****

**Министерство образования и науки Российской Федерации**

**Федеральное Государственное Бюджетное Образовательное Учреждение**

**Высшего Образования**

**«Московский Государственный Технологический Университет «СТАНКИН»**

**Методические указания по лабораторной работе №1**

**Дисциплина: «Разработка приложений для встраиваемых и мобильных устройств»**

Москва  
2019

# Установка и настройка интегрированной среды разработки Android Studio.

1. Переходим на сайт <https://developer.android.com/studio> и скачиваем последнюю версию IDE.

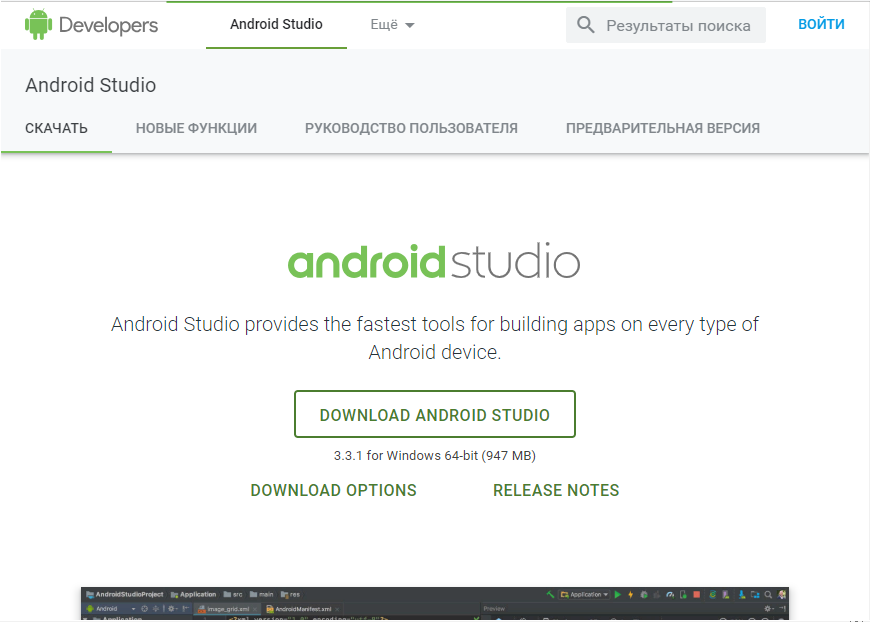


Рис 1. Сайт Android Studio.

1. Выберите установочный файл Android Studio и следуйте дальнейшим инструкциям для установки IDE

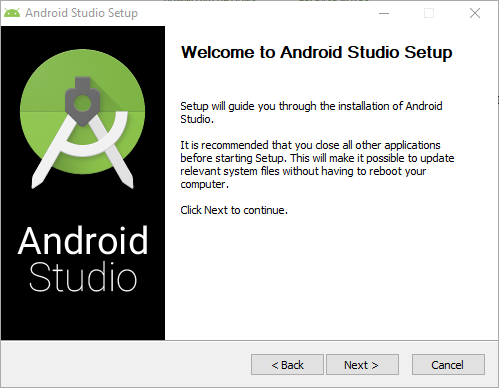


Рис 2. Окно установки Android Studio.

Обязательно выберите установку Android Virtual Device (сокр. AVD) так как этот компонент необходим для эмуляции работы Android устройства при отсутствии физического.

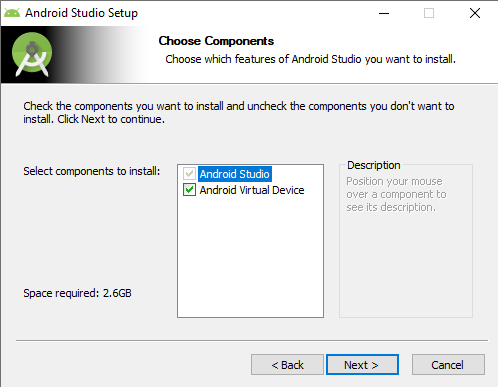


Рис 3. Окно 2 установки Android Studio

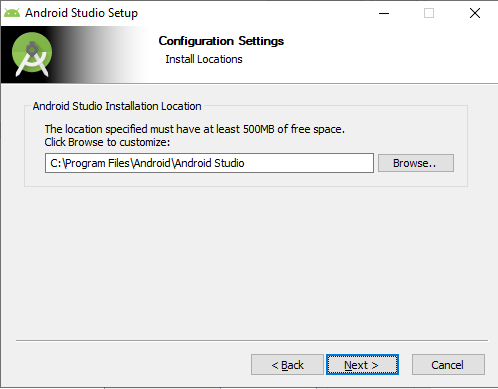


Рис 4. Окно 3 установки Android Studio.

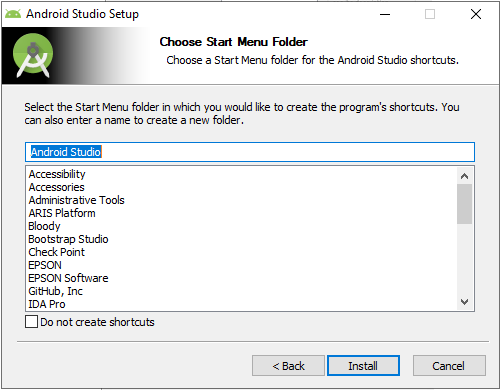


Рис 5. Окно 4 установки Android Studio.

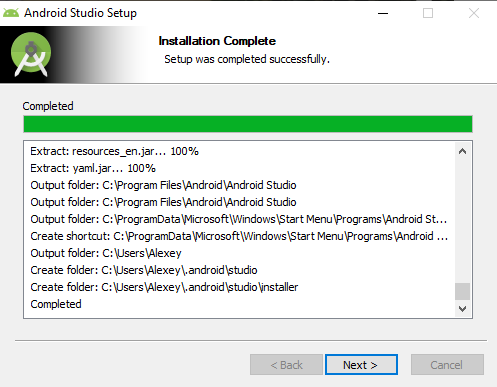


Рис 6. Окно 5 установки Android Studio.

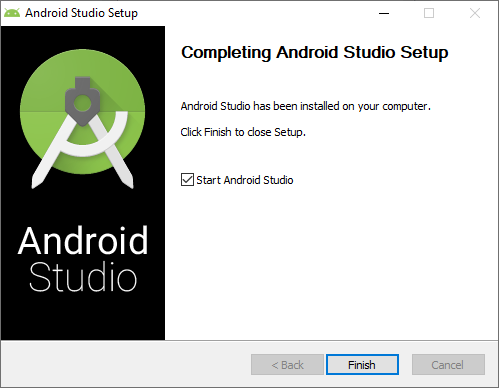


Рис 7. Окончательное окно установки Android Studio.

