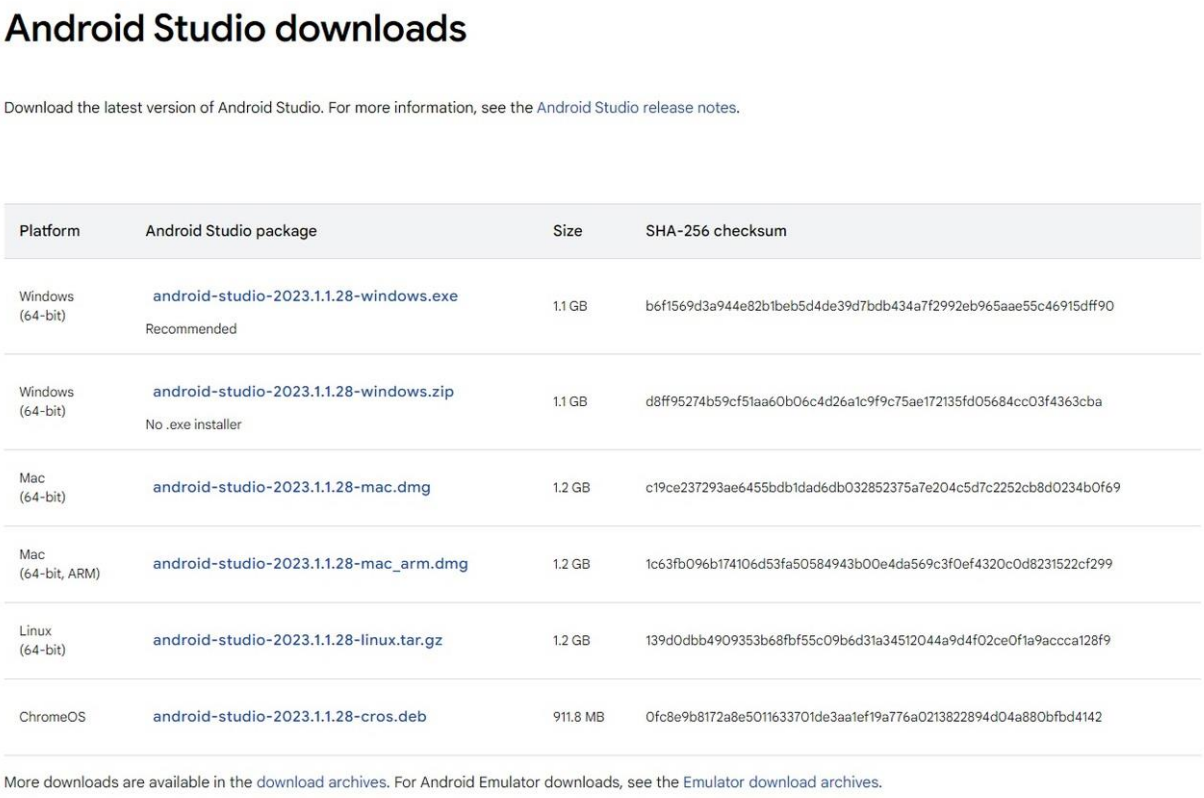
Часть 1. Установка Android Studio

Множество сред разработки поддерживают создание приложений для Android, но наиболее предпочтительной считается Android Studio. Она разработанна специально для Android, предлагает интегрированные инструменты и оптимизированный рабочий процесс, что делает ее наиболее предпочтительным выбором для разработчиков. Для начала работы с Android Studio, необходимо скачать установщик с официального сайта: https://developer.android.com/studio



Android Studio downloads

Download the latest version of Android Studio. For more information, see the Android Studio release notes.

Platform	Android Studio package	Size	SHA-256 checksum
Windows (64-bit)	android-studio-2023.1.1.28-windows.exe Recommended	1.1 GB	b6f1569d3a944e82b1beb5d4de39d7bdb434a7f2992eb965aae55c46915dff90
Windows (64-bit)	android-studio-2023.1.1.28-windows.zip No .exe installer	1.1 GB	d8ff95274b59cf51aa60b06c4d26a1c9f9c75ae172135fd05684cc03f4363cba
Mac (64-bit)	android-studio-2023.1.1.28-mac.dmg	1.2 GB	c19ce237293ae6455bdb1dad6db032852375a7e204c5d7c2252cb8d0234b0f69
Mac (64-bit, ARM)	android-studio-2023.1.1.28-mac_arm.dmg	1.2 GB	1c63fb096b174106d53fa50584943b00e4da569c3f0ef4320c0d8231522cf299
Linux (64-bit)	android-studio-2023.1.1.28-linux.tar.gz	1.2 GB	139d0dbb4909353b68fbf55c09b6d31a34512044a9d4f02ce0f1a9accca128f9
ChromeOS	android-studio-2023.1.1.28-cros.deb	911.8 MB	0fc8e9b8172a8e5011633701de3aa1ef19a776a0213822894d04a880bfbd4142

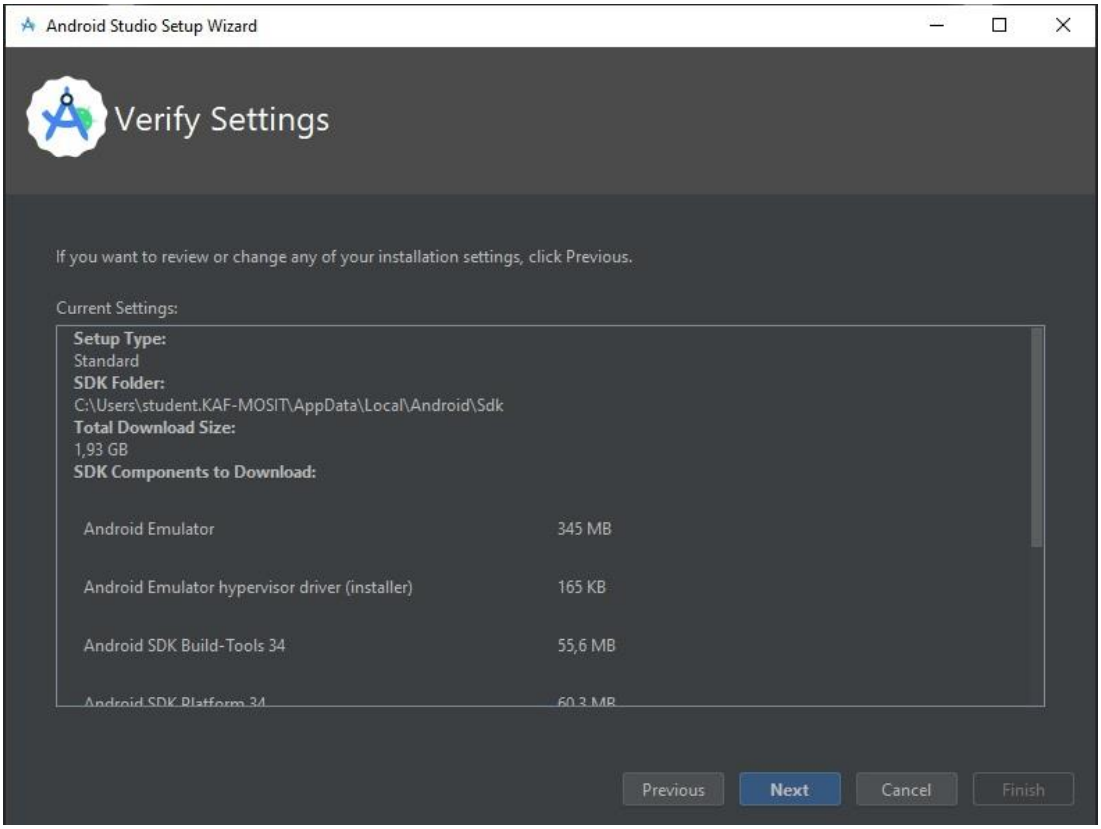
More downloads are available in the download archives. For Android Emulator downloads, see the Emulator download archives.

Примечание: среда Android Studio требует для своей работы много системных ресурсов. Как следствие, для комфортной работы программиста рекомендуется достаточно мощный компьютер (так, например, рекомендуемый объем оперативной памяти - 8 ГБ).

Для разработки на Android в дополнение к Android Studio вам потребуется также установить Android Software Development Kit (SDK). Этот набор инструментов является обязательным для создания приложений, поскольку включает в себя библиотеки API, эмуляторы устройств, инструменты для отладки и многое другое.

