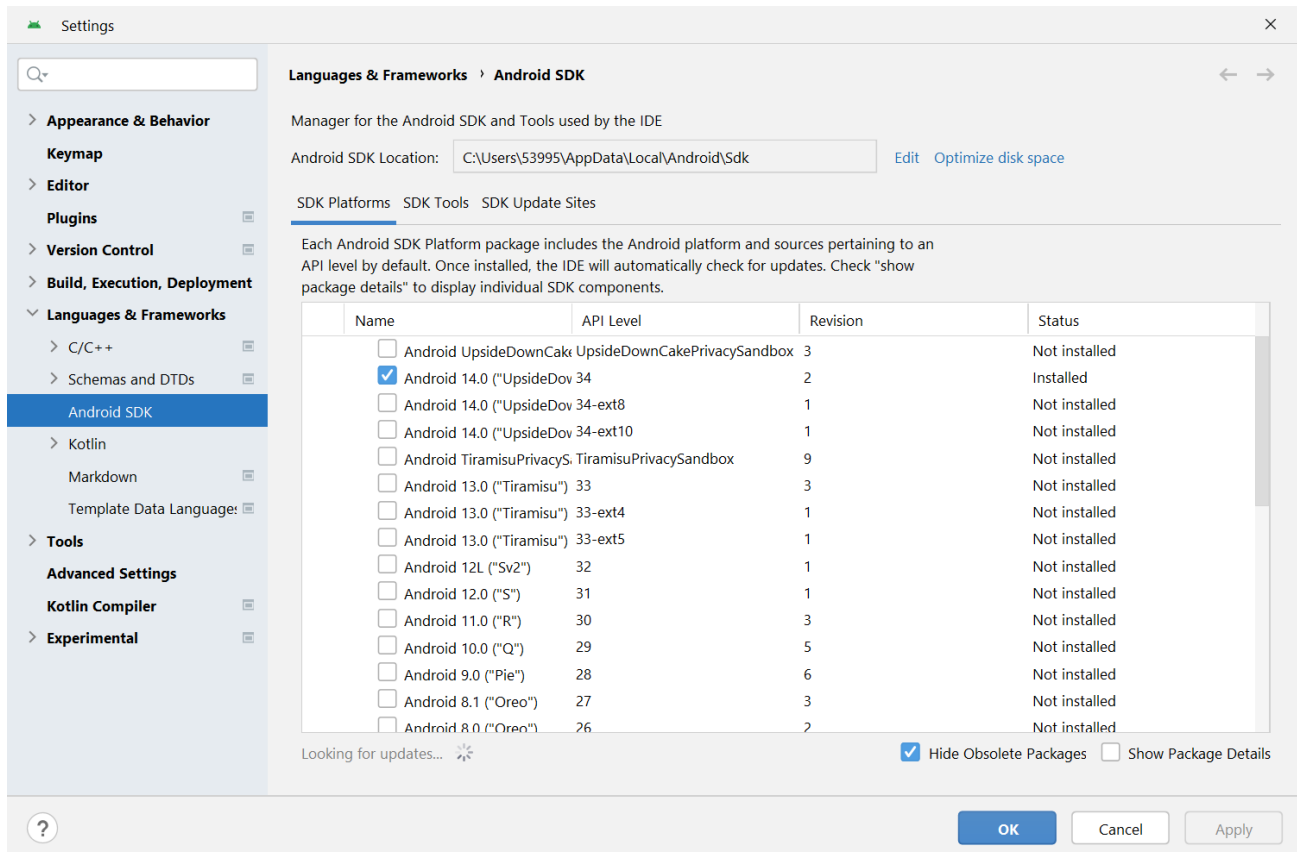


## № 1 Установка и настройка среды

### 1. Установите Android Studio

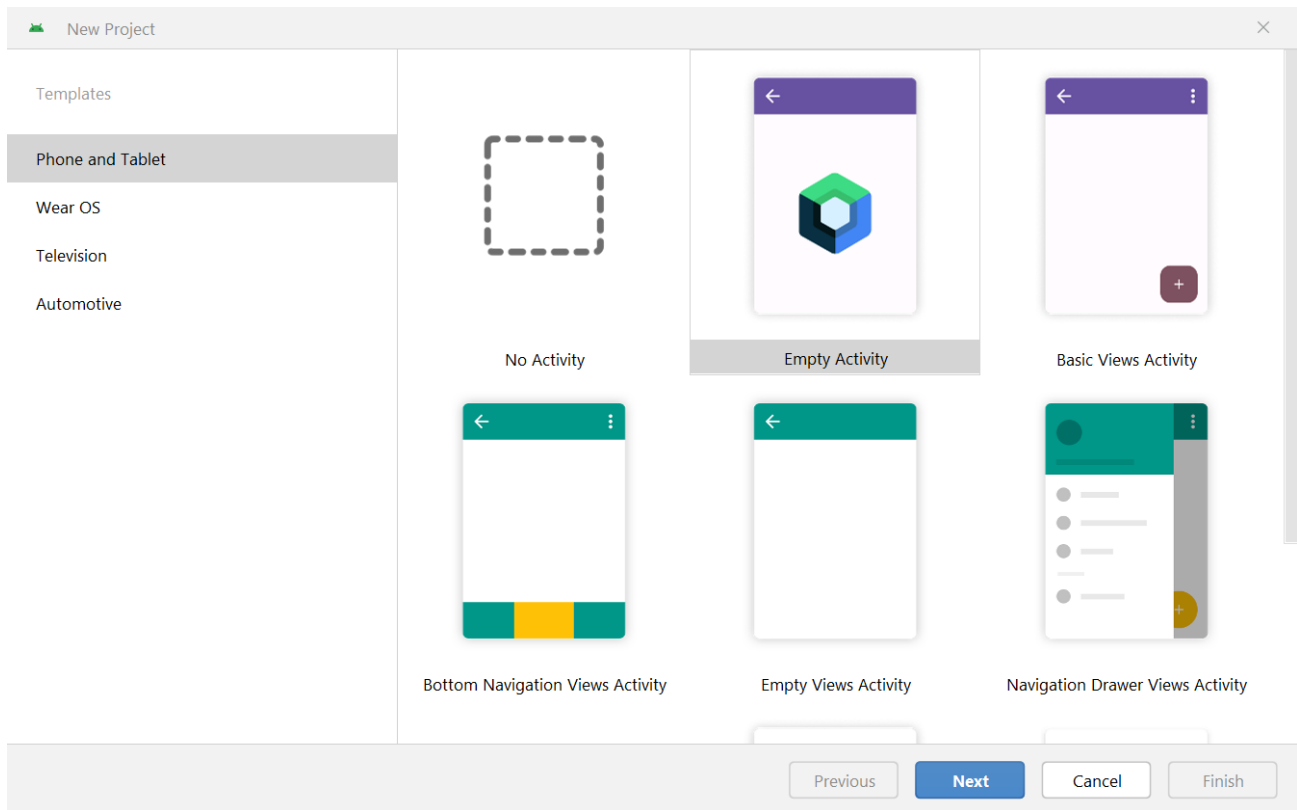
[Download Android Studio & App Tools - Android Developers](#)

2. Установите дополнительные компоненты. (Можно пропустить по умолчанию устанавливается SDK)



Компоненты любой платформы можно получить при помощи Android SDK Manager. В Android Studio выполните команду Tools\ SDK Manager. (Меню Tools отображается только при наличии открытого проекта. Если вы еще не создали проект, SDK Manager также можно вызвать с экрана Welcome; выберите вариант Configure\SDK Manager) Тут можно выбрать SDK которые понадобятся.

3. Пробное приложение. Создайте новый проект. Выбирается шаблон. Как пример из простейших на Java это **Basic Views Activity** и **Empty Views Activity**.



После выбора можно будет заполнить информацию о проекте где указывается имя проекта, имя проекта место хранения приложения, язык и минимальную версию SDK.

**Empty Views Activity**

Creates a new empty activity

Name: My Application

Package name: by.uni.myapplication

Save location: G:\AndroidLab

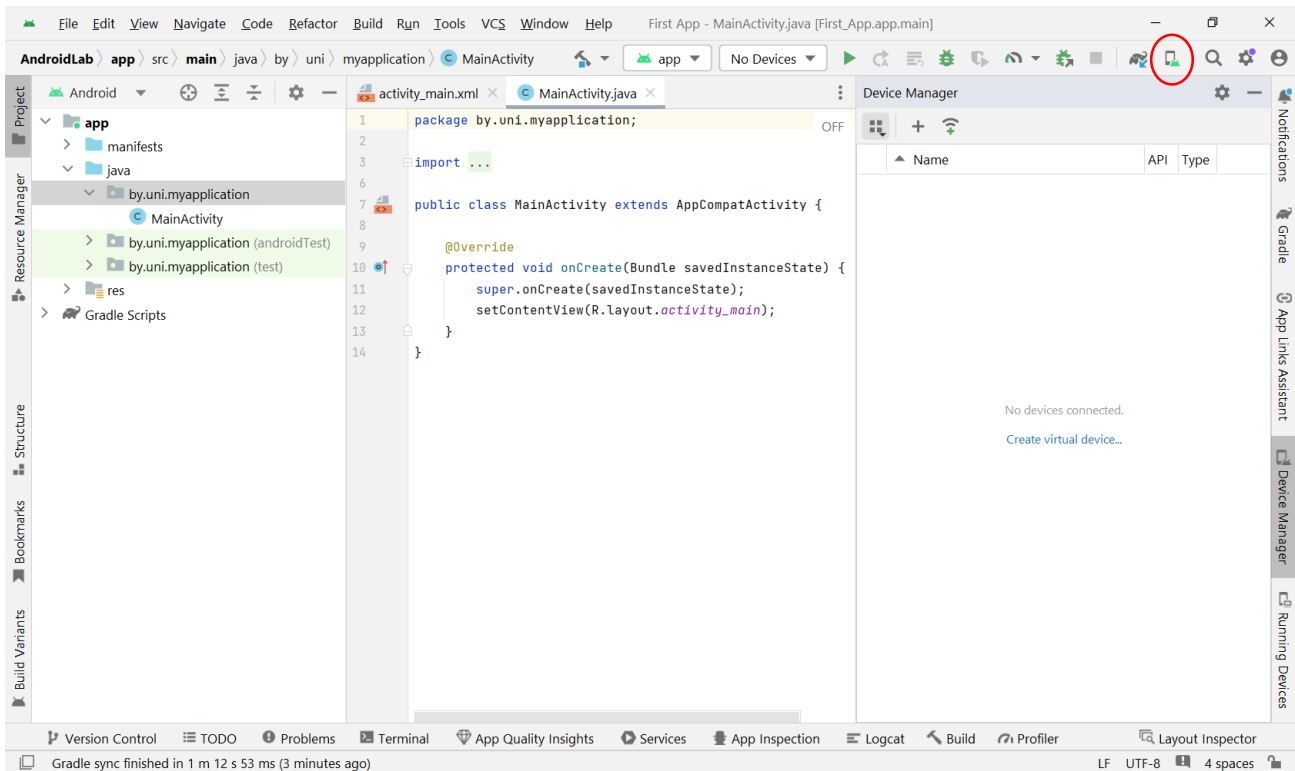
Language: Java

Minimum SDK: API 24 ("Nougat"; Android 7.0)

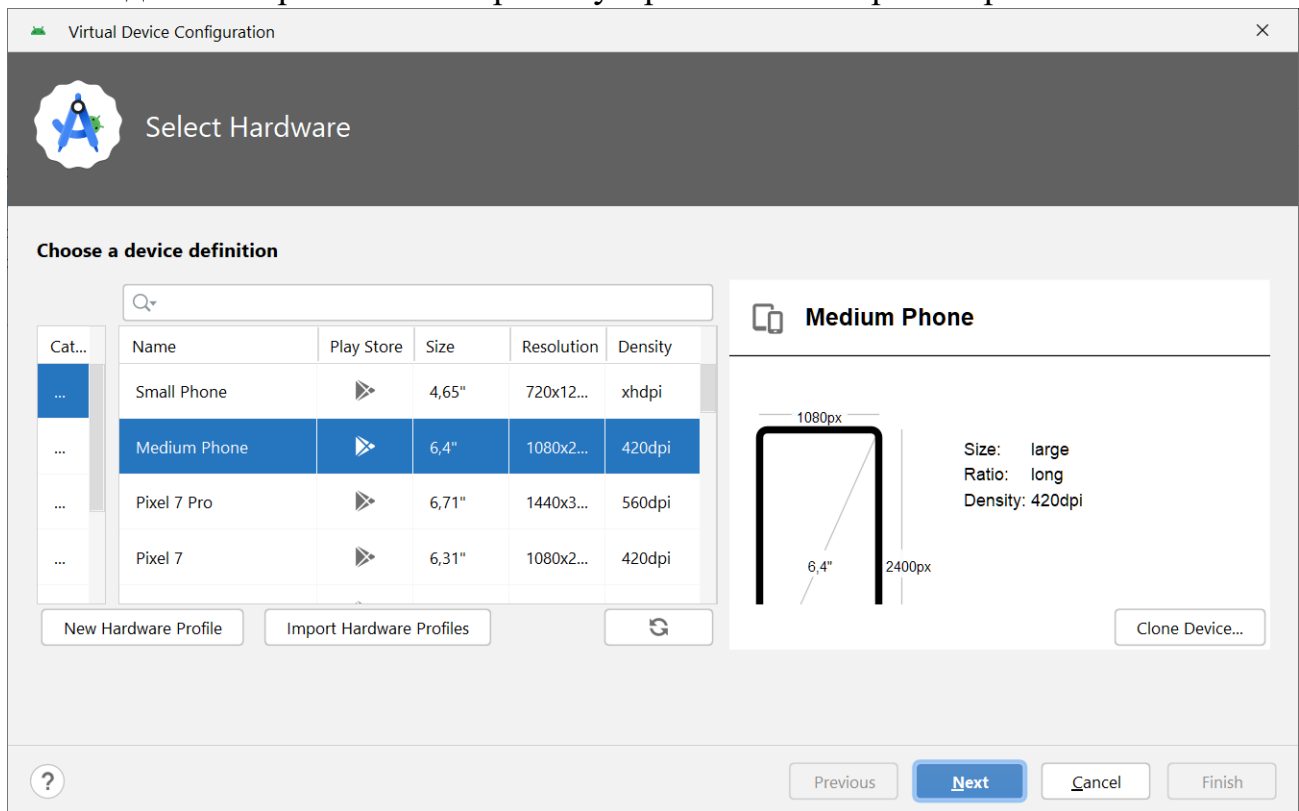
*i* Your app will run on approximately **96,3%** of devices.  
[Help me choose](#)