После установки IDE запускаем ее с иконки в меню «Пуск» и переходим к настройке среды.

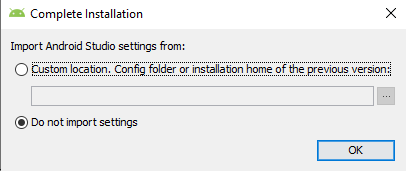


Рис 8. Окно первого запуска IDE

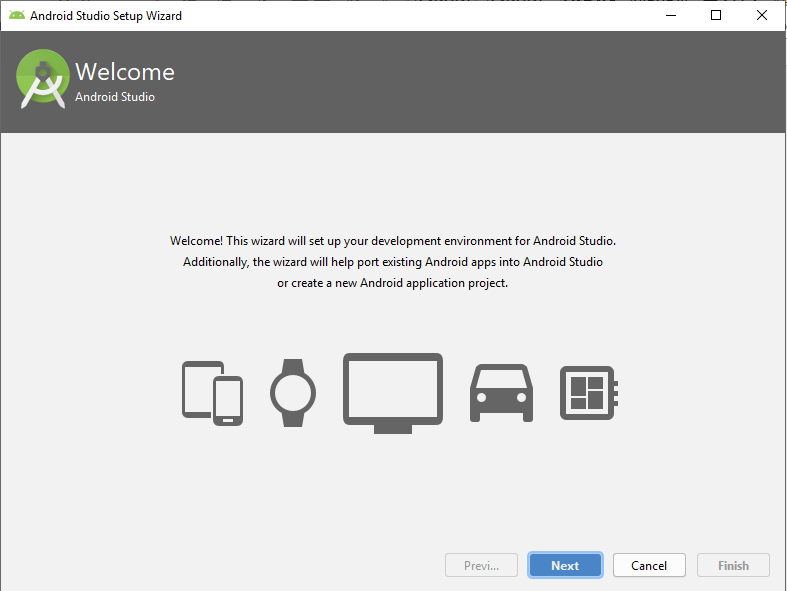


Рис 9. Окно настройки IDE.

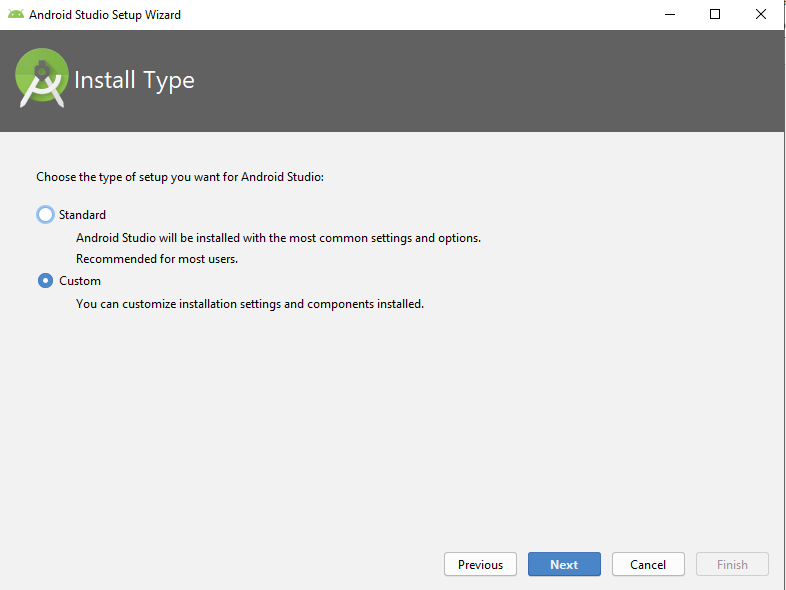


Рис 10. Окно 2 настройки IDE.

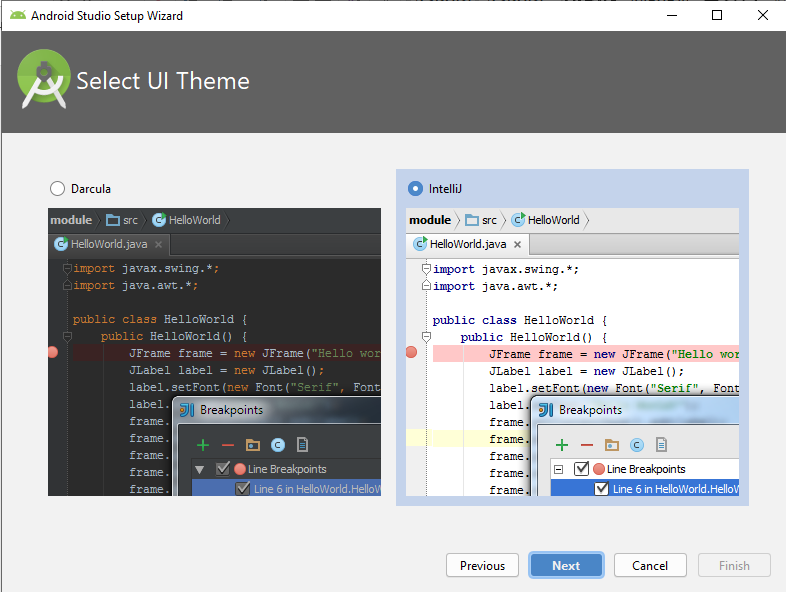


Рис 11. Окно 3 настройки IDE.