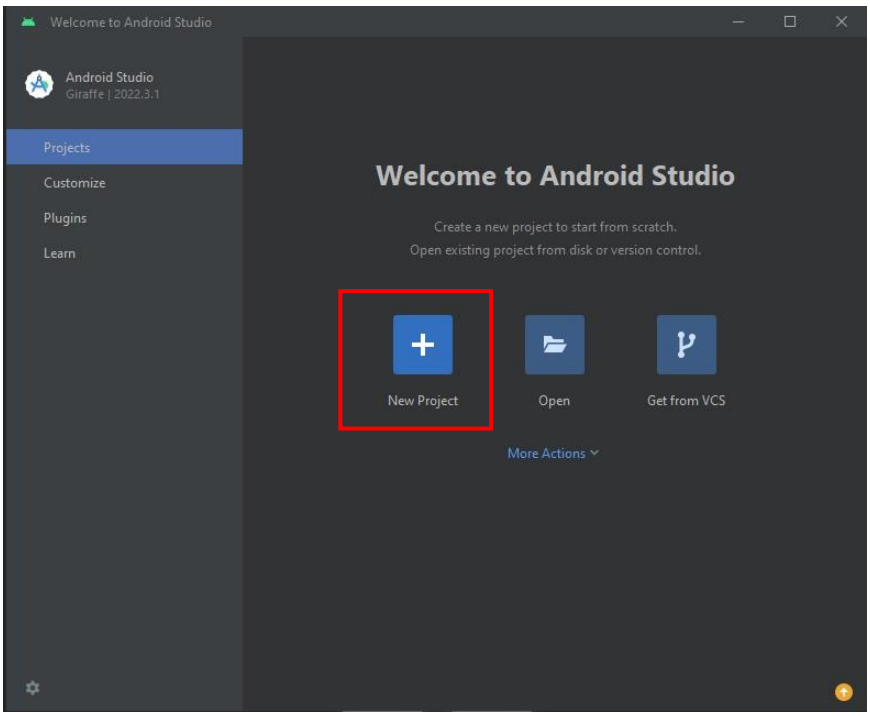
Если вы впервые работаете с Android Studio и у вас еще не установлен Android SDK, сама среда разработки предложит это сделать при первом запуске. Она автоматически определит необходимые компоненты и предложит их установить. К

этим компонентам, помимо основного Android SDK, могут относиться различные дополнения и инструменты, которые улучшат и облегчат процесс разработки. Например, это могут быть дополнительные пакеты API, инструменты для работы с базами данных, утилиты для профилирования производительности и другие полезные ресурсы, которые вы можете установить позже самостоятельно.

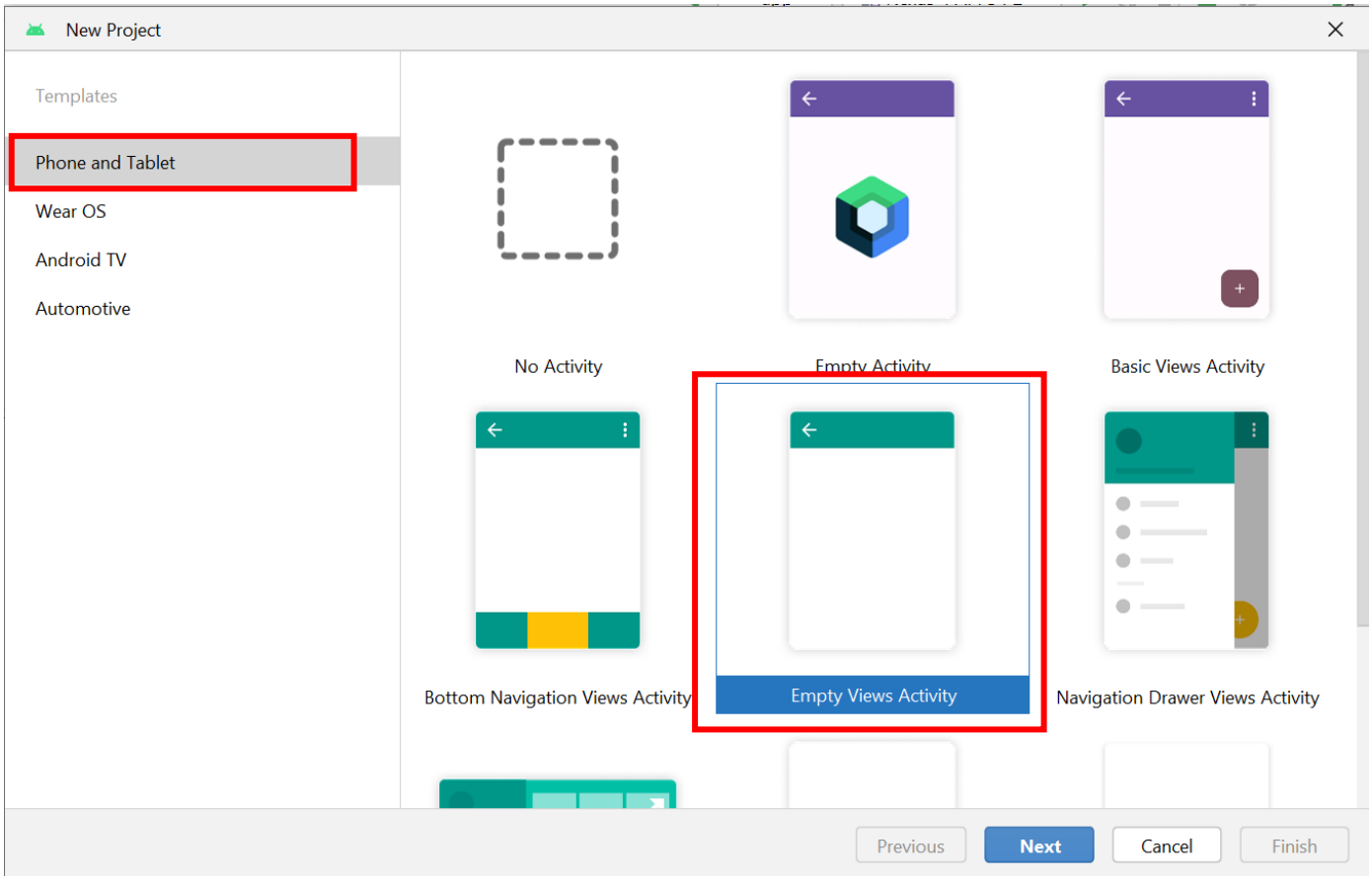


Часть 2. Создание проекта

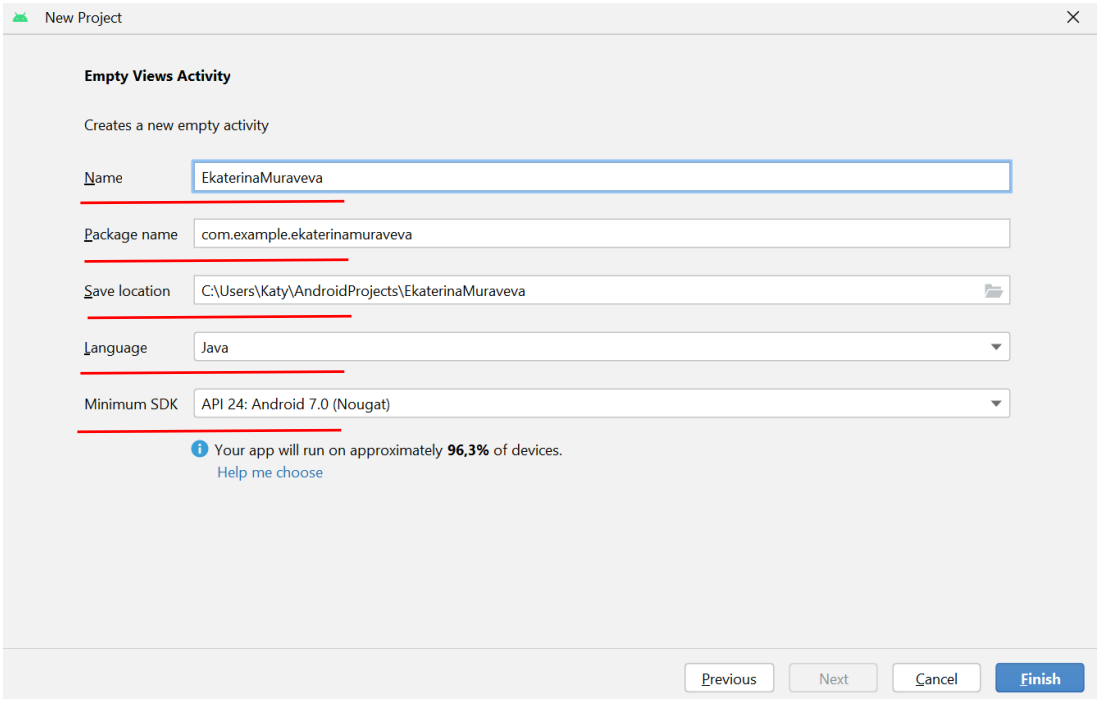
После установки и запуска Android Studio создайте новый проект. Для этого на начальном экране выберите кнопку "Create New Project".