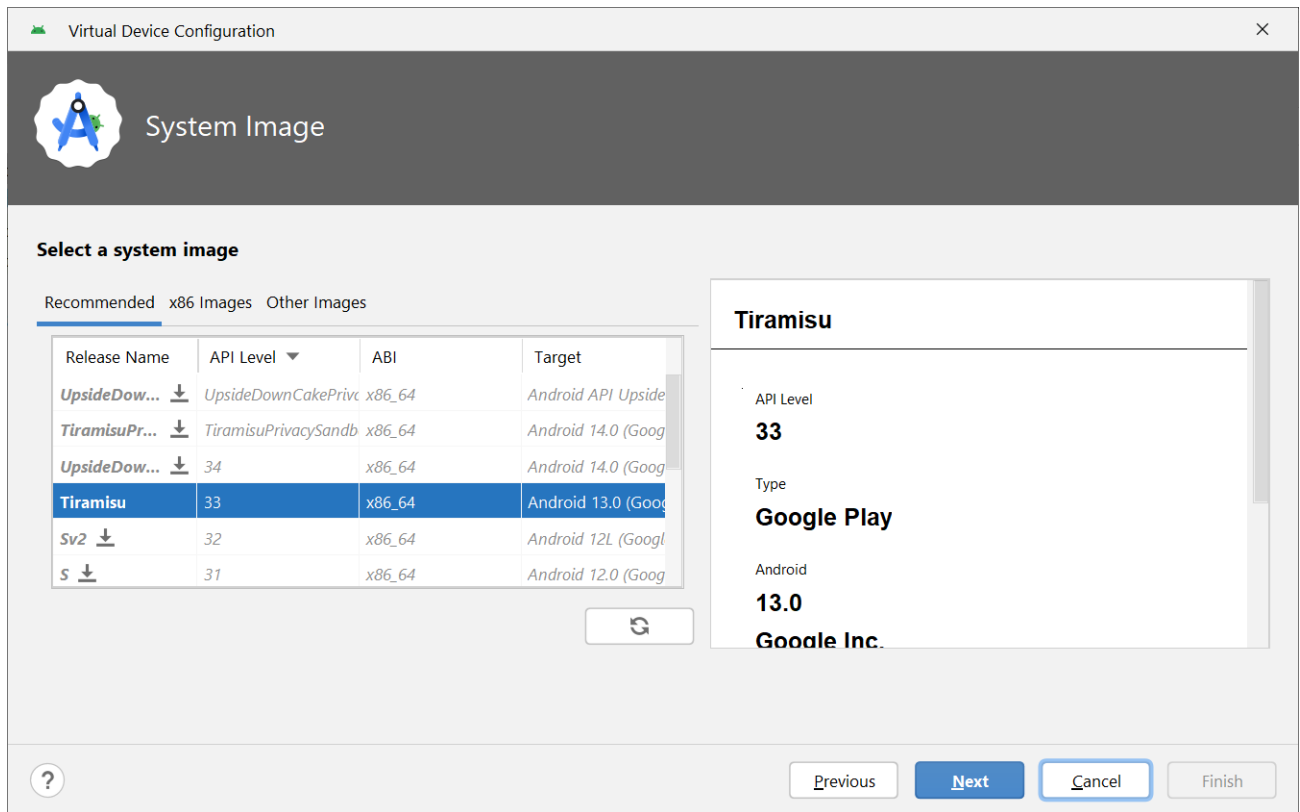
Build configuration language *?* Kotlin DSL (build.gradle.kts) [Recommended]

#### 4. Вид экрана после запуска приложения

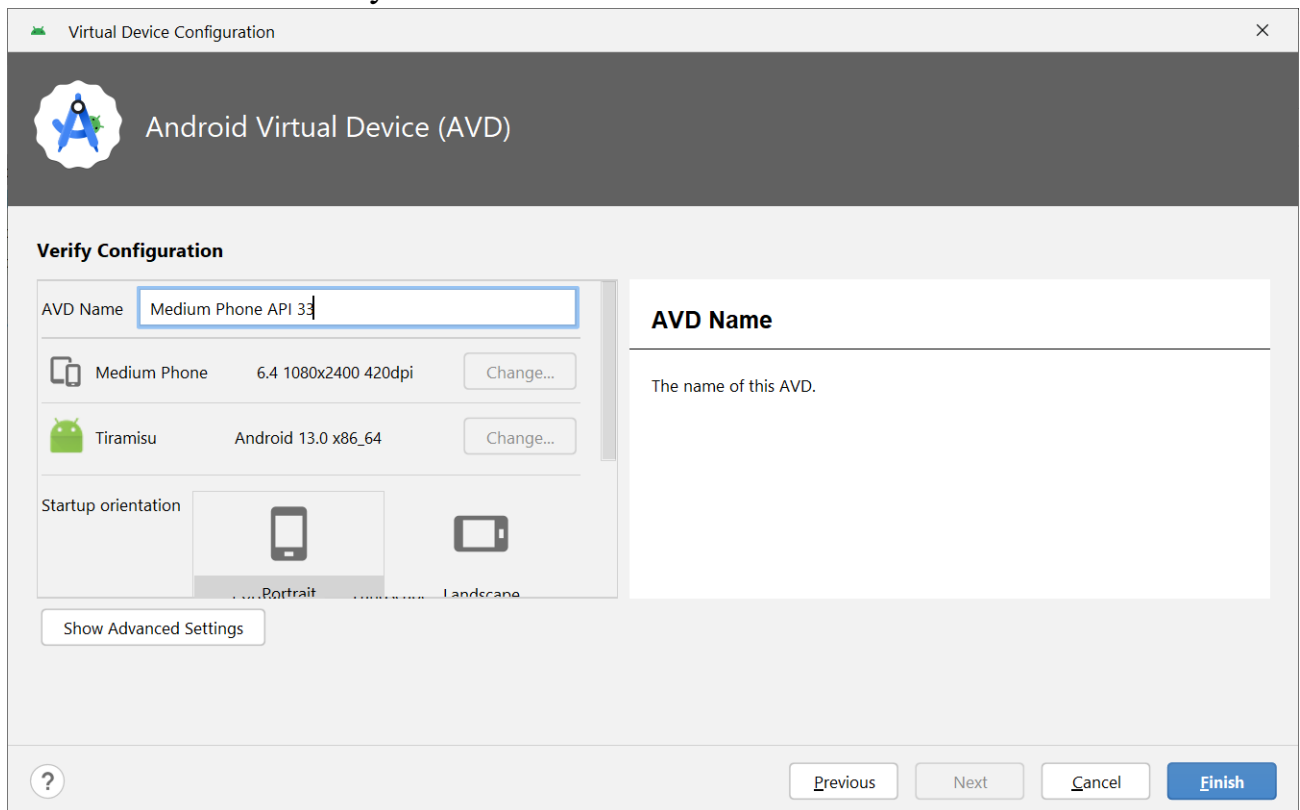


5. Так как в данный момент нет устройств для запуска или приложения необходимо либо добавить эмулятор либо подключить по USB или Wifi физическое устройство для отладки. Для добавления необходимо нажать на кнопку, отмеченную красным выше. Далее необходимо выбрать или настроить устройство и выбрать версию Android.

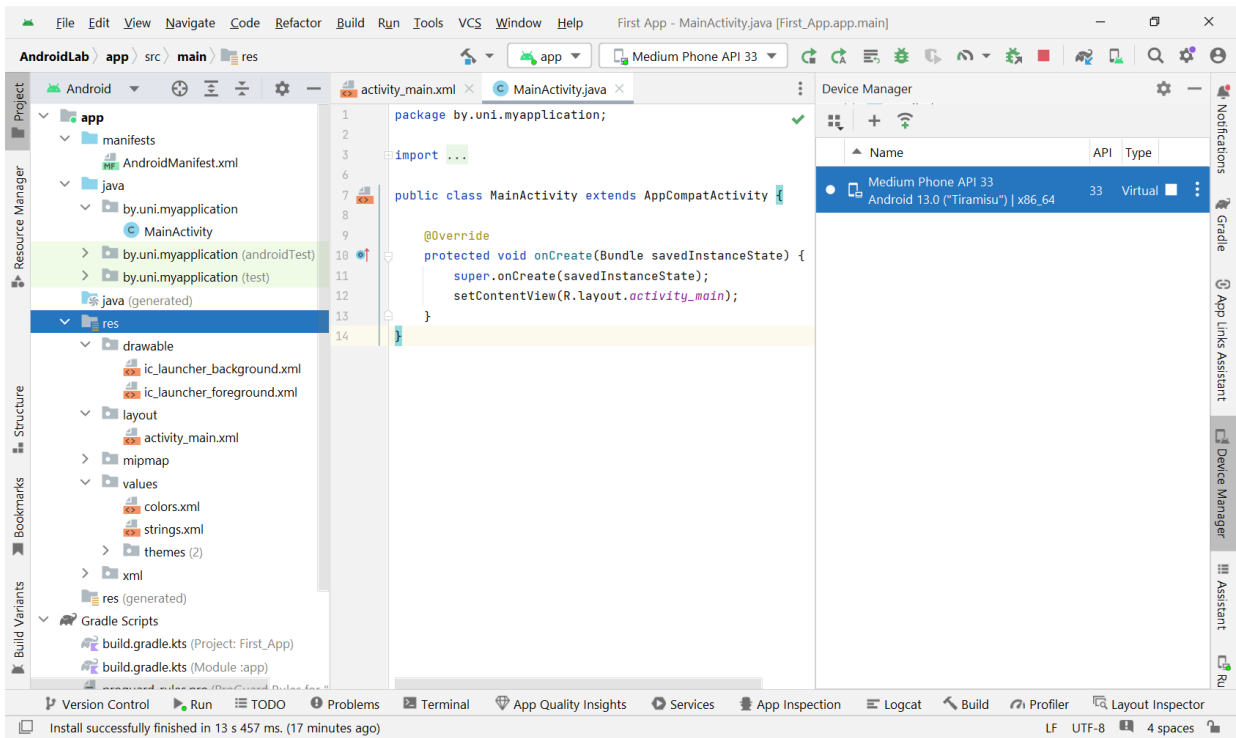




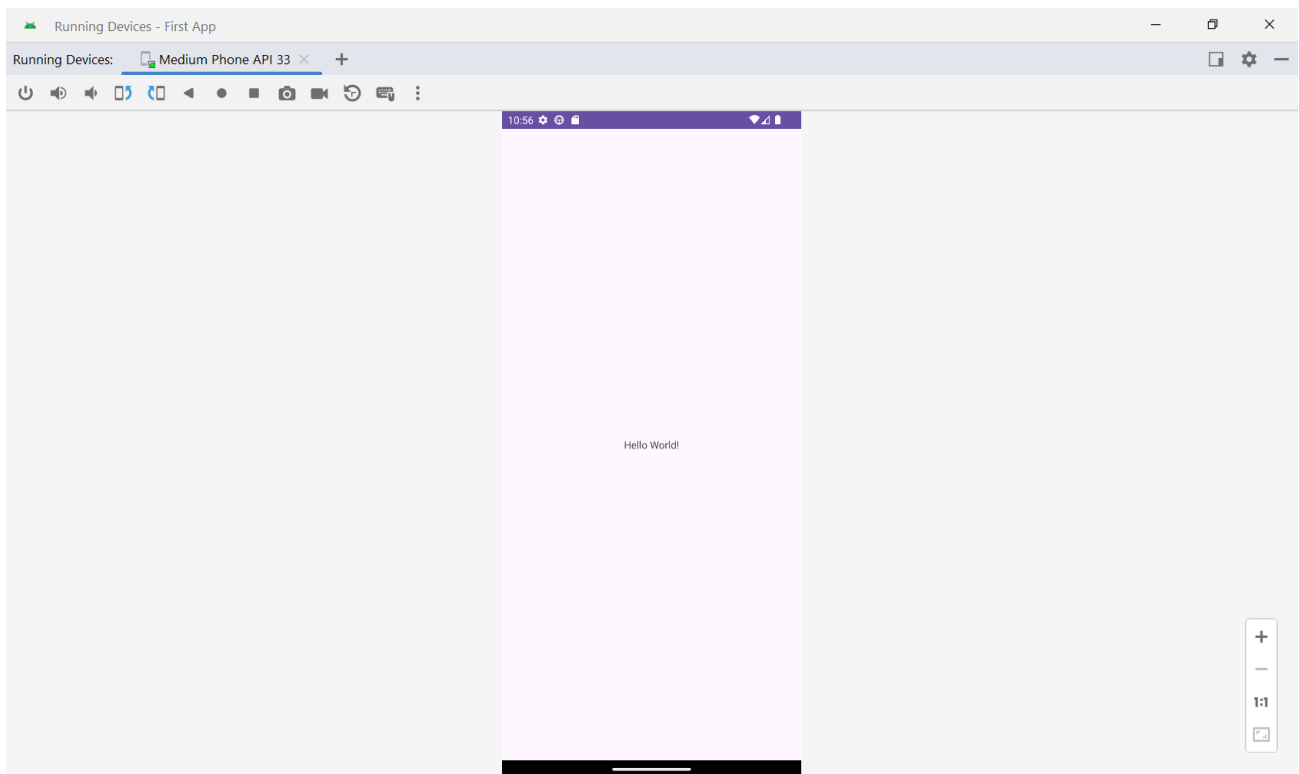
В ходе создания также будет предложено изменить имя устройства и поменять некоторые настройки в Advanced Settings. В данный случае можно просто просмотреть настройки выставленные по умолчанию.