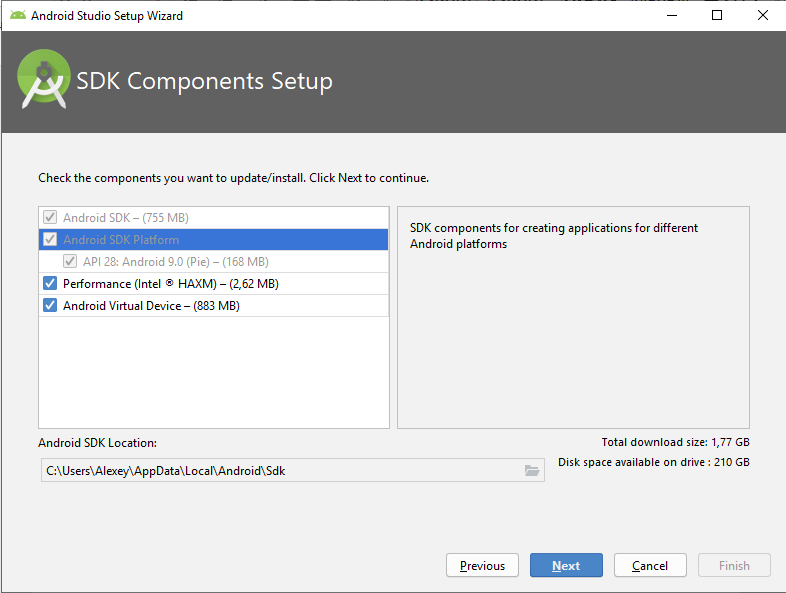


Рис 12. Окно 4 настройки IDE

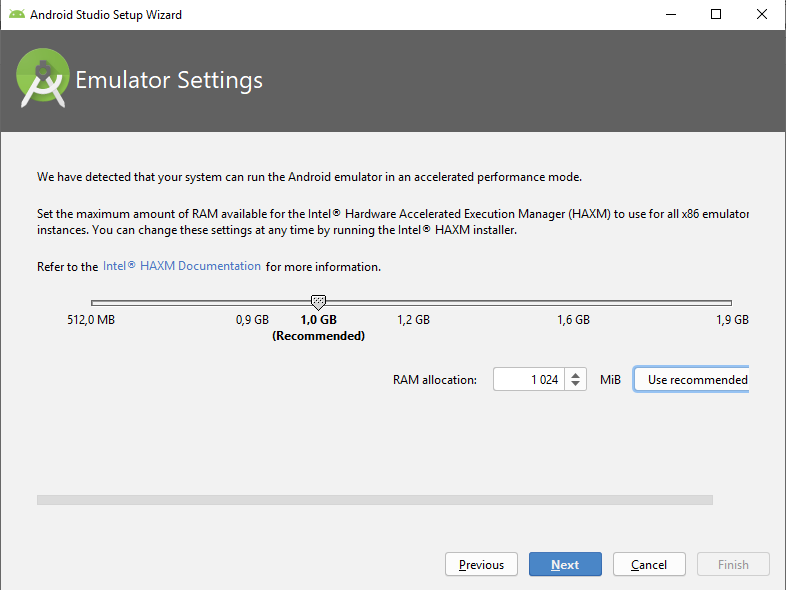


Рис 13. Окно 5 настройки IDE

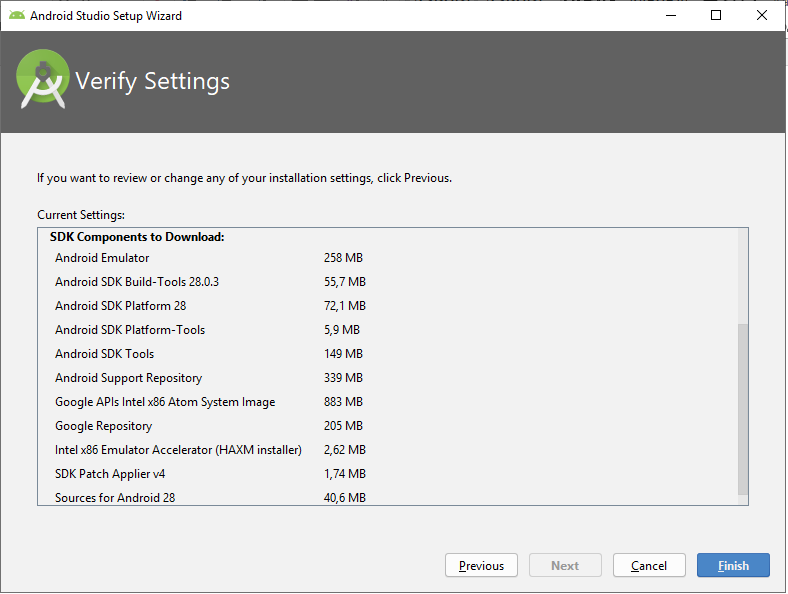


Рис 14. Окно 6 настройки IDE.

Далее начнется загрузка необходимых компонентов.

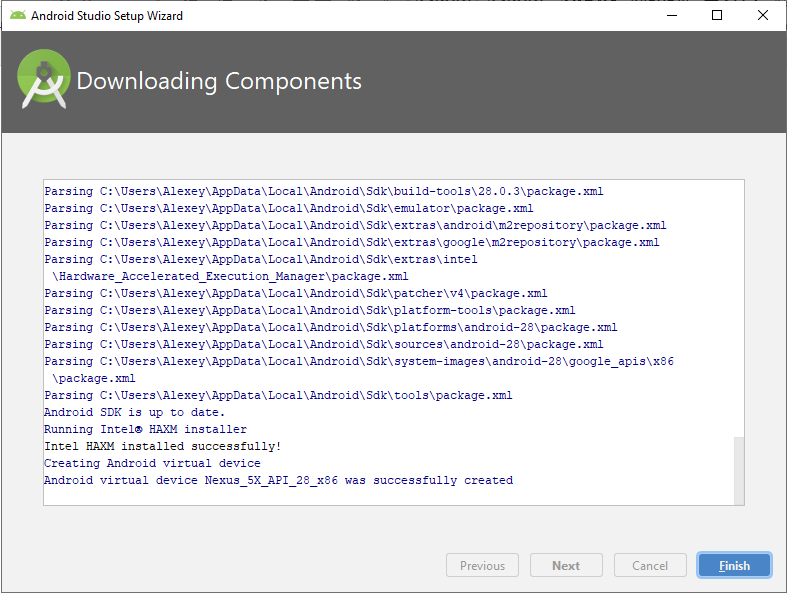


Рис 15. Окно окончания настройки IDE

Далее необходимо установить дополнительные SDK и драйверы для USB и можно приступать к разработке.

После завершения загрузки появится стартовое окно запуска Android Studio (Рис. 16). Необходимо выбрать из выпадающего меню “Configure” подменю SDK Manager после чего вы попадете в окно настроек Android SDK (Рис. 17).