Android Studio предлагает различные шаблоны для вашего нового проекта. Для простого приложения выберите тип шаблона "Phone and Tablet" и шаблон "Empty View Activity", который представляет собой пустой шаблон проекта с одним экраном.



Далее необходимо заполнить настройки проекта.

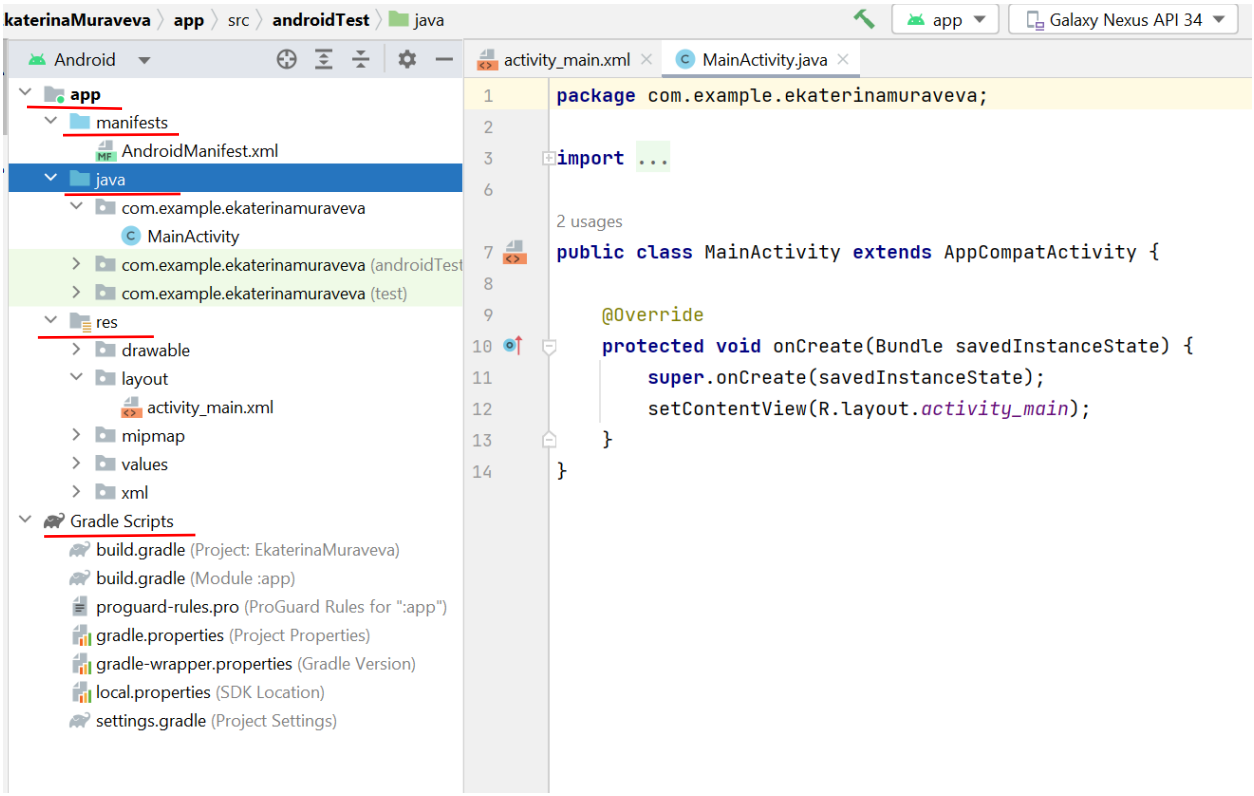


- Name: Отвечает за название проекта;
- Package name: Название пакета. Уникальный идентификатор вашего приложения, обычно следует формату com.example.myapp;
- Save location: Место на компьютере, где будет сохранен проект;
- Language: Язык программирования, на котором будет написан проект;
- Minimum SDK: Минимальая версия Android, которую будет поддерживать ваше приложение.

После настройки нажмите кнопку "Finish" и дождитесь загрузки проекта.

Часть 3. Структура проекта

Теперь рассмотрим структуру проекта.



Модуль "app" является основным компонентом проекта Android и содержит файлы, необходимые для сборки проекта. В нём хранятся следующие элементы:

1) manifests: содержит файл AndroidManifest.xml где описаны основные характеристики приложения — компоненты (активности, службы и т.д.), требуемые разрешения (на отправку уведомлений, интернет соединение и др.) и настройки приложения (название, тема, иконка и т.д.).
2) java: содержит три пакеты, отвечающих за: код приложения, код для инструментальных тестов, выполняющихся на Android устройствах и код для модульных тестов, выполняющихся на вашем компьютере.
3) res: содержит все ресурсы, не связанные с кодом, такие как XML-макеты, UI элементы (кнопки, текстовые поля), строки, изображения (drawables), стили и темы.

Модуль "Gradle Scripts" управляет процессом сборки вашего проекта. В нём хранятся следующие элементы:

1) build.gradle (Project: «Название вашего проекта»): Этот файл на уровне проекта содержит конфигурацию, применяемую ко всем модулям в проекте, а также включает ссылки на плагины Gradle, используемые проектом.

2) build.gradle (Module :app): Этот файл на уровне модуля содержит конфигурацию сборки, специфичную для данного модуля. Здесь определяются настройки конкретного модуля, такие как версия SDK, версии зависимостей, конфигурации сборки (например, отладка и выпуск) и другие настройки Android, специфичные для модуля.
3) settings.gradle: Содержит ссылки на модули, включенные в ваш проект. Каждый модуль, который должен быть собран как часть вашего проекта, должен быть заявлен здесь.

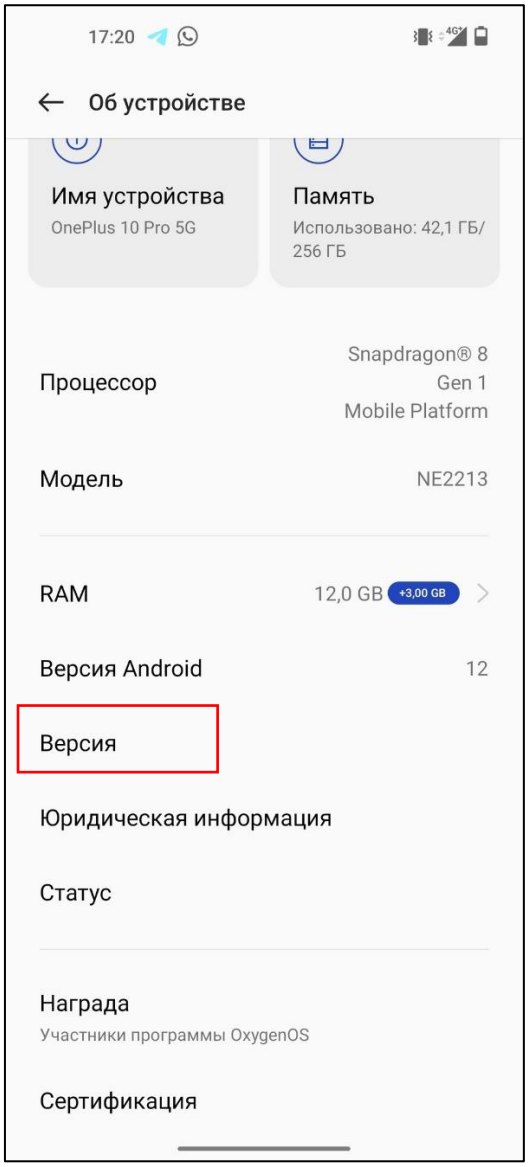
Часть 4. Запуск проекта

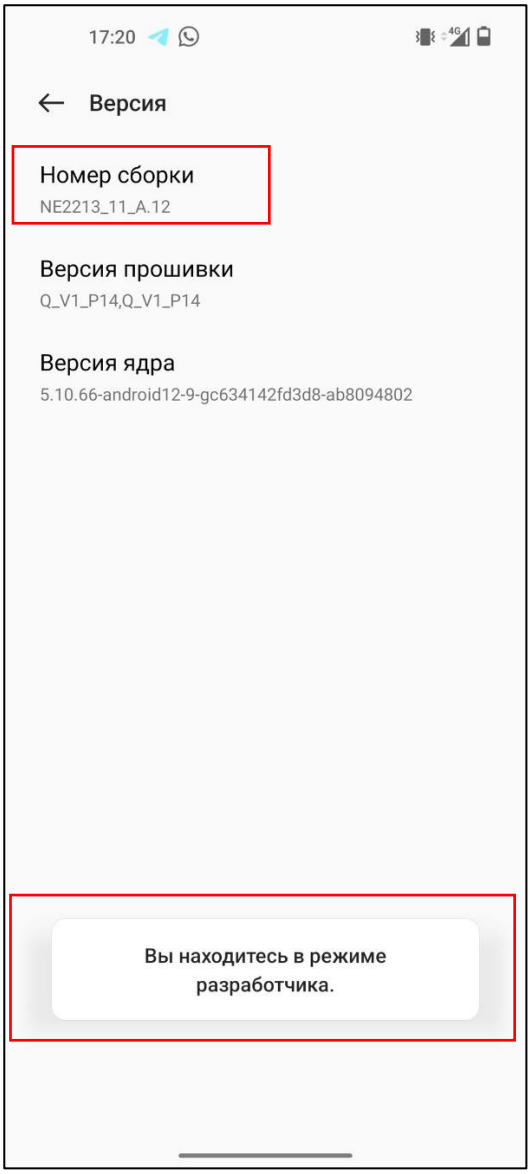
Созданный проект можно запустить как на реальном, так и на виртуальном устройстве. Рассмотрим оба варианта.