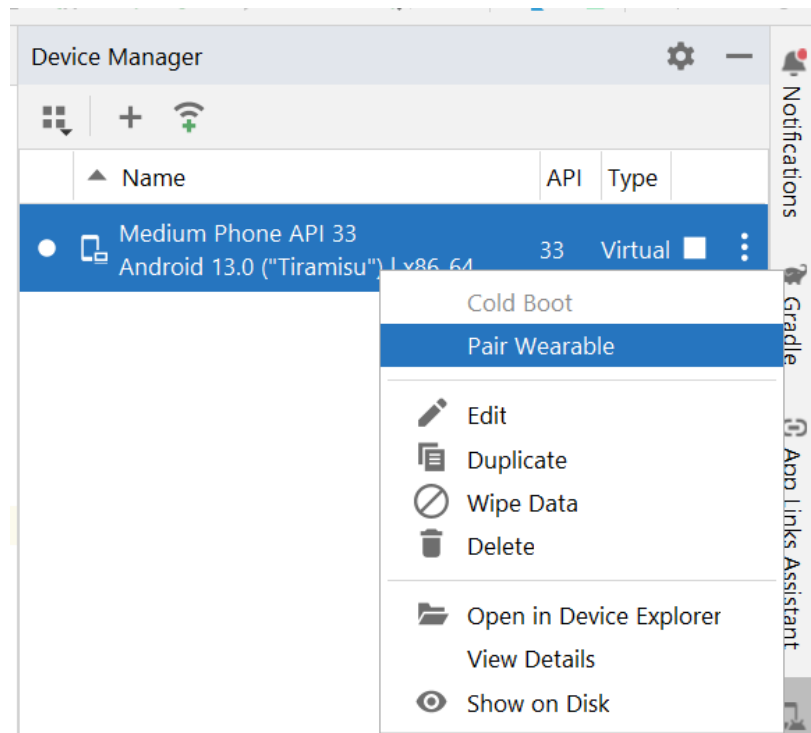
6. После добавления можно будет увидеть устройство в Device Manager



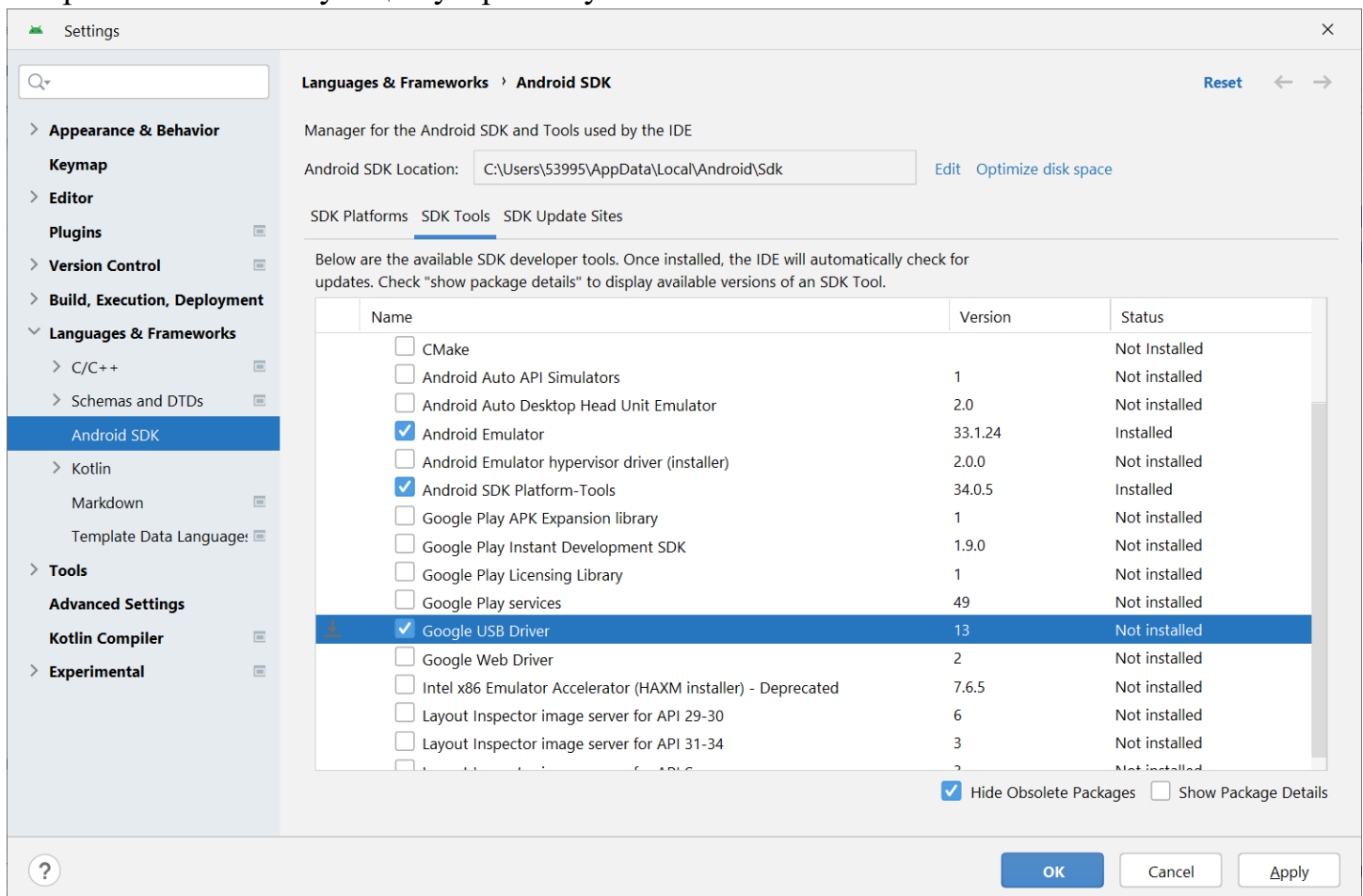
7. Следующим шагом можно считать запуск на эмуляторе приложения по умолчанию.



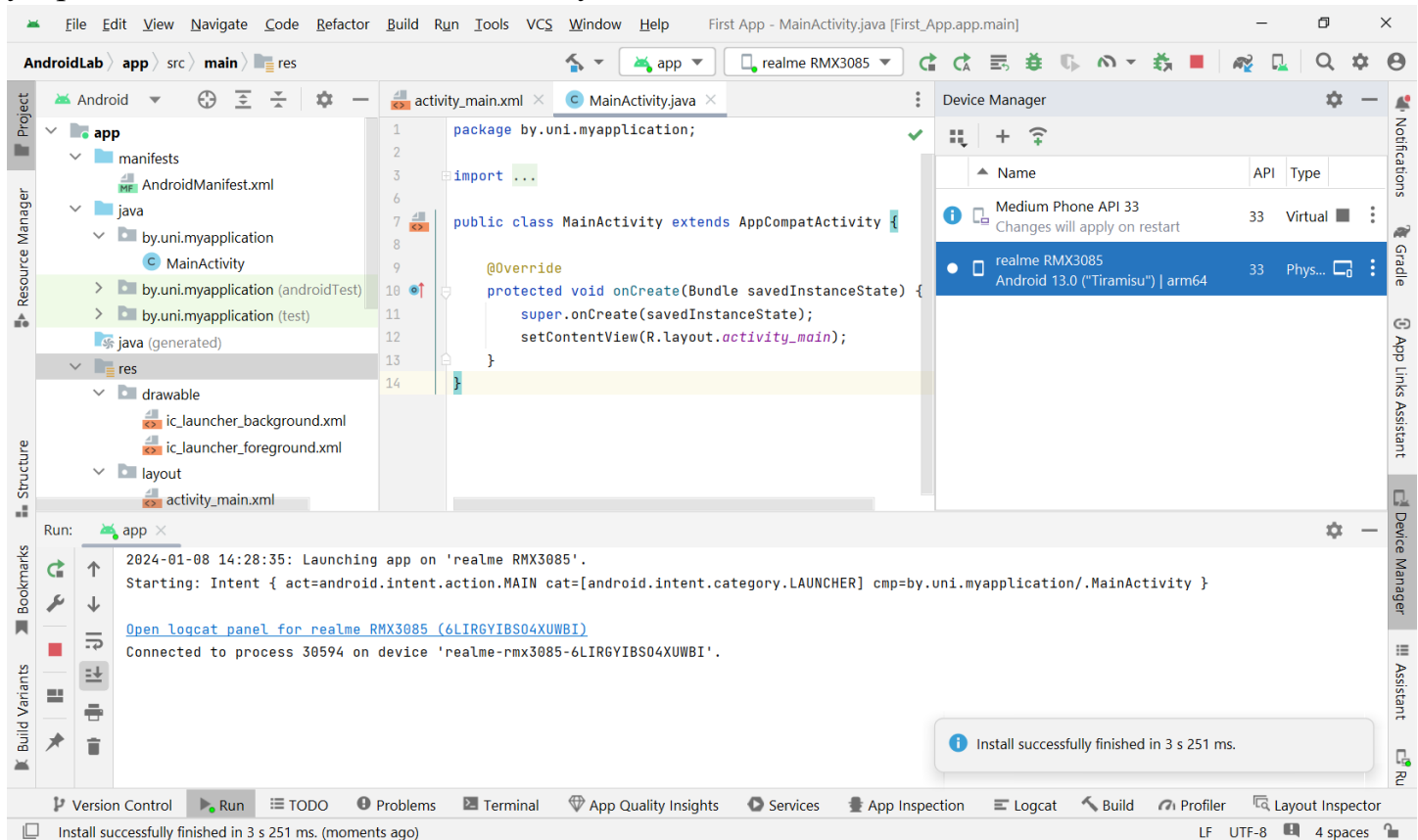
8. Кроме того, с помощью меню в Device Manager можно изменить протубликовать и очистить сохраненные данные устройства.



9. Кроме того, как уже было указано возможно отлаживать приложения на физическом устройстве для этого предварительно необходимо установить Google USB Driver в Sdk tools находящимся в SDK Manager. Кроме того необходимо установить SDK с версией соответствующей устройству.

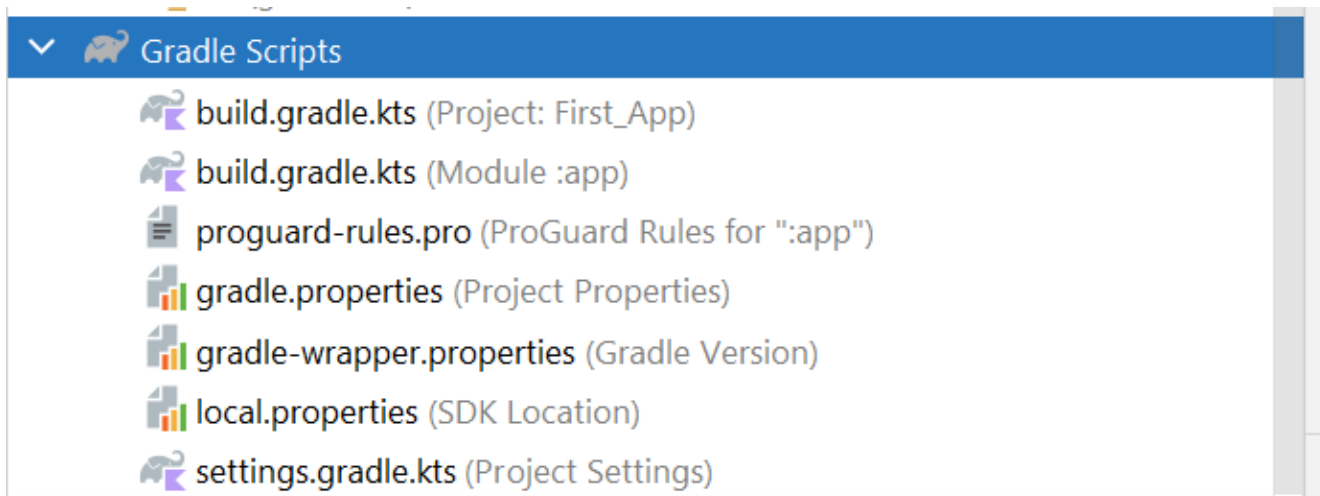


10. Также кроме этого необходимо активировать режим разработчика на устройстве и в нём включить отладку по Usb или Wifi.



11. **Gradle** процесс построения и зависимости. Gradle – инструмент автоматической сборки проектов Android

Откройте Gardle scripts и local.properties. Там прописано место установкиAndroid SDKы



Перейдите в gradle.properties - он содержит свойства процесса построения. Перейдите в proguard-rules.pro -там содержатся правила обфускации и оптимизации проекта. Обычно они закомментированы. build.gradle - в нем содержится информация о версии SDK, используемой для компиляции, минимальной версии, версии приложения и тп.

## 12. Вид файла build.gradle

```
1 plugins { this: PluginDependenciesSpecScope
2     id("com.android.application")
3 }
4
5 android { this: BaseAppModuleExtension
6     namespace = "by.uni.myapplication"
7     compileSdk = 34
8
9     defaultConfig { this: ApplicationDefaultConfig
10         applicationId = "by.uni.myapplication"
11         minSdk = 24
12         targetSdk = 34
13         versionCode = 1
14         versionName = "1.0"
15
16         testInstrumentationRunner = "androidx.test.runner.AndroidJUnitRunner"
17     }
18
19     buildTypes { this: NamedDomainObjectContainer<ApplicationBuildType>
20         release { this: ApplicationBuildType
21             isMinifyEnabled = false
22             proguardFiles(getDefaultProguardFile( name: "proguard-android-optimize.txt"), "proguard-rules.pro")
23         }
24     }
25
26     compileOptions { this: CompileOptions
27         sourceCompatibility = JavaVersion.VERSION_1_8
28         targetCompatibility = JavaVersion.VERSION_1_8
29     }
30 }
```

13. Еще один раздел – зависимости В нем содержится информация о включенных в libs jar файлов. При включении новой зависимости он а будет автоматически здесь прописана

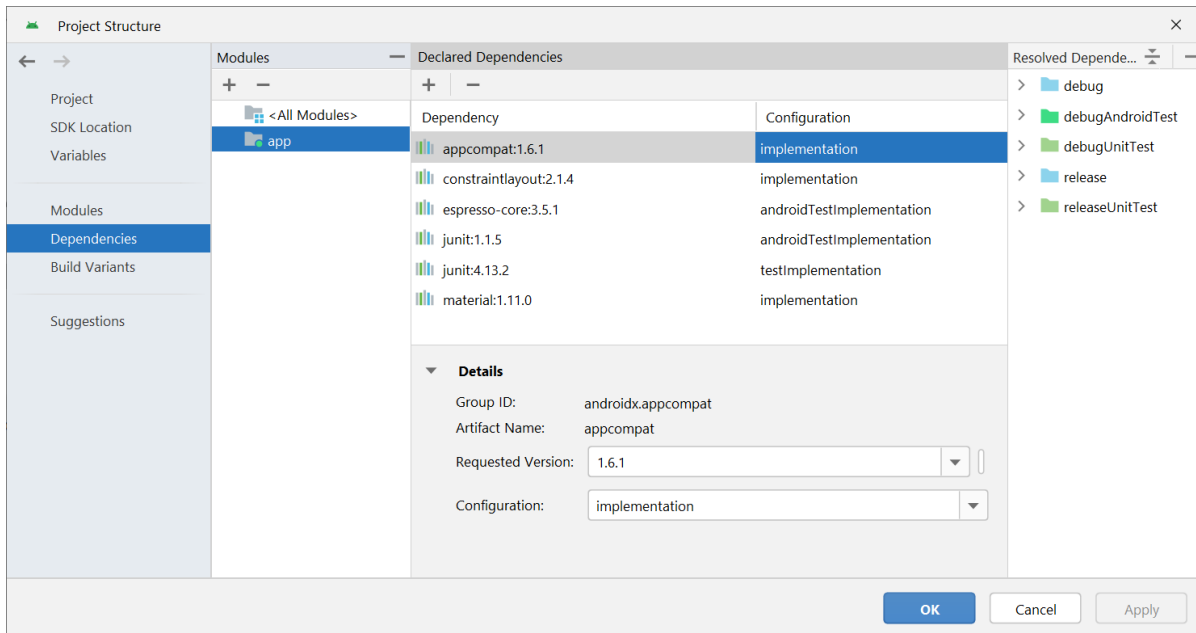
```
27
28
29
30
31 dependencies { this: DependencyHandlerScope
32
33     implementation("androidx.appcompat:appcompat:1.6.1")
34     implementation("com.google.android.material:material:1.11.0")
35     implementation("androidx.constraintlayout:constraintlayout:2.1.4")
36     testImplementation("junit:junit:4.13.2")
37     androidTestImplementation("androidx.test.ext:junit:1.1.5")
38     androidTestImplementation("androidx.test.espresso:espresso-core:3.5.1")
39 }
```



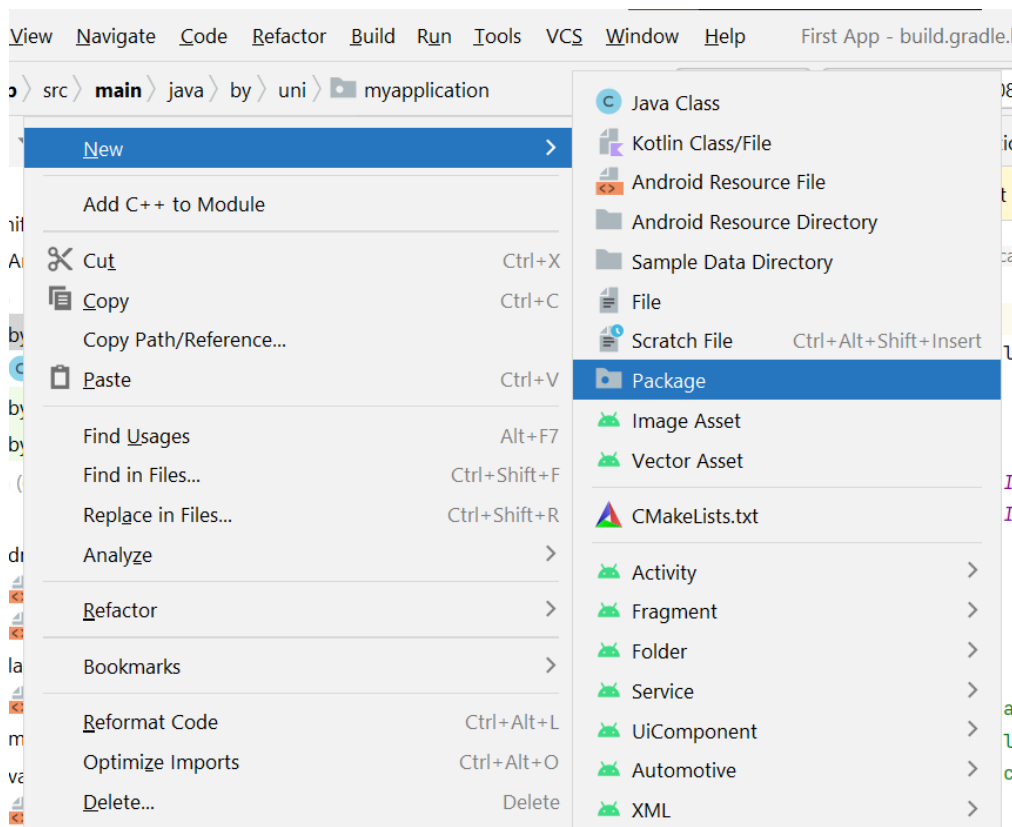
Каждый модуль имеет свой файл build.gradle, который определяет конфигурацию построения проекта, специфичную для данного модуля.

Для подключения новых зависимостей можно использовать также File-> Project Structure. И знаками + - добавлять или удалять их

### ***Вид окна Project Structure***

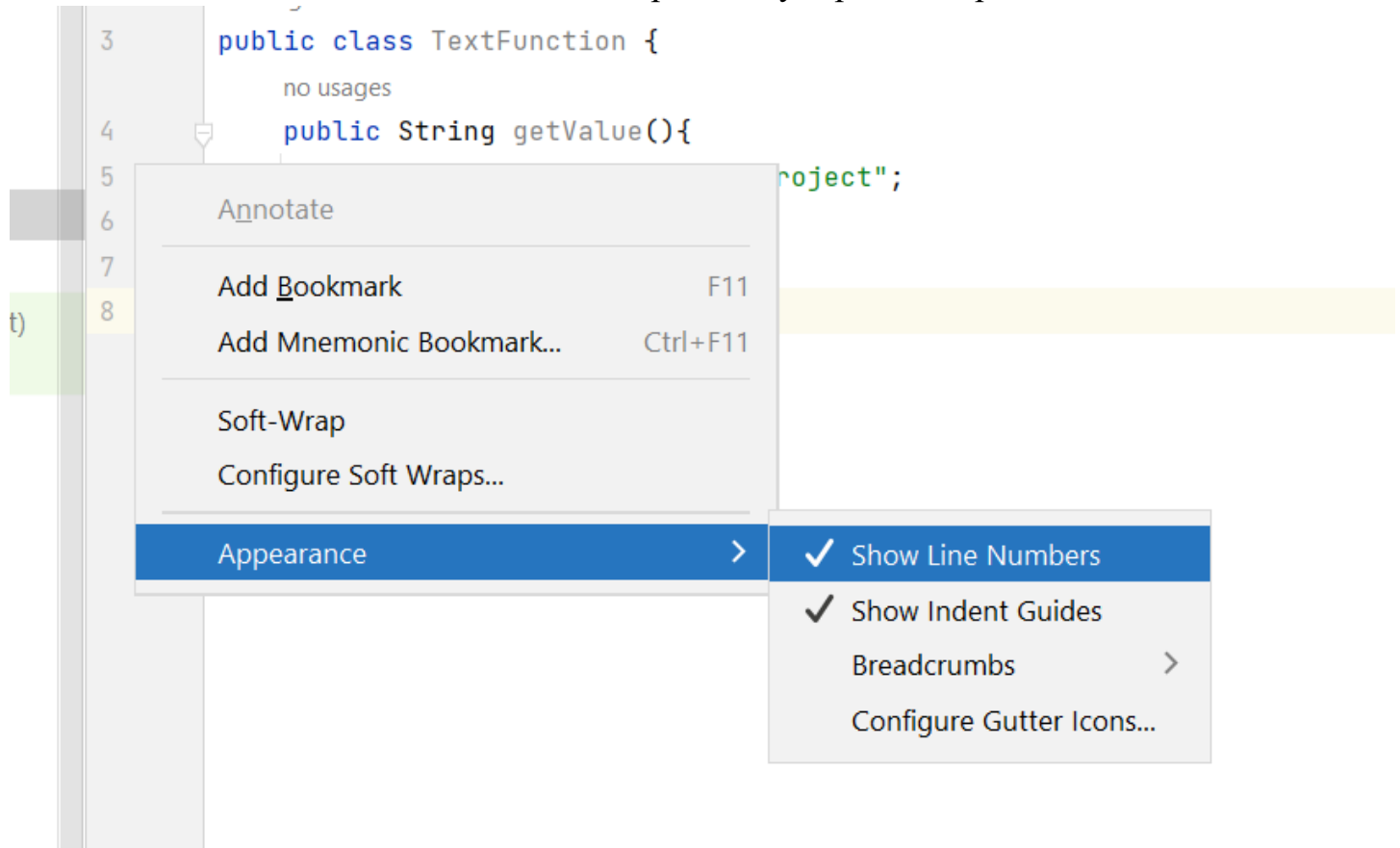


**14.** Добавление нового пакета и классов . При нажатии правой кнопкой мыши по папке с кодом (MainActivity). Следует добавить новый пакет и в нём же новый класс включающий в себя код представленный ниже.

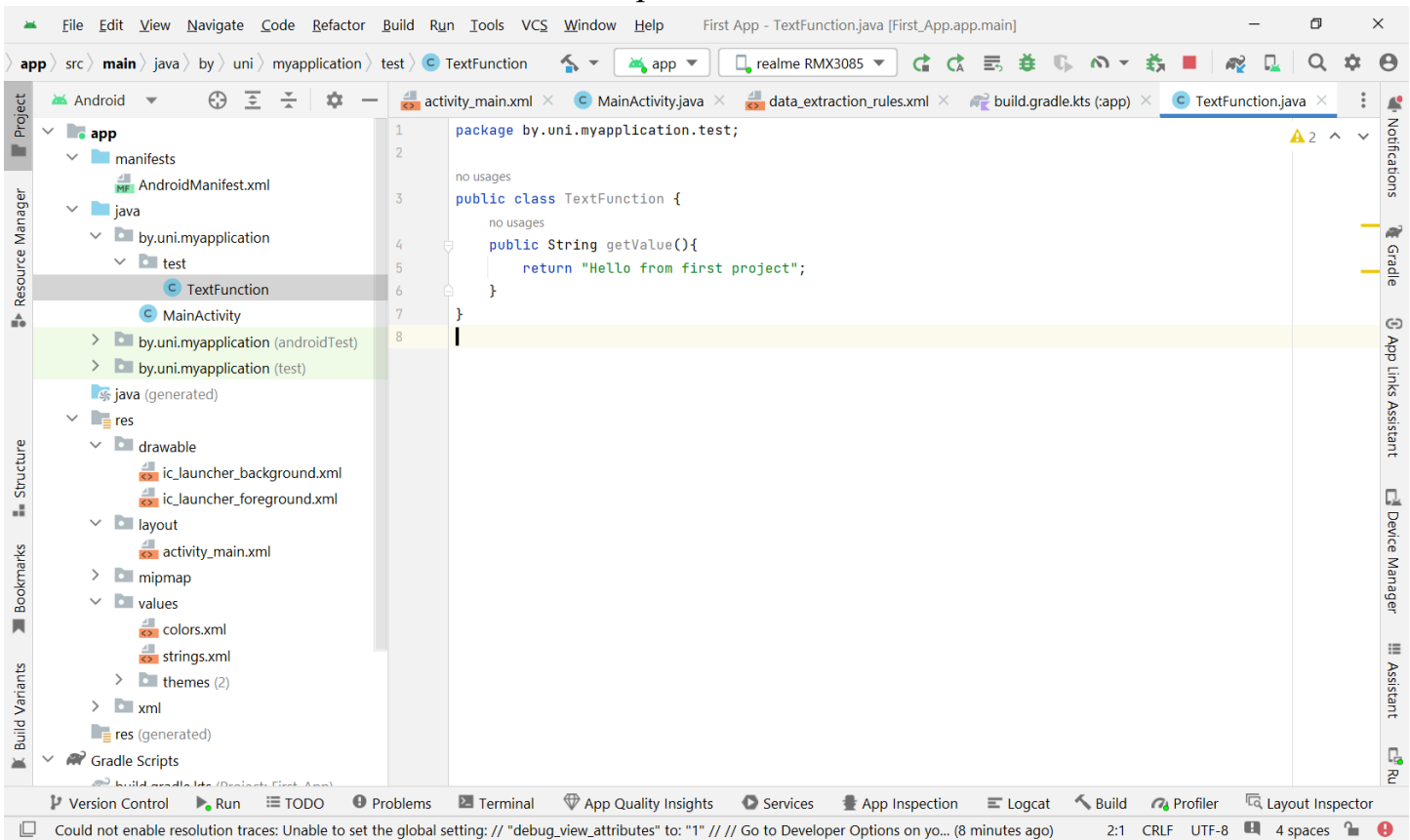


```
public class TextFunction {  
    public String getValue(){  
        return "Hello from first project1";  
    }  
}
```

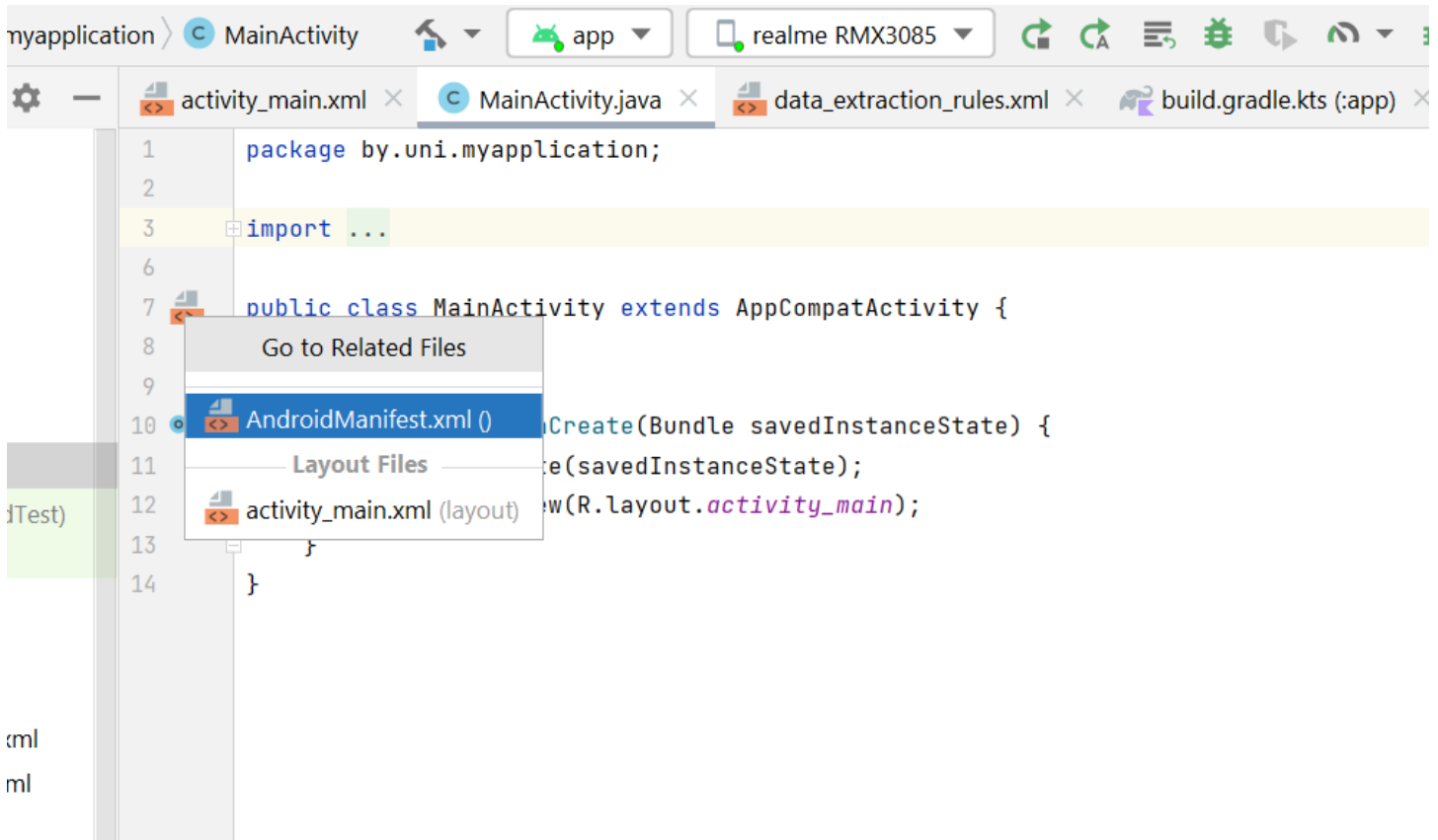
**15.** Также может быть полезно отобразить нумерацию строк если не включена.



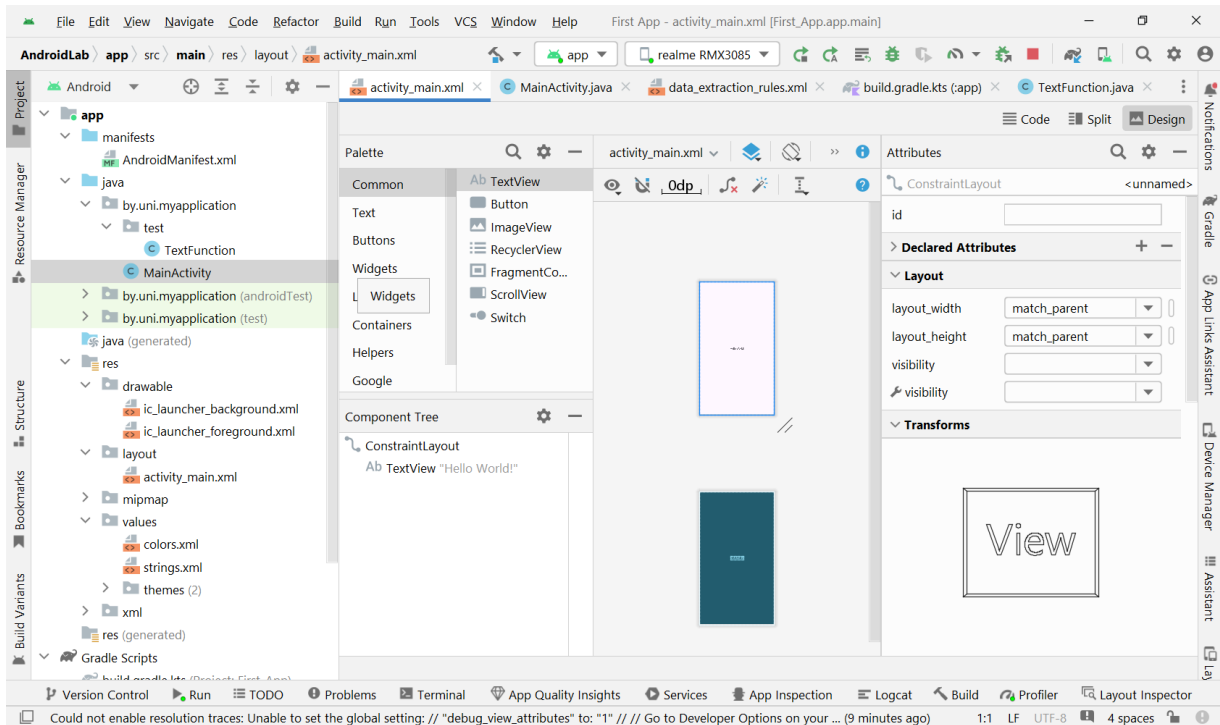
## 16. Вид после добавления кода в проект



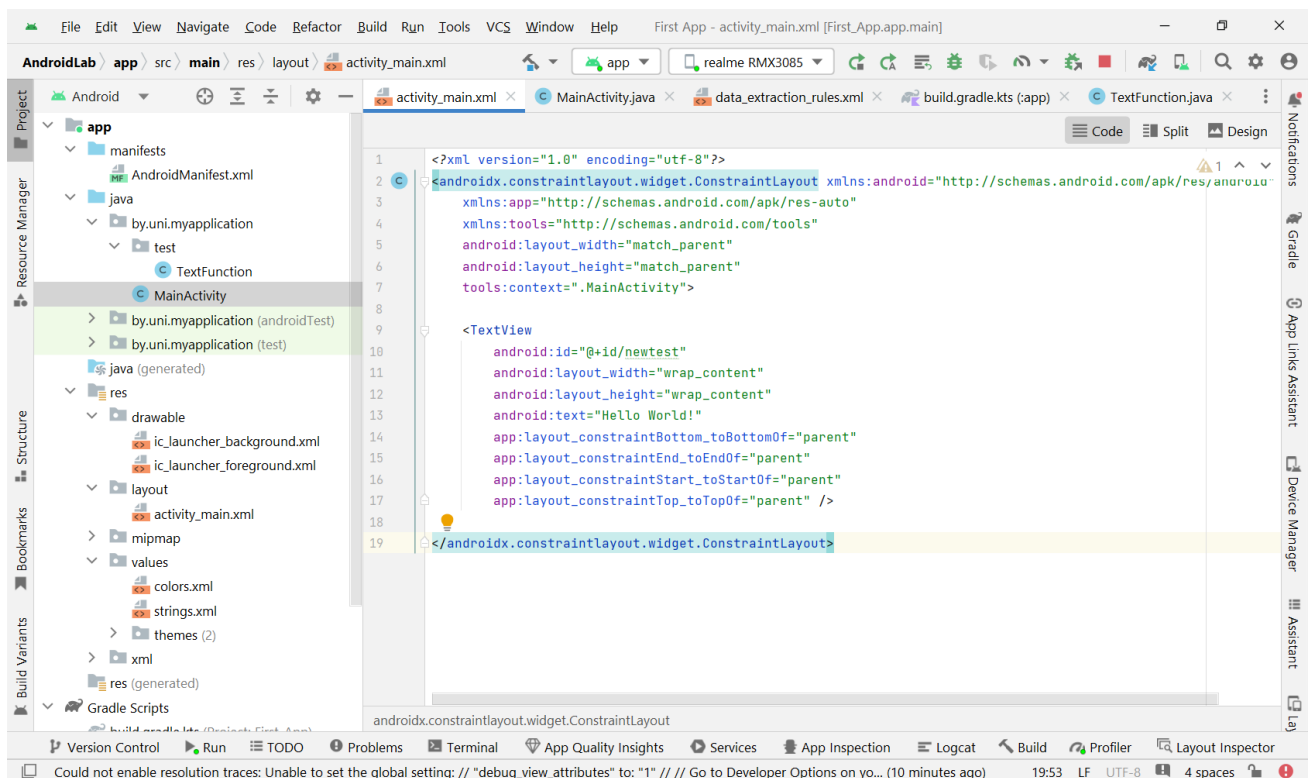
17. Следующим шагом будет добавить id для блока текста который был создан по умолчанию. Для этого можно из MainActivity по значку связанных файлов перейти к связанному файлу xml и добавить в него одну строку.



18. Стоит также помнить что по умолчанию скорее всего выбран режим отображения Design в нём также можно задать id но в данном случае рекомендую открыть режим Code и добавить в код текстового поля следующий код «`android:id="@+id/newtest"`».



И получим примерно следующее



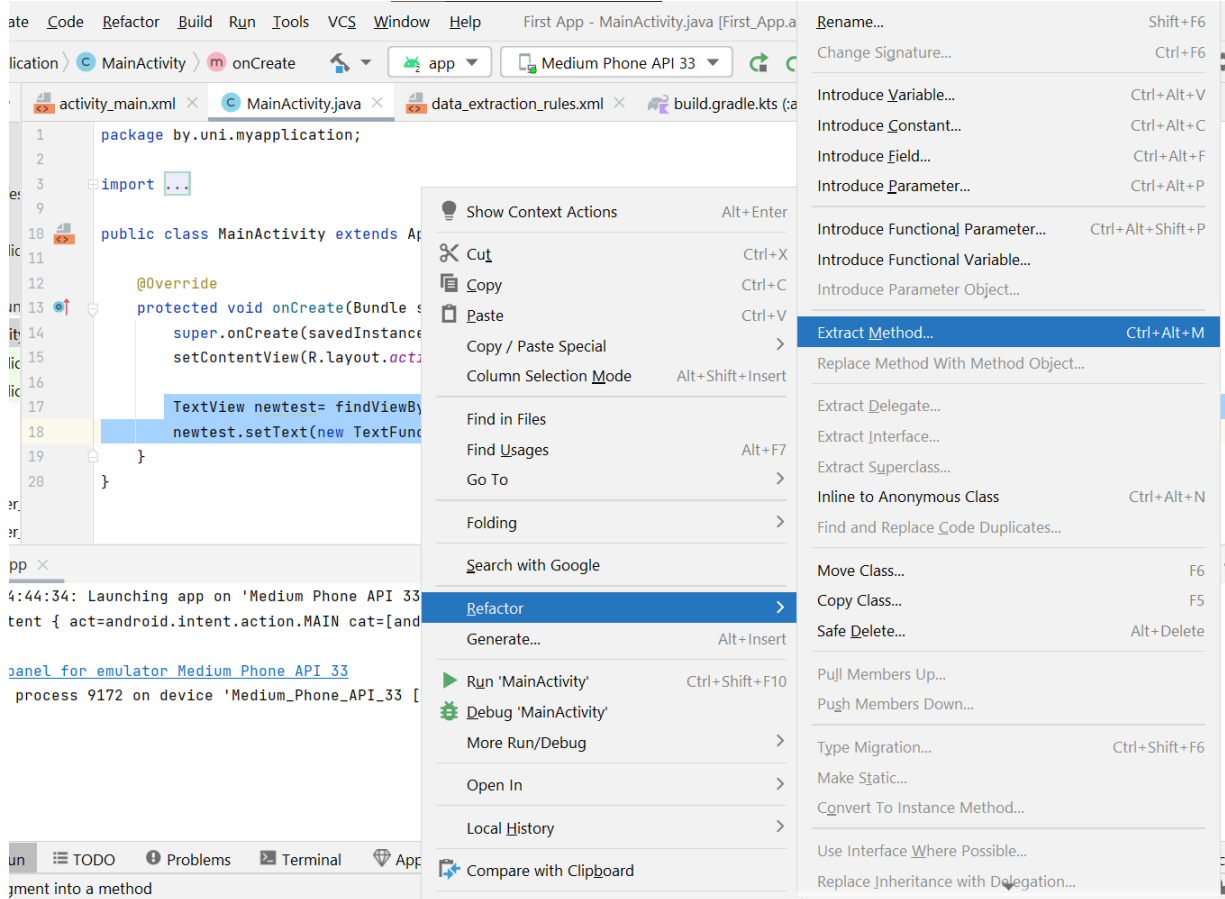
19. После добавления id можно получить доступ к этому элементу. Для этого в MainActivity можно дописать следующий код.

```
TextView nt= findViewById(R.id.newtest);
nt.setText(new TextFunction().getValue());
```

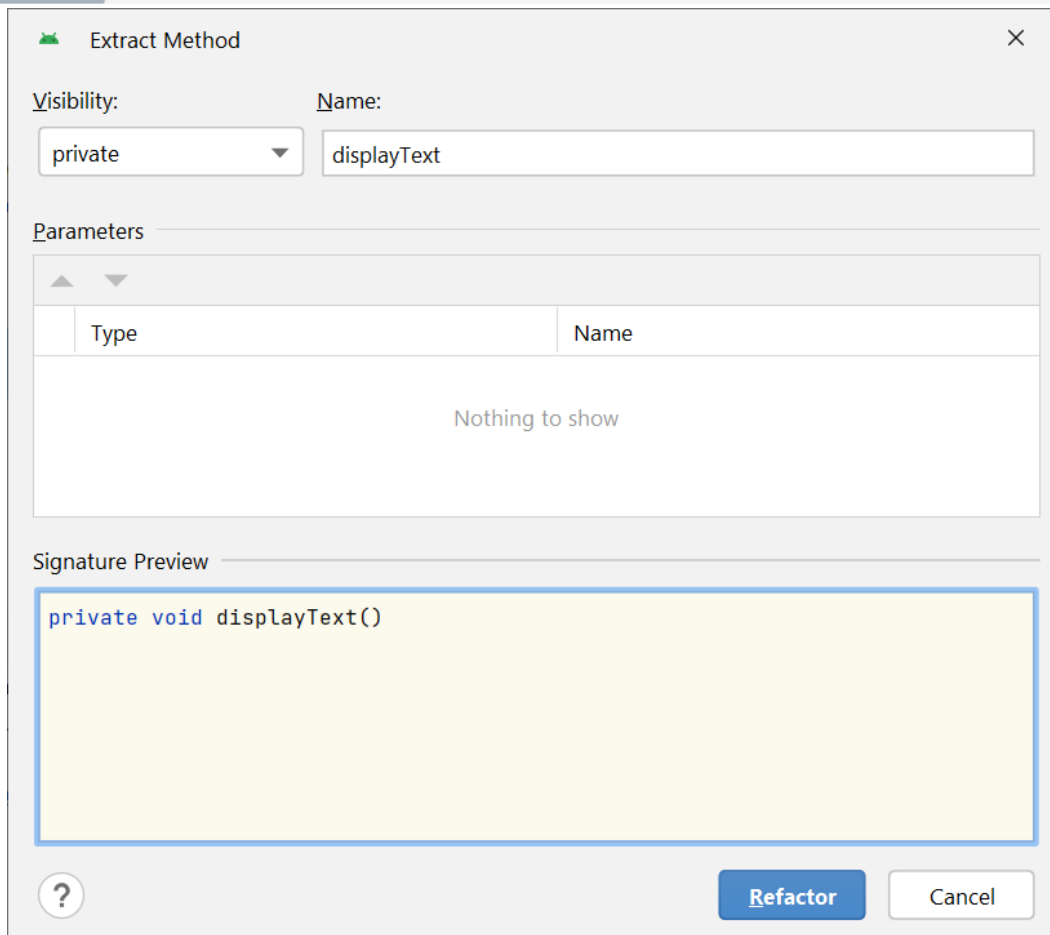
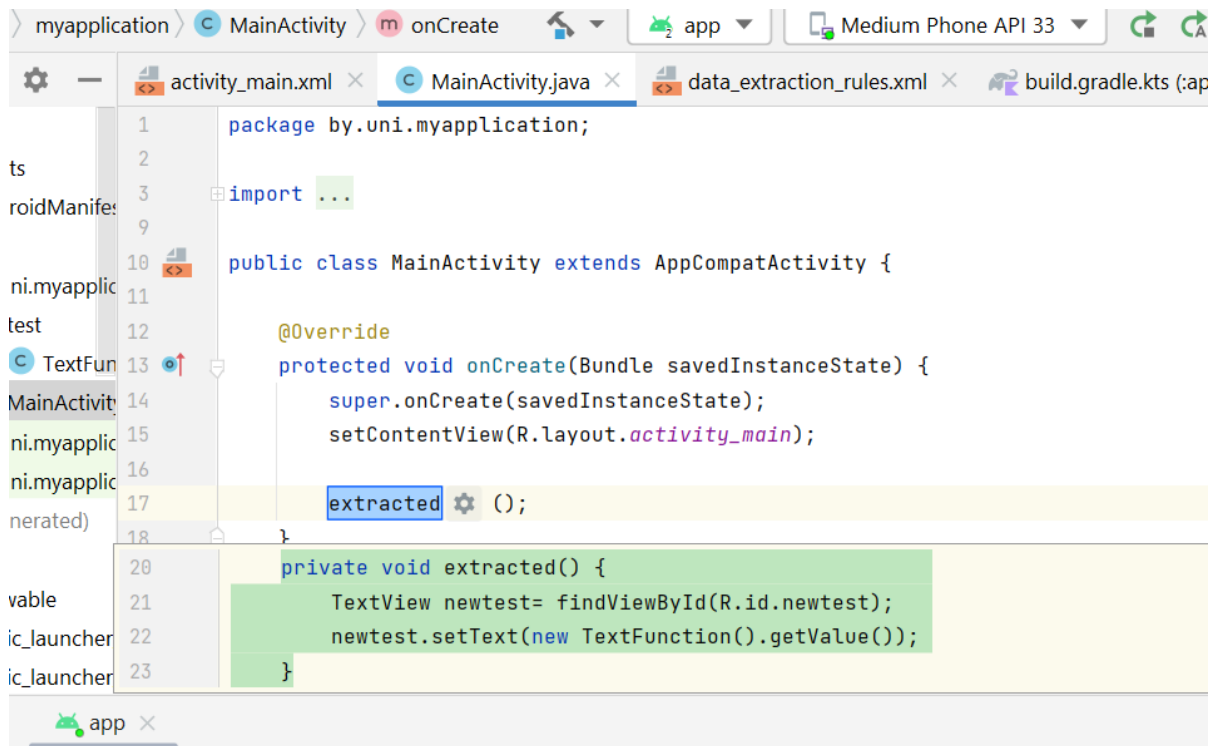


Результат запуска исследуйте сами.

20. После запуска и проверки работоспособности код можно слегка улучшить выделив написанный фрагмент и проведя рефакторинг. Refactor -> Extract-Method.



И после нажатия можно будет задать имя новому методу (выделено синим). Или воспользоваться значком шестерёнки чтобы открыть меню с параметрами метода.

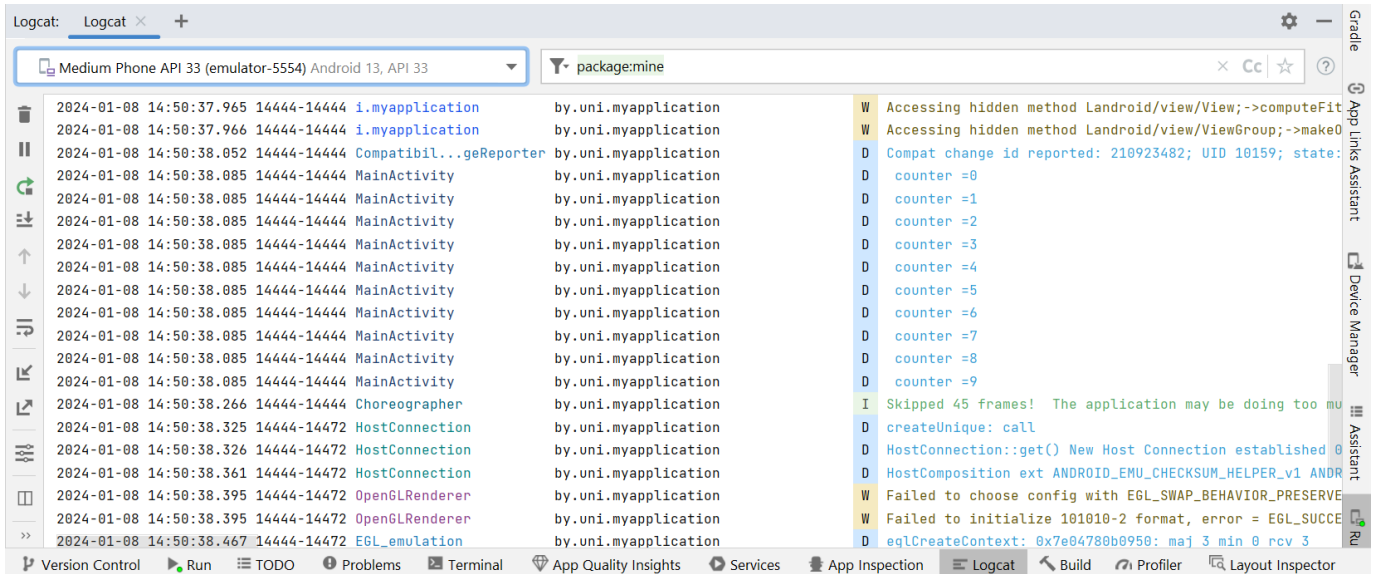


**21.** Также в ходе разработки часто необходима отладка приложений. Для этого можно использовать логирование как пример добавим цикл выводящий в лог итерацию цикла. Для этого можно просто в методе onCreate вписать for и нажать Enter.