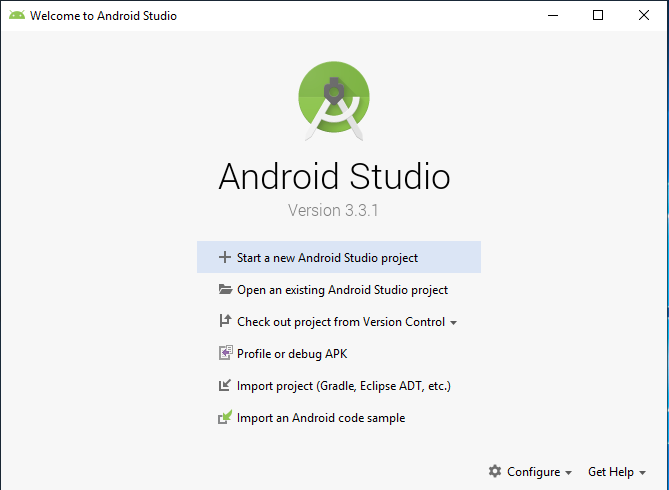


Рис 16. Стартовое окно IDE.

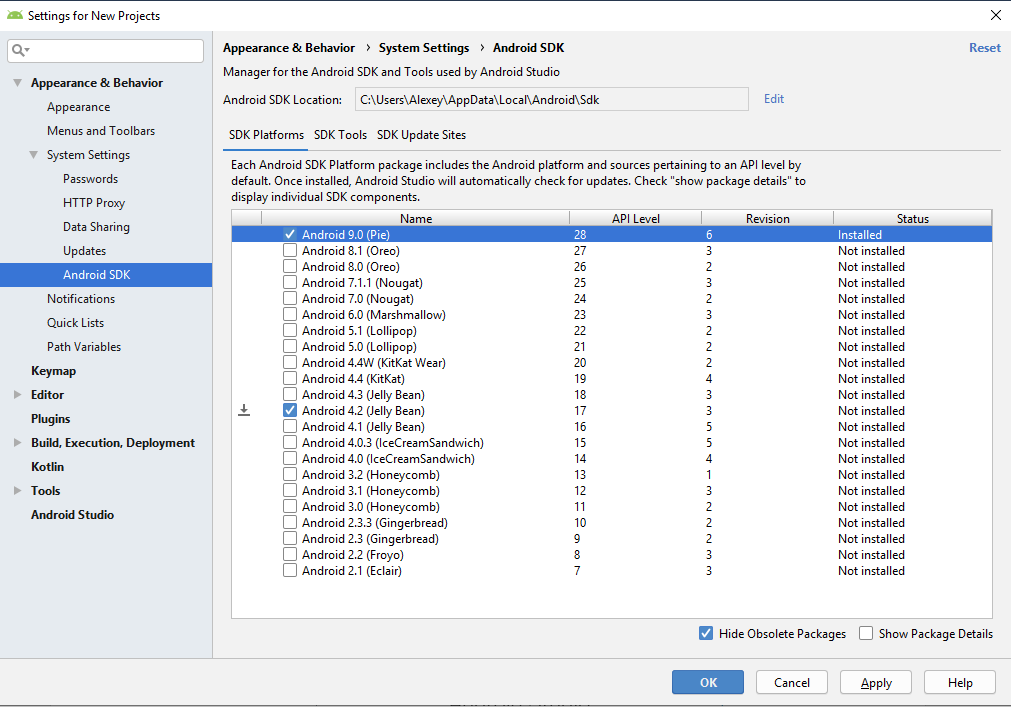


Рис 17. Окно настроек Android SDK.

На вкладке “SDK Platforms” нужно выбрать необходимые версии SDK. Изначально будет установлен последний SDK для Android 9.0. Важно подметить то, что при выборе версии SDK необходимо ориентироваться на минимальную версию Android, которую вы собираетесь поддерживать. То есть если выбрать SDK для Android 4.2, то все последующие версии Android смогут запустить ваше приложения, а все, которые ниже 4.2 нет (P.S. Формально у Android SDK работает обратная совместимость поэтому если не использовать функции и методы, которые появились только с выбранной версии, то можно попытаться их запустить и на устройствах с версией ниже выбранной, но лучше ориентироваться на устройства с уровнем API выше, чем у выбранного SDK).

Далее перейдите на вкладку “SDK Tools” и отметьте “Google USB Driver” для скачивания и нажмите “Apply” для продолжения установки (Рис 18.)

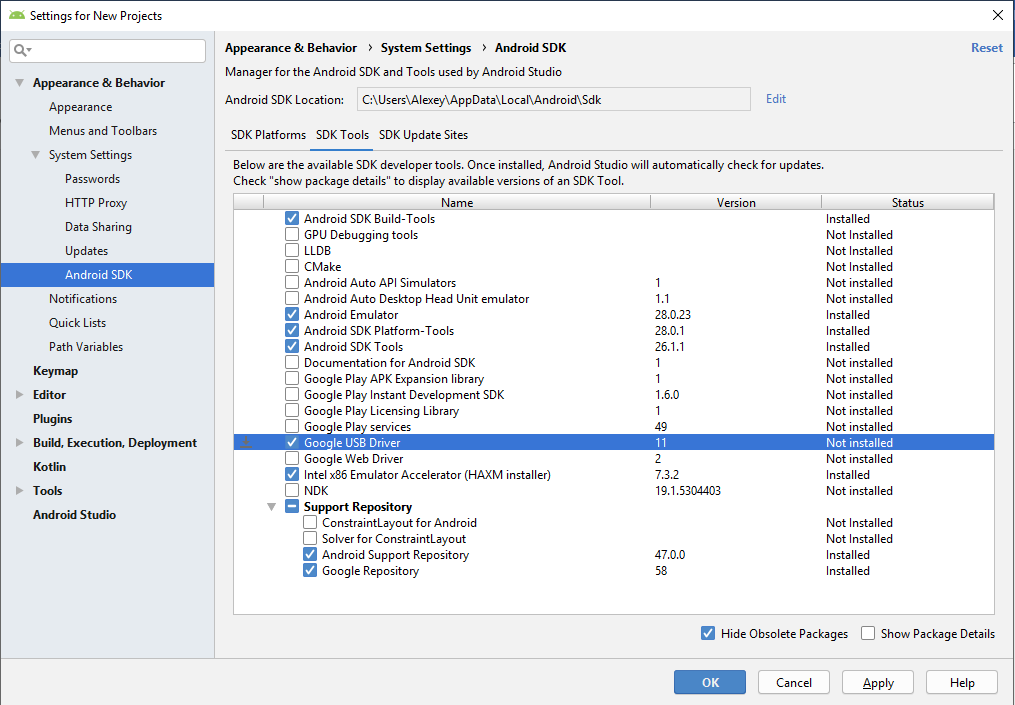


Рис 18. Окно настроек Android SDK

Далее необходимо подтвердить начало загрузки кнопкой «ОК», принять лицензионное соглашение и дождаться загрузки выбранных компонентов. После окончания загрузки нажимаем «Finish» и закрываем окно настроек SDK.