Для запуска проекта на физическом устройстве необходимо активировать режим разработчика и разрешить отладку по USB.

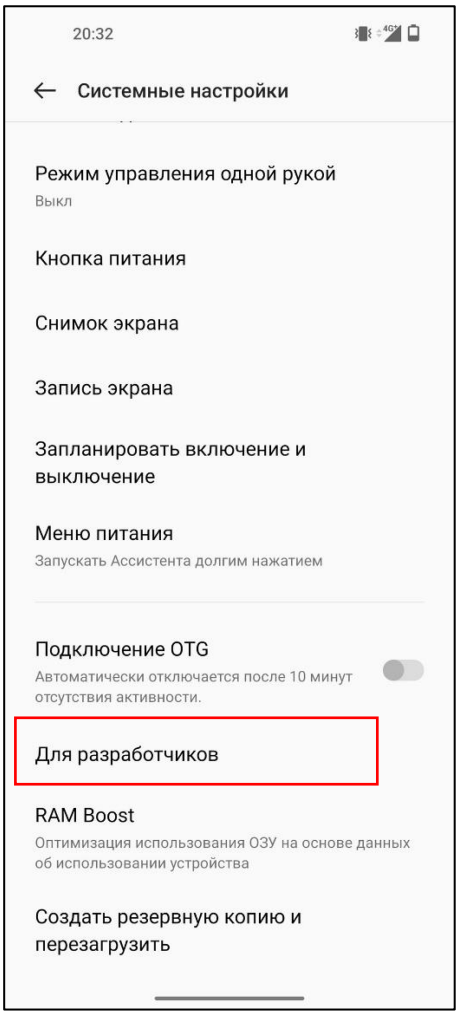
Примечание: ниже указана последовательность действий для телефона OnePlus 10pro для активирования режима разработчика, для Вашей модели последовательность шагов и название элементов настройки данного режима может отличаться.

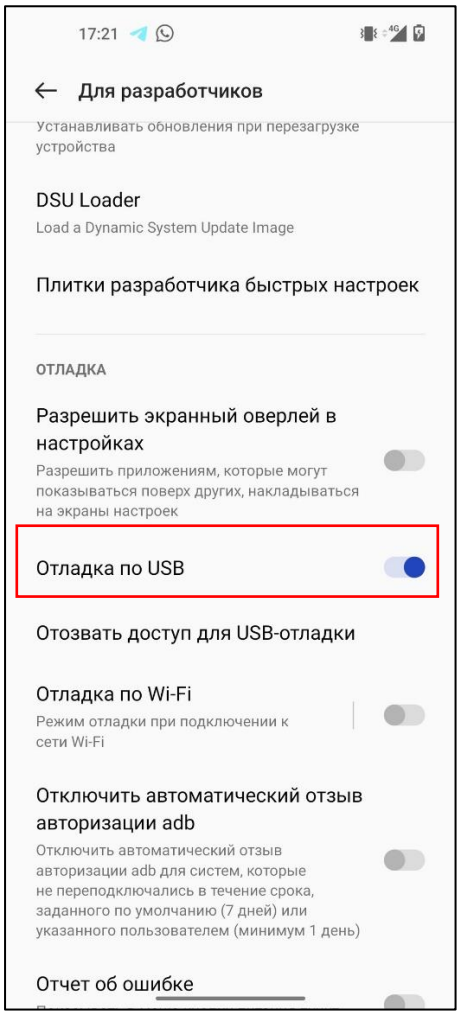
Чтобы это сделать вам нужно перейти в «Настройки», далее выбрать пункт «Об устройстве» и в разделе «Версия» и 7 раз нажать на пункт «Номер версии» («Номер сборки» в других версиях), после чего должно появиться всплывающее уведомление о том, что режим разработчика включен.



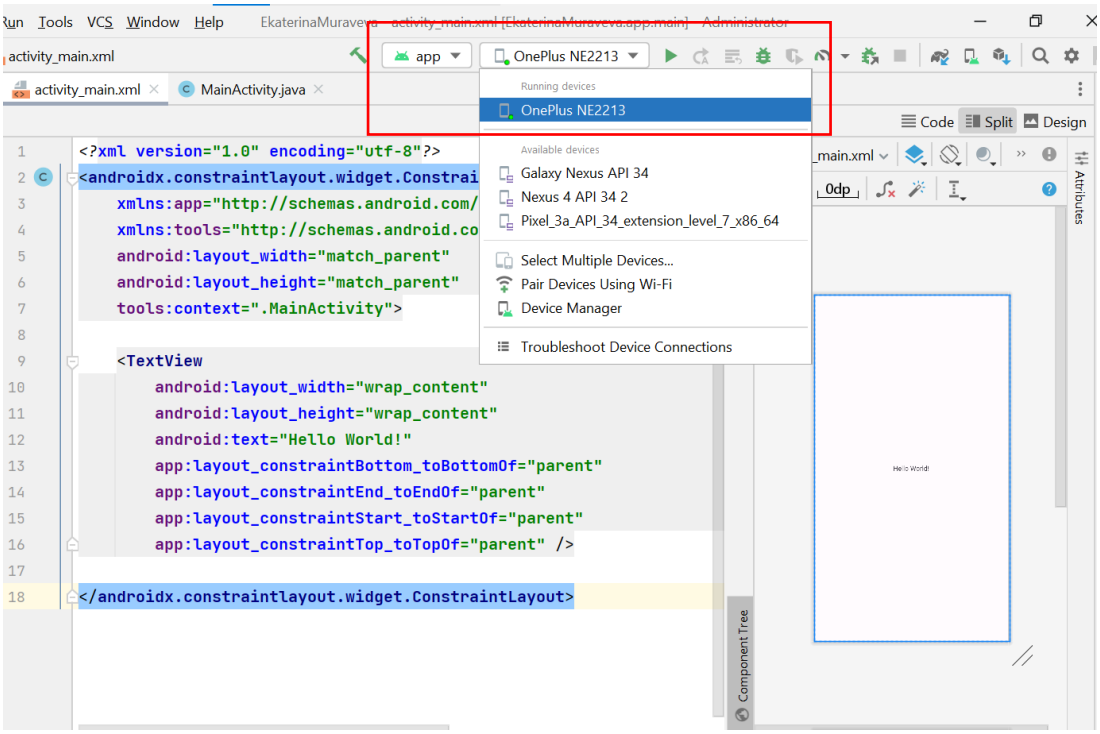


Теперь, для включения отладки по USB вернитесь в «Настройки» и в разделе «Дополнительные настройки» выберите пункт «Возможности разработчика», где необходимо включить пункт «Отладка по USB».

