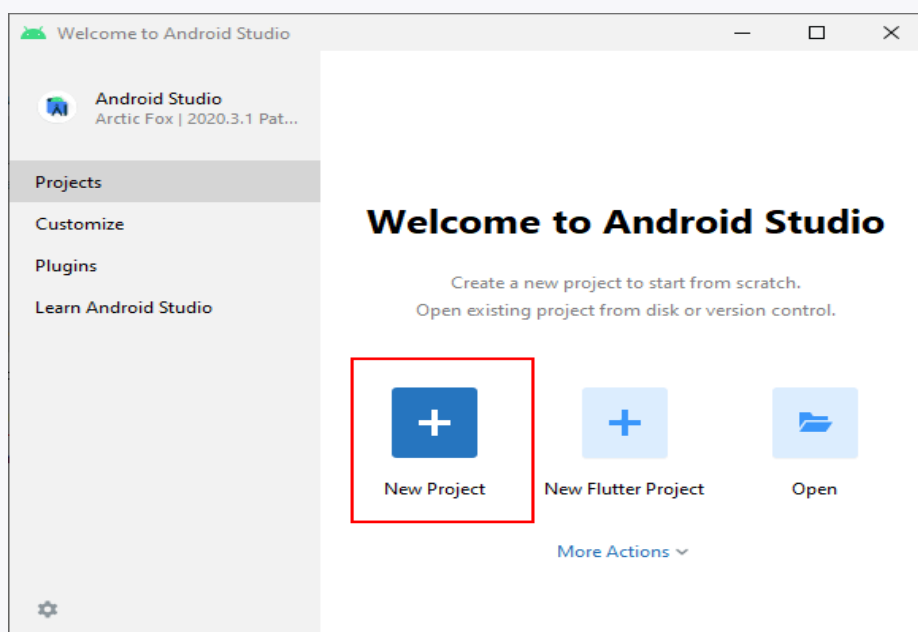


## 7 лабораторная работа

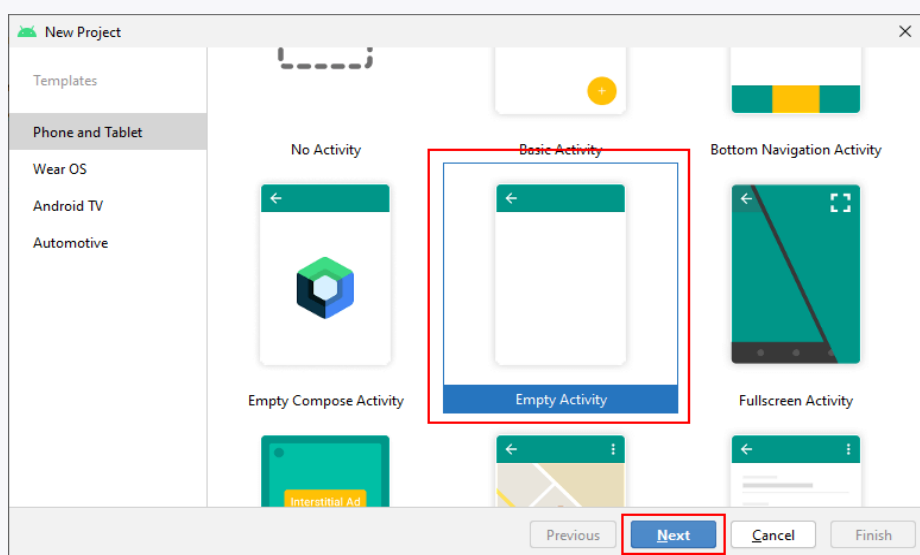
**Тема: Напишите первое приложение в Android Studio и протестируйте его на эмуляторе.**

**Цель работы: Умение писать первое приложение на Android Studio и компиляция на AVD.**

Теперь создадим первое приложение в среде Android Studio для операционной системы Android. Откроем Android Studio и на начальном экране выберем пункт **New Project**:



При создании проекта Android Studio вначале предложит нам выбрать шаблон проекта:



Android Studio предоставляет ряд шаблонов для различных ситуаций. Выберем в этом списке шаблон **Empty Activity**, который предоставляет самый простейший функционал, необходимый для начала, и нажмем на кнопку **Next**.

После этого отобразится окно настроек нового проекта:

**New Project**

**Empty Activity**

Creates a new empty activity

Name: HelloApp

Package name: com.example.helloapp

Save location: C:\Users\Eugene\AndroidStudioProjects\Java\HelloApp

Language: Java

Minimum SDK: API 21: Android 5.0 (Lollipop)

*i* Your app will run on approximately **94,1%** of devices.  
[Help me choose](#)

☐ Use legacy android.support libraries *?*  
Using legacy android.support libraries will prevent you from using the latest Play Services and Jetpack libraries

**Выберем Java**

[Previous](#) [Next](#) [Cancel](#) **Finish**

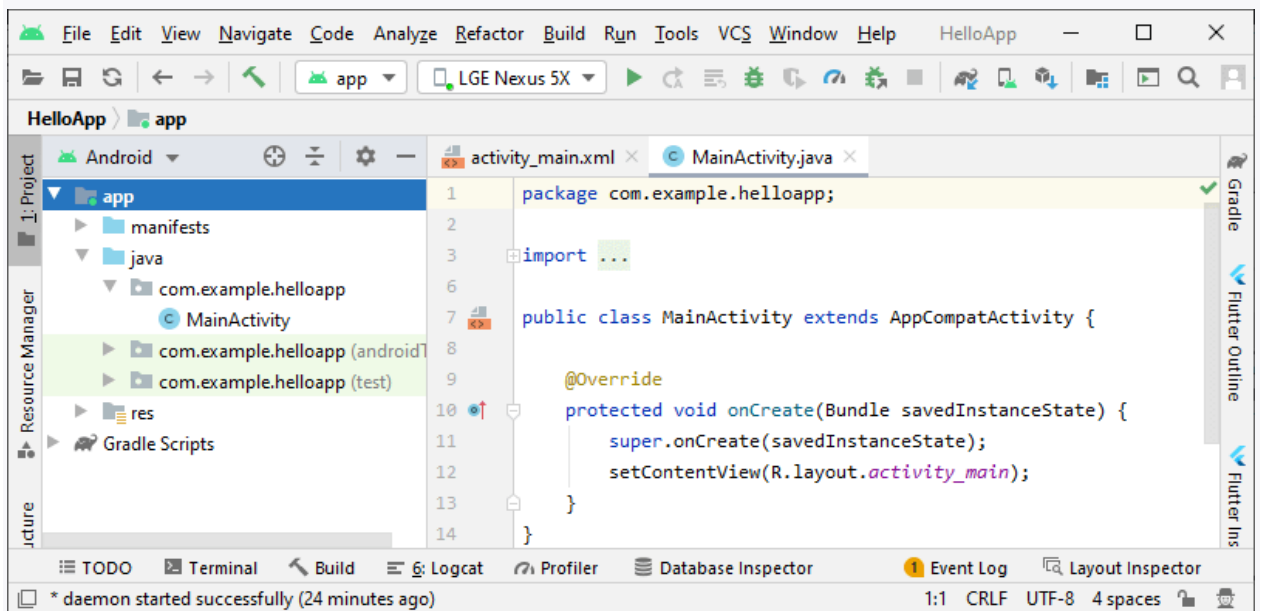
В окне создания нового проекта мы можем установить его начальные настройки:

- В поле **Name** вводится название приложения. Укажем в качестве имени название **HelloApp**
- В поле **Package Name** указывается имя пакета, где будет размещаться главный класс приложения. В данном случае для тестовых проектов это значение не играет большого значения, поэтому установим **com.example.helloapp**.
- В поле **Save Location** устанавливается расположение файлов проекта на жестком диске. Можно оставить значение по умолчанию.