На данном этапе первоначальная настройка среды закончена и можно приступить к созданию первого приложения.

# Создание шаблона приложения

Вернувшись на стартовое окно Android Studio нажмите на кнопку “Start a new Android Studio project” для создания нового проекта. В следующем окне на вкладке “Phone and Tablet” выберите “Empty Activity” и нажмите “Next” (Рис. 19.).

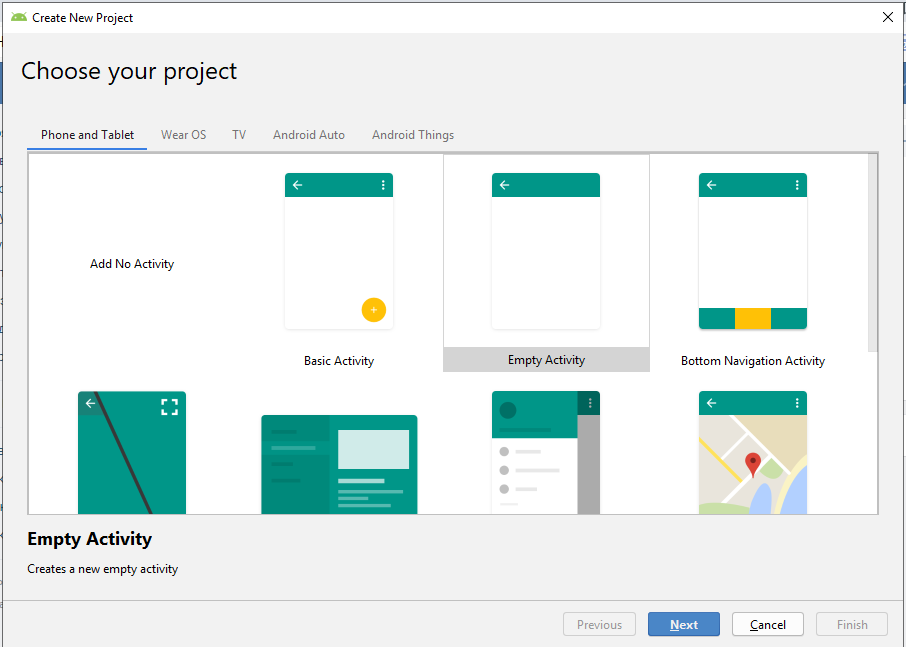


Рис. 19. Окно создания проекта

Далее необходимо настроить основные параметры проекта (Рис. 20)

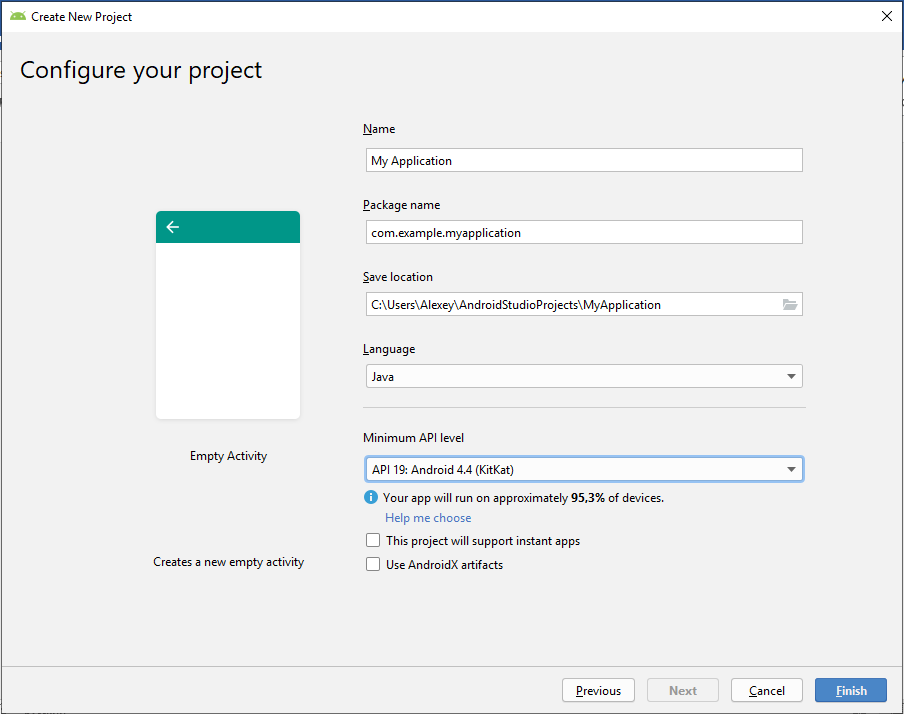


Рис. 20. Окно создания проекта

**Name** – Имя вашего приложения (напр. Lab 1. Необходимо вводить имя латинскими буквами).

**Package name** – Уникальный идентификатор вашего приложения на устройстве и в Google Play Market (обычно имеет формат вида: «домен сайта». «название организации». «имя приложения» напр. ***com.adobe.photoshop (Adobe Photoshop)***).

**Save location** – Место хранения исходного кода вашего приложения (Лучше избегать кириллицы в пути).

**Language** – Язык на котором будет разрабатываться приложение (В нашем случае выбрать Java).

**Minimum API level** – минимальный уровень API, который будет поддерживать ваше приложение (Выберите то, который вы скачали при загрузке дополнительных пакетов).

**This project will support instant apps** – данная галочка отвечает за поддержку такого типа приложений, которые не требуют установку на устройство (что то вроде демоверсии, пока не отмечать).

**Use AndroidX artifacts –** включает интеграцию новой библиотеки поддержки, которая добавляет новые функции и изменяет некоторые старые. (пока не отмечать).