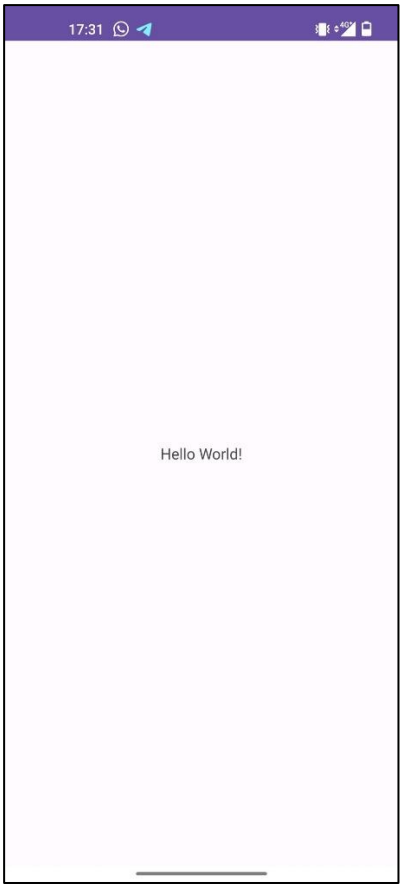




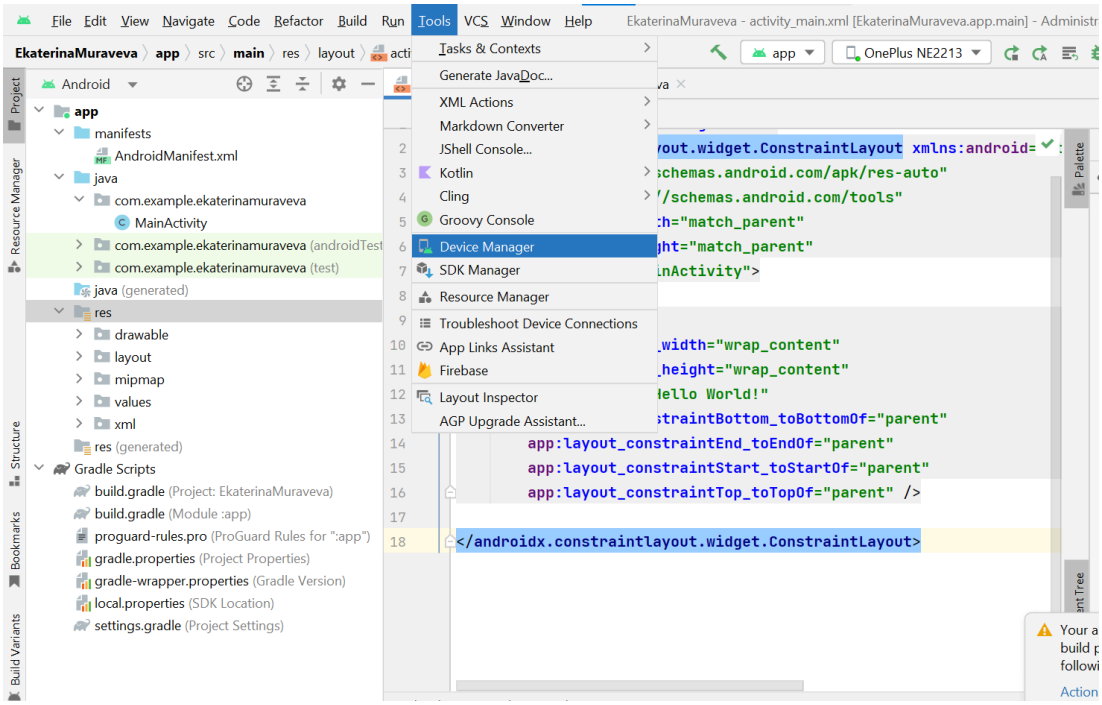
Далее подключаем устройство к компьютеру с открытым Android Studio. Программа должна автоматически обнаружить устройство и отобразить его в разделе "Available devices". Запускаем проект, нажав зеленую кнопку справа.



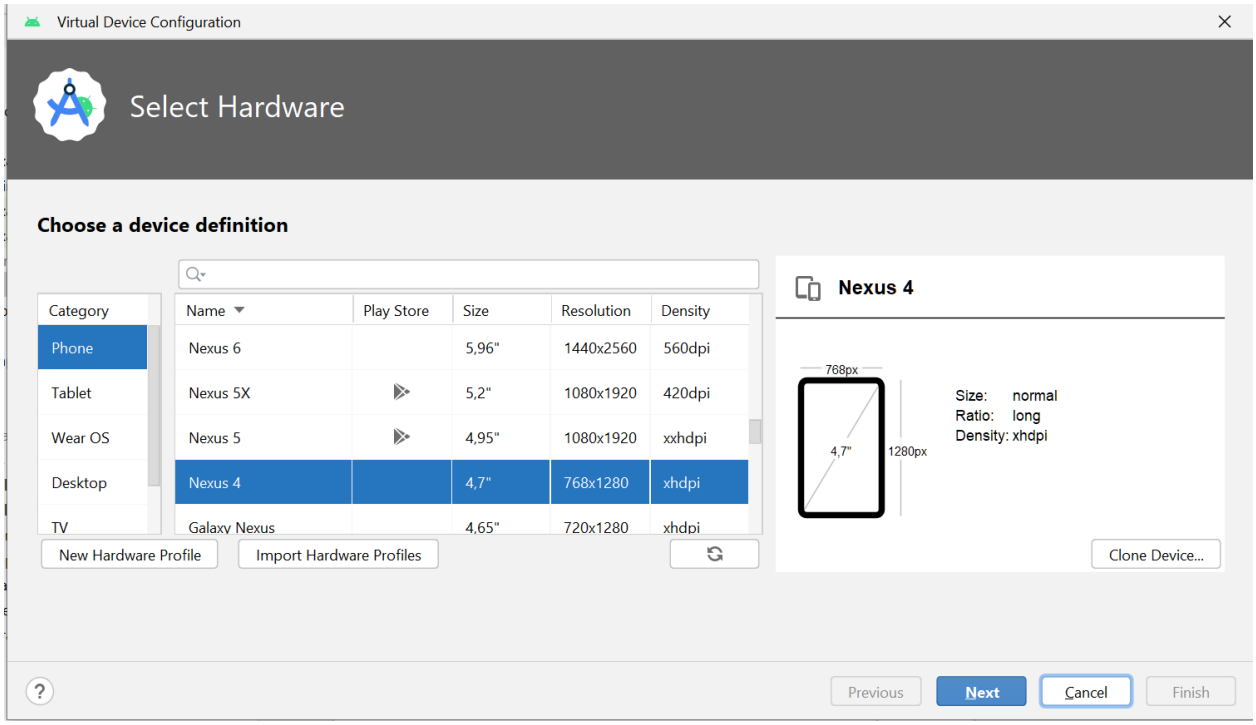
После сборки приложения вы сможете увидеть запущенное приложение на экране вашего телефона.



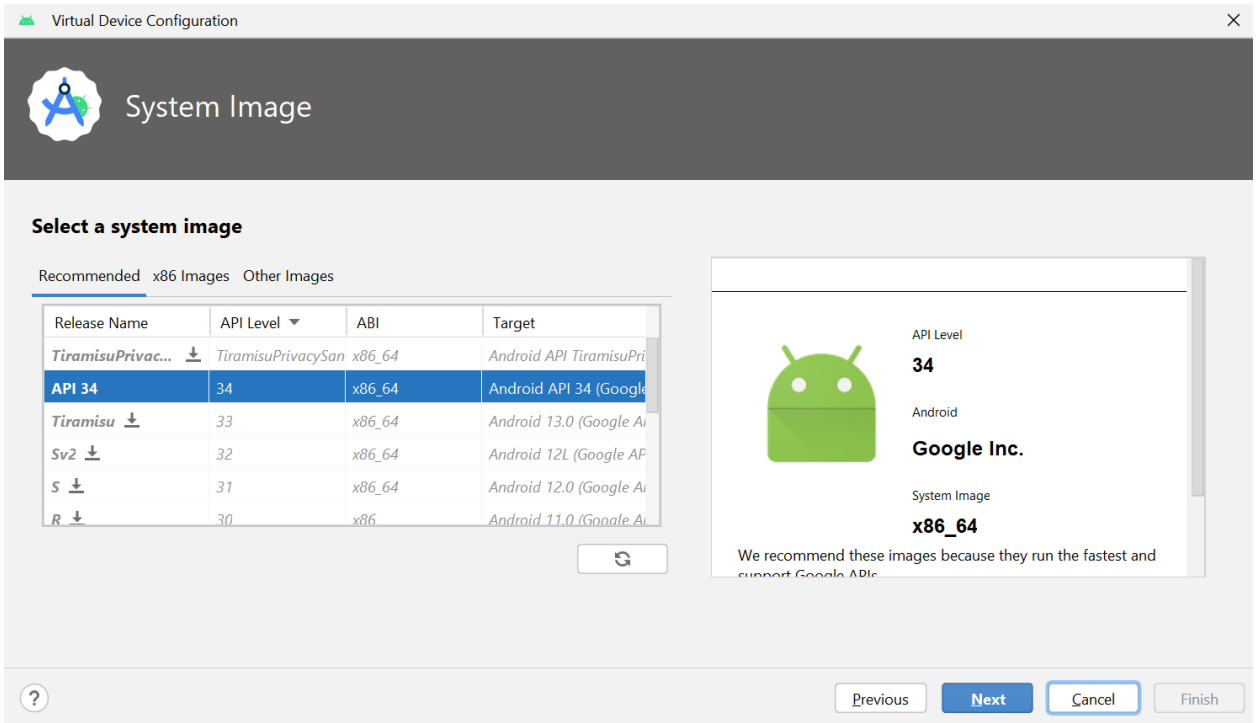
Для запуска на виртуальном устройстве необходимо создать его в Android Studio. Чтобы это сделать перейдите в пункт "Tools" и выберите раздел "Device Manager". Затем, в открывшемся боковом окне нажмите «+» для создания нового устройства