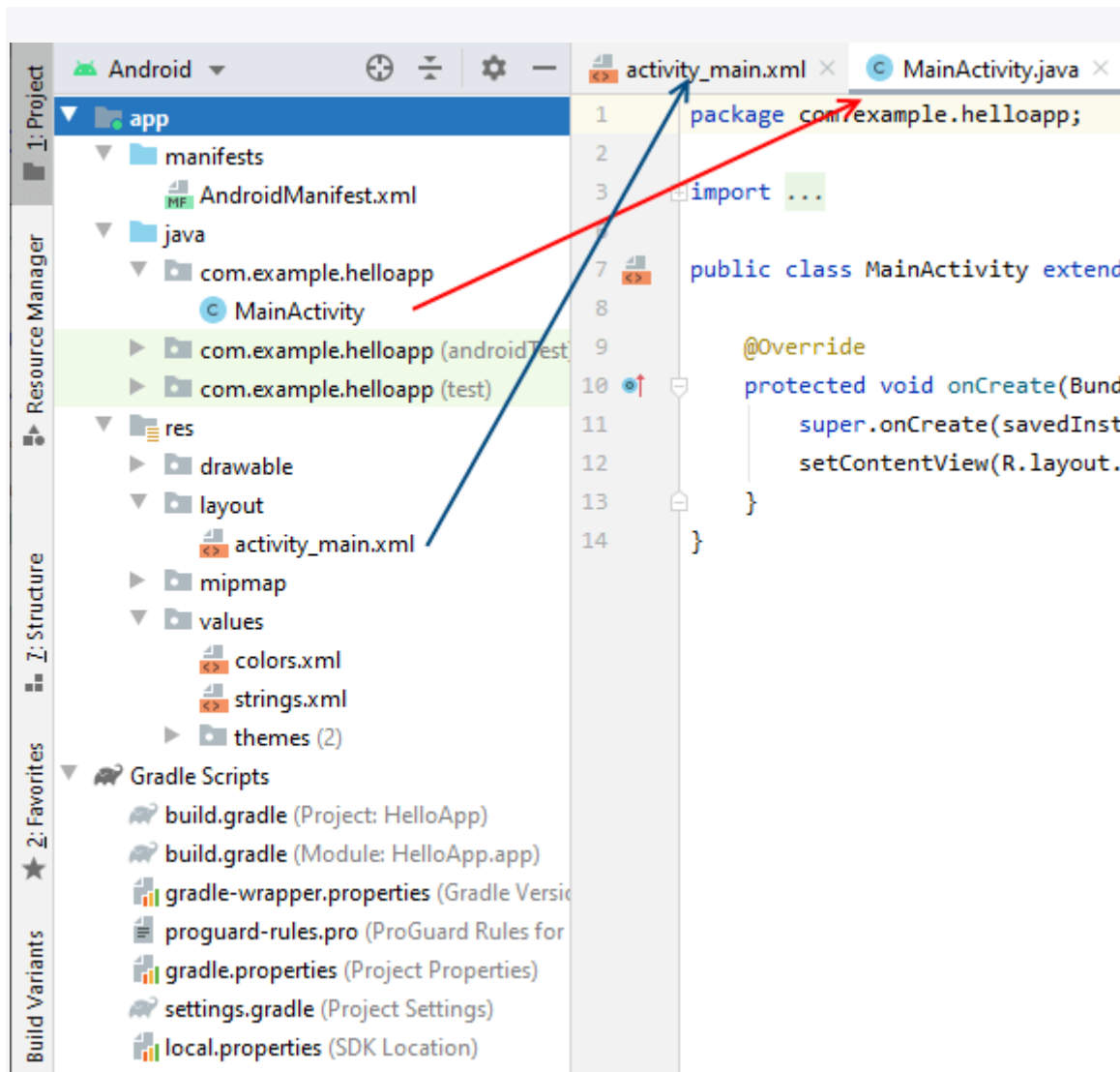
- В поле **Language** в качестве языка программирования укажем **Java** (будьте внимательны, так как по умолчанию в этом поле стоит Kotlin)
- В поле **Minimum SDK** указывается самая минимальная поддерживаемая версия SDK. Оставим значение по умолчанию - **API 21: Android 5.0 (Lollipop)**, которая означает, что наше приложение можно будет запустить начиная с Android 5.0, а это 94% устройств. На более старых устройствах запустить будет нельзя.

Стоит учитывать, что чем выше версия SDK, тем меньше диапазон поддерживаемых устройств.

Далее нажмем на кнопку Finish, и Android Studio создаст новый проект:



Вначале вкратце рассмотрим структуру проекта, что он уже имеет по умолчанию



Проект Android может состоять из различных модулей. По умолчанию, когда мы создаем проект, создается один модуль - **app**. Модуль имеет три подпапки

- **manifests**: хранит файл манифеста **AndroidManifest.xml**, который описывает конфигурацию приложения и определяет каждый из компонентов данного приложения.
- **java**: хранит файлы кода на языке java, которые структурированы по отдельным пакетам. Так, в папке **com.example.helloapp** (название которого было указано на этапе создания проекта) имеется по умолчанию файл **MainActivity.java** с кодом на языке Java, который представляет класс **MainActivity**, запускаемый по умолчанию при старте приложения
- **res**: содержит используемые в приложении ресурсы. Все ресурсы разбиты на подпапки.

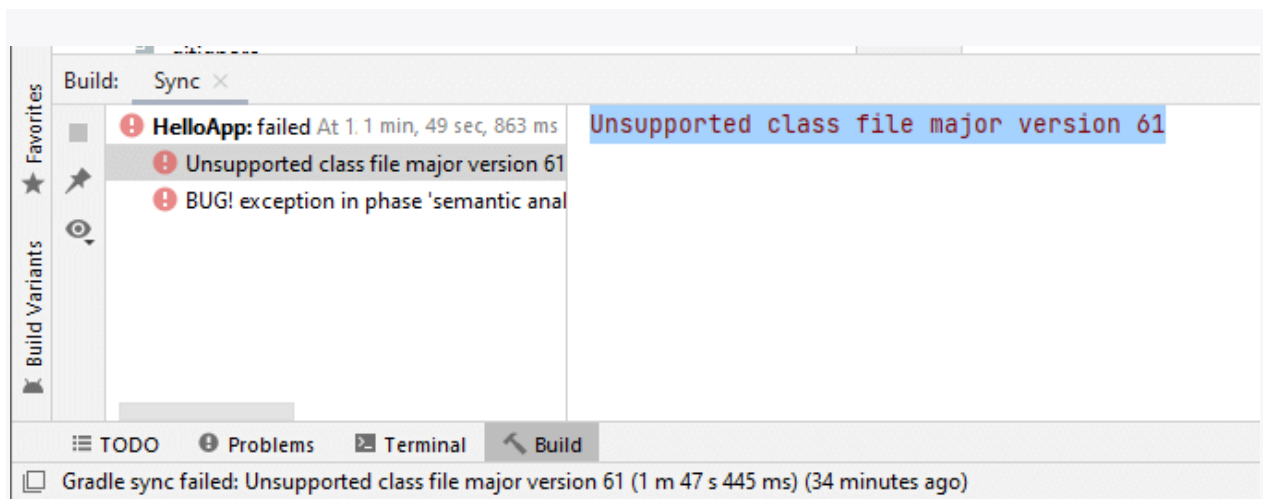
- папка **drawable** предназначена для хранения изображений, используемых в приложении
- папка **layout** предназначена для хранения файлов, определяющих графический интерфейс. По умолчанию здесь есть файл **activity\_main.xml**, который определяет интерфейс для класса MainActivity в виде xml
- папки **mipmap** содержат файлы изображений, которые предназначены для создания иконки приложения при различных разрешениях экрана.
- папка **values** хранит различные xml-файлы, содержащие коллекции ресурсов - различных данных, которые применяются в приложении. По умолчанию здесь есть два файла и одна папка:
  - файл **colors.xml** хранит описание цветов, используемых в приложении
  - файл **strings.xml** содержит строковые ресурсы, используемые в приложении
  - папки **themes** хранит две темы приложения - для светлую (дневную) и темную (ночную)

Отдельный элемент **Gradle Scripts** содержит ряд скриптов, которые используются при построении приложения.

Во всей этой структуре следует выделить файл MainActivity.java, который открыт в Android Studio и который содержит логику приложения и собственно с него начинается выполнение приложения. И также выделим файл **activity\_main.xml**, который определяет графический интерфейс - по сути то, что увидит пользователь на своем смартфоне после загрузки приложения.

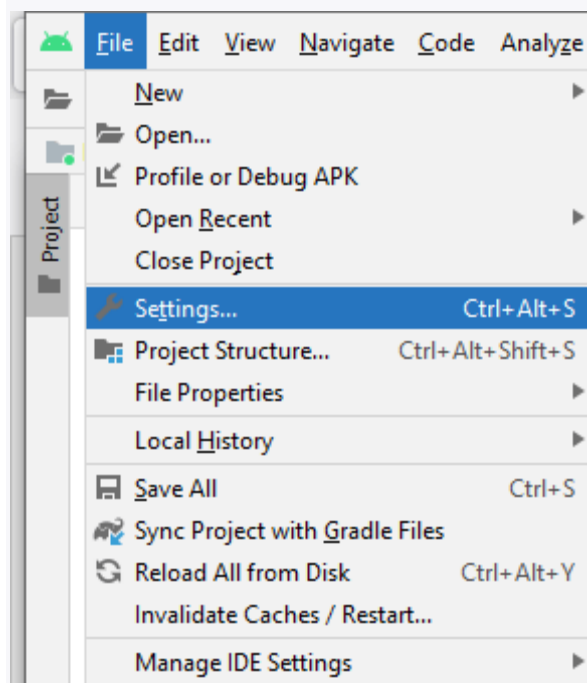
### Возможные проблемы

Для создания приложения используется Java. А для построения приложения применяется инфраструктура Gradle. Однако текущая используемая версия Gradle может быть несовместима с выбранной по умолчанию версией JDK. И в этом случае Android Studio может отображать ошибки, например, ошибку **Unsupported class file major version 61**:



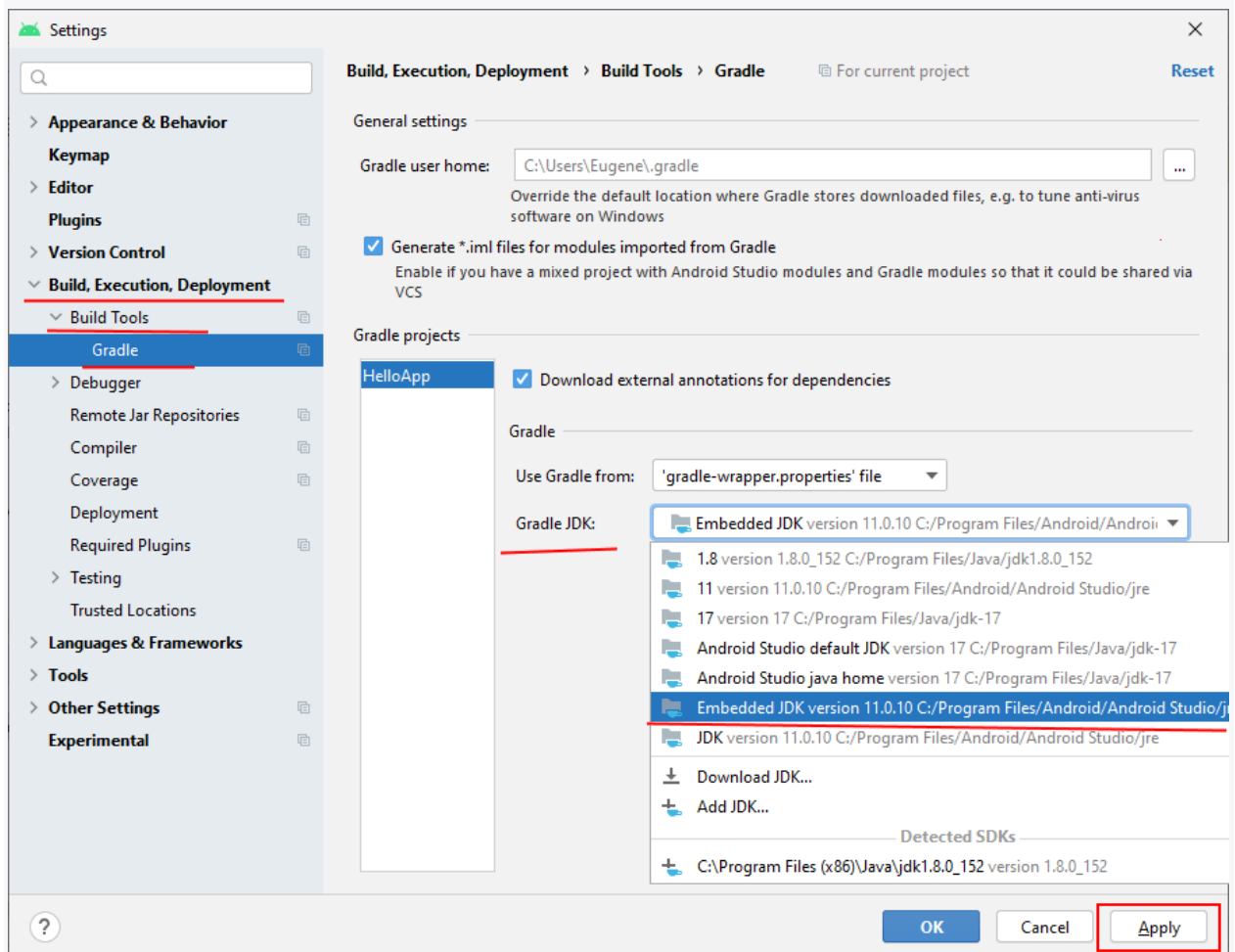
Эта ошибка говорит о том, что версия JDK 17 несовместима с текущей версией Gradle. И надо использовать меньшую версию.

Для решения этой проблемы перейдем в студии к меню **File ->Settings** (на MacOS это пункт **Android Studio -> Preferences**)



Затем в открывшемся окне настроек перейдем к пункту меню **Build, Execution, Deployment -> Build Tools -> Gradle** и далее найдем поле **Gradle**

**JDK**, где изменим версию JDK. Она должна иметь версию 11 и выше. Как правило, вместе с Android Studio устанавливается и поддерживаемая версия JDK - на данный момент это JDK 11. И ее можно выбрать в списке JDK:



Наиболее оптимальный пункт для выбора версий JDK, которая идет вместе с Android Studio, называется **Embedded JDK version....** Как видно на скриншоте, это версия 11, но при последующих обновлениях Android Studio эта версия может измениться.

После сделанных изменений сначала нажмем на кнопку **Apply**, а затем на кнопку **ОК**. И повторим запуск проекта.

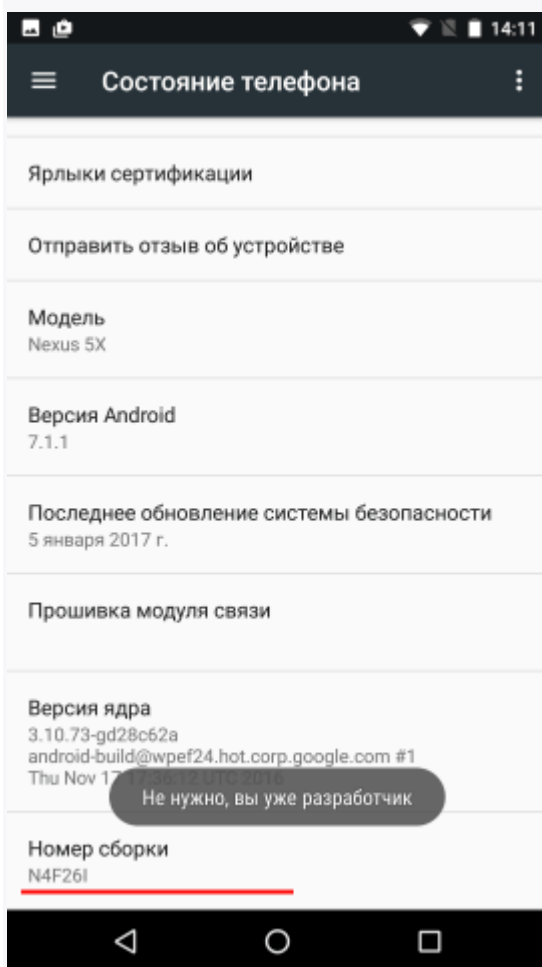
## Запуск проекта

Созданный выше проект уже содержит некоторый примитивный функционал. Правда, этот функционал почти ничего не делает, только выводит на экран строку "Hello world!". Тем не менее это уже фактически приложение, которое мы можем запустить.

Для запуска и тестирования приложения мы можем использовать эмуляторы или реальные устройства. Но в идеале лучше тестировать на реальных устройствах. К тому же эмуляторы требуют больших аппаратных ресурсов, и не каждый компьютер может потянуть требования эмуляторов. А для использования мобильного устройства для тестирования может потребоваться разве что установить необходимый драйвер.

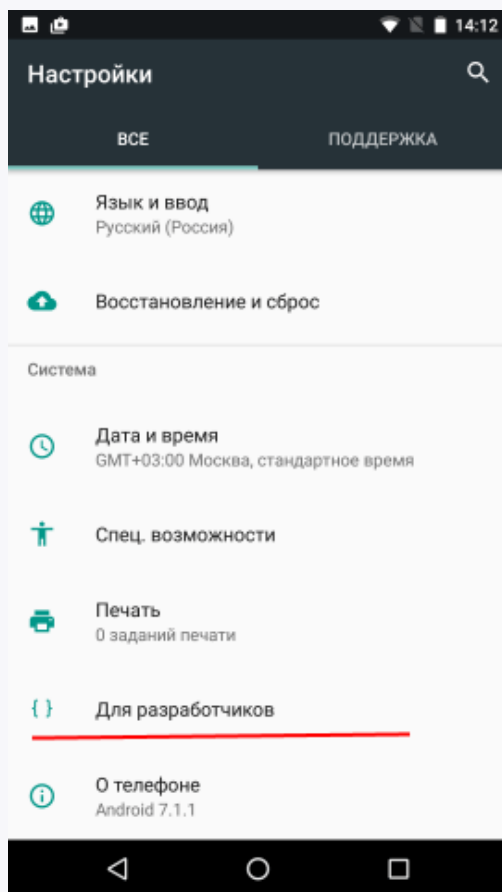
## Режим разработчика на телефоне

По умолчанию опции разработчика на смартфонах скрыты. Чтобы сделать их доступными, надо зайти в **Settings > About phone (Настройки > О телефоне)** (в Android 8 это в **Settings > System > About phone (Настройки > Система > О телефоне)**) и семь раз нажать **Build Number (Номер сборки)**.

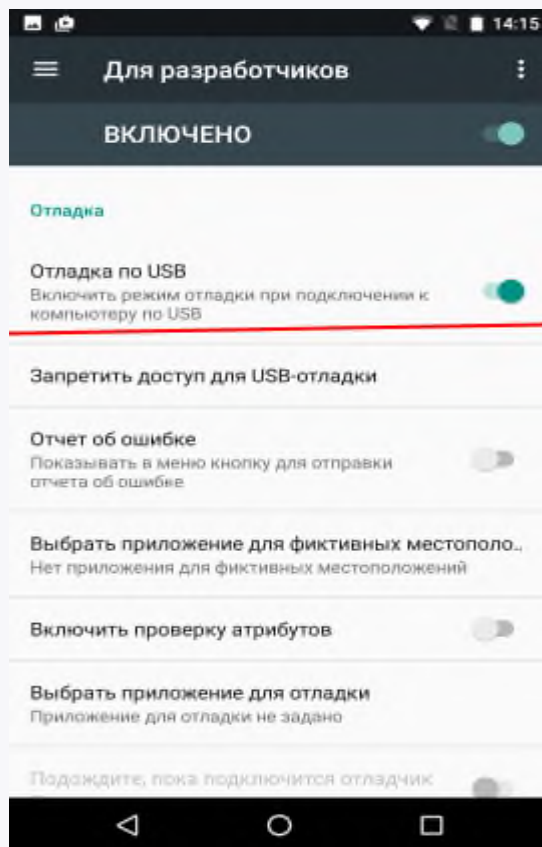


Теперь необходимо включить отладку по USB. Для этого перейдем в **Settings > System > Advanced > Developer options** или **Настройки > Система > Дополнительно > Для разработчиков** (в Android 8 это в **Settings > System > Developer options** или **Настройки > Система > Для разработчиков**).



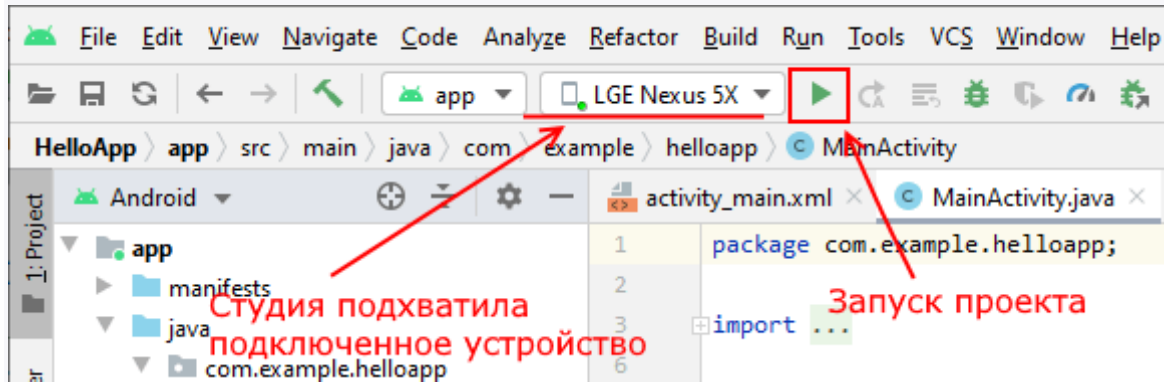


И включим возможность отладки по USB:



## Запуск приложения

Подключим устройство с ОС Android (если мы тестируем на реальном устройстве) и запустим проект, нажав на зеленую стрелочку на панели инструментов.



Выберем устройство и нажмем на кнопку ОК. И после запуска мы увидим наше приложение на экране устройства:



Hello World!