После заполнения всех полей нажимаем “Finish” и ждем пока наш проект создастся. Текущие выполняемые задачи можно увидеть в правом нижем углу IDE (Рис. 21)

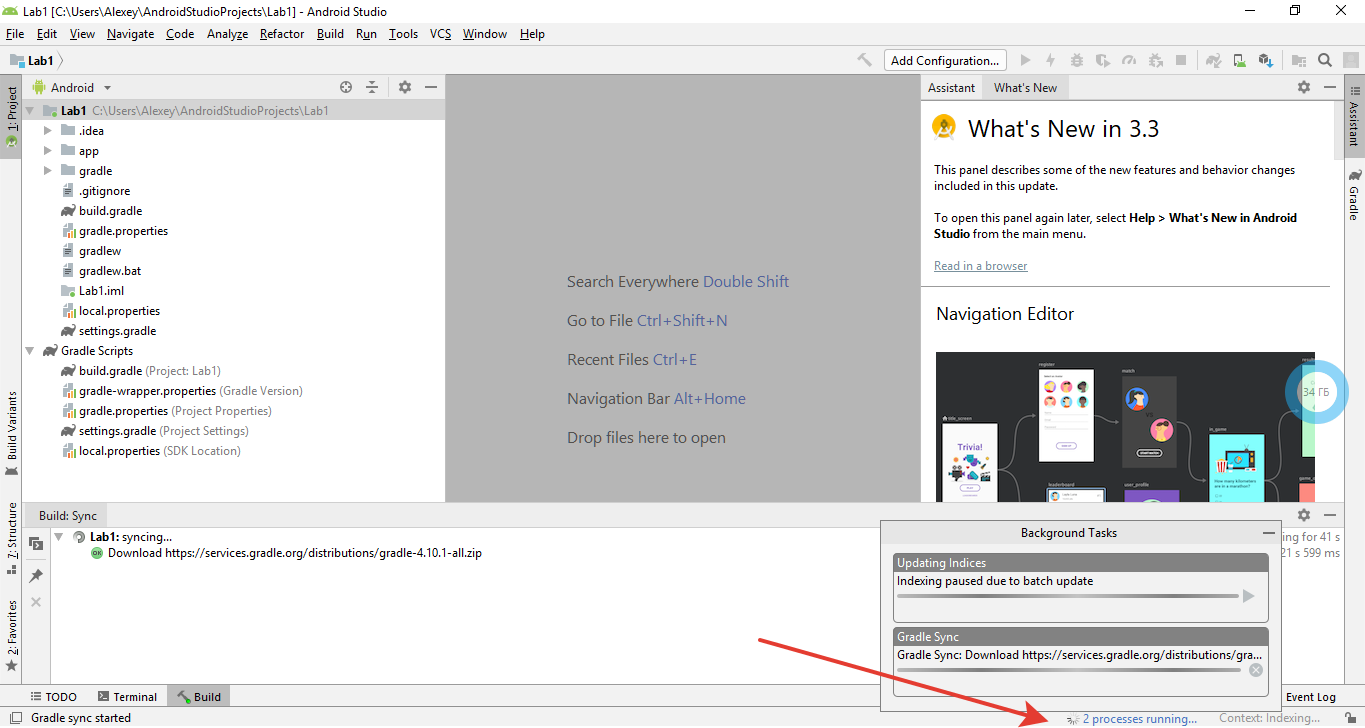
****

Рис. 21 Основное окно Android Studio.

Когда процессы индексации и выполнения задач Gradle завершатся на экране будет отображено активное окно с двумя вкладками (Рис. 22).

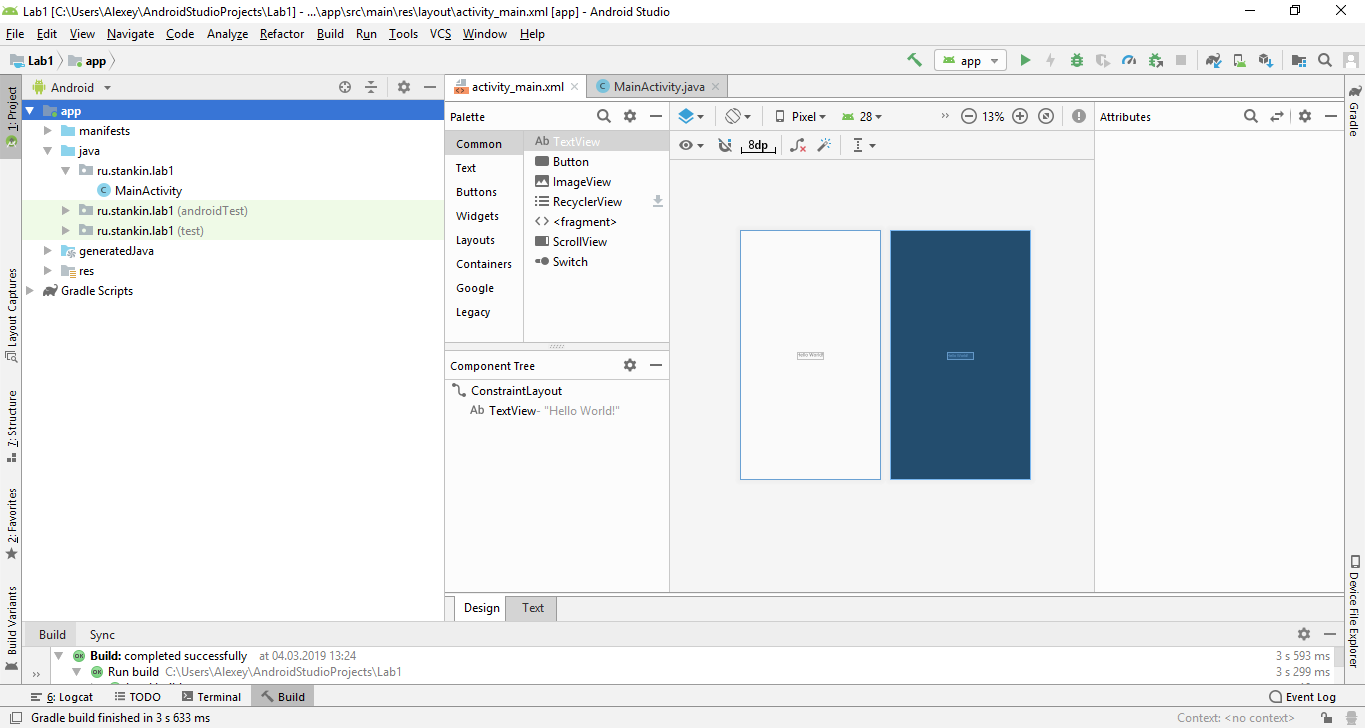


Рис. 22. Основное окно среды разработки

**Activity\_main.xml** – в данном файле хранится интерфейс Activity

**MainActivity.java** – это файл хранящий непосредственно функционал Activity (функции, методы, переменные и т.д.)

Изначально на Activity будет только TextView (простое отображение текста). Если выделить это элемент, то справа появится окно его атрибутов (Рис. 23)