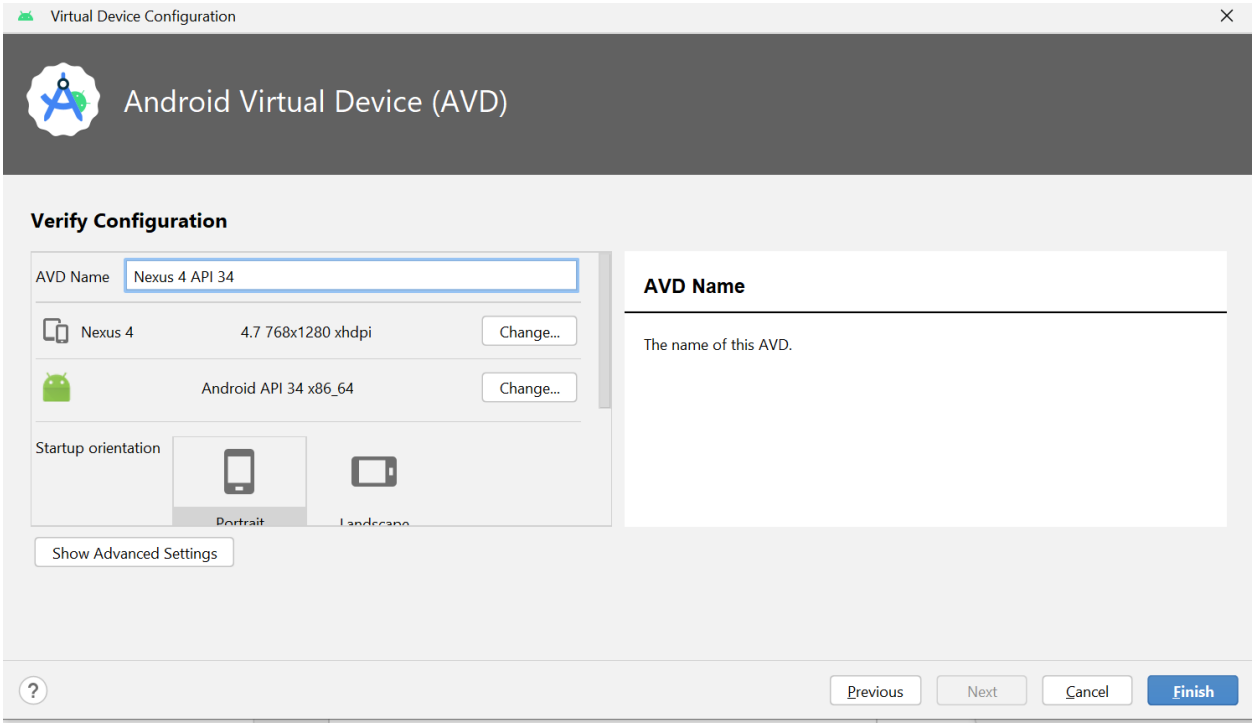
Теперь выберите параметры виртуального устройства. Сначала настройте тип устройства и его размеры.



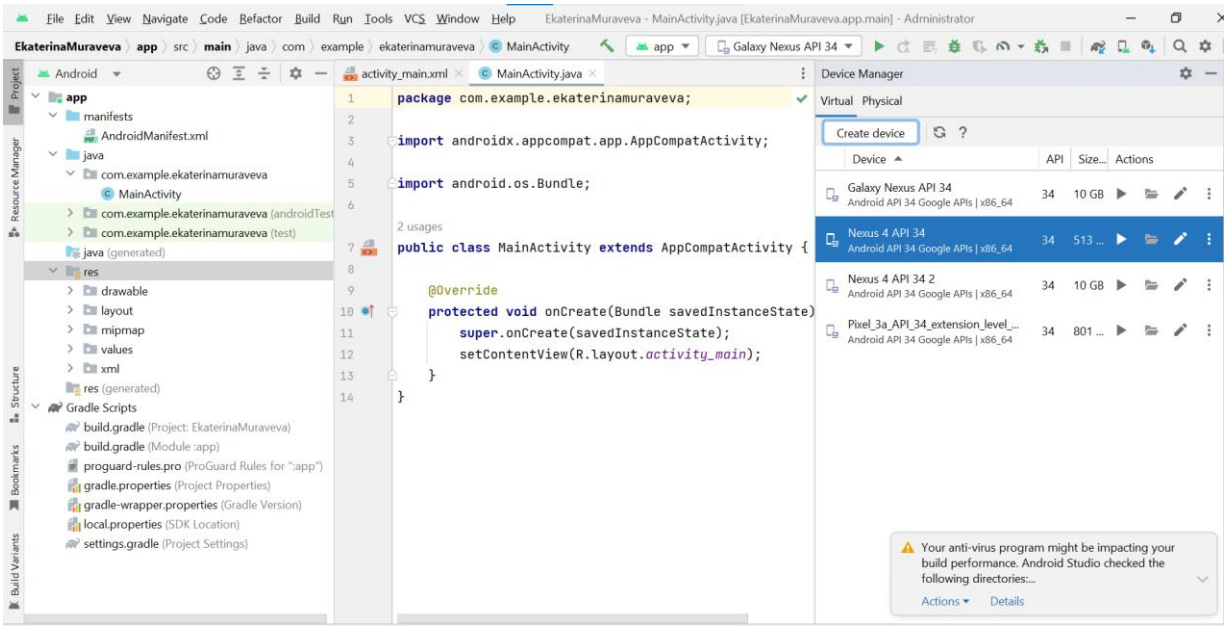
Далее выберите версию ОС Android, которая будет установлена на устройство, лучше выбирать самую последнюю. Если создаете устройство впервые, то перед выбором ОС ее нужно будет скачать, для этого нажмите на кнопку загрузки, рядом с версией ОС.



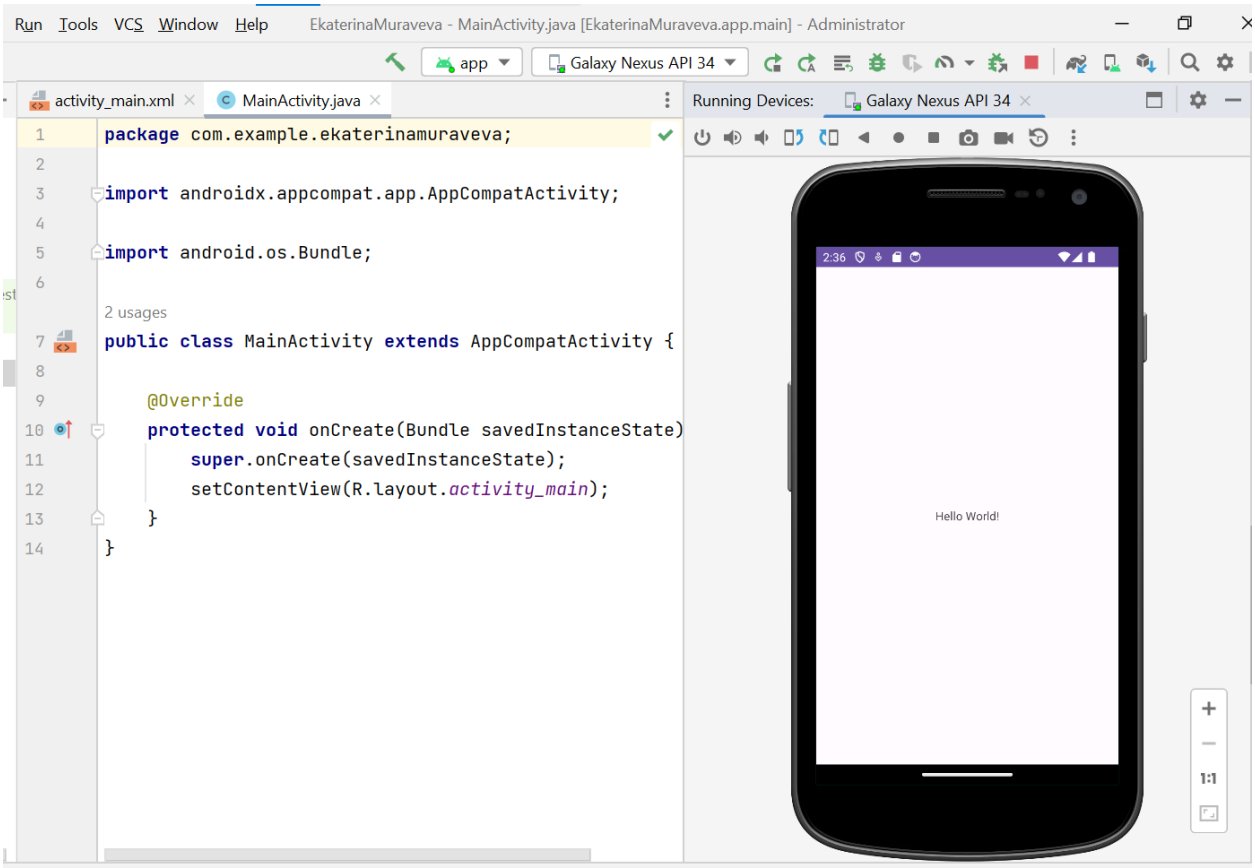
После выбора устройства и версии Android вы можете еще раз посмотреть выбранные характеристики, выбрать тип ориентации экрана и создать устройство



Когда устройство создано, вы можете выбрать его в качестве основного в разделе "Available devices" и запустить приложение на нем.



Запущенное устройство будет доступно в пункте "Running Devices", который автоматически откроется при окончании сборки проекта. Функционал приложения будет полностью идентичен с физическим устройством.



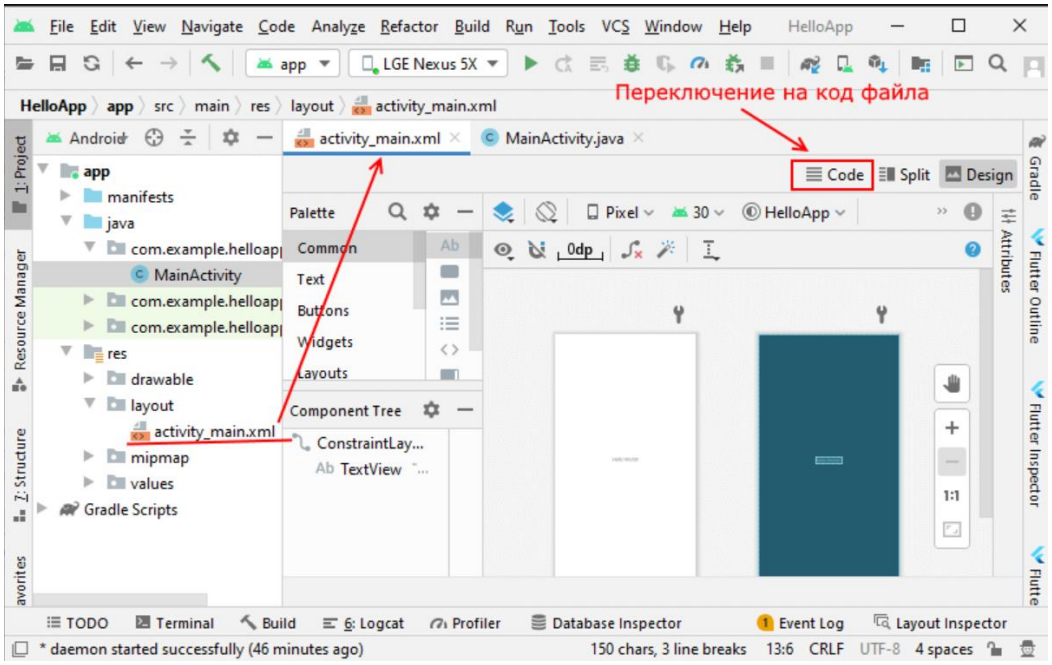
Часть 5. Создание графического интерфейса

Выполнение приложения Android по умолчанию начинается с класса MainActivity, который по умолчанию открыт в Android Studio.

Каждый отдельный экран или страница в приложении описывается таким понятием как activity. В литературе могут использоваться различные термины: экран, страница, активность. Если запустить приложение, то на экране мы, по сути, увидим

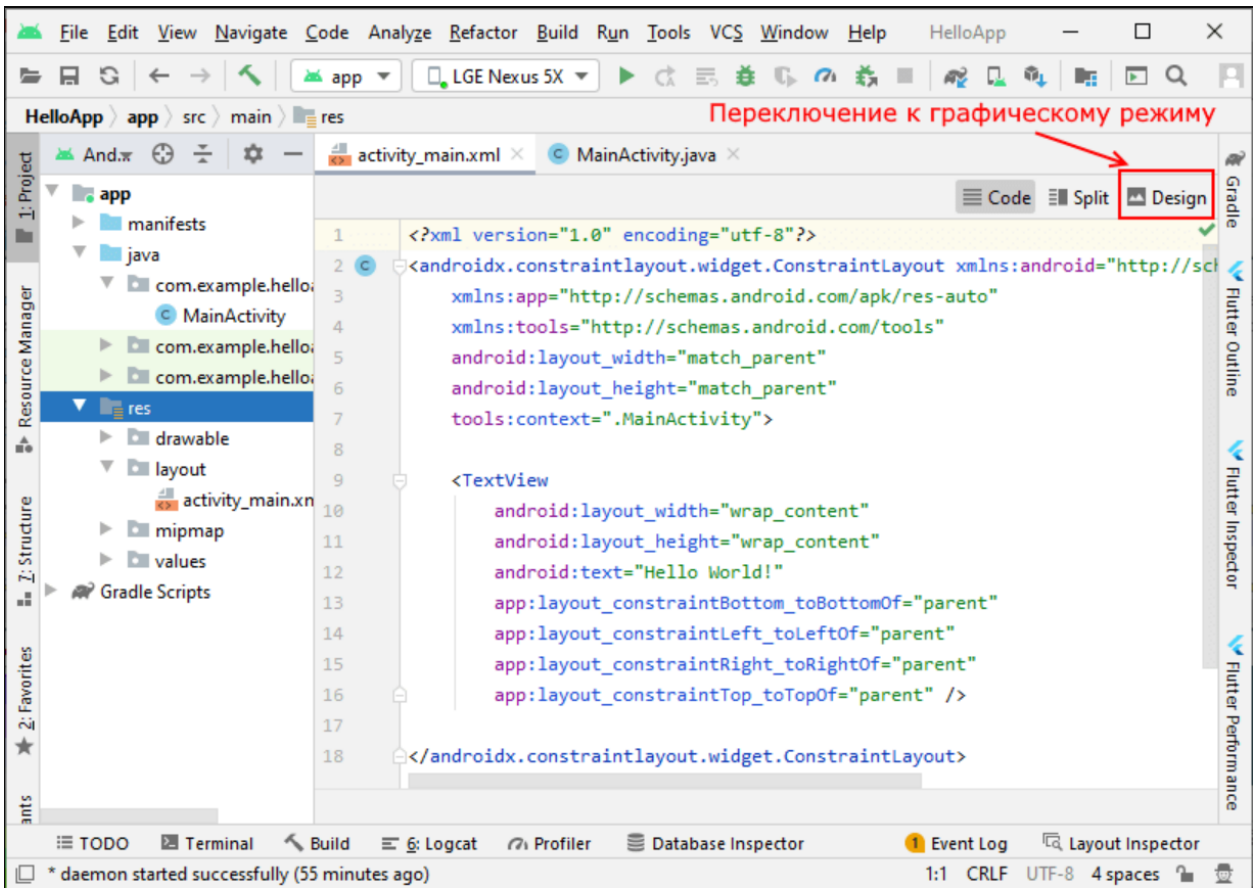
определенную activity. По умолчанию MainActivity содержит только один метод onCreate(), в котором фактически и создается весь интерфейс приложения.

В метод setContentView() передается ресурс разметки графического интерфейса. Именно здесь и решается, какой именно визуальный интерфейс будет иметь MainActivity.



Android Studio позволяет работать с визуальным интерфейсом как в режиме кода, так и в графическом режиме. Так, по умолчанию файл открыт в графическом режиме, и мы наглядно можем увидеть, как у нас примерно будет выглядеть экран приложения. И даже набрасать с панели инструментов какие-нибудь элементы управления, например, кнопки или текстовые поля.

Но также мы можем работать с файлом в режиме кода, поскольку activity_main.xml — это обычный текстовый файл с разметкой xml. Для переключения к коду нажмем на кнопку Code над графическим представлением. (Дополнительно с помощью кнопки Split можно переключиться на комбинированное представление код + графический дизайнер)



Большинство визуальных элементов, наследующихся от класса View, такие как кнопки, текстовые поля и другие, располагаются в пакете android.widget.

При определении визуального интерфейса есть три стратегии:

- Создать элементы управления программно в коде java.
- Объявить элементы интерфейса в XML.
- Сочетание обоих способов - базовые элементы разметки определить в XML, а остальные добавлять во время выполнения.

Часть 6. Верстка в Android. Язык XML

Как правило, для определения ресурсов, а в том числе и визуального интерфейса, в проектах под Android используются специальные файлы xml. Эти файлы являются ресурсами разметки и хранят определение визуального интерфейса в виде кода XML.

Объявление пользовательского интерфейса в файлах XML позволяет отделить интерфейс приложения от кода. Что означает, что мы можем изменять определение интерфейса без изменения кода java. Кроме того, объявление разметки в XML позволяет легче визуализировать структуру интерфейса и облегчает отладку.

XML — это язык свободного описания структур документов. То есть, если

необходимо, чтобы в документе присутствовал какой-либо элемент, то мы для него определяем некоторый тег (маркер в тексте). Например, для описания элемента «текстовая строка» можно условиться использовать тег <string>, где первая метка указывает начало описания элемента, а вторая (со знаком /) — конец описания. Между парой тегов помещается текстовое представления содержимого элемента. Для каждого элемента применяется своя пара тегов, при этом однотипные элементы описываются одинаковой парой тегов. Таким образом, для описания двух строк нужны две пары тегов:

```
<string>это первая строка</string>
<string>это вторая строка</string>
```

В открывающем теге можно поместить атрибуты описываемого элемента, такие как цвет, размер, начертание, выравнивание и т. п., то есть описать особенности формируемого элемента. Атрибут — это свойство описываемого элемента. При этом у однотипных элементов полный набор атрибутов будет совпадать, но в описании можно использовать не все свойства. Каждому имени атрибута присваивается значение, записанное в виде текстовой строки, то есть заключенное в двойные кавычки. Разделяются свойства пробелом либо переносом строки. Вернемся к рассматриваемому примеру:

```
<string color = "red" align = "center">это первая строка</string>
```

Данная разметка описывает текстовую строку, написанную красным шрифтом (начертание и размер установлены по умолчанию, поскольку эти свойства не указаны при описании) с выравниванием в центре страницы.

Каким бы свободным не был стиль XML-документа, все-таки существуют правила его формирования.

- В языке XML все теги парные. Это значит, что у каждого открывающего тега обязательно должен присутствовать закрывающий тег. Это правило позволяет описывать вложенные элементы, то есть помещать внутри одного элемента другие. Если тело тега пусто, то два тега записываются в один, который завершается косой чертой;
- Документ может содержать декларацию — строку заголовка, в которой указывается версия языка и используемая текстовая кодировка;
- Имена тегов могут содержать буквы, цифры и специальные знаки, такие

как знак подчеркивания (_), но должны начинаться с буквы. Теги записываются с соблюдением регистра, поскольку XML регистрозависим;

- Если возникает необходимость использования одинаковых имен элементов для разного типа структур документа, применяют понятие пространства имен. Чтобы различать такие элементы, необходимо задать соответствие — специальный уникальный идентификатор ресурса или URI с конкретным именем элемента. В качестве идентификатора чаще всего используется адрес своего (необязательно реально существующий) ресурса. Пространство имен определяется благодаря атрибуту xmlns в начальном теге элемента;
- В XML-тексте комментарии выделяются тегами <! >;

```
<androidx.constraintlayout.widget.ConstraintLayout xmlns:ar
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

Элемент — это структурная единица XML-документа. Границы элементов маркируются одинаковыми начальным и конечным тегами. Внутри этой границы может быть текстовая строка значения элемента. Элемент может быть также представлен пустымтегом, то есть не включающим в себя другие элементы и/или символьные данные.

Помимо текстового значения элемент может включать другие элементы. Такие элементы называются дочерними (child) элементами. Дочерних элементов может быть несколько. Элемент, который окружает дочерний элемент, называется родительским (parent). По естественным причинам у дочернего элемента может быть только один родительский. Важно, чтобы любой дочерний элемент располагался

целиком внутри родительского. То есть пары открывающих и закрывающих тегов всех дочерних элементовдолжны быть заключены (окружены) парой открывающего и закрывающего тегов родительского элемента. В случае нарушения этого правила любая программа не сможет прочитать ваш документ и выдаст сообщение об ошибочности. Автор документа, вкладывая одни элементы в другие, задает иерархическую структуру внутри документа.

Часть 7. Ресурсы в Android

Создавая приложение для Android, помимо написания программ на языке Java необходимо также работать с ресурсами. В экосистеме Android принято отделять такие файлы, как изображения, музыка, анимации, стили, макеты окон, строковые константы — в общем все части оформления GUI (Graphical User Interface — графический интерфейс пользователя) от программного кода. Большая часть ресурсов (за исключением мультимедийных) хранятся во внешних XML-файлах. При создании и развитии программного проекта внешние ресурсы легче поддерживать, обновлять и редактировать.

Как уже было показано, каждое приложение на Android содержит каталог для ресурсов res/. Доступ к информации в каталоге ресурсов из приложения осуществляется через класс R, который автоматически генерируется средой разработки.

В общем случае ресурсы представляют собой файл (например, изображение) или значение (например, заголовок программы), связанные с создаваемым приложением по имени ресурса. Удобство использования ресурсов заключается в том, что их можно заменять/изменять без изменения программного кода приложения или компиляции. Поскольку имена файлов для ресурсов фактически будут использованы как имена констант в R, то они должны удовлетворять правилам написания имен переменных в Java. Так как разработка ведется на различных ОС (Windows, Mac, Linux), то также есть еще ограничения. В итоге имена файлов должны состоять исключительно из букв в нижнем регистре, чисел и символов подчеркивания.

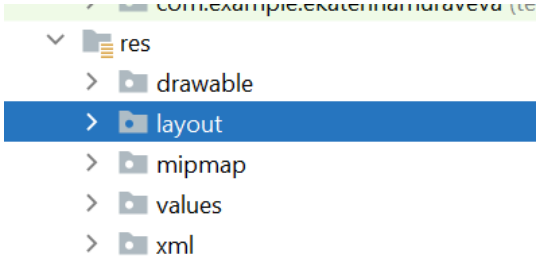
В Android используются два подхода к процессу создания ресурсов — первыйподход заключается в том, что ресурсы задаются внутри файла и тогда его имя

задается в месте его описания. Второй подход — ресурс задается в виде самого файла, и тогда имя файла уже и есть имя ресурса.

Для различных типов ресурсов, определенных в проекте, в каталоге res создаются подкаталоги. Поддерживаемые подкаталоги:

- animator/: xml-файлы, определяющие анимацию свойств
- anim/: xml-файлы, определяющие tween-анимацию
- color/: xml-файлы, определяющие список цветов
- drawable/: Графические файлы (.png, .jpg, .gif)
- dimensions/: xml-файлы, определяющие размерности элементов
- mipmap/: Графические файлы, используемые для иконок приложения подразличные разрешения экранов
- layout/: xml-файлы, определяющие пользовательский интерфейс приложения
- menu/: xml-файлы, определяющие меню приложения
- raw/: различные файлы, которые сохраняются в исходном виде
- values/: xml-файлы, которые содержат различные используемые в приложении значения, например, ресурсы строк
- xml/: Произвольные xml-файлы
- font/: файлы с определениями шрифтом и расширениями .ttf, .otf или .ttc, либофайлы XML, который содержат элемент <font-family>

Чаще других используют следующие ресурсы: разметка (layout), строки (string),цвета (color) и графические рисунки (bitmap, drawable).



Часть 8. Создание интерфейса пользователя при помощи ресурсов

Графический интерфейс создается с помощью представлений (View) и групп представлений (ViewGroup). Эти элементы размещаются на активности, их описания помещаются в файл манифеста, а действия с объектами прописываются программно в файле кода MainActivity.java в виде методов классов, наследуемых от классов View

и ViewGroup или атрибутивно в файле разметки *layout/activity_main.xml*. У файла разметки также имеется графический вид Graphical layout — системная имитация мобильного устройства.

В файле определяются все графические элементы и их атрибуты, которые составляют интерфейс. При создании разметки в XML следует соблюдать некоторые правила: каждый файл разметки должен содержать один корневой элемент, который должен представлять объект View или ViewGroup.

По умолчанию при создании проекта с пустой activity уже есть один файл ресурсов разметки activity_main.xml. В нем корневым элементом является элемент ConstraintLayout, который содержит элемент TextView. Как правило, корневой элемент содержит определение используемых пространств имен XML.

Часть 9. Компоненты разметки

Компоненты разметки имеют конкретный внешний вид иконкретные задачи. В SDK Android находятся множество различных компонентов, однако будет рассмотрен основной перечень из них:

- TextView – компонент, предназначенный для простого вывода текста на экран. Он просто отображает текст без возможности его редактирования.
- EditText – является подклассом класса TextView. Он также представляет текстовое поле, но теперь уже с возможностью ввода и редактированиятекста. Таким образом, в EditText мы можем использовать все те же возможности, что и в TextView.
- Button – один из часто используемых компонентов. Ключевой особенностьюкнопок является возможность взаимодействия с пользователем через нажатия.
- ImageView – является базовым элементом-контейнером для использования графики. Можно загружать изображения из разных источников, например, изресурсов программы, контент-провайдеров.

Задание

Создать проект с ФИО студента в названии.

Реализовать несколько файлов разметки и создать там базовые компоненты: Текст, Кнопка, Поле ввода, Картинка. (При желании можно расширить перечень собственными компонентами).

Источники

1) https://developer.android.com/studio
2) https://metanit.com/java/android/1.1.php
3) https://metanit.com/java/android/1.2.php
4) https://developer.android.com/studio/projects