

Лабораторная работа №1. Создание первого приложения

Задание: скачать и установить последние версии Java Development Kit и Android Studio, установить необходимые компоненты SDK, сконфигурировать эмулятор, создать новый проект, изменить TextView с надписью "Hello world" на свои ФИО и запустить проект на эмуляторе.

Для того, чтобы начать программировать под Android, необходимо скачать и установить две вещи: Java Development Kit (JDK + JRE) и Android Studio. Тут дело в том, что на многих компьютерах уже установлена Java какой-то версии, поэтому первый шаг необязателен. Но лучше всего все-таки установить последнюю версию, на всякий случай, чтобы получить доступ к последней версии языка и development kit-а. Для JRE нужно перейти по ссылке <https://www.java.com/ru/download/>. Для JDK нужно перейти по адресу oracle.com – Products – Hardware and Software – Java, затем на новой странице – кнопка Download Java – выбираем свою ОС и разрядность (x86, x64) и нажимаем на ссылку, чтобы скачать файл установки (в моем случае это были файлы jre-8u421-windows-x64.exe и jdk-22_windows-x64_bin.exe) и установить с параметрами по умолчанию, для установки нужны права администратора.

Вторым шагом идет установка Android Studio. Тут нужно отметить, что последние (свежие, вышедшие недавно) версии Android Studio, как правило, нестабильные, и в них очень много багов – как визуальных, так и на уровне программирования. Поэтому, если вы не боитесь багов и нестабильности, перейдите по адресу <https://developer.android.com/studio/>, нажмите на кнопку «Загрузить Android Studio (Koala Feature Drop на данный момент)», примите лицензию, скачайте и установите приложение с параметрами по умолчанию (1.1 Гб, версия 2024.1.2.12, Android Studio Koala Feature Drop, август 23, 2024, см. рис. 1), для установки нужны права администратора. Если же вам больше подходит стабильность, отсутствие багов, надо найти какую-нибудь уже проверенную (не сырую) версию Android Studio под названием Electric Eel (последний патч для которой вышел в 2023 году), и установить её, например, отсюда: <https://redirector.gvt1.com/edgedl/android/studio/install/2022.1.1.20/android-studio-2022.1.1.20-windows.exe>. Electric Eel позволяла выбирать язык программирования между Java и Kotlin даже для шаблона Empty Activity. Если хотите, на сайте <https://teamandroid.com/> можно найти ещё более старую версию Android Studio, например Dolphin или Arctic Fox, или даже использовать другую IDE типа IntelliJIDEA, Eclipse или даже Visual Studio – для прохождения данного курса этого будет вполне достаточно. Но вы должны установить и сконфигурировать соответствующие компоненты в сторонних IDE самостоятельно, если будете использовать их – Android SDK + Android Emulator.

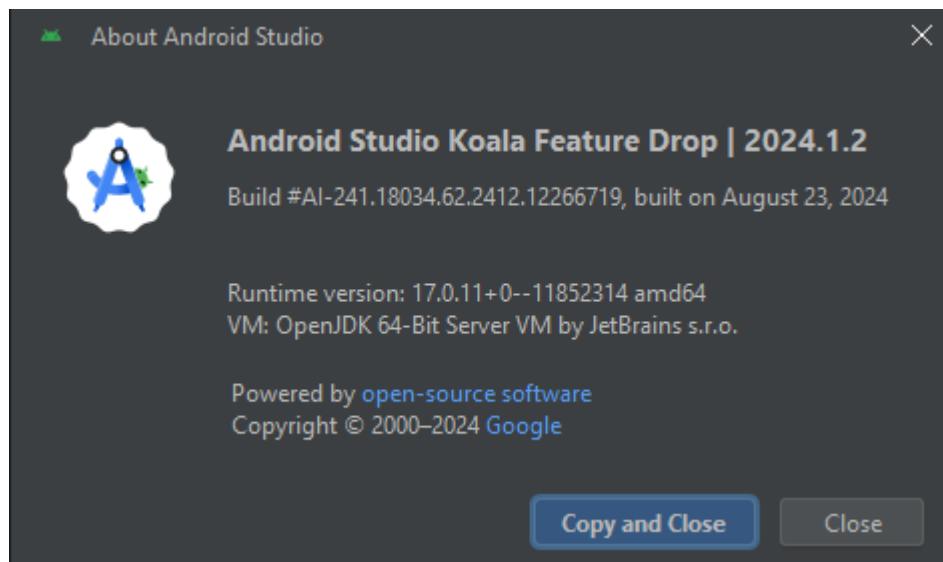


Рисунок 1 – Версия Android Studio Koala

После скачивания файла установки запустите его, и вы увидите стандартные окна установщика, спрашивающие вас, куда установить приложение, говорящие, что понадобится минимум 3.6 Гб свободного места, и т.д. Установите Android Studio и запустите ее.

Другой вариант установки возможен, если вы устанавливаете не Android Studio Koala. Тогда, после запуска установщика, возможно, вы увидите окно, показанное на рисунке 2, и последующие окна, см. рисунки 3 – 10. Окно на рисунке 2 появится, если у вас ранее была установлена или ещё установлена какая-то другая версия Android Studio. Можно выбрать опцию не импортировать настройки, как показано на рисунке 2, и нажать кнопку OK.

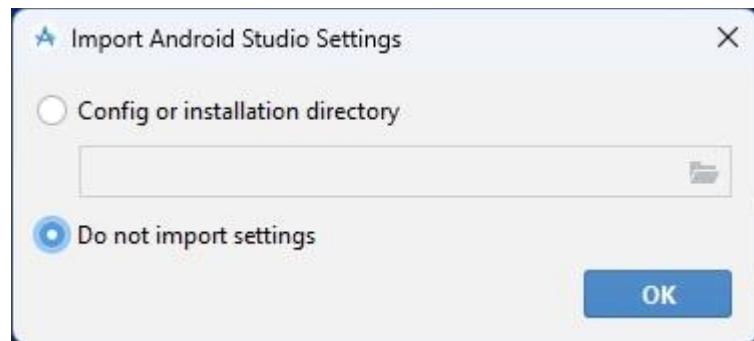


Рисунок 2 – Окно импорта настроек из предыдущей версии Android Studio

Должно появиться окно сбора статистики использования Android Studio, изображённое на рисунке 3. Здесь вам нужно сделать выбор, посыпать ли эту статистику в Google, или нет.

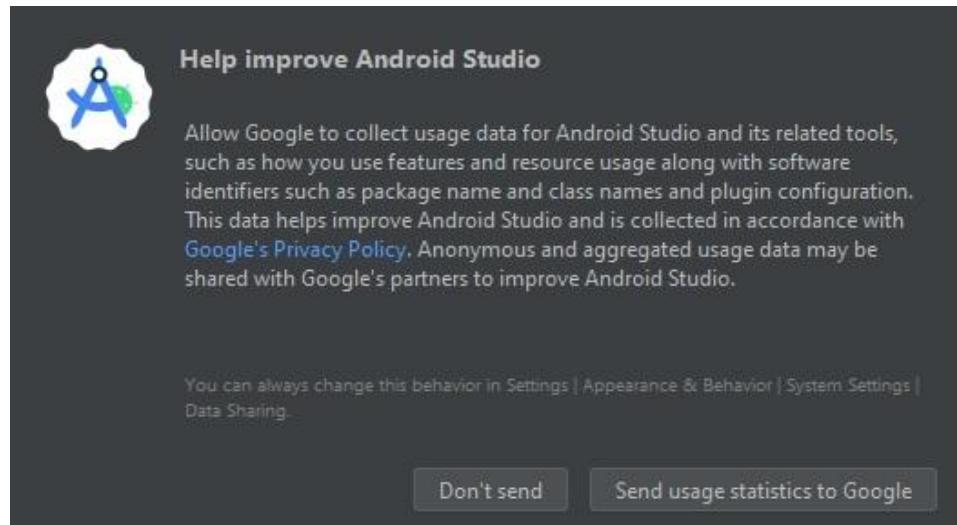


Рисунок 3 – Окно сбора статистики использования среды

Далее появится окно приветствия установщика Android Studio, показанное на рисунке 4. На нём можно видеть условные обозначения устройств, для которых можно разрабатывать приложения в Android Studio. Нажмите кнопку Next.

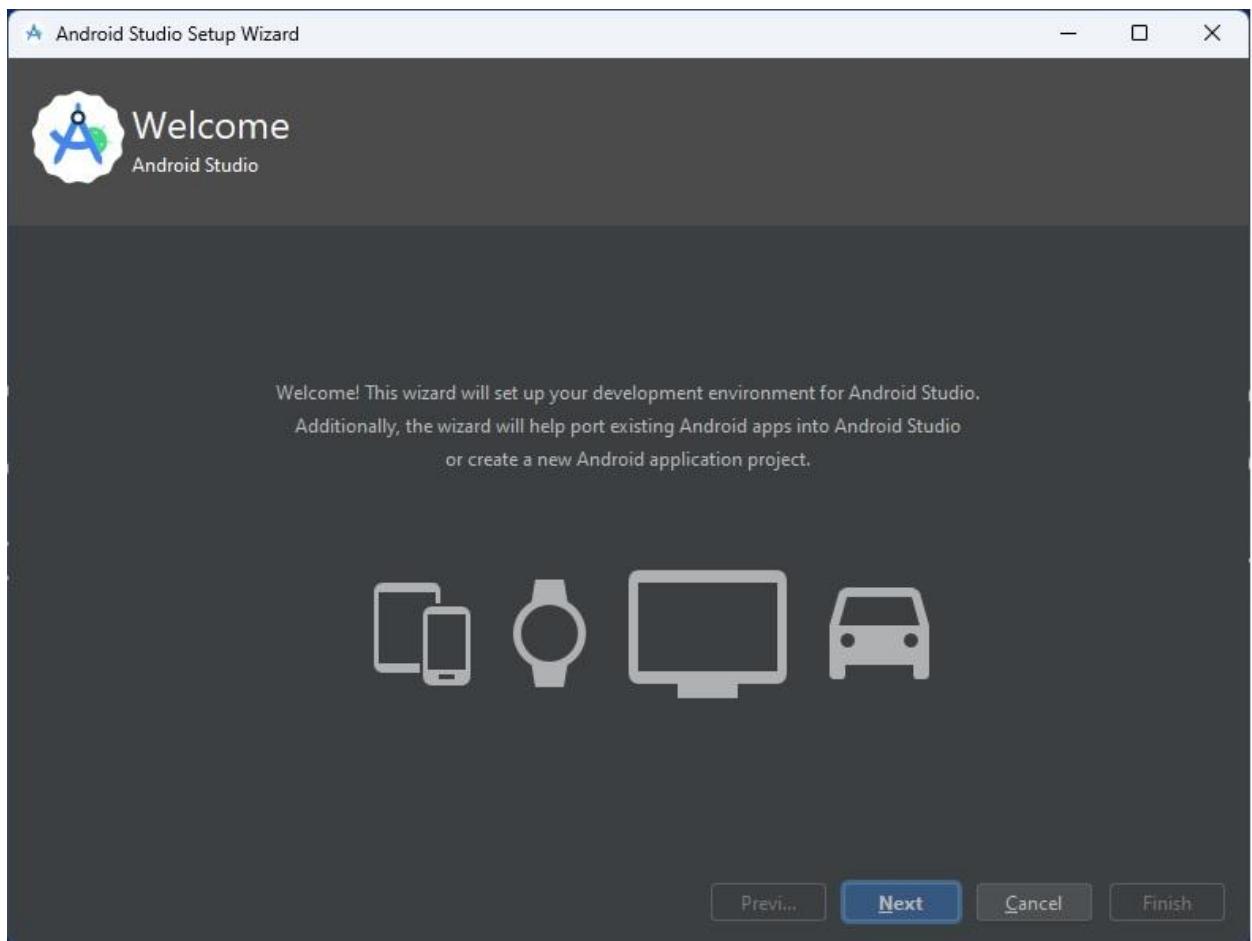


Рисунок 4 – Окно приветствия установщика

В следующем окне, приведённом на рисунке 5, необходимо выбрать тип установки. Рекомендуемый тип – стандартный, он стоит по умолчанию. Однако если вам уже на этом этапе необходимо выбрать что-то ещё, можно выбрать тип Custom (пользовательский). Нажмите на кнопку Next.

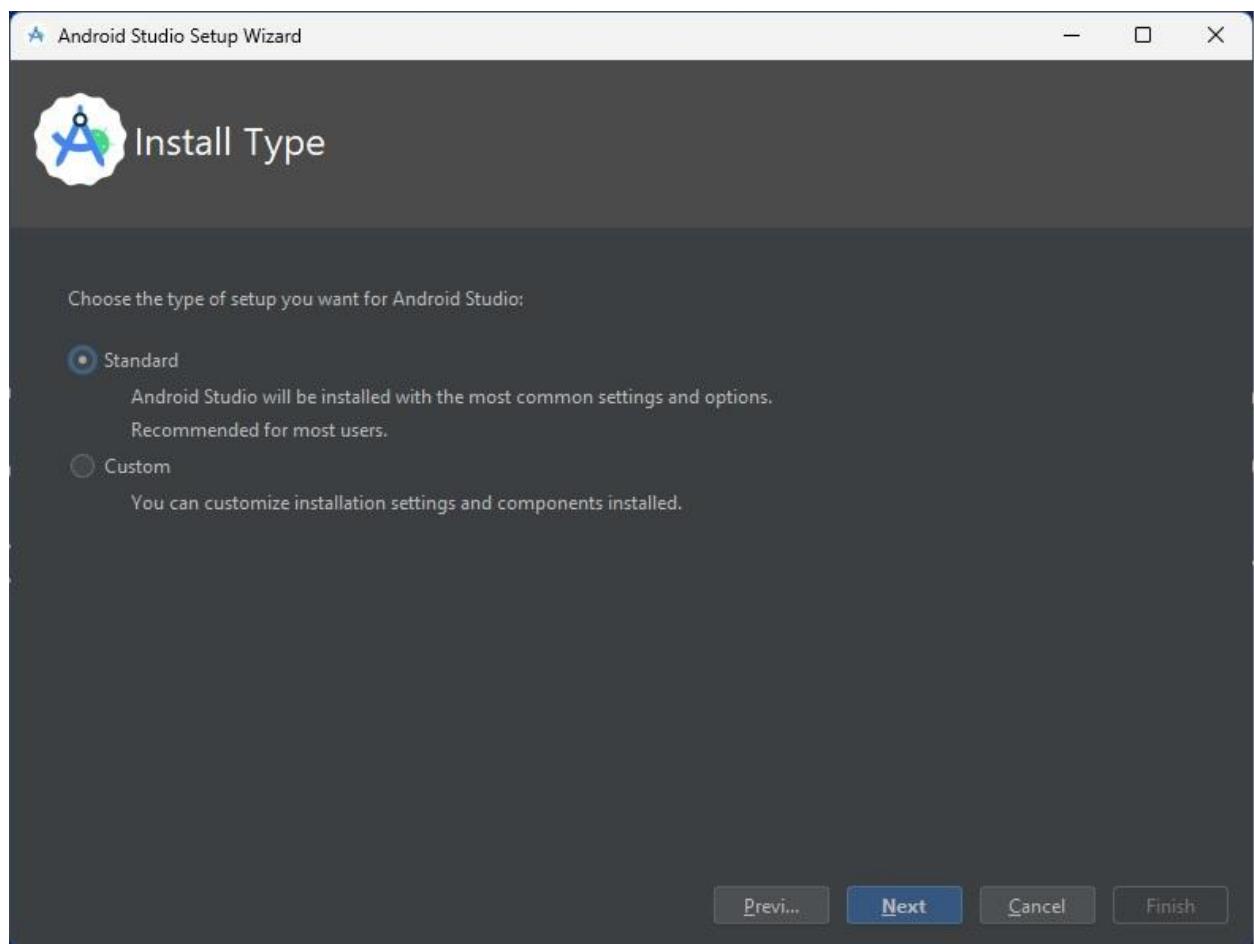


Рисунок 5 – Окно выбора типа установки

В следующем окне вы можете выбрать, как будет выглядеть среда разработки, в основном это касается цвета фона, на котором будет код, см. рисунок 6. Снова нажмите кнопку Next после выбора темы.

В очередном окне, показанном на рисунке 7, установщик показывает вам всю необходимую информацию, собранную им для установки. Можно снова нажать на кнопку Next.

Далее надо согласиться с лицензией. Установщик представляет возможность подробнее изучить лицензию, см. рисунок 8. Обратите внимание, что Accept надо выбирать для каждой категории лицензий, показанных в окне слева, т.е. в данном случае – три раза, а не один, см. рисунок 9. Пока вы не подтвердите своё согласие со всеми лицензиями, вы не сможете нажать на кнопку Finish (рисунок 8), на которую надо нажать, чтобы установщик приступил к работе (рисунок 9).

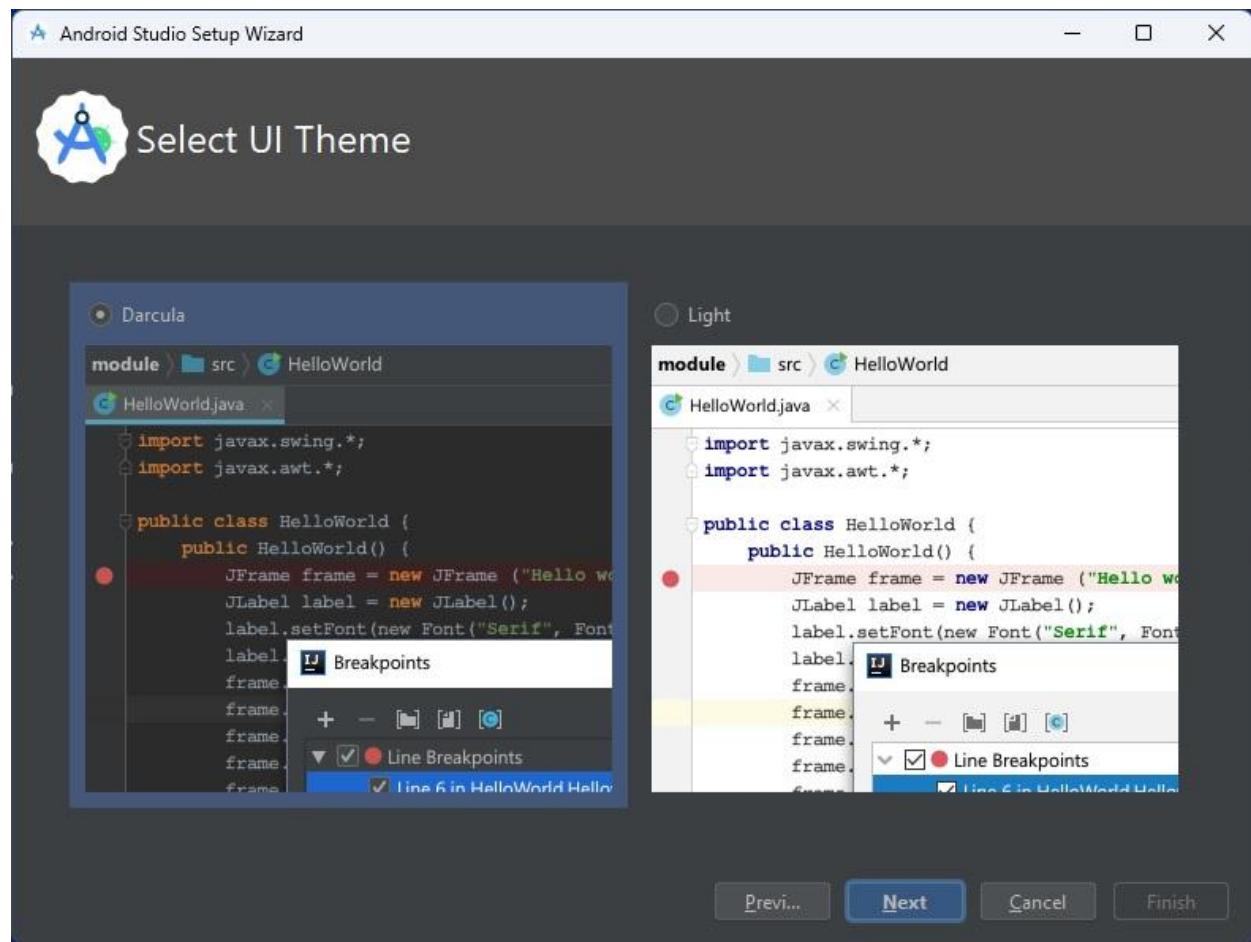


Рисунок 6 – Выбор визуальной темы интерфейса среды разработки

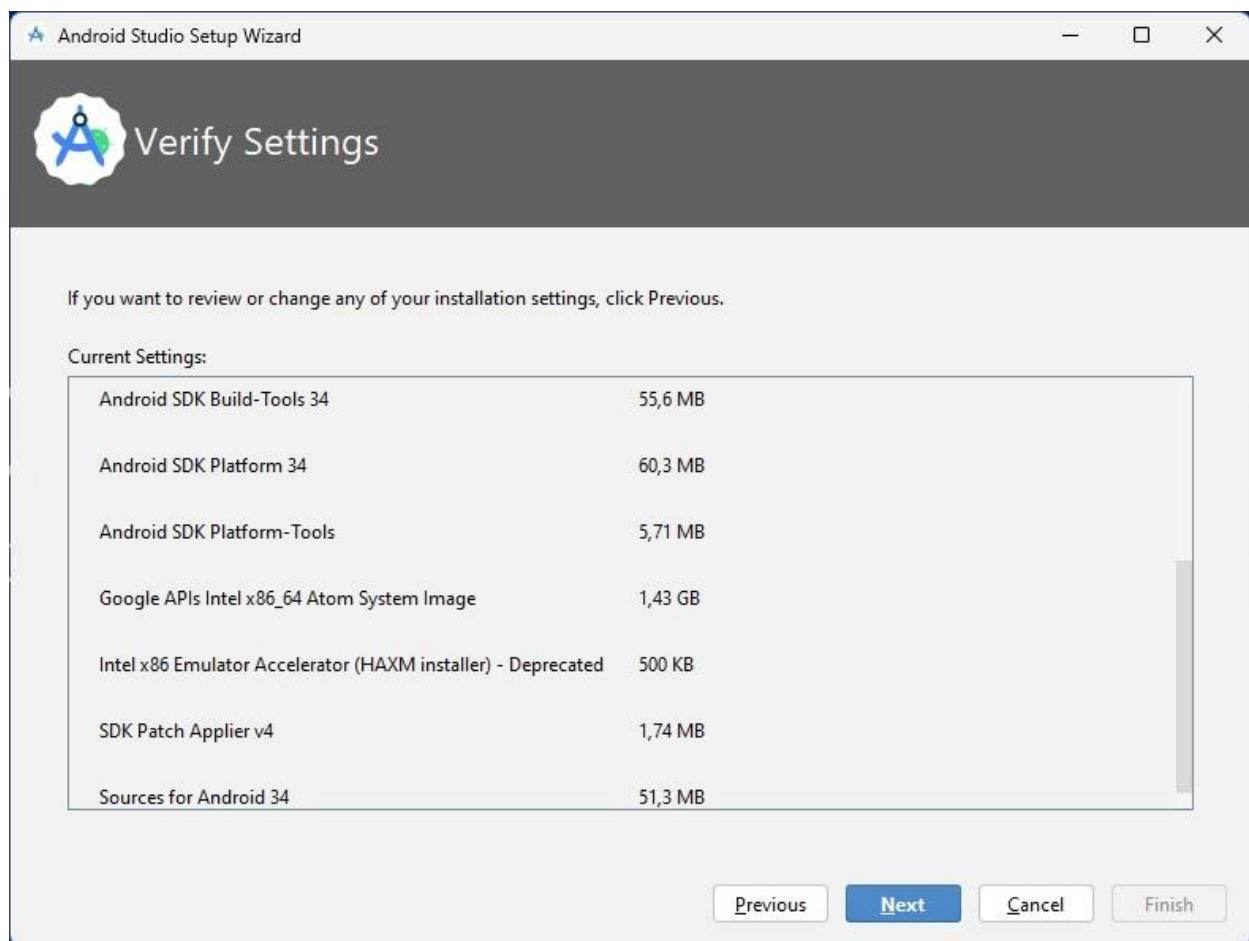


Рисунок 7 – Подтверждение выбранных компонентов для установки

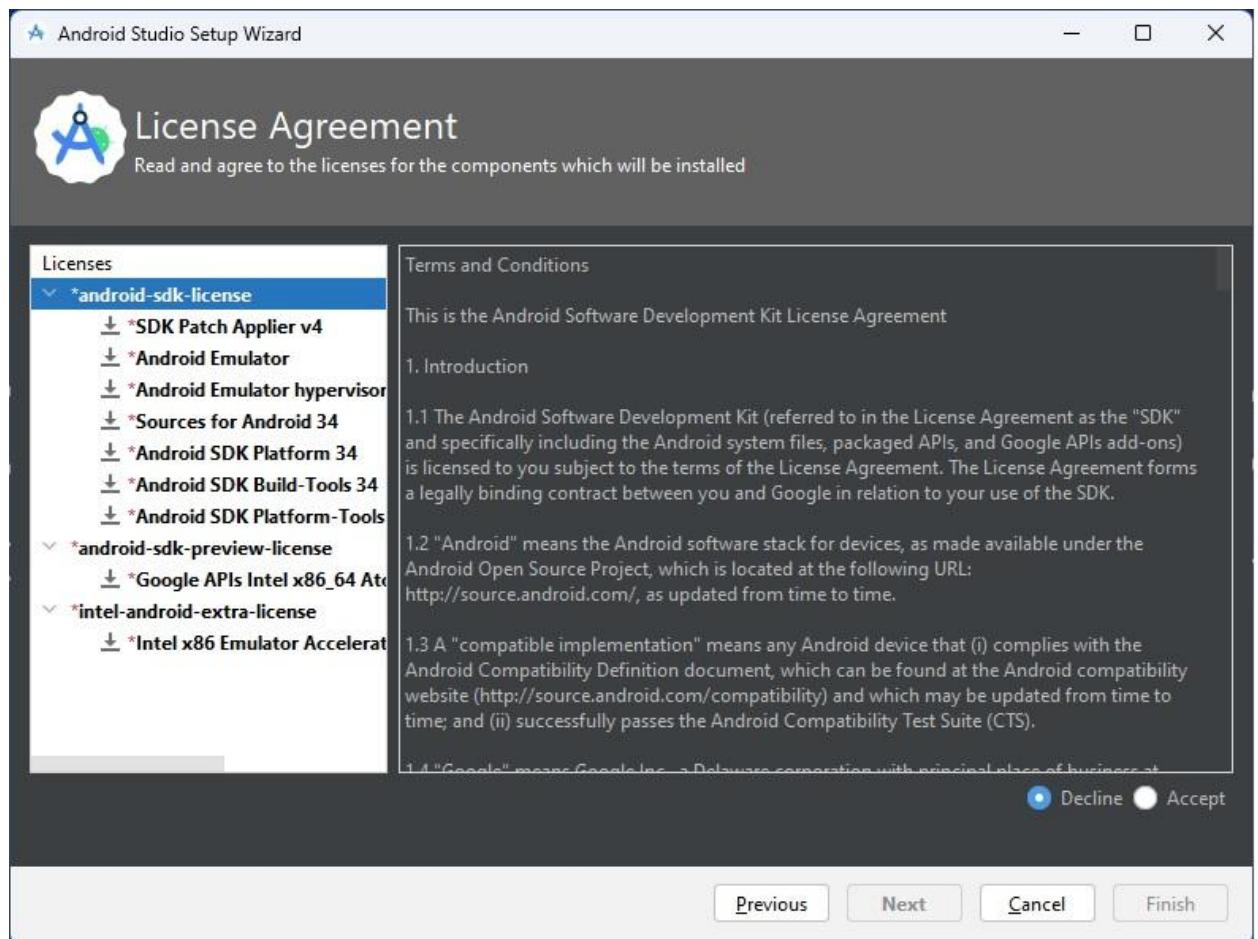


Рисунок 8 – Подтверждение согласия с лицензией

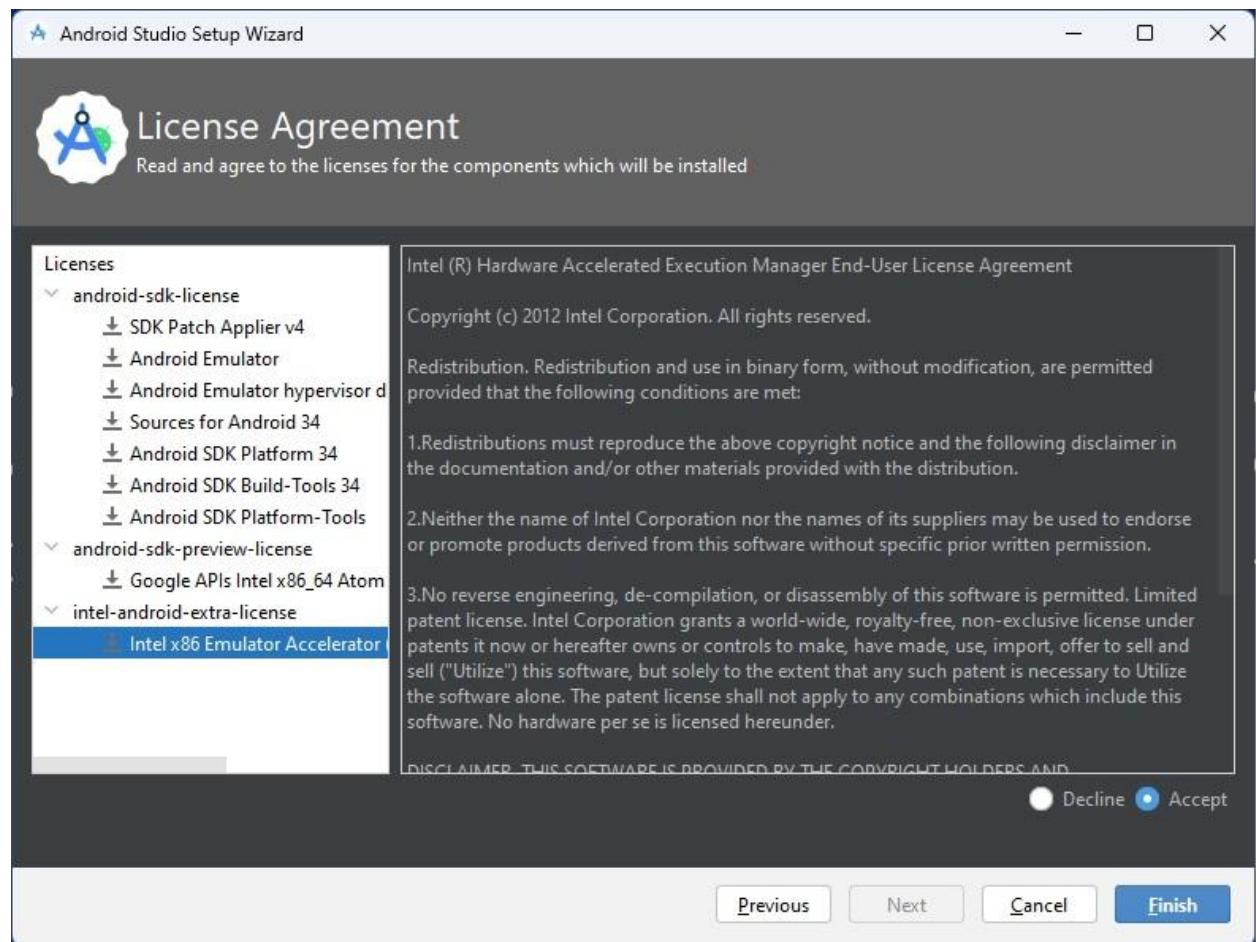


Рисунок 9 – Подтверждение согласия с лицензией – со всеми лицензиями

После того, как вы нажмёте кнопку Finish, начнётся установка выбранных компонентов, как показано на рисунке 10. По завершении установки надо снова нажать на кнопку Finish.

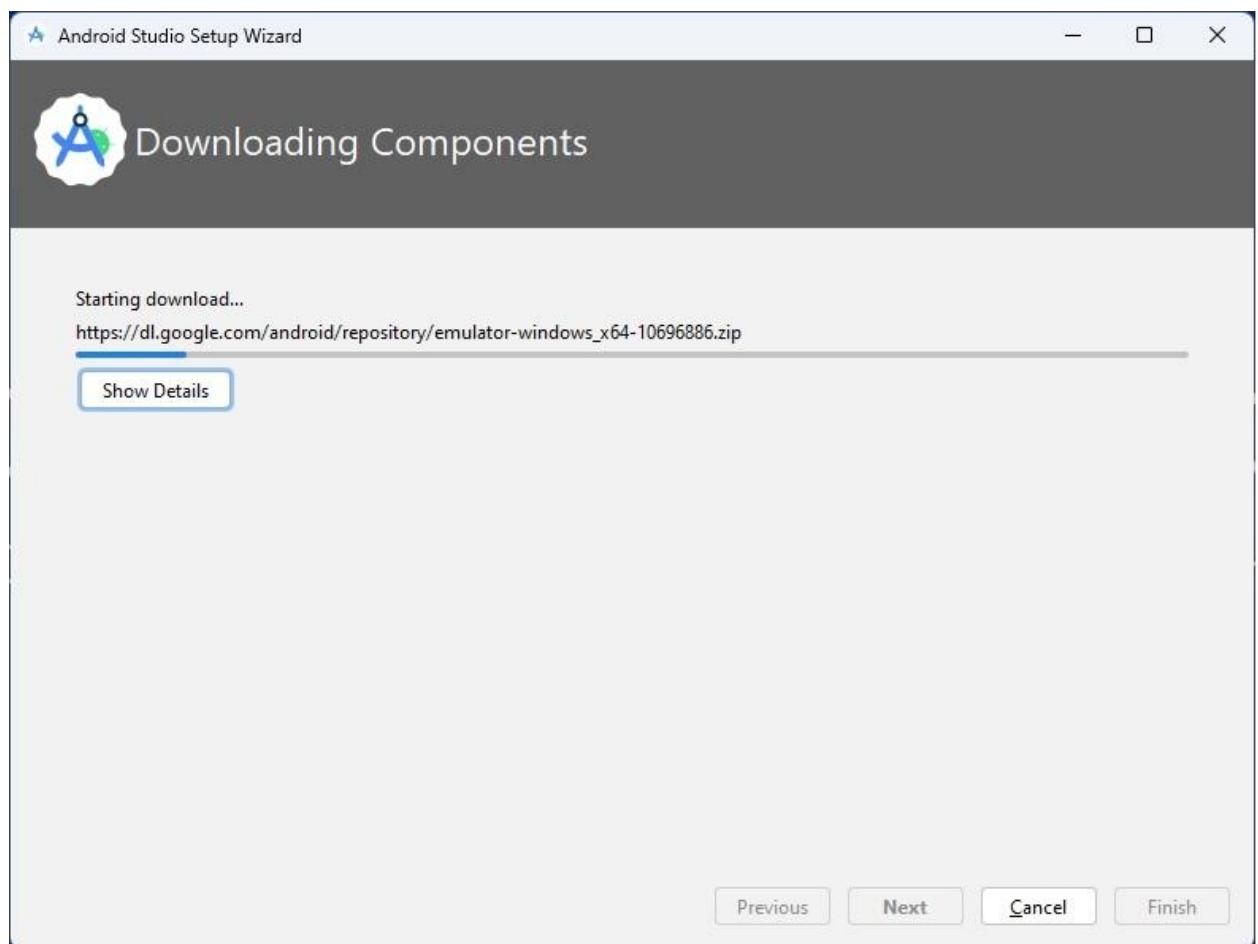


Рисунок 10 – Процесс установки

После установки появится другое окно приветствия, которое показано на рисунке 11 и принадлежит уже самой Android Studio, а не её установщику.

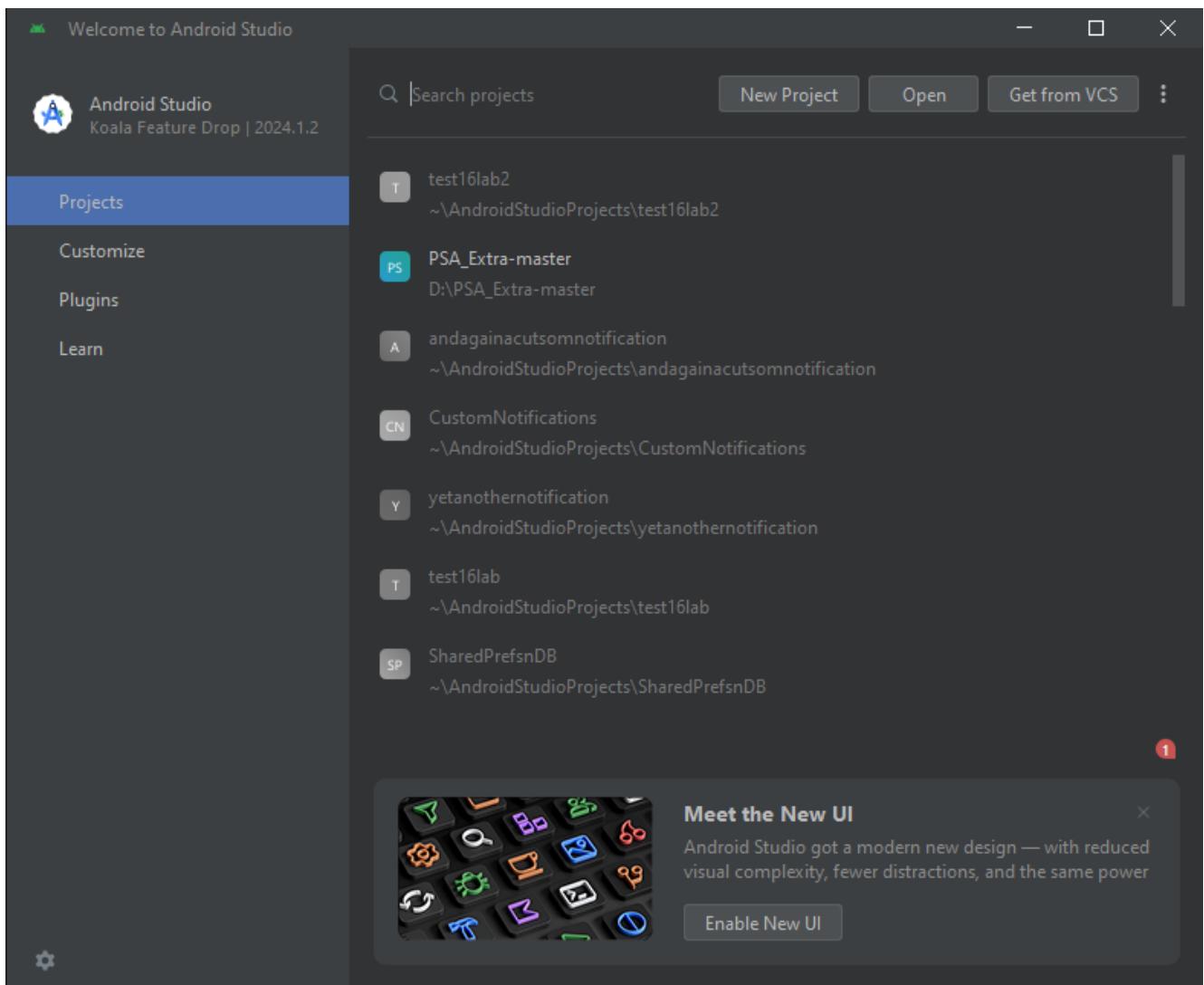


Рисунок 11 – Окно приветствия в Android Studio

В этом окне нас интересует кнопка **New Project**, но вообще в этом окне есть различные дополнительные возможности. Например, если нажать на ссылку **More Actions**, см. рисунок 12, можно понять, что можно, например, сразу перейти в **SDK Manager**, чтобы скачать пакет одной из последних версий API, можно сразу создать эмулятор (**Virtual Device Manager**), или импортировать проект.

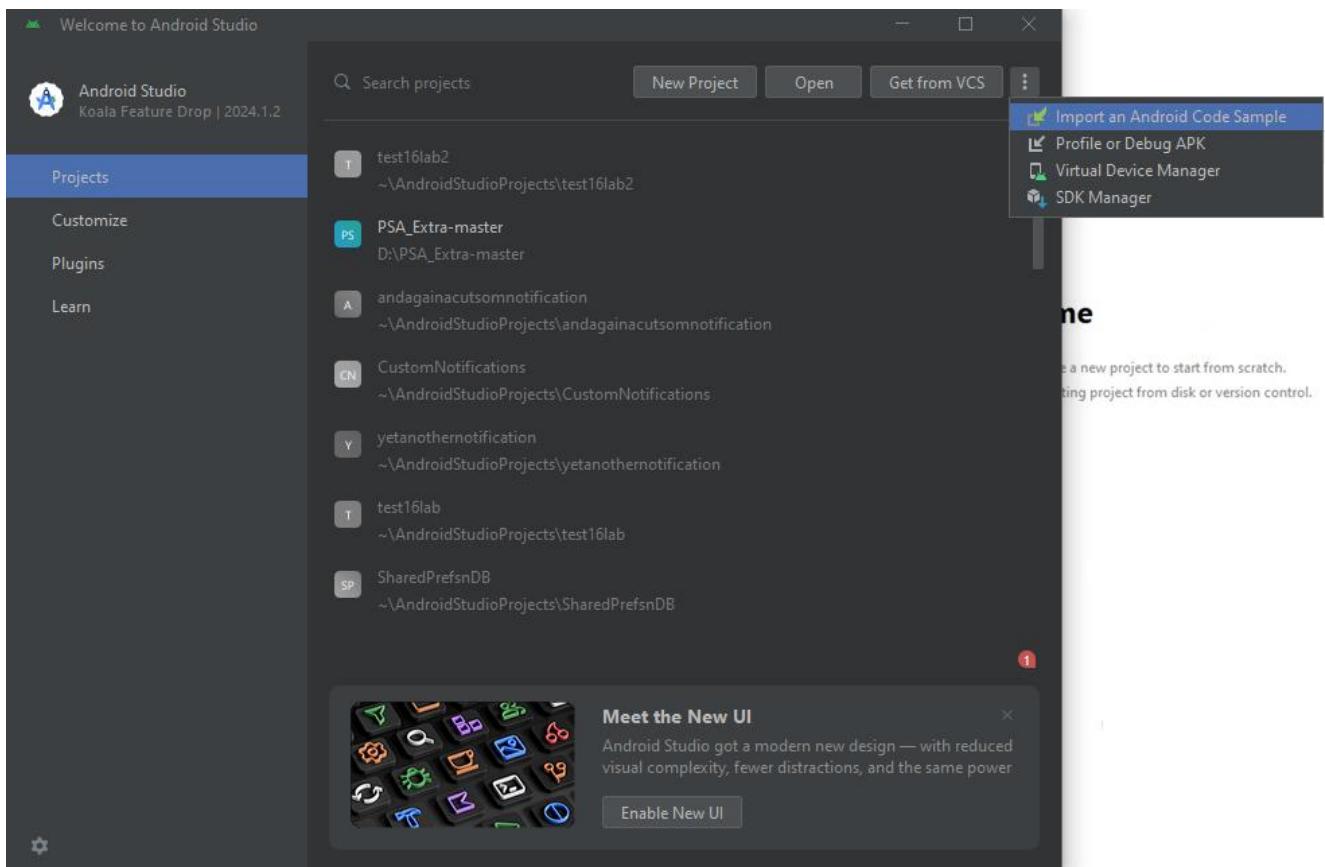


Рисунок 12 – Результат нажатия на More Actions

Если перейти на вкладку **Customize**, рисунок 13, можно настроить внешний вид среды разработки: поменять тему, синхронизировать её с темой, установленной в операционной системе, поменять шрифт и размер программного кода, и многое другое. Я поменял тему на IntelliJ Light, так как привык к белому фону.

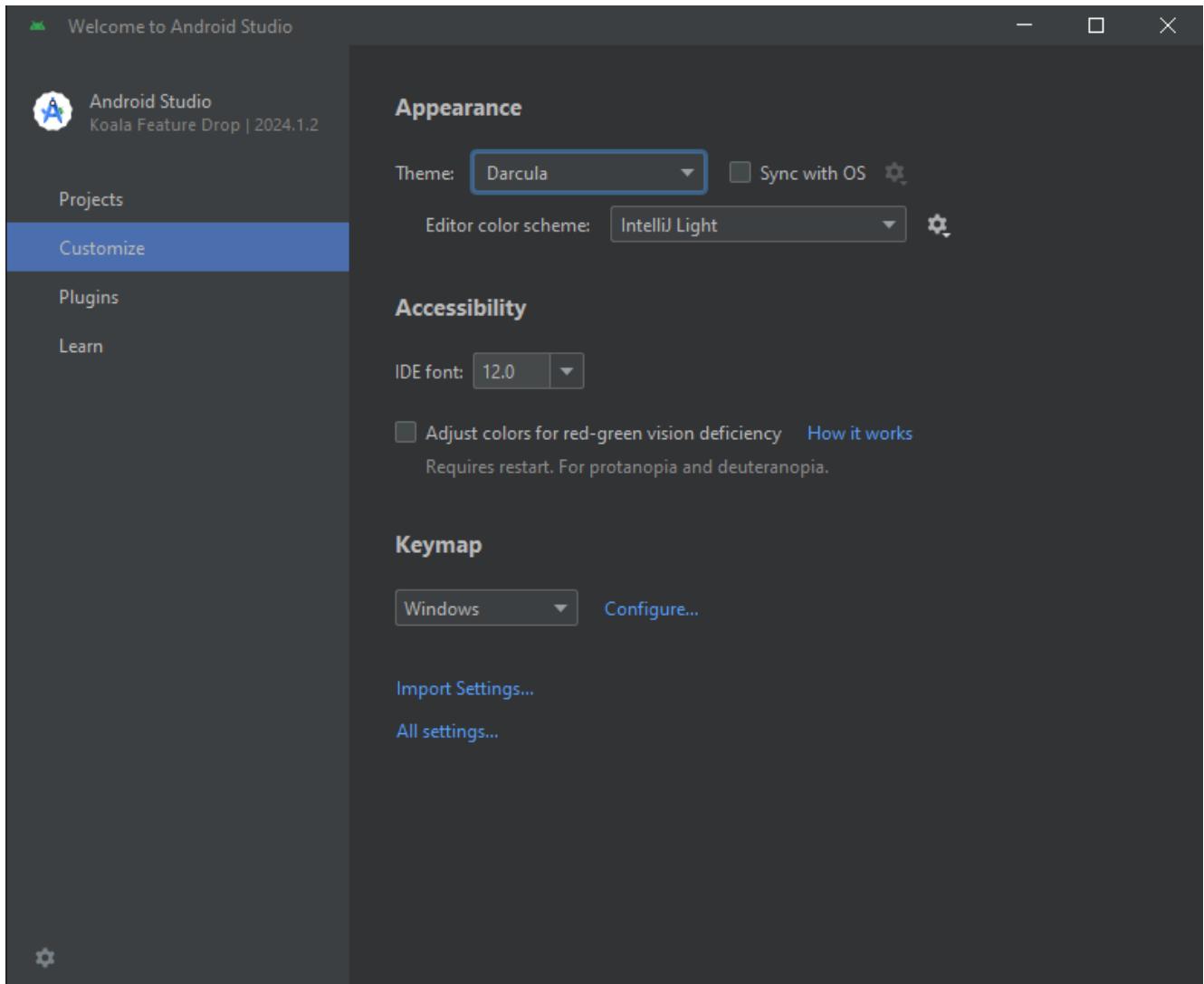


Рисунок 13 – Вкладка Customize в окне приветствия

На вкладке Plugins, см. рисунок 14, до недавнего времени можно было выбрать и установить дополнительные плагины для среды разработки, прочитать их краткое описание и т.д. Например, можно было установить Flutter прямо из этого окна. Однако с начала сентября установка плагинов в Android Studio отключена, см. рисунок 15, поэтому нужно выходить в интернет из другой страны, если вам нужен какой-то плагин.

И, наконец, если открыть вкладку Learn, изображённую на рисунке 16, можно получить доступ к обучающим материалам по работе с Android Studio. Среди прочего, можно перейти по ссылке к обучающим материалам на платформе Youtube.

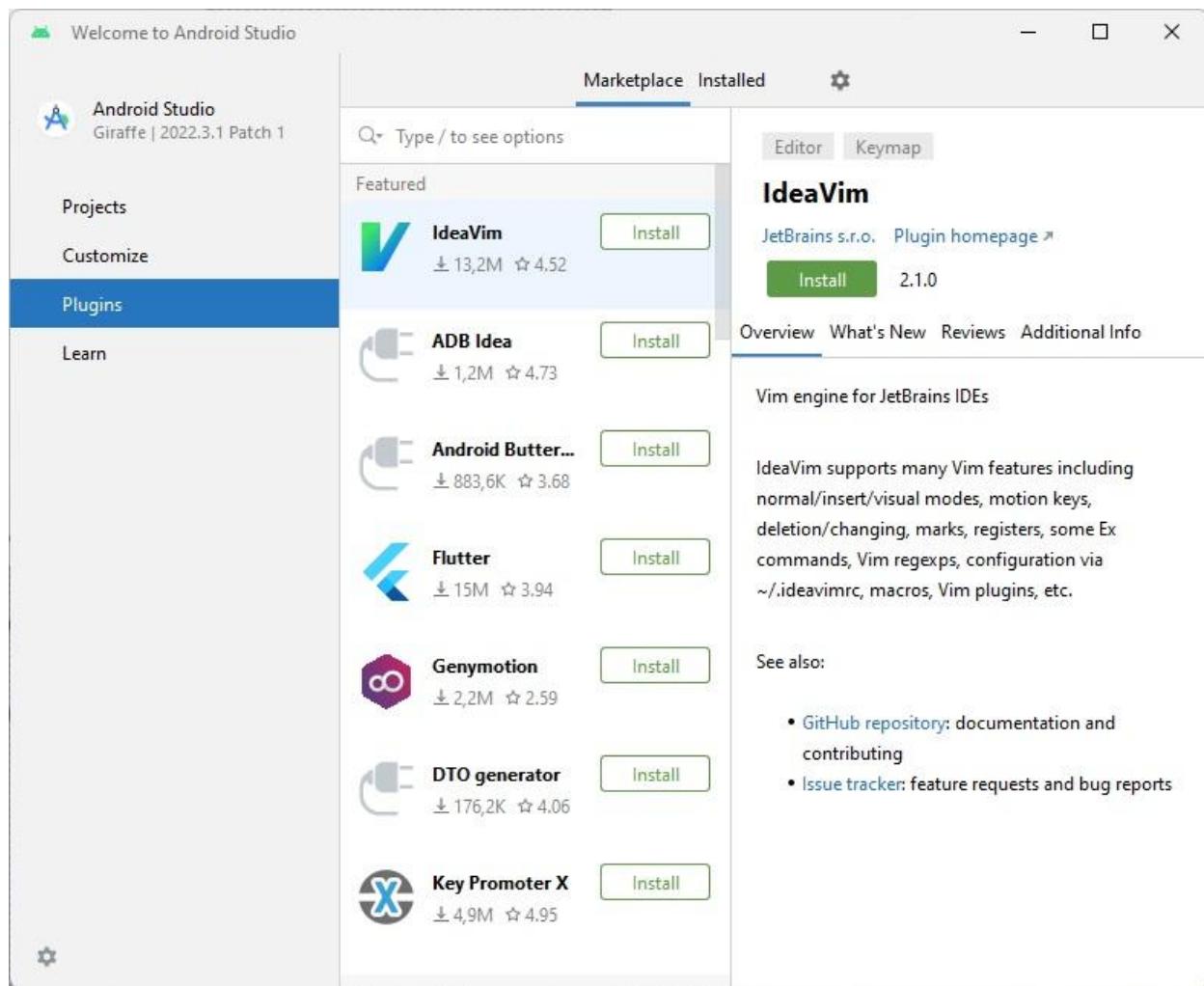


Рисунок 14 – Вкладка Plugins в окне приветствия

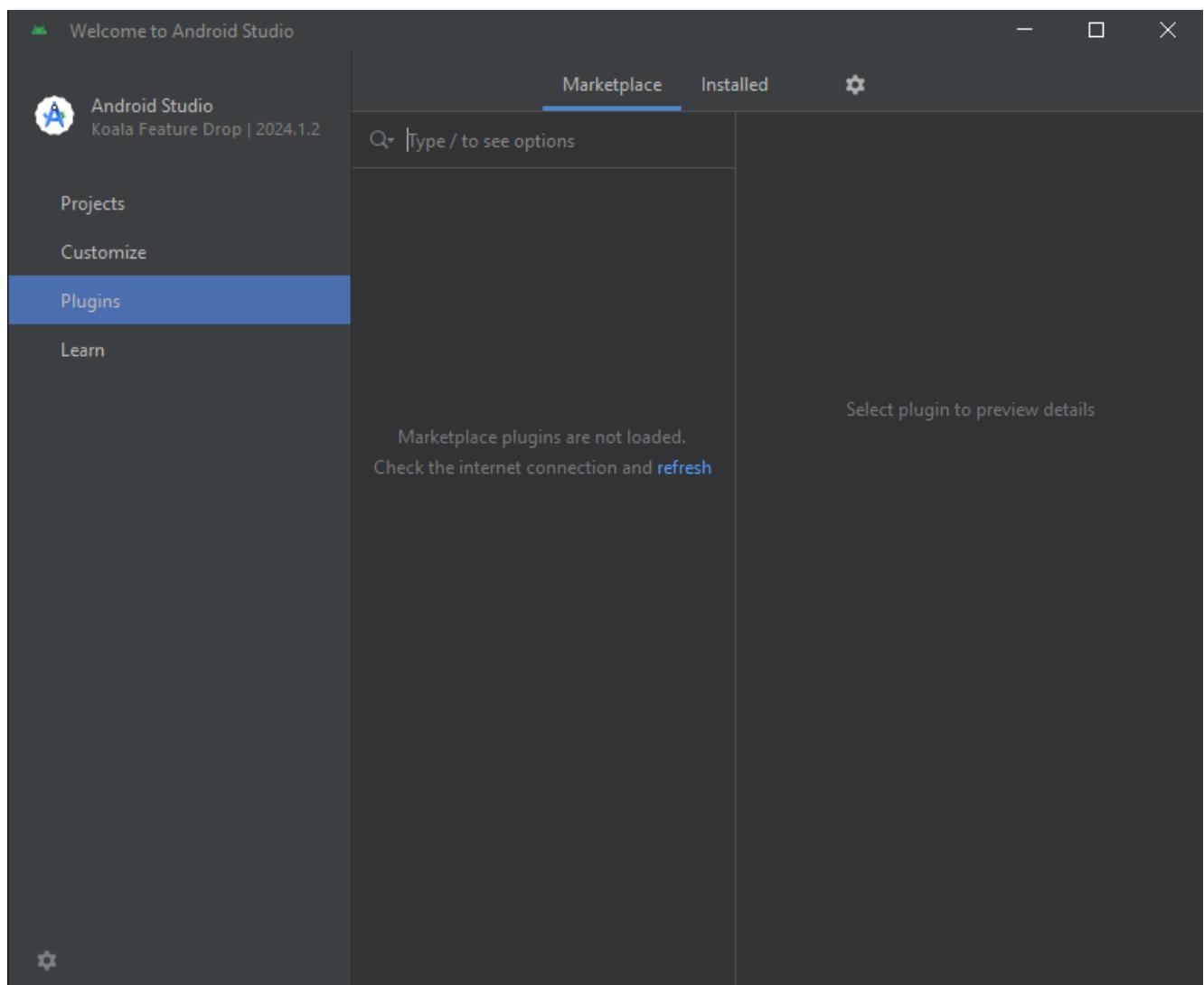


Рисунок 15 – Пустая вкладка Plugins в окне приветствия (с начала сентября 2024 года)

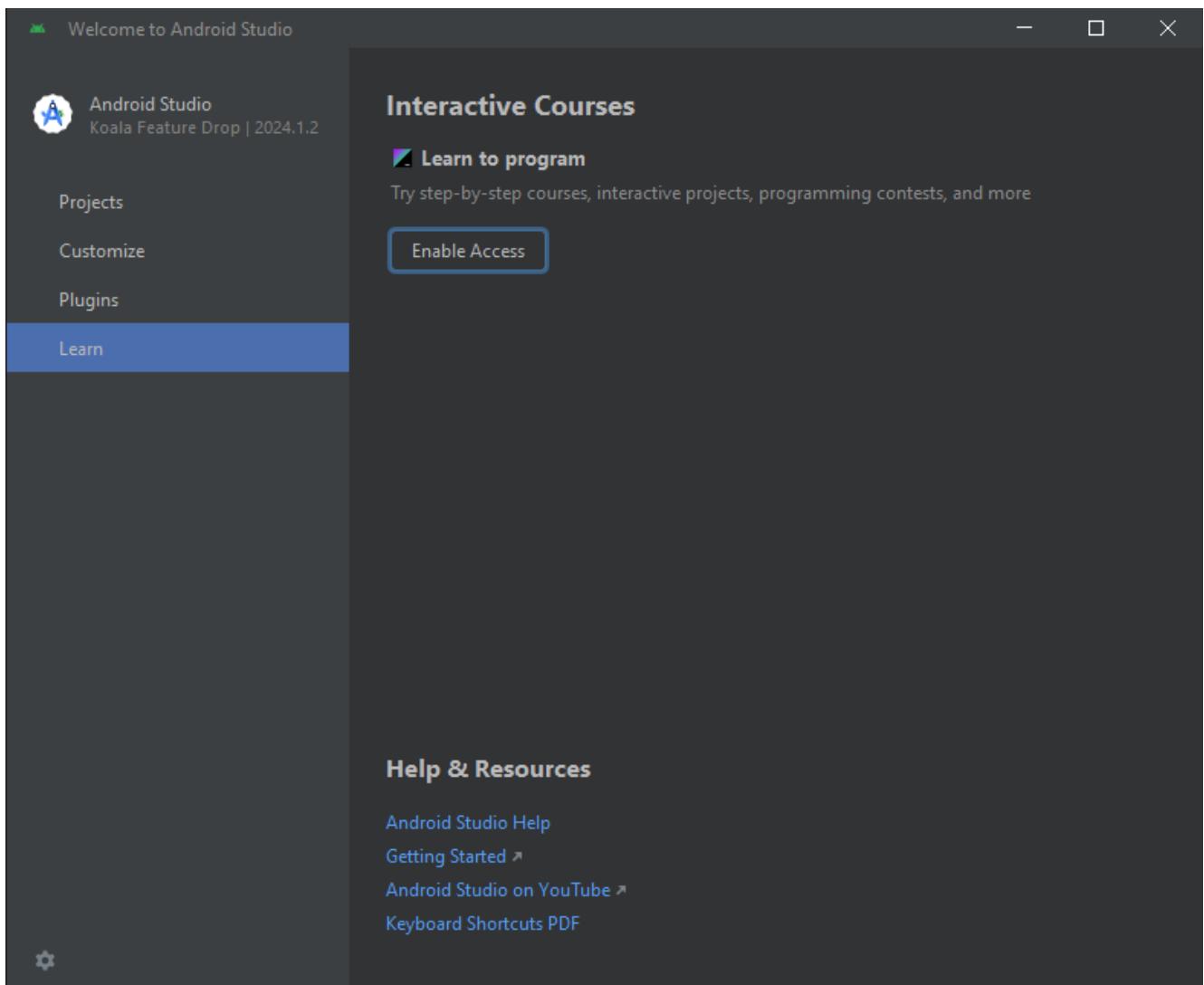


Рисунок 16 – Вкладка Learn в окне приветствия

Вернемся на вкладку Projects и нажмём на кнопку New Project. Откроется окно, изображенное на рисунке 17. В этом окне можно выбрать платформу, для которой вы хотите создать проект: традиционно это смартфон или планшет (Phone and Tablet), носимое устройство (часы, Wear OS), телевизор (Television) или автомобильное устройство (Automotive). Нас, разумеется, интересует первый выбор, стоящий по умолчанию: Phone and Tablet. Кроме этого, в данном окне можно выбрать шаблон нового приложения, которое вы собираетесь создать. В большинстве лабораторных работ, согласно реалиям последней версии Android Studio, нужно выбирать Empty Views Activity. Если выбрать шаблон Empty Activity, как было раньше (в Electric Eel, например), вы не сможете выбрать язык Java для вашего приложения. Если выбрать No Activity, язык Java вы выбрать сможете, но вам придётся самостоятельно создавать Activity (окно) для вашего приложения. Вариант Empty Views Activity содержит как выбор языка, так и уже добавленную в интерфейс Activity, на которой ничего нет, и поэтому является предпочтительным – именно его надо выбирать в наших приложениях.

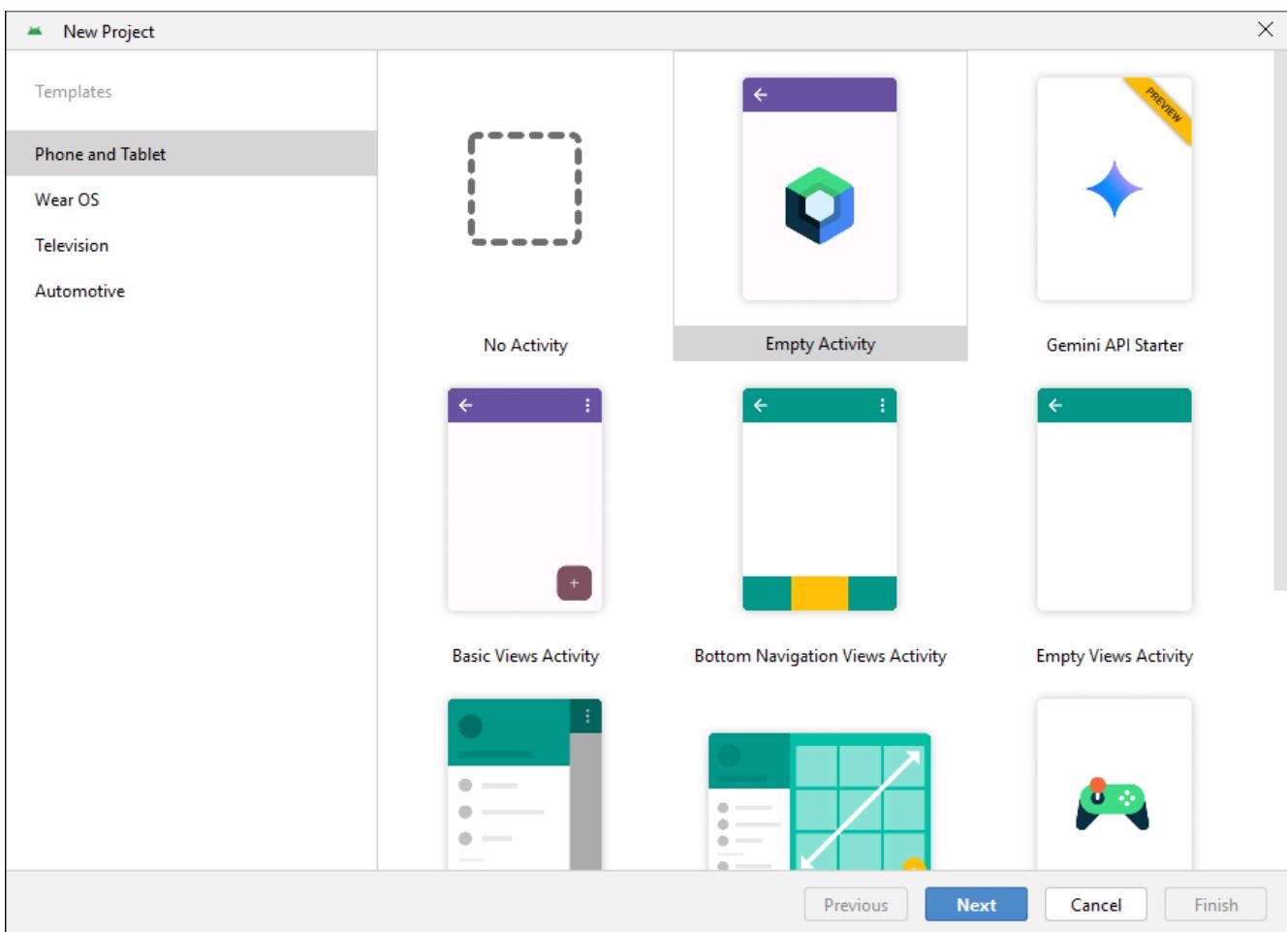


Рисунок 17 – Создание нового проекта

Нажмите Next, и придумайте название проекта, см. рисунок 18. Укажите название приложения в поле Name. Корневой пакет приложения в поле Package Name, который является идентификатором приложения, по умолчанию формируется автоматически в виде com.example.myapplication, меняется в зависимости от значения поля Name. Укажите также путь для сохранения проекта Save location (лучше оставить по умолчанию). Далее можно выбрать язык (Language) и минимальную версию SDK, на которой приложение будет работать (или по-другому – минимальную версию ОС Android). Пример заполнения показан на рисунке 18. Google рекомендует использовать как можно меньшую версию API, чтобы приложение можно было установить на большее количество устройств. Но не надо выбирать, скажем, версию ниже Oreo (Android 8), так как в ней появился другой механизм уведомлений, например. С другой стороны, по умолчанию Android Studio ставит самую последнюю версию OS (API 34 UpsideDownCake в данном случае), которая доступна менее, чем на 12.1% устройств, по сравнению, например, с API 21 (99,7% устройств). Для корректного выполнения лабораторных работ не рекомендуется использовать API меньше 21, далее выбор за вами. Если у вас есть устройство под управлением Android, было бы логично выбрать соответствующую версию API, которая стоит на вашем смартфоне. Соответствие версий API (SDK) и версий ОС Android можно посмотреть, например, здесь: <http://socialcompare.com/en/comparison/android-versions-comparison>. Например, Tiramisu – это Android 13 и, соответственно, 33 версия API/SDK. В качестве Build configuration language я рекомендую выбрать Groovy DSL, если язык Java, и Kotlin DSL, если вы выбрали язык программирования Kotlin.

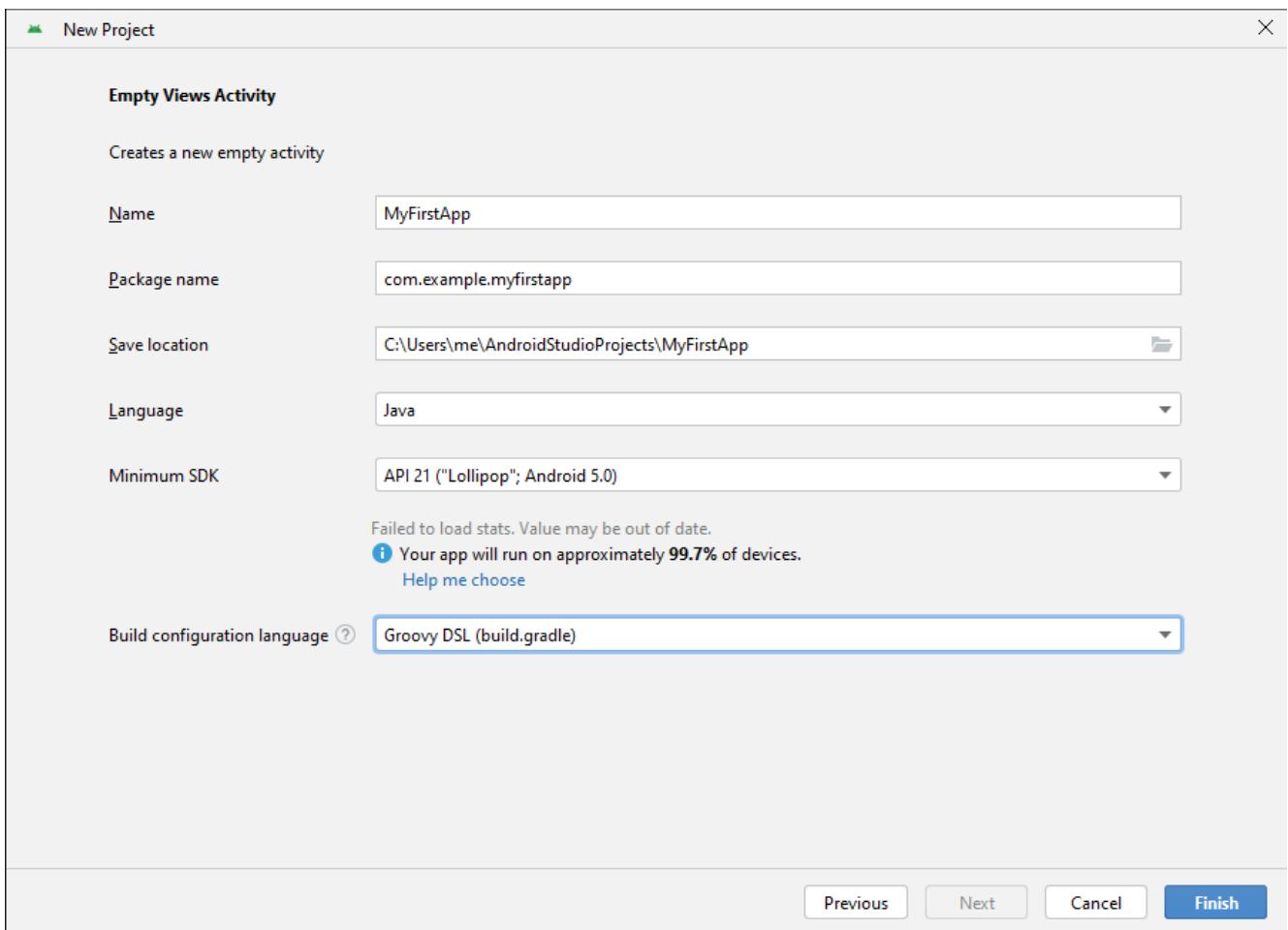


Рисунок 18 – Выбор названия проекта и других параметров

Нажмите «Finish», и через некоторое время, нужное на загрузку необходимых компонентов и построение проекта, отобразится главное окно среды разработки, которое показано на рисунке 19.

Здесь сразу же могут возникнуть ошибки. У меня, например, возникла ошибка Could not install Gradle distribution from 'https://services.gradle.org/distributions/gradle-8.7-bin.zip'. Reason: java.net.SocketException: Connection reset by peer. Это связано с тем, что версия Android Studio последняя, и, как правило, сырья. Эта ошибка лечится принудительным скачиванием пакета по указанной ссылке в любом браузере, и затем копированием этого zip-файла в папку .gradle текущего пользователя компьютера по адресу C:\Users\имя_пользователя\.gradle\wrapper\dists\gradle-8.7-bin\bhs2wmbdwecv87pi65oeuq5iu (имя последней папки уникальное, у вас оно будет другим, но в ней вы увидите файл gradle-8.7-bin.zip.part, что значит, что это файл не может докачаться через Android Studio) и затем перезапуском Sync или всей Android Studio, см. рисунок 20.