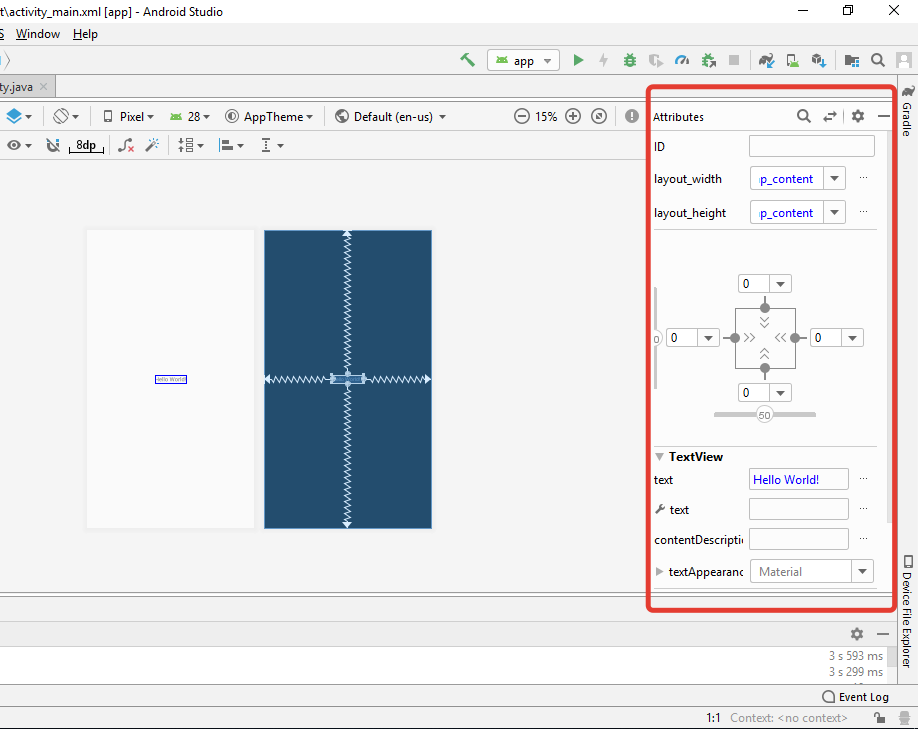


Рис 23. Атрибуты элемента.

Атрибуты:

**text** – данный атрибут отвечает за текст, который будет отображён в это элементе

**textAppearance** – если развернуть этот атрибут, то появятся расширенные настройки отображения текста такие как: Шрифт, Размер шрифта, Цвет текста и т.д.

Попробуйте «поиграться» со значениями атрибутов и посмотрите какой эффект это даст.

# Запуск приложения.

Для того что бы запустить наше приложения необходимо либо устройство под управлением Android, либо эмулятор. Давайте рассмотрим оба варианта.

Для того что бы запустить ваше приложение на физическом устройстве необходимо включить отладку по USB в настройках устройства. (Обычно эта опция находится на вкладке «Для разработчиков». Подробнее о том, как включить отладку ищите на сайте производителя). После того как отладка включена, подключите устройство по USB к компьютеру, на котором установлена Android Studio. Сначала должны установиться необходимые драйвера, после чего нужно нажать на кнопку “Run” (Shift + F10), выбрать ваше устройство в списке «Connected Devices» и нажать «ОК» (Рис. 24)

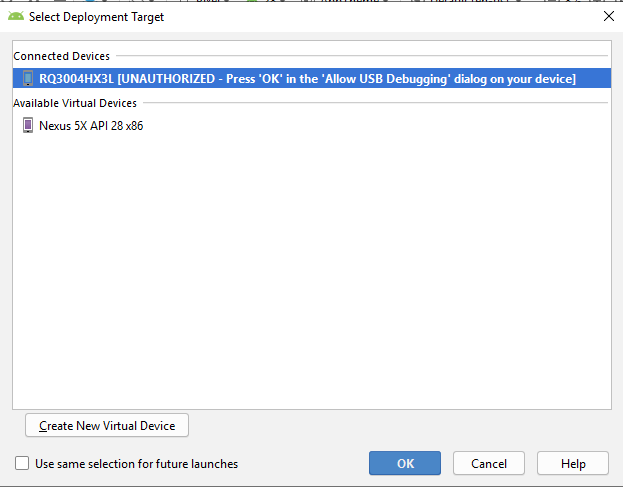


Рис. 24. Окно выбора устройства для отладки

После запуска процесса отладки необходимо разрешить данному компьютеру отладку на вашем устройстве (Рис. 25)

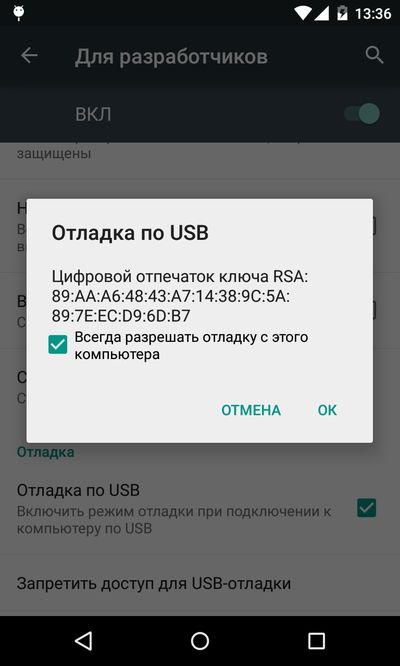


Рис. 25. Пример окна разрешения отладки на устройстве

Когда приложение соберется и установится на устройство, то сразу откроется главное Activity приложения (Рис. 26).

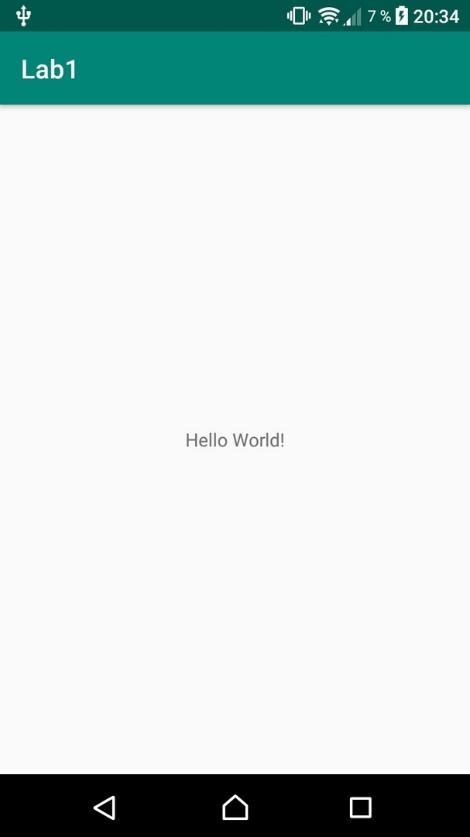


Рис. 26. Запущенное приложение.

Если у вас нету устройства на Android или вы хотите протестировать ваше приложение на устройстве с API выше чем у вас есть, то можно воспользоваться эмулятором, поставляемым вместе с Android Studio.

Для этого перейдем в меню Tools -> AVD Manager (Рис. 27)

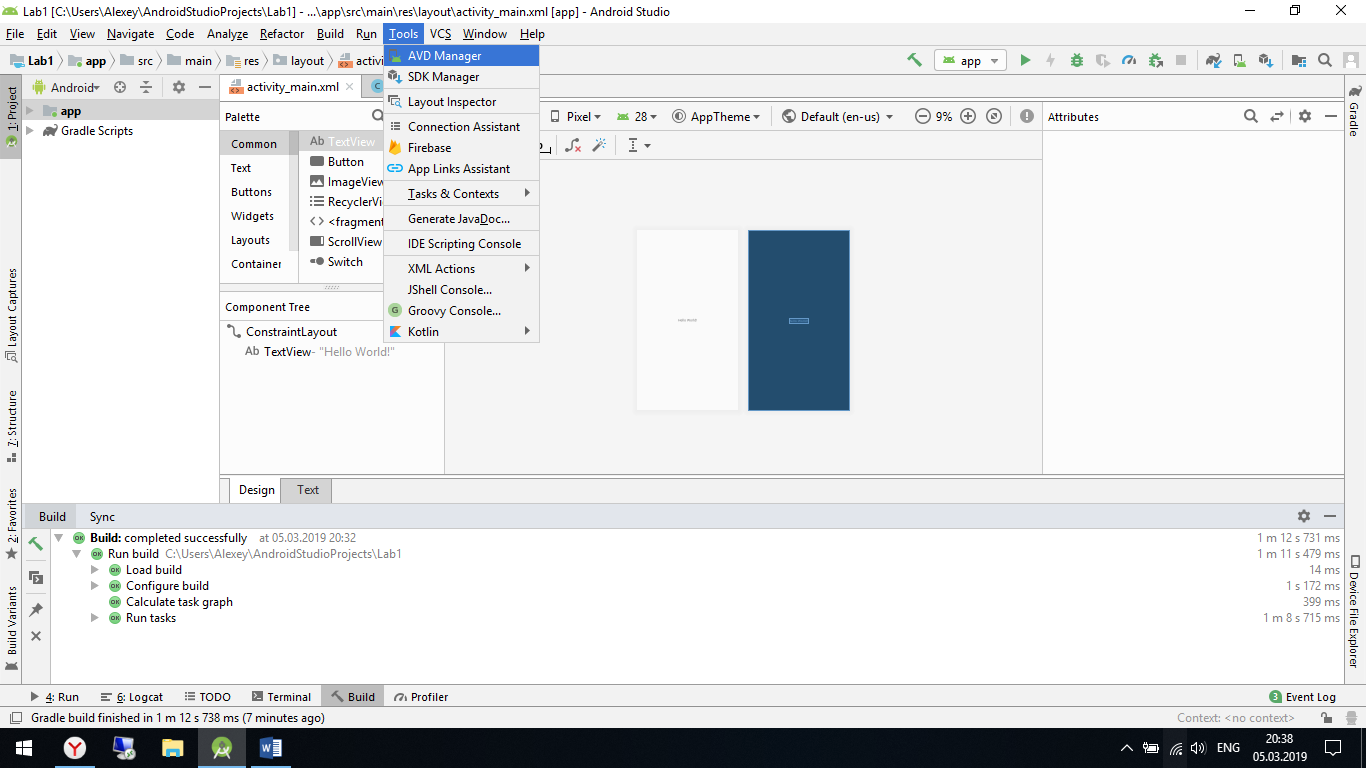


Рис. 27.

Далее жмем «Create Virtual Device..», а потом либо выбираем устройство из предложенных шаблонов (отличаются они все типом устройства: Планшет/Телефон/Носимое устройства/Android TV, размером экрана в дюймах, Разрешением, количеством выделяемой оперативной памяти, и прочими опциями, подробнее с которыми вы можете ознакомится по [данной](https://developer.android.com/studio/run/managing-avds) ссылке) (Рис. 28)

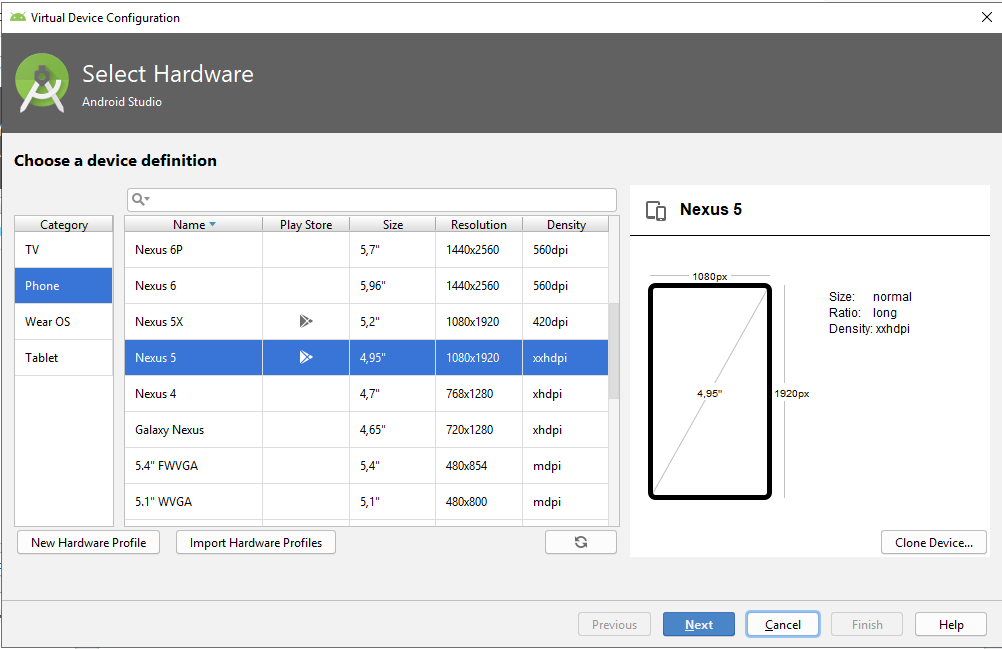


Рис. 28.

На следующем этапе необходимо выбрать образ системы, которая будет запущенна на виртуальном устройстве. Выберите, тот образ, который больше всего вам подходит, а затