 (Snow Cone). Даже сейчас, в преддверии выхода Android 13 (Tiramisu) 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 [4].

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

Основные технологии и инструменты можно увидеть на дорожной карте (рис. 4) Android-разработчика.



*Рис.4 - Дорожная карта Android-разработчика*

Основным инструментом для Android-разработчиков считается программа Android Studio. Её достоинствами является то, что она постоянно обновляется и совместима с любой операционной системой. Программа доступна для бесплатного скачивания на сайте [developer.android.com](https://developer.android.com/).

Android Studio позволяет как написать приложение с нуля, так и использовать многочисленные макеты:

* 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.

Еще одним преимуществом Android Studio является то, что в неё встроен редактор макетов и есть функции для командной работы, тестирования и рефакторинга кода, поддерживаются языки Kotlin и Java [5].

Не менее важным является наличие системы контроля версий Git и хостинги, такие как Github и Gitlab. Для любых проектов это полезный и часто незаменимый инструмент, так как помогает следить за версиями проекта и быстро переключаться между ними. Особенно помогает Git в больших командных проектах, где используются его «ветки». С их помощью можно разделить проект между разработчиками, а в последующем просто совместить все части между собой в единую программу.

## **4. Java или Kotlin**

Android-разработчики могут использовать для создания приложений такие языки программирования, как Java, Kotlin, Python, C/C++, JavaScript, Dart, C#. Однако предпочтительными языками считаются Java и Kotlin. Так какой язык все же лучше выбрать? Споры по этому вопросу то затихают, то разгораются с новой силой. Когда в мае 2019 года компания Google объявила Kotlin предпочтительным языком для Android-разработки, этому нашлось как немало сторонников, так и огромное число противников. У каждого из этих языков программирования есть ряд факторов, дающих ему преимущества. Поэтому выбирать нужно исходя из целей и задач, поставленных перед разработчиком. Но личные предпочтения и опыт работы в определенном языке также очень важны. Любая мелочь может склонить чашу весов в пользу выбора Java или Kotlin. Поэтому давайте сравним их по нескольким факторам.

1. *Возраст языка*

Язык Java появился в 1995 году. У него большое комьюнити, много библиотек, решений и уже готовых модулей. Если вы столкнетесь с какой-либо проблемой, то сможете легко найти документацию, посмотреть видеоролик или быстро получить ответ на свой вопрос от других разработчиков.

А вот в случае с Kotlin всё сложнее. Язык появился в 2011 году. Молодость языка и связанных библиотек, таких как Kotlinx.Serialization или Exposed, заставят вас потратить немало времени на поиск решений. Даже документация языка часто помогает не сильно, хотя в большинстве случаев задачи решаются как на Java, но с небольшими изменениями.

1. *Количество кода*

Хоть эти языки внешне очень похожи, но вместе с тем и сильно отличаются друг от друга. Синтаксис Java предполагает, что код будет более громоздким (рис. 5), чем в Kotlin, и поэтому его написание занимает больше времени. Кроме того, в нем необходимо следить за мелкими деталями.

Kotlin позволяет написать такую же программу, что и на Java, используя меньше времени и меньшее количество строк кода (Extensions, Null Safety, Data Classes). Это является важным фактором, когда надо создать проект в сжатые сроки.



*Рис.5 - Сравнение кода на Java и Kotlin*

1. *Безопасность*

Настоящей проблемой для Java-разработчиков является так называемая «нулевая безопасность» или исключение NullPointerExceptions. В Java не поддерживается гарантия нулевой безопасности при объявлении переменных. То есть не поддерживается оценка того, могут ли переменные иметь значение NULL во время компиляции. В то же время в Android используется null для представления отсутствия значений, поэтому такая ситуация может попросту уничтожить программу.

В Kotlin по умолчанию встроена null-безопасность, поэтому если программист решит присвоить чему-то значение null — во время компиляции просто произойдёт сбой. Для разработчиков null-безопасность языка позволяет не писать дополнительный код или обнаружить проблему заранее. Это является еще одним преимуществом перед языком Java.

1. *Среда разработки*

В Java для разработки приложений под Android давно применяется специализированная среда разработки IDE - Android Studio. Она изначально заточена под работу с Java, поэтому код можно писать буквально по одной букве — IDE самостоятельно подтянет всё необходимое и закончит строку.

Поскольку компания JetBrains является создателем Kotlin и IntelliJ Idea,

а среда разработки Android Studio базируется именно на IntelliJ Idea, то получается, что разработчики получают актуальную IDE, тесно связанную с языком программирования и которая покрывает все их запросы. Это очень упрощает разработку приложения.

1. *Цели разработки*

Большинство существующих Android-приложений написаны на Java, и маловероятно, что их кто-то решит переписать на Kotlin. По этой причине многие компании всё же отдают предпочтение языку Java.

В случае, если предполагается, что приложение будет иметь долгий срок жизни (например, мобильный банк), то предпочтительным вариантом считается Kotlin. Особенно, если не стоит вопрос о кроссплатформенной разработке, поскольку Kotlin не зависит от версии Android так, как Java 8. Kotlin полностью совместим с Java, это дает возможность в рамках одного проекта использовать оба языка. Но учитывая простоту разработки и написания кода, а также наличие множества фишек, которых нет в Java, будет оптимальнее все же выбрать язык Kotlin.

1. *Перспективы*

С течением времени универсальный язык Java точно не потеряет свою популярность. Ему отдают предпочтение благодаря его кроссплатформенности и возможности создавать на нём программное обеспечение для чего угодно — хоть для будильника, хоть для космического корабля. И умирать в ближайшие годы этот язык программирования не собирается.

Kotlin — молодой язык программирования, который во многом идёт вразрез с консервативными устоями. Все чаще многие современные компании переходят на этот язык, но всё ещё неизвестно, что будет с ним завтра. Возможно, скоро он обгонит по популярности язык Java, но может и повторить судьбу мёртвого языка Perl. Вряд ли в обозримом будущем он заменит Java, но вполне может сосуществовать, постепенно набирая популярность. Пока Kotlin развивается достаточно предсказуемо и ориентирован именно на разработку мобильных приложений.

Так что выбрать — Java или Kotlin? Если вы начинающий программист, то на старте лучше выбрать Java: этот язык более старый, у него большое комьюнити, много библиотек, уже готовых модулей и решений. Этот язык стоит изучить и в случае, если вы хотите заниматься чем-то большим, чем разработкой мобильных приложений и понимать все нюансы документации. Сейчас трудно представить Android без Kotlin, но документация Kotlin предполагает, что вы знаете Java, поэтому лучше начать изучение с Java, а после переходить на Kotlin. Популярность языка Kotlin резко взлетела В 2019 году (рис. 6). Имея множество фишек, этот язык лучше всех других подойдет для разработки мобильных приложений. Сейчас наблюдается нехватка Kotlin-разработчиков, а потому язык весьма востребован. Имейте в виду, что только освоив оба языка программирования, вы станете востребованным специалистом и обеспечите себе хорошее будущее [6,7].

*Рис.6 - Использование языков Kotlin и Java*

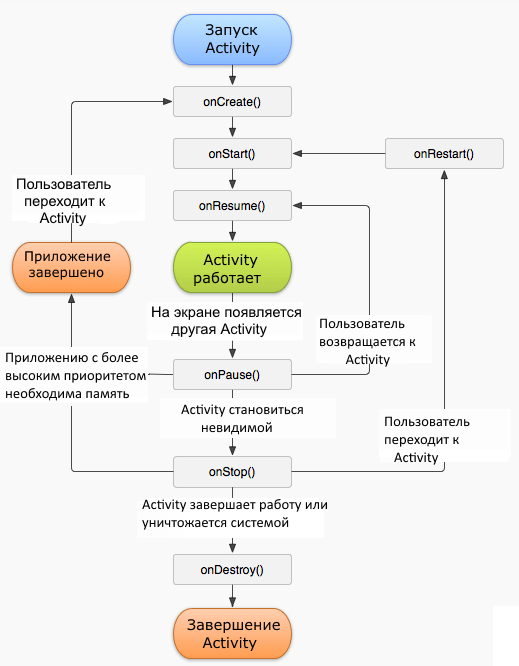
## **5. Основные компоненты и коллбэки**

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 [8].

Activity в Android — это по сути окно приложения. Простые приложения состоят из одного окна, сложные — из двух и более. Жизненный цикл Activity (рис. 7) 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.

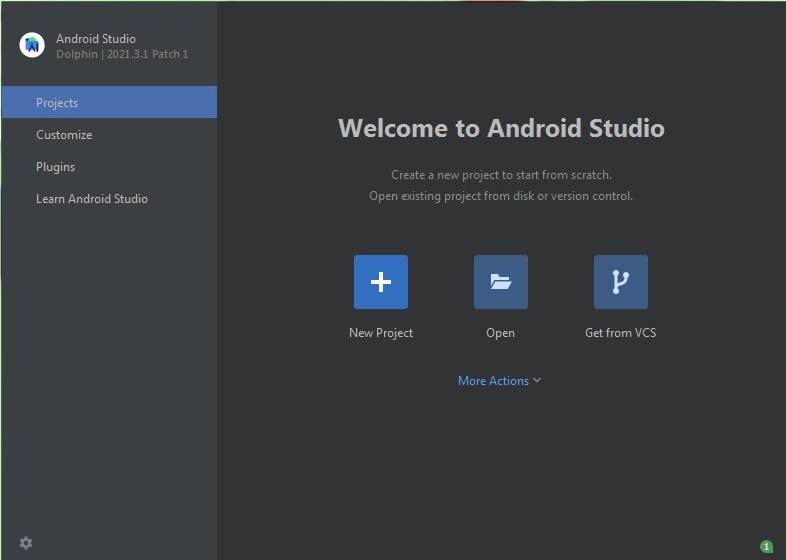


*Рис.7 - Жизненный цикл Activity*

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

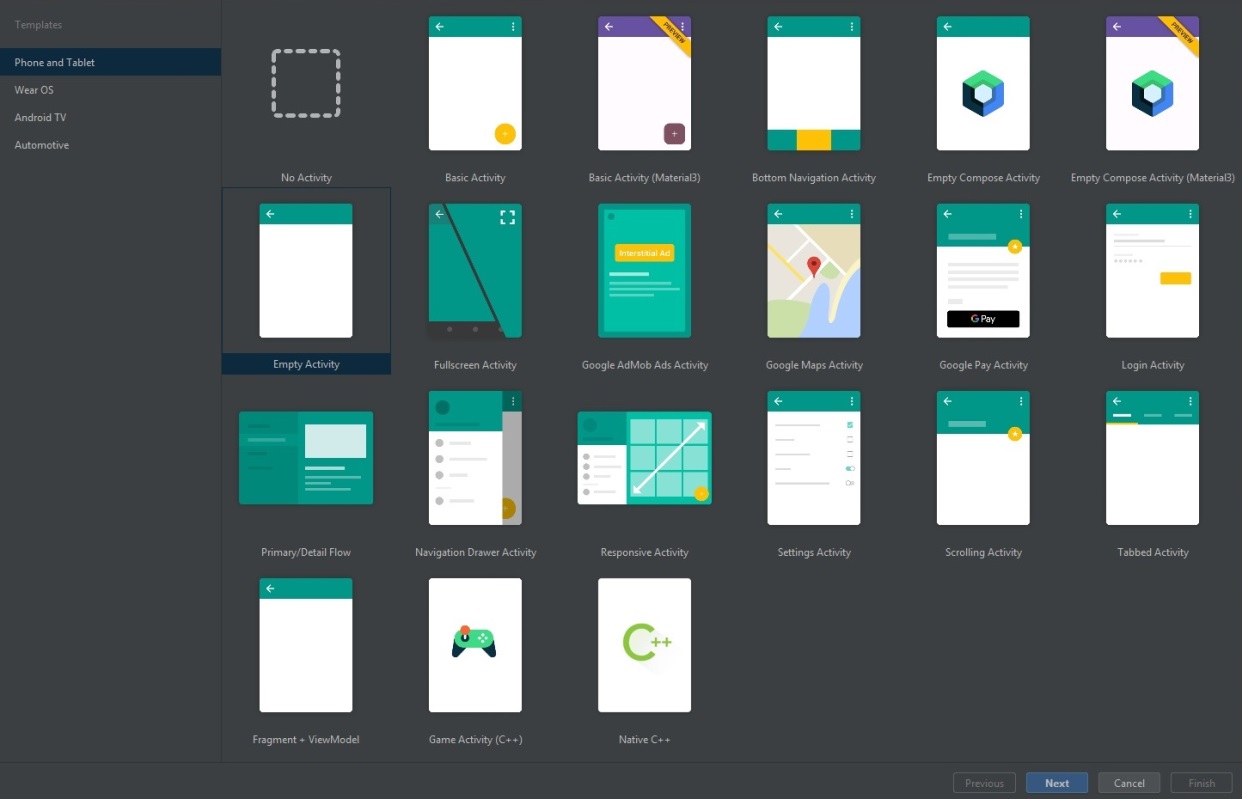
Работа начинается с создания проекта Android. Проект Android содержит файлы, из которых состоит приложение. Чтобы создать новый проект, откройте Android Studio.

Если Android Studio запускается на вашем компьютере впервые, на экране по­является диалоговое окно с приветствием (рис. 8).



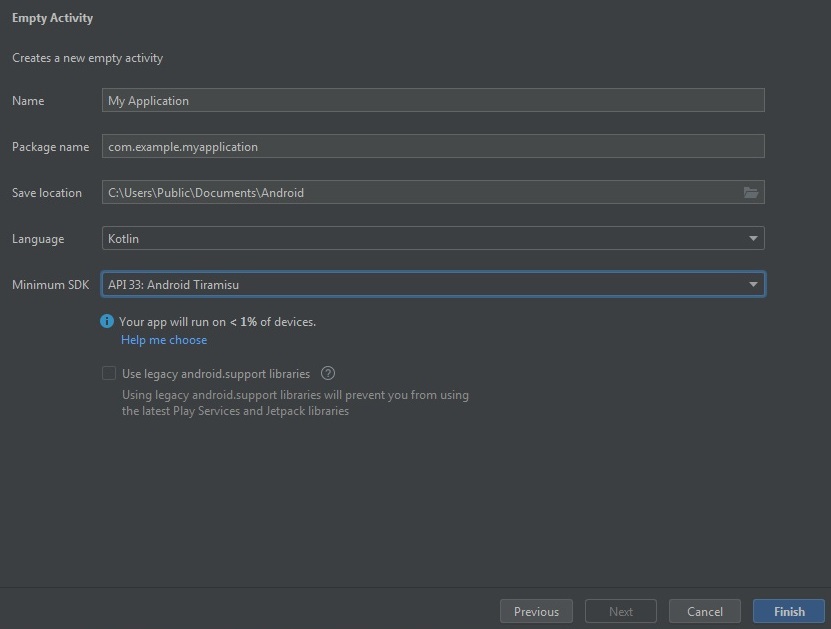
*Рис.8 - Добро пожаловать в Android Studio*

В окне приветствия нажмите кнопку *New Project*. Перед вами откроется мастер создания нового проекта (рис. 9). Убедитесь, что выбрана вкладка *Phone and Tablet*, выберите шаблон *Empty Activity* и нажмите кнопку *Next*.



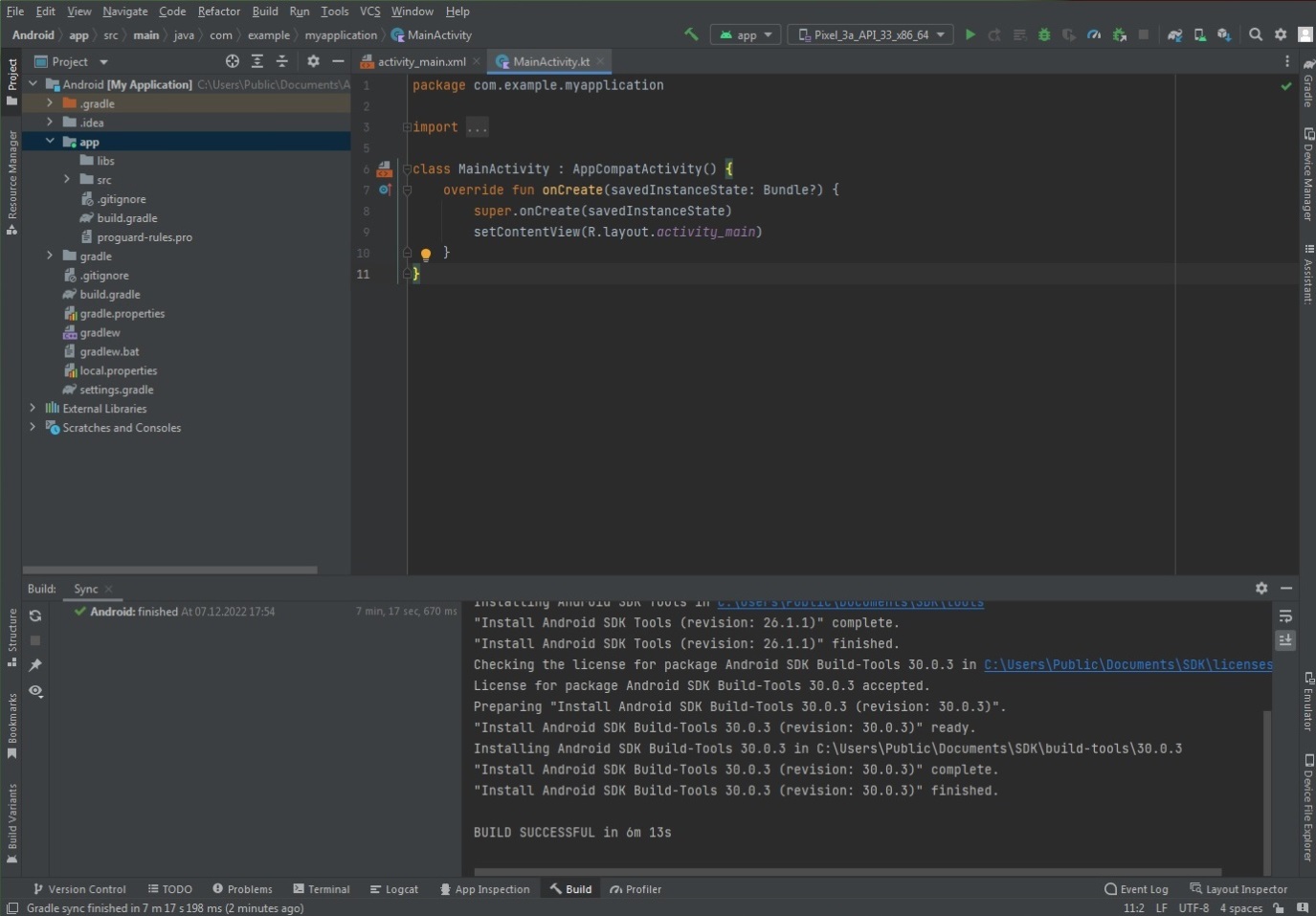
*Рис.9 – Выбор шаблона проекта*

Теперь перед вами появится экран настройки проекта (рис. 10). Задайте для нового приложения имя, имя и путь расположения проекта (используйте любой каталог в вашей файловой системе). Выберите язык программирования Java или Kotlin, а также версию Android.



*Рис.10 – Настройка нового проекта*

Нажмите кнопку *Finish*, после чего Android Studio создаст и откроет ваш новый проект (рис. 11). Различные части окна проекта называются *панелями*. Слева располагается *панель Project*. На ней выполняются операции с файлами, относящимися к проекту. Снизу находится *панель Build*, где можно просмотреть подробную ин­формацию о процессе компиляции и состоянии сборки. Создавая новый про­ект, Android Studio автоматически собирает его. На *панели Project* раскройте пункт *app*: Android Studio автоматически откроет файлы activity\_main.xml и MainActivity.kt на основной панели, называемой *панелью редактора* или просто *редактором.*



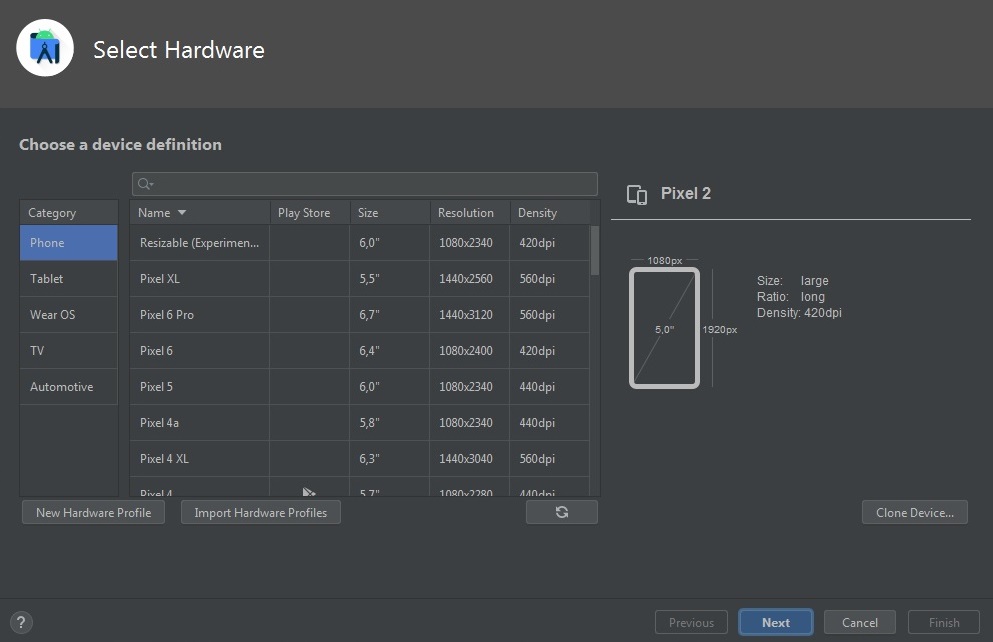
*Рис.11 – Окно нового проекта*

Отображением различных панелей можно управлять, щелкая по вкладкам с со­ответствующими названиями у левого, правого или нижнего края экрана. Также для многих панелей определены специальные комбинации клавиш.

**Запуск приложения**

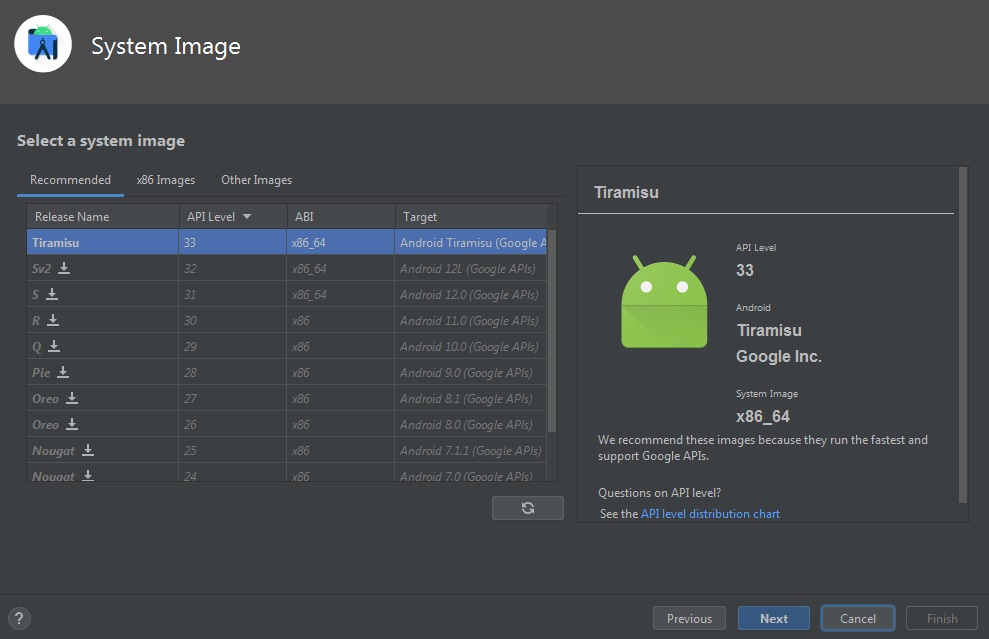
Для запуска Android-приложений необходимо устройство — физическое или виртуальное. Виртуальные устройства работают под управлением эмулятора Android, включенного в поставку средств разработчика.

Чтобы создать виртуальное устройство Android (AVD, Android Virtual Device), выполните команду *Tools ⇒ Device Manager* и щелкните мышью по кнопке *Create Device*. Открывается диалоговое окно (рис. 12) с многочисленными параметрами настройки вир­туального устройства.



*Рис.12 – Выбор виртуального устройства*

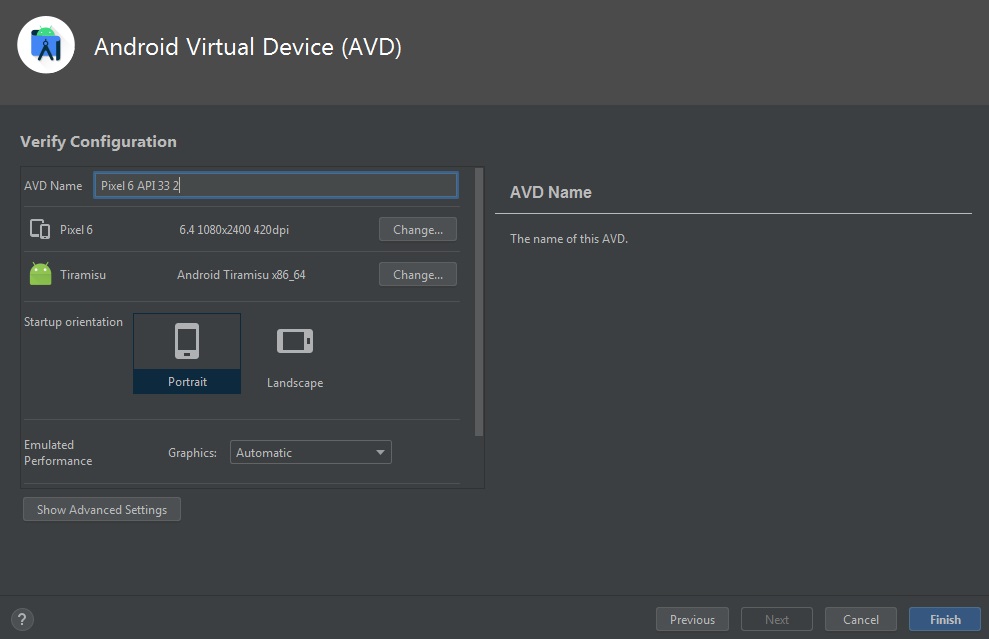
На следующем экране (рис. 13) выберите версию Android, на основе которого будет работать эмулятор.



*Рис.13 – Выбор версии Android*

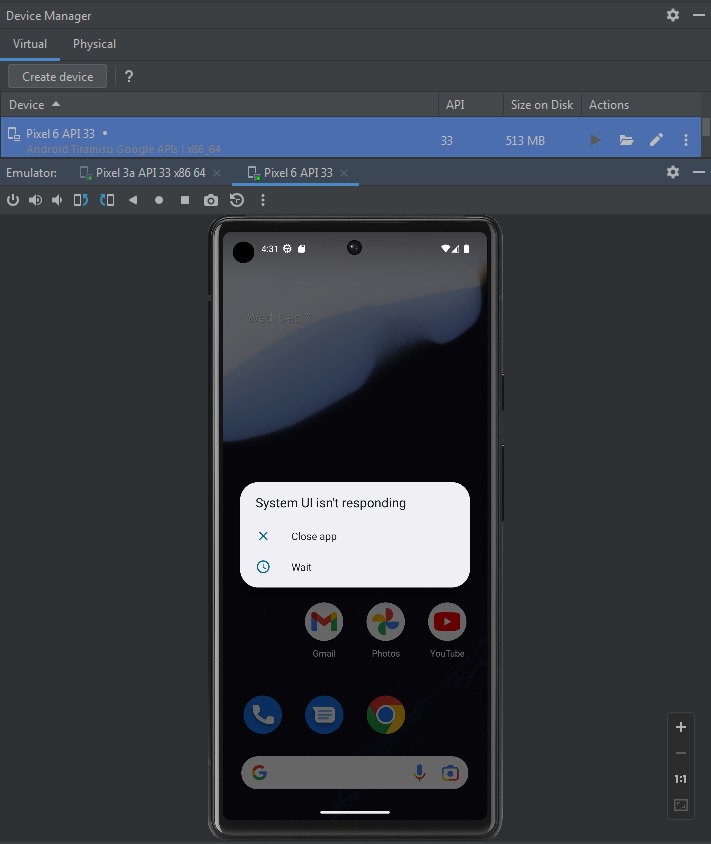
Присвойте своему эмулятору имя, по которому вы сможете узнать его в будущем, выберите портретную или ландшафтную (альбомную) ориентацию.

и нажмите кнопку Finish (рис. 14).



*Рис.14 – Настройка свойств эмулятора*

Когда виртуальное устройство будет создано, в нем можно запустить эмулятор. На панели инструментов Android Studio нажмите кнопку Run (символ «воспроизведение»). Android Studio находит созданное виртуальное устройство (рис. 15), устанавливает на нем пакет приложения и запускает его.



*Рис.15 – Работа эмулятора*

Эмулятор полезен, но тестирование на реальном устройстве дает более точные результаты [9].

# Заключение

Подводя итоги, можно сказать, что для того, чтобы стать Android-разработчиком, необходимо не только знать языки программирования Java или Kotlin и иметь навыки работы в специальных программах, таких как Android Studio и Gradle, но и уметь работать в веб-сервисах для хостинга проектов Github и Gitlab. Кроме того, предстоит разобраться в работе с паттернами, изучить создание гибких интерфейсов, знать особенности работы с многопоточностью и параллелизмом и многими другими аспектами. Стоит упомянуть, что людям, собирающимся работать Android-разработчиком, необходимо быть готовым к постоянному самообучению, повышению своей квалификации, быть стрессоустойчивыми и немного перфекционистами. По моему мнению, современный Android-разработчик – это сплав хорошего профессионала, вечного студента и постоянно стремящегося к совершенству идеалиста.

Сейчас направление Android-разработки очень популярно, так как мобильные телефоны есть практически у каждого, и у школьников и даже у некоторых дошкольников. Рынок устройств на операционной системе Android растет и расширяется, внедряется в торговое оборудование и в автомобильном рынке. Кроме того, растет доля смарт-устройств и бытовых роботов, а также самых разнообразных умных устройств. Специалисты android-разработки в данный момент очень востребованы на рынке, поэтому найти работу будет довольно легко.

# Список литературы

1. Профессия: мобильный разработчик: [Электронный ресурс] // Хабр. URL: https://habr.com/ru/company/ruvds/blog/495422/ (Дата обращения: 04.12.2022)
2. Сколько зарабатывают Андроид-разработчики на приложениях и играх в 2022 году: [Электронный ресурс] // УчисьОнлайн.ру. URL: https://uchis-online.ru/blog/professii/skolko-zarabativayut-android-razrabotchiki (дата обращения: 03.11.2022).
3. История Android: эволюция самой популярной операционной системы в мире: [Электронный ресурс] // Высокие технологии Trashbox.ru. URL: https://trashbox.ru/link/history-of-android/ (дата обращения: 26.10.2022).
4. Все названия версий Android от 1 до 13: [Электронный ресурс] // China mobile Все о китайских гаджетах. URL: https://china-mobile.ru/vse-nazvaniya-versij-operatsionnyh-sistem-android/ (дата обращения: 17.11.2022).
5. Чем занимаются андроид-разработчики: [Электронный ресурс] // Код. Журнал Яндекс Практикума. URL: https://thecode.media/android-developer/ (дата обращения: 20.11.2022).
6. Java vs Kotlin для Android-разработки: ответы «за» и «против» : [Электронный ресурс] // Tproger. URL: https://tproger.ru/articles/java-vs-kotlin/ (дата обращения: 23.11.2022).
7. Java vs Kotlin: что выбрать в 2022 году: [Электронный ресурс] // Checkroi. URL: https://checkroi.ru/blog/java-ili-kotlin-chto-vybrat/ (дата обращения: 04.12.2022).
8. Как сделать приложение для Android самостоятельно: [Электронный ресурс] // Skillbox Media. URL: https://skillbox.ru/media/code/kak\_sdelat\_prilozhenie\_dlya\_android/ (дата обращения: 11.11.2022).
9. Android. Программирование для профессионалов/Билл Филлипс, Крис Стюарт, Кристин Марсикано, Брайан Гарднер. - 4-е изд. - СПб.: Питер, 2021. – 704 с.