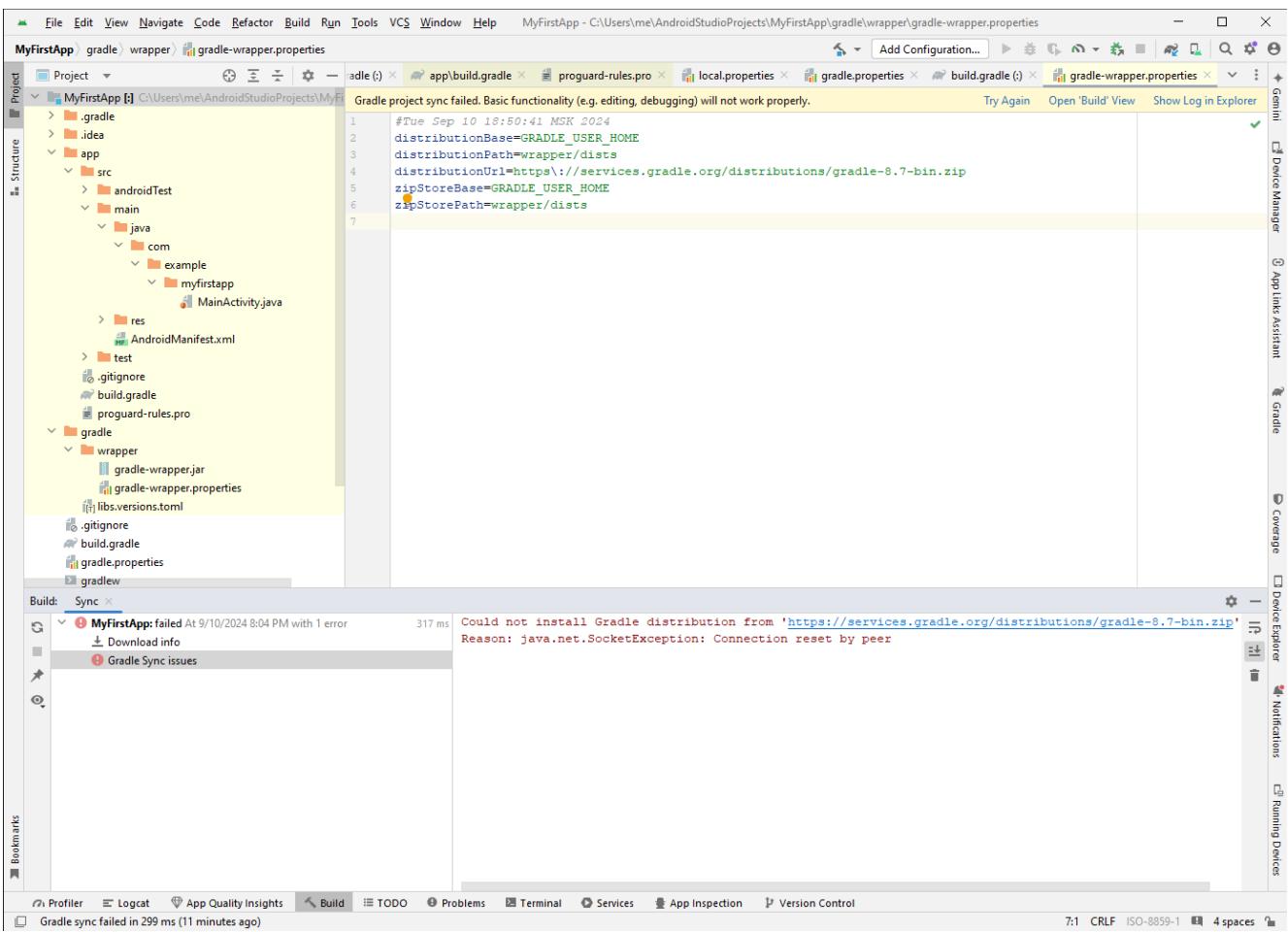


Рисунок 19 – Рабочая область Android Studio с ошибкой

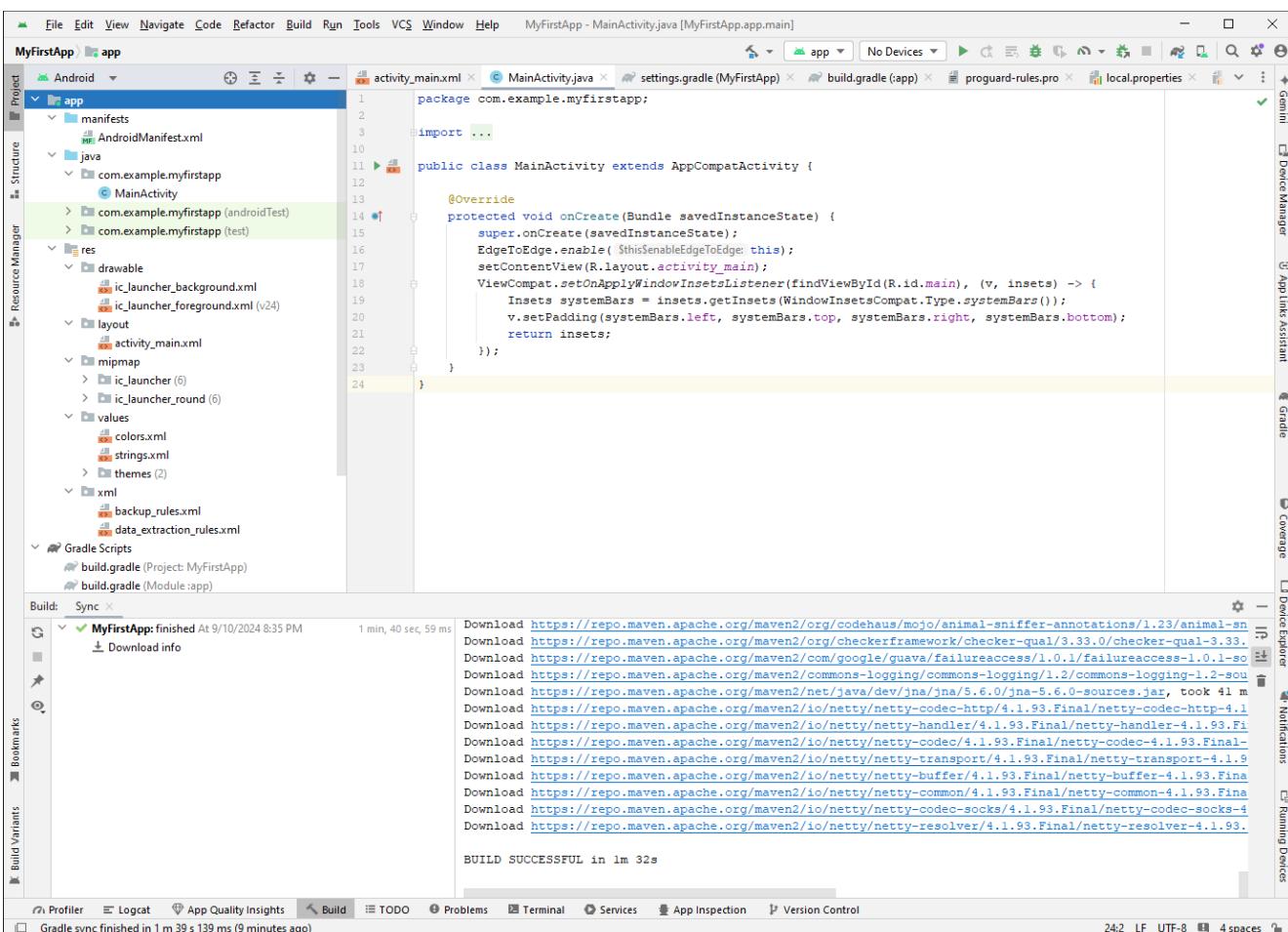


Рисунок 20 – Рабочая область Android Studio (Java)

Основные файлы и папки проекта (слева вверху, представление Android):

- **manifests/AndroidManifest.xml** - данный файл предоставляет основную информацию о программе системе и является центральным для приложения, в нём перечисляются все используемые Activity, атрибуты и разрешения, которыми обладает приложение, указывается его иконка, название и т.д.
- **java/com.example.myapplication** – содержит файлы с исходным кодом на языке Java. Именно в этой папке размещаются все классы, создаваемые в процессе разработки приложения
- **res/** – содержит структуру папок ресурсов приложения, рассмотрим некоторые из них:
 - **layout** - в данной папке содержатся xml-файлы, которые описывают внешний вид форм и их элементов, пока там находится только **activity_main.xml** – интерфейс **MainActivity**;
 - **values** - содержит xml-файлы, которые определяют простые значения таких ресурсов, как строки, числа, цвета, темы, стили, которые можно использовать в данном проекте.

Для языка Kotlin, соответственно, надо выбрать Kotlin на рисунке 18 (и для Gradle) и затем получить вид рабочей области Android Studio, показанный на рисунке 21.

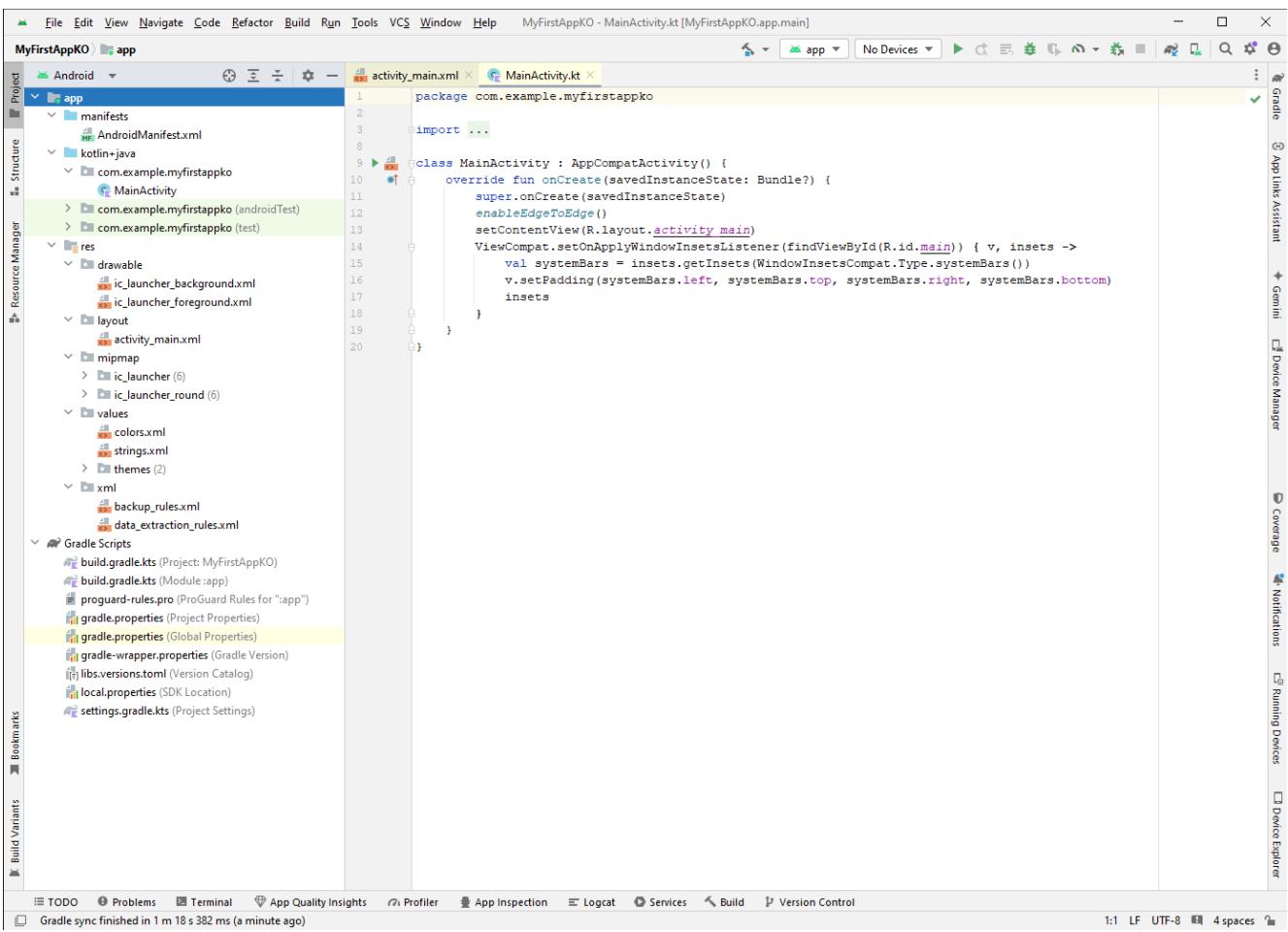


Рисунок 21 – Рабочая область Android Studio (Kotlin)

При создании проекта сразу же открывается файл `MainActivity.java/MainActivity.kt`, в котором находится исходный код главного класса проекта. Чтобы открыть интерфейс приложения, нужно два раза кликнуть на `res/layout/activity_main.xml` файл, содержащий интерфейс первого экрана (Activity) приложения, или просто кликнуть на соседнюю с `MainActivity.java/MainActivity.kt` уже открытую закладку. В зависимости от версии Android Studio для отображения интерфейса приложения может понадобиться дополнительно скачать и установить различные компоненты. Поскольку ранее мы выбрали Empty Views Activity, в окне интерфейса приложения ничего нет, кроме `TextView` с надписью «Hello World!», который лежит в контейнере `ConstraintLayout`. Добавлять интерфейсные элементы можно двумя способами: в графическом (справа вверху кнопка «Design») и текстовом (справа вверху кнопка «Code») представлениях (можно и совмещать – кнопка «Split»). Интерфейсные элементы расположены в контейнере `Palette` слева от отображения интерфейса приложения. В контейнере элементы интерфейса сгруппированы по категориям, например, Common (наиболее используемые), Text, Button и т.д. В другом контейнере ниже, Component Tree, расположено дерево всех элементов интерфейса, которые есть в данный момент. На экране уже есть компонент `TextView` со значением «Hello World!». Если его не видно, значит, у вас возникли проблемы с рендерингом окна дизайна. Справа вверху есть воскликательный знак в красном кружке, если нажать его, вы увидите суть проблемы.

Нажмите один раз левой кнопкой мыши по компоненту `TextView`, и справа появится окно, в котором можно задать текст и id для данного элемента. Поле «text» уже заполнено фразой «Hello world!». Если нажать на крошечную кнопку справа от этого поля, откроется окно ресурсов, показанное на рисунке 22. Дело в том, что в файле `res/values/strings.xml` хранятся все текстовые данные приложения, что позволяет довольно быстро изменять названия различных элементов, а также с легкостью создавать локализацию под различные языки. В это окно выводятся значения ресурсов, в том числе и из этого файла. Если нажать кнопку + и выбрать `String Value` слева вверху, можно с помощью

диалоговых окон создать новый строковый ресурс (рисунок 23). После нажатия на кнопку ОК ссылка на ресурс автоматически добавляется в поле text элемента TextView.

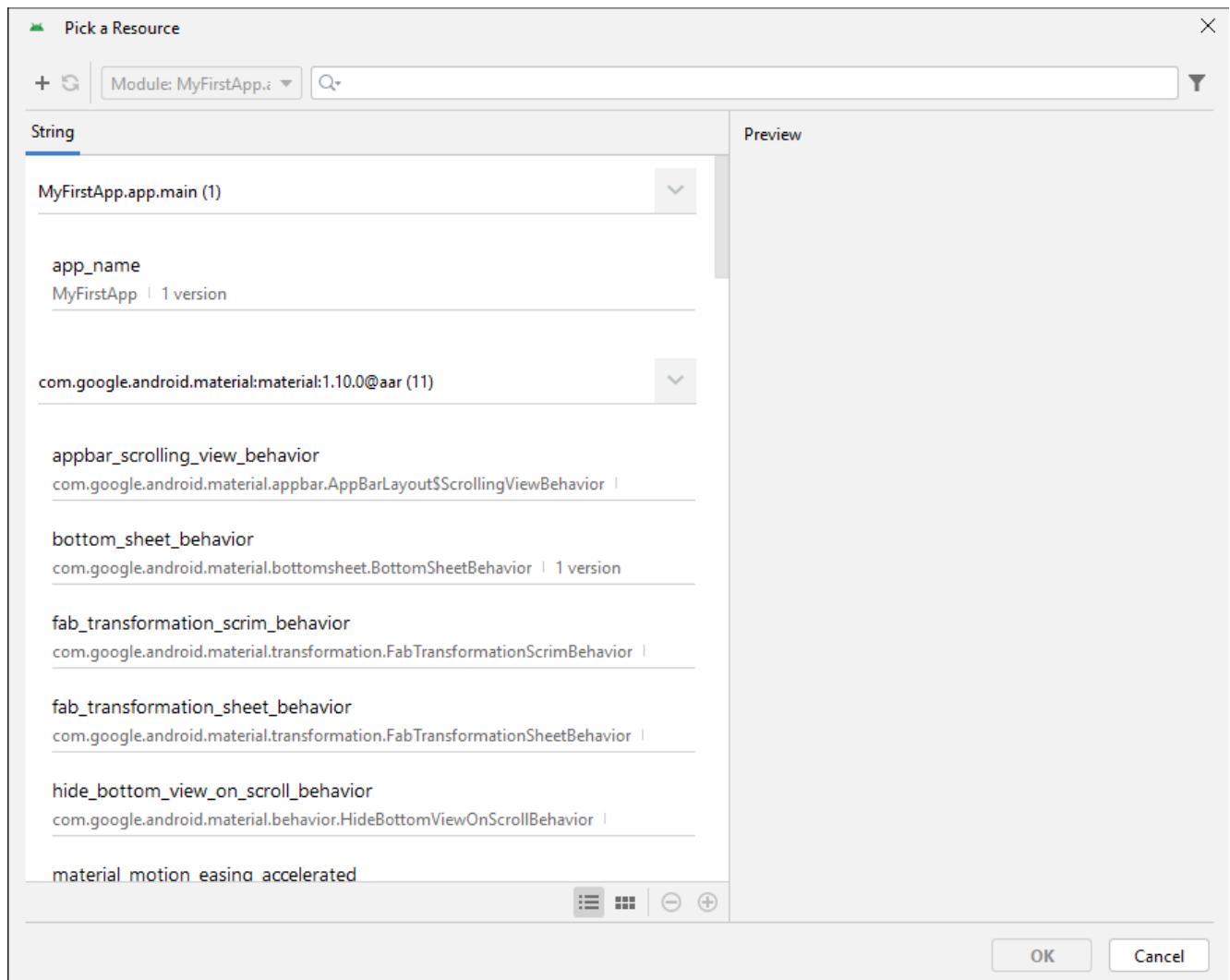


Рисунок 22 – Окно выбора ресурсов

Второй способ создания строкового ресурса – это редактирование непосредственно файла strings.xml. Для этого в файл res/values/strings.xml добавьте новую надпись с именем «fio», для этого необходимо написать следующее:

```
<string name="fio">Иванов Иван Иванович, гр. БПИ 171</string>
```

После того, как вы создали новую строковую переменную с помощью редактирования файла strings.xml, перейдите в файл activity_main.xml и измените название надписи на «@string/fio», а в поле id вставьте «fio» и нажмите Enter.

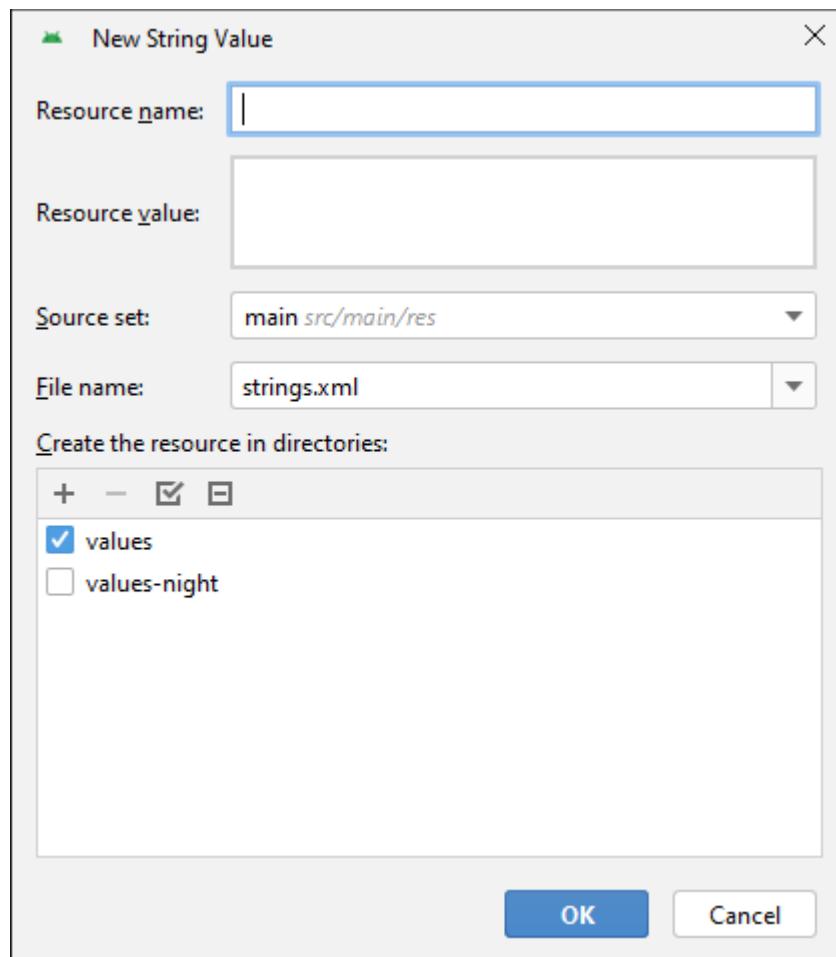


Рисунок 23 – Окно создания нового строкового ресурса

Если всё сделано правильно, то теперь на экране смартфона должны отобразиться ФИО и группа. Осталось запустить приложение на эмуляторе, но для начала эмулятор необходимо создать. Нажмите в главном меню наверху Tools -> Device Manager, затем в появившейся панели справа нажмите кнопку «+» сверху, и выберите пункт «Create Virtual Device», и в открывшемся окне (рисунок 24) выберите модель телефона (Phone), которую хотите эмулировать, например, Pixel 7, нажмите «Next», выберите образ определённой версии Android и скачайте его, нажав на Download (рисунок 25). Опыт показывает, что чем меньше версия Android, тем быстрее работает эмулятор; но это касается в основном ноутбуков со слабыми процессорами и графикой. Если у вас мощный компьютер с хорошим процессором и видеокартой, можно выбрать последние версии Android: образы из вкладки Recommended или тот, который соответствует вашему реальному устройству под управлением Android. Закладка x86 Images служит для счастливых обладателей компьютеров на базе процессоров Intel – для этих эмуляторов существует встроенное средство для десятикратного ускорения работы эмулятора (Intel HAXM – Hardware Accelerated Execution Manager). Однако, начиная с версии 4.1 Android Studio, даже если у вас процессор AMD, работа эмулятора будет ускорена за счёт ресурсов видеокарты, поэтому не важно, на базе какого процессора построен ваш компьютер или ноутбук. Вариант образа, у которого в скобках написано Google APIs, служит для запуска приложений, использующих Google Play Services; нам это пока не требуется, но вообще – потребуется, так что в идеале лучше выбрать эмулятор с Google APIs и/или Google Play (значком). Если у эмулятора есть значок Google Play (Store), значит, на эмуляторе можно будет использовать этот магазин для установки новых приложений, и на эмуляторе установлены Google Play Services.

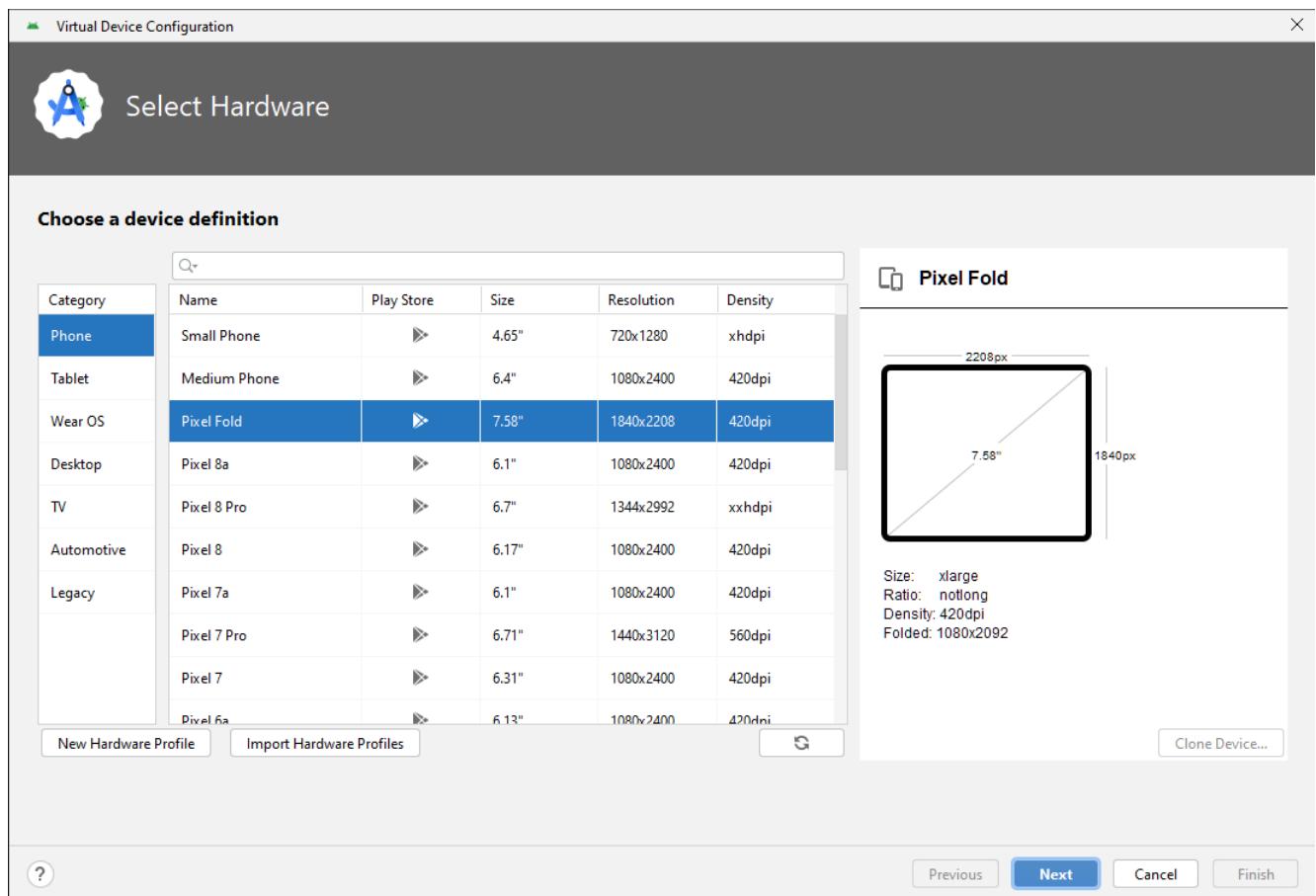


Рисунок 24 – Окно создания эмулятора в Android Studio

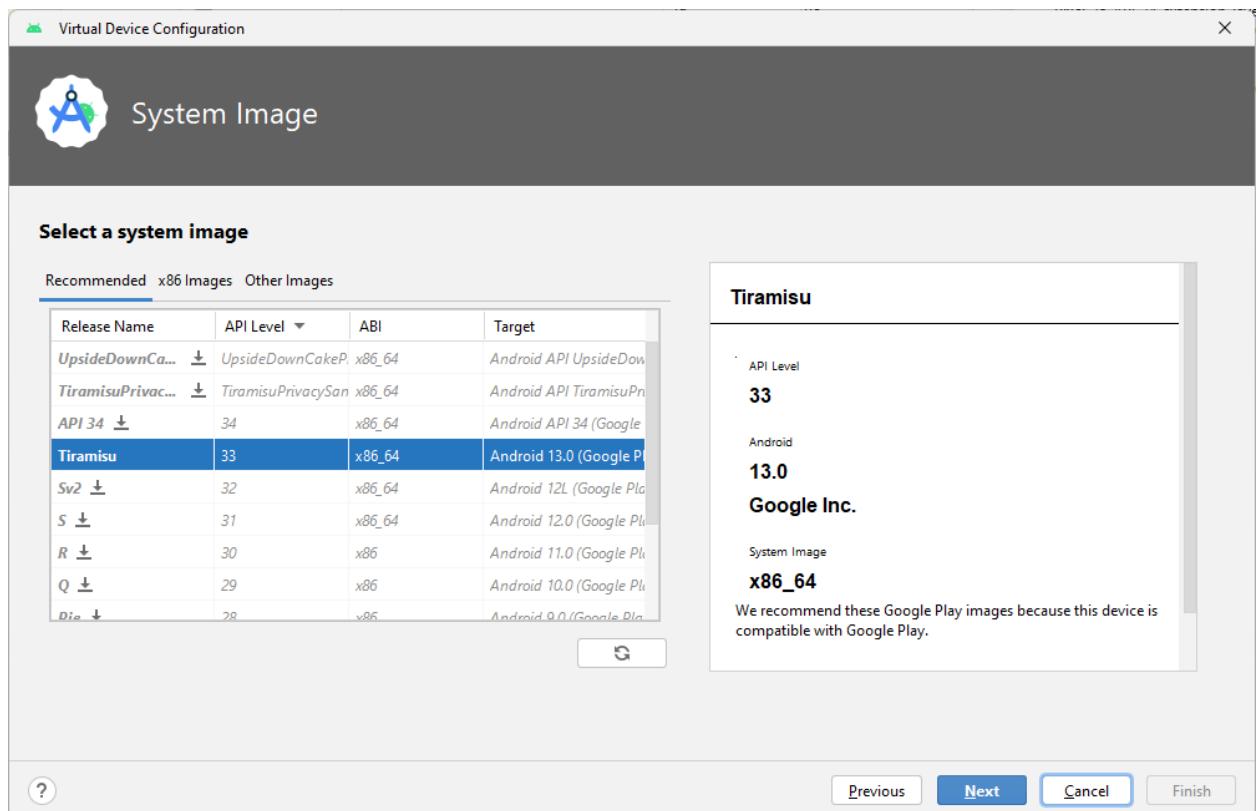


Рисунок 25 – Окно выбора (версии ОС) и скачивания образа эмулятора

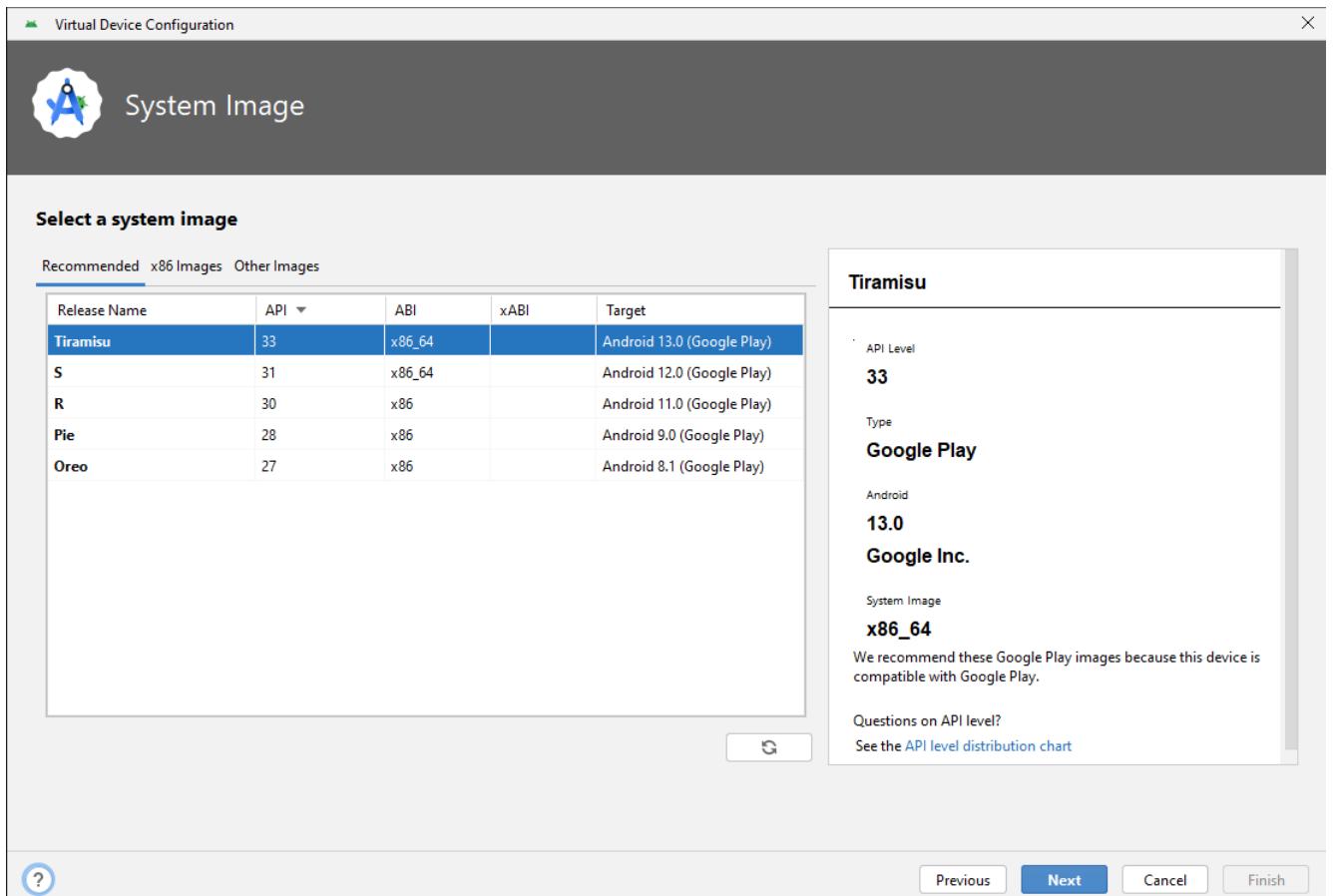


Рисунок 26 – Окно выбора (версии ОС) после ограничений сентября 2024 года

Возможно, вы окажетесь в ситуации, которая изображена на рисунке 26 – но ничего страшного, как уже говорилось выше, можно скачать SDK из другой страны, а того, что вы видите в этом случае более чем достаточно, чтобы успешно пройти этот курс.

Чтобы скачать образ, надо нажать на стрелку в первой колонке (см. рисунок 25). После скачивания образа устройства и нажатия кнопки **Finish** в окне скачивания выберите из списка скачанный образ, который теперь выделяется жирным шрифтом (см. рисунок 25). Нажмите «**Next**», в следующем окне нажмите кнопку «**Show Advanced Settings**» в левом нижнем углу и убедитесь, что стоят галочки «**Enable keyboard input**» и «**Enable device frame**» (рисунок 27). Остальное менять нет необходимости, нажмите «**Finish**» и сверните панель **Device Manager** кнопкой «**–**» (Hide) справа вверху панели. Осталось выбрать эмулятор из списка слева от зелёной кнопки «**Run ‘app’**» вверху на панели инструментов, нажать эту кнопку «**Run ‘app’**» (или - Shift+F10 на клавиатуре) и ждать загрузки эмулятора. После того, как эмулятор загрузится, разблокируйте экран смартфона (или нажмите кнопку OK), и сразу же должно запуститься приложение, как показано на рисунке 28. Справа располагается панель управления эмулятором, где можно менять его ориентацию в пространстве, управлять звуком, нажимать аппаратные кнопки, вводить координаты локации и многое другое. Если возникли проблемы с константой `compileSdk 33`, она находится в файле `build.gradle` (Module: app), там её можно поменять на другое значение и синхронизировать проект.

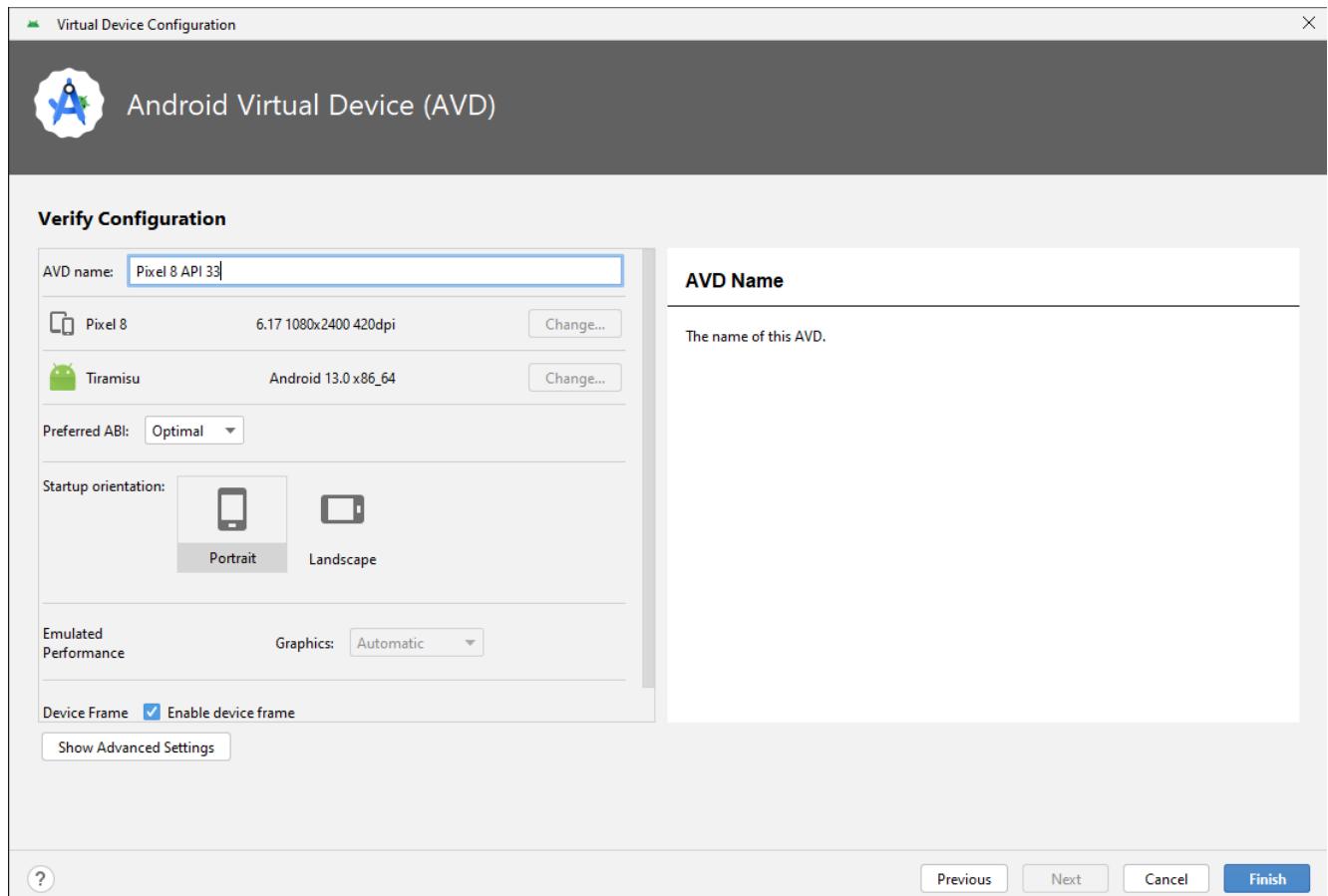


Рисунок 27 – Окно дополнительных свойств эмулятора

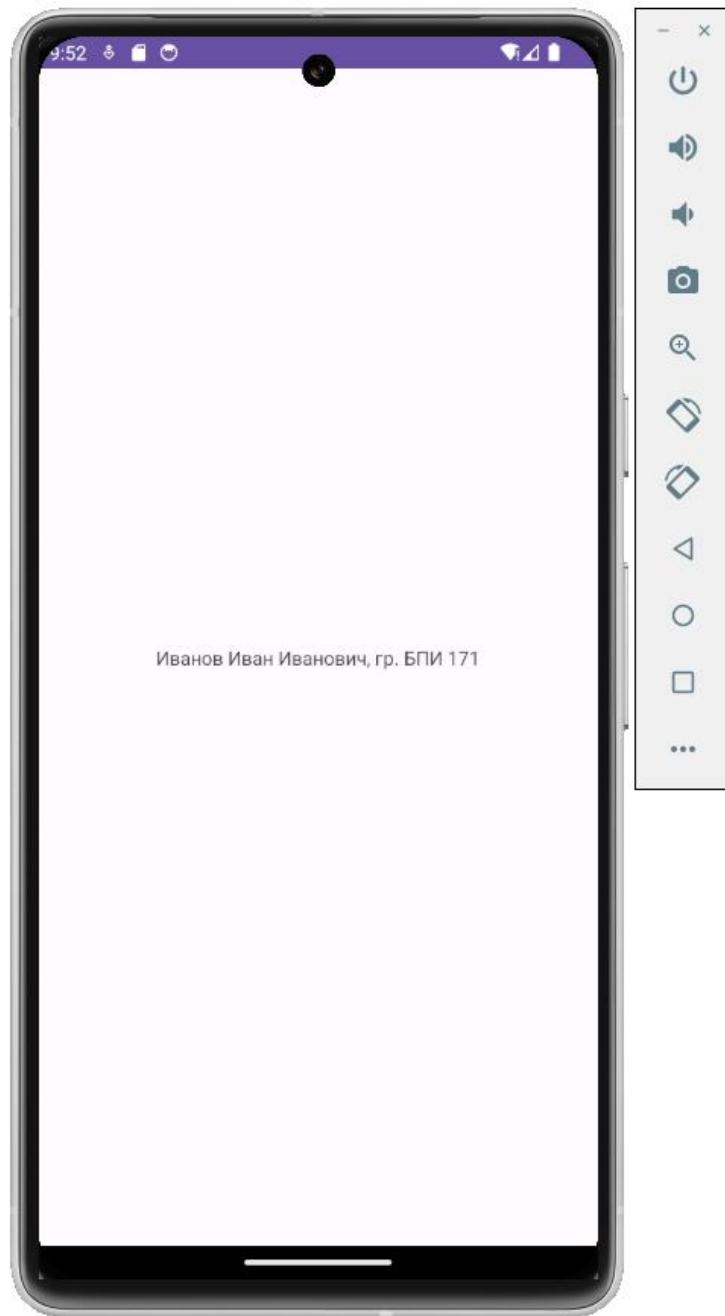


Рисунок 28 – Эмулятор Android с запущенным приложением

Теперь необходимо проверить установленные компоненты SDK. Для этого перейдите Tools -> SDK Manager и сверьтесь со следующим списком. Если каких-то компонентов не хватает, то следует их установить.

- SDK Tools
 - Android SDK Build-Tools 34 (последней версии)
 - Android SDK Platform-tools
 - Android Emulator
 - Android Emulator hypervisor driver (installer)
 - Google Play services
 - Intel x86 Emulator Accelerator (HAXM installer)
 - Google USB Driver

- Google Play Instant Development SDK (если хотите использовать Android Instant Apps)
- SDK Platforms
- Android API 34 (или той версии, которая у вас на реальном устройстве или на эмуляторе)

В файле `MainActivity.java` в этой лабораторной мы ничего вообще не делали, однако, я удалил оттуда всё лишнее, таким образом, для работы приложения достаточно 2 строк кода в методе `onCreate()`:

```
public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}
```

То же самое касается файла `MainActivity.kt`:

```
class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
    }
}
```