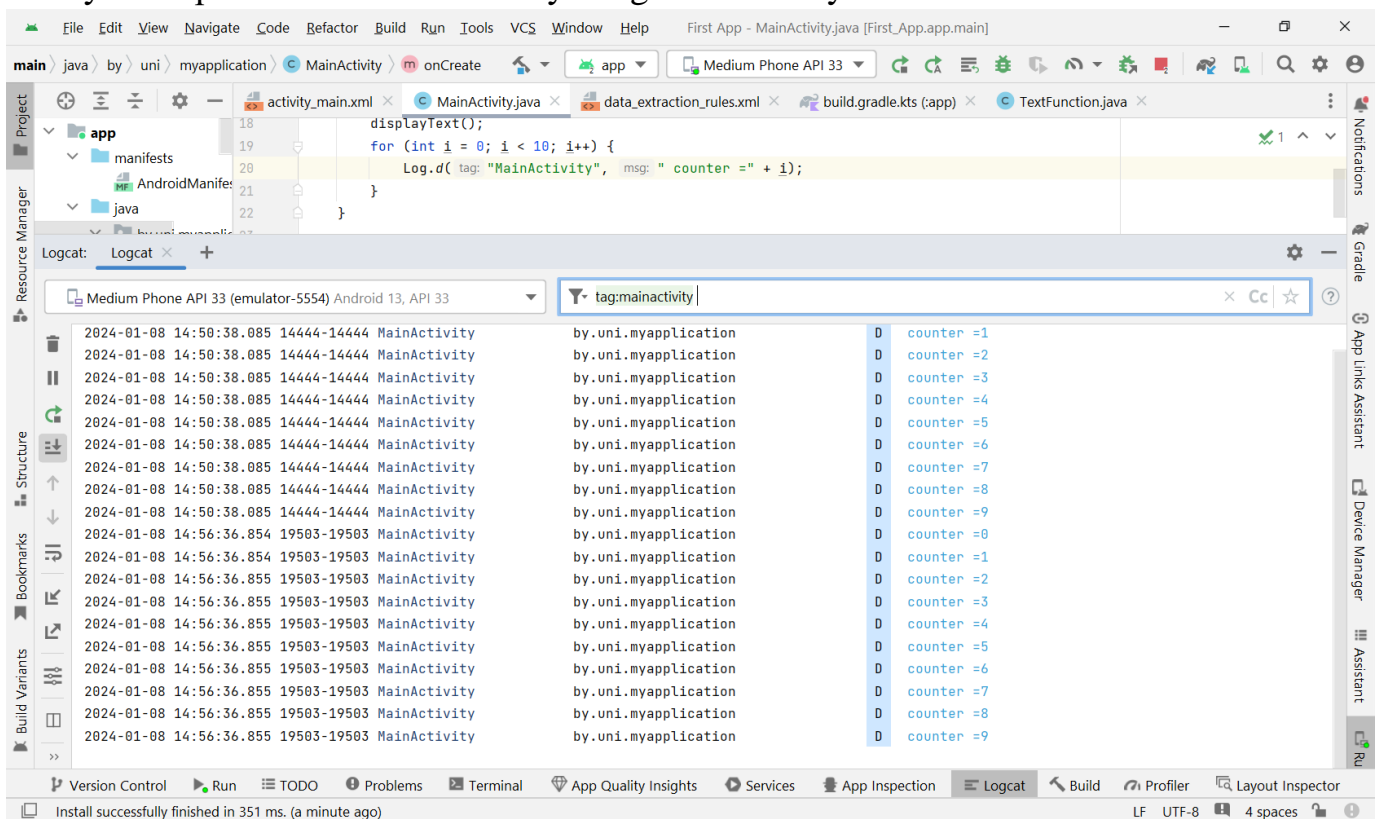




23. Потом найдя вкладку Logcat на нижней панели можно найти логи.

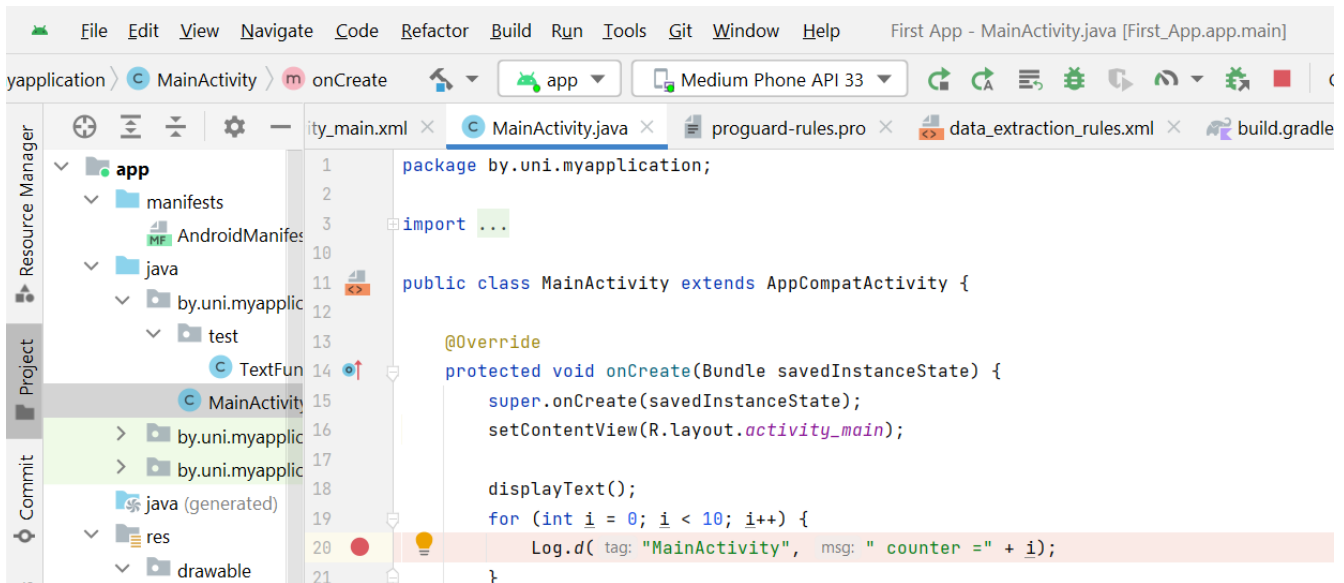


24. Чтобы уменьшить число логов можно прописать условия поиска логов в данном случае лучше прописать поиск по тэгу. «tag:mainactivity»

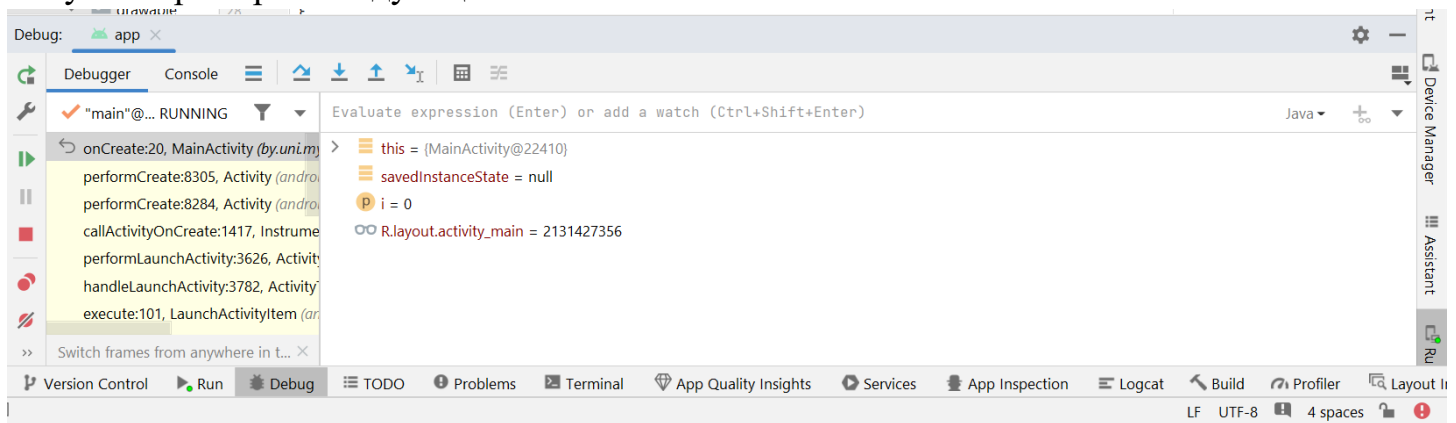


25. Следующим шагом добавим точку останова в цикл. Просто нажать возле номера строки

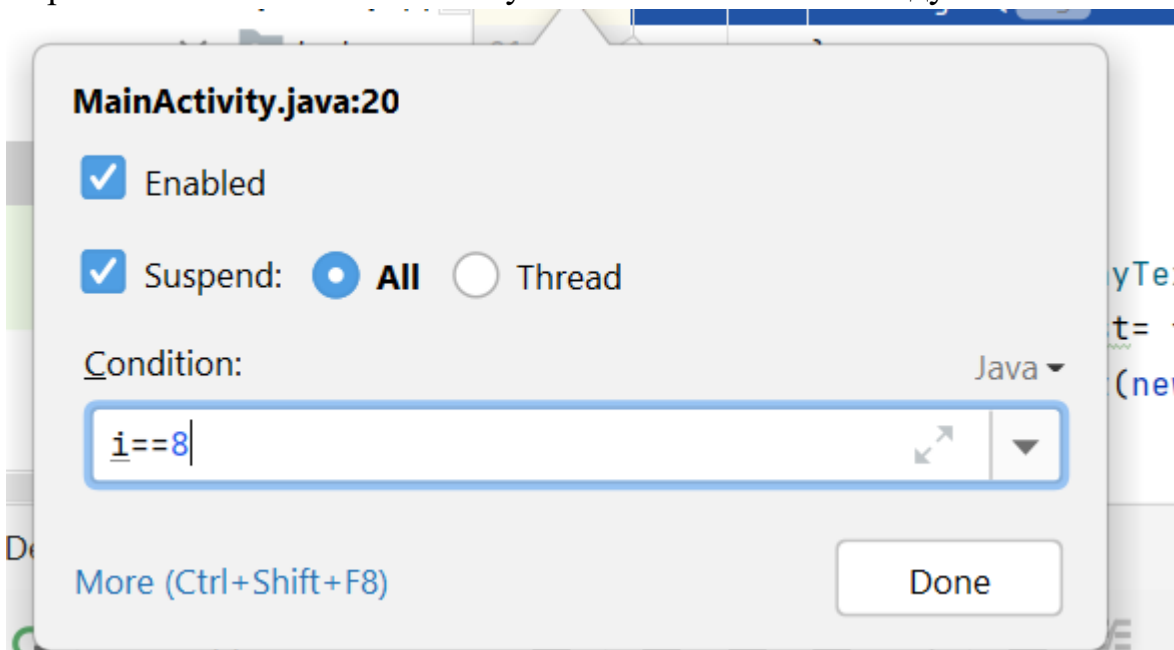




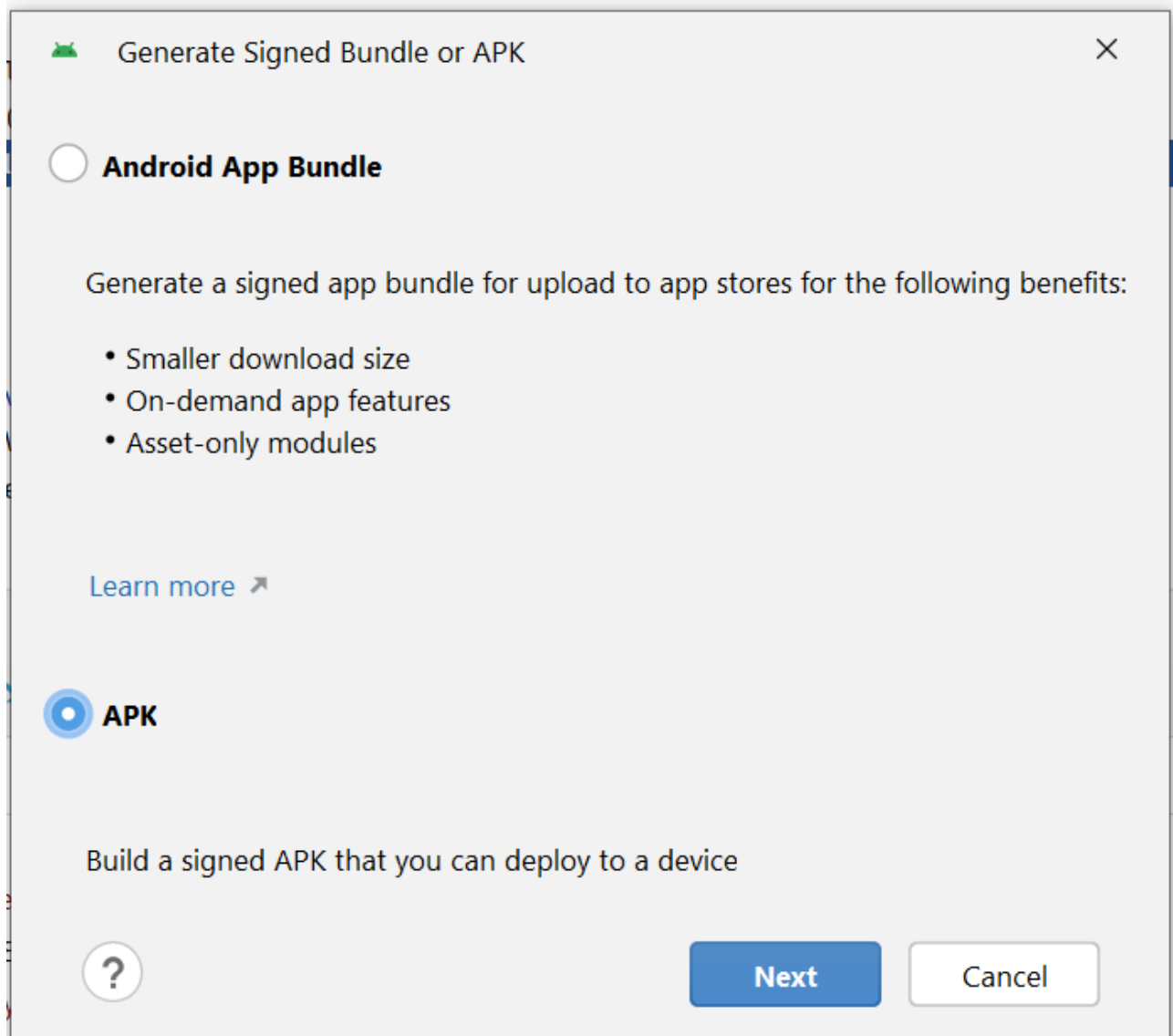
26. При запуске в режиме отладки(значок жука или Shift-f9 или Run->Debug) получим примерно следующее




27. По клику на точку останова (красный круг) можно установить условие при которым сработает точка останова. Результат выполнения исследуйте сами.



## 28. Публикация приложения. Из меню Build –Generate Signed APK



29. Далее из за отсутствия ключа его необходимо сгенерировать

 **Generate Signed Bundle or APK** ✕

Module

First\_App.app

Key store path

G:\key.jks

Create new...

Choose existing...

Key store password

.....

Key alias

key0

Key password

.....

☐ Remember passwords

?

Previous

Next

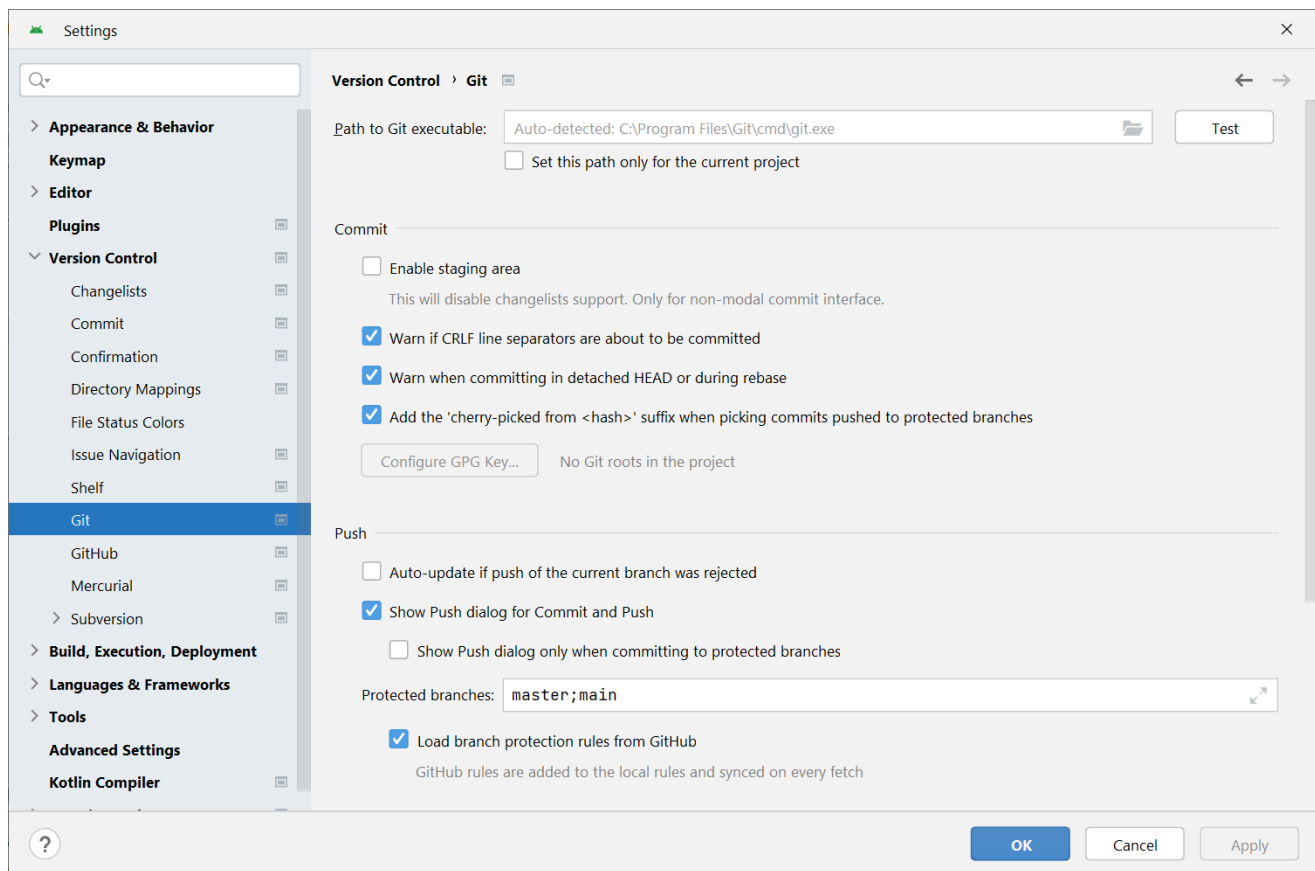
Cancel

30. Ну и как результат можно найти файл Apk в папке с проектом.

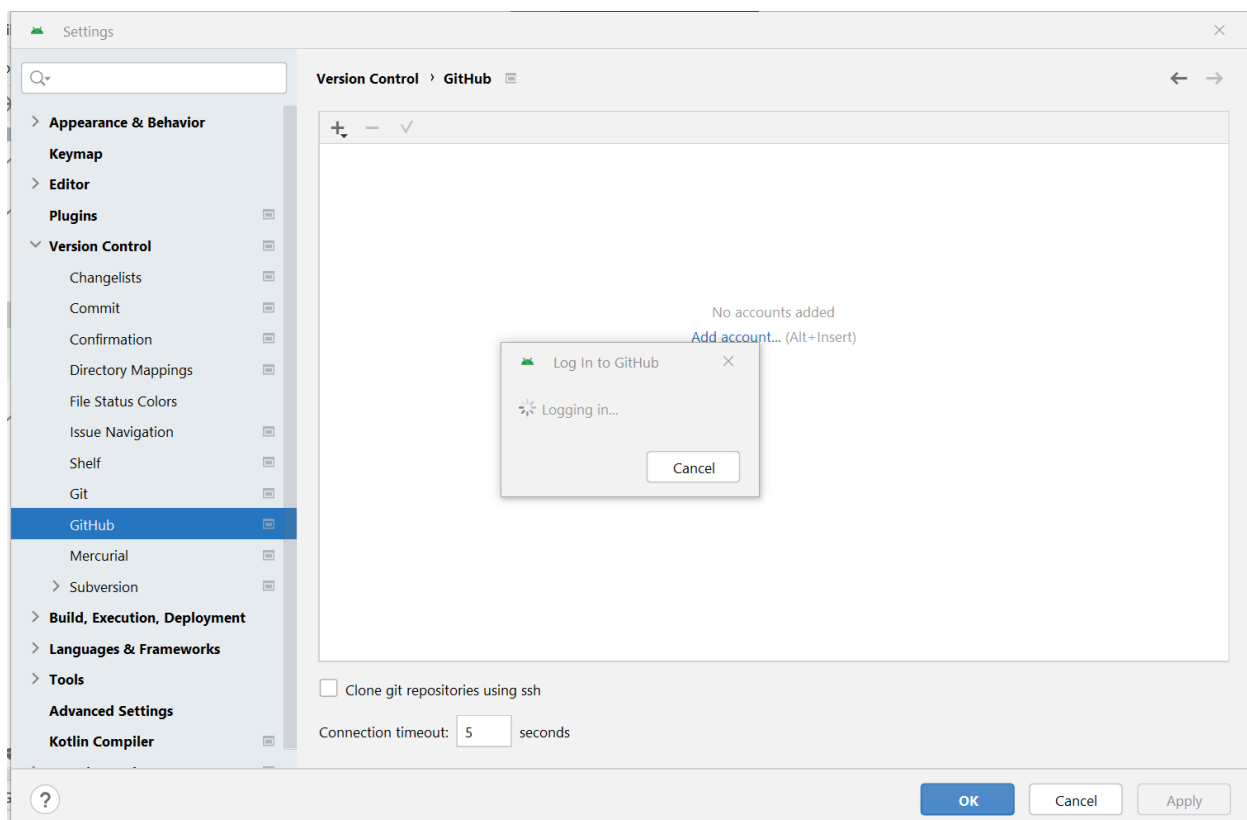
Имя	Дата изменения	Тип	Размер
app-release.apk	08.01.2024 15:07	Файл "APK"	4 496 КБ
output-metadata.json	08.01.2024 15:07	Исходный файл J...	1 КБ

### 31. Интеграция с Git

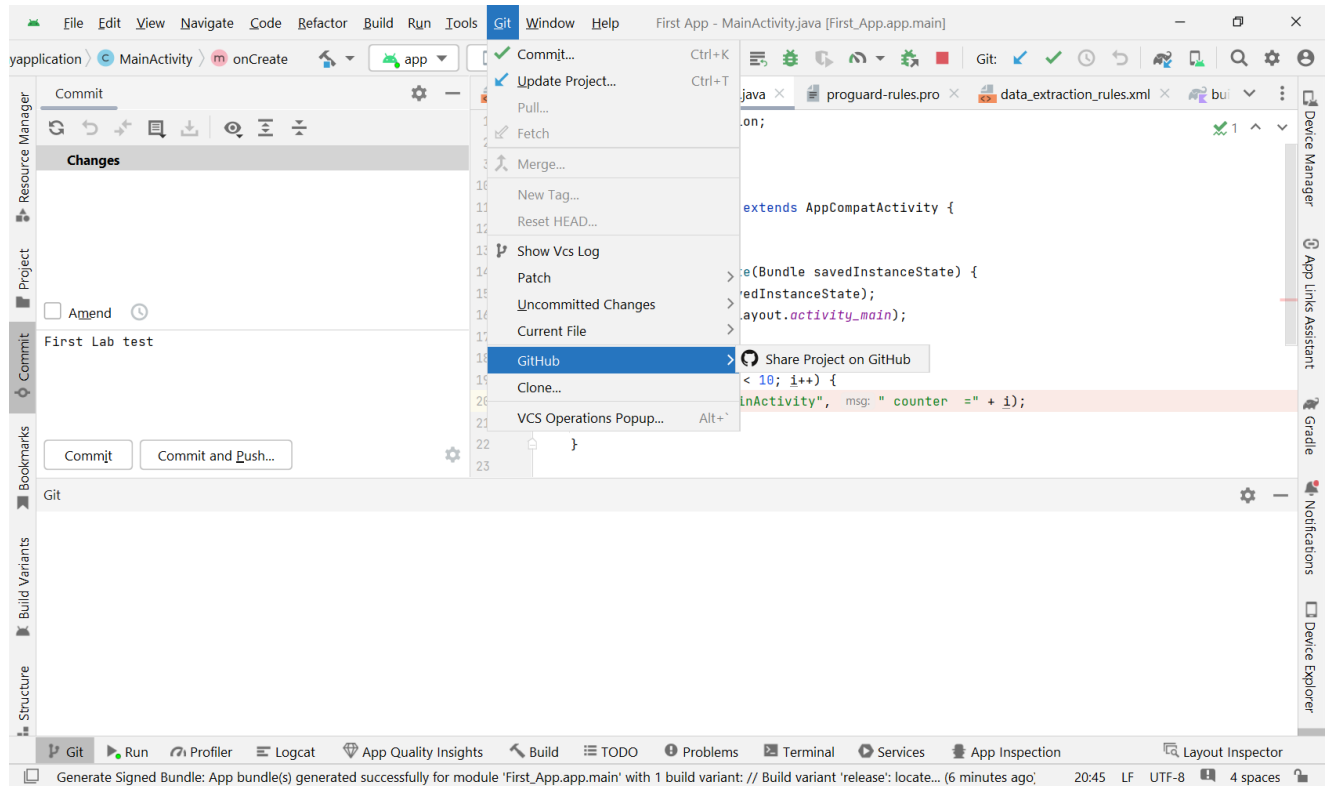
Перейдите в File->Settings->Version Control->Git И нажмите на кнопку Test чтобы проверить что он установлен или нажмите кнопку установить появится после нажатия на кнопку test.



После перейдя в меню в пункт **GitHub** и нажав на плюс вас перебросит на сайт JetBrains где нажав на кнопку **Authorize in GitHub** можно авторизоваться.



**32.** После успешного входа с помощью вкладки VCS или Git можно загрузить проект в систему контроля версий.



**Share Project On GitHub**

Repository name:  ☒ Private

Remote:

Description:

Share by:  [Add account](#)

## Задание к выполнению

1. Установить Android Studio
2. Создать новый проект Empty Views Activity
3. Изучить содержимое папок res, manifests, Gradle Script.
4. Подключите виртуальное устройство для отладки. Исследуйте возможности взаимодействия с виртуальными устройствами и их настройки.
5. Поясните назначение (левая панель) Structure, Bookmarks, Resource Manager, Build Variants.
6. Поясните назначение вкладок нижней панели Build, Run, ToDo, LogCat
7. Добавьте пакет (Package в проект).
8. Добавьте класс (Package в проект) и изучите возможности рефакторинга для класса. Код класса представлен ниже.

```
public class TextFunction {  
    public String getValue(){  
        return "Hello from first project1";  
    }  
}
```

Переместите созданный класс в пакет (можно просто перетаскиванием) . Обратите внимание как поменялось имя package

9. Добавьте id элементу TextView и добавьте код для изменения текста этого элемента в MainActivity.

```
TextView nt= findViewById(R.id.newtest);  
nt.setText(new TextFunction().getValue());
```

10. Проведите рефакторинг кода из задания 9 с выделением его в отдельный метод.
11. Исследуйте возможности отладки на примере точки останова а также задайте условие для точки останова и логирования на пример с выводом в цикле лога с итерацией. Отсортируйте логи по имени тега.
12. Сгенерируйте и опубликуйте приложение в Apk.

## ***Задание по желанию***

1. *Настройте пользовательский интерфейс. File->Settings (ctrl-alt-s) затем Appearance & Behavior*
2. *Подключите физическое устройство для отладки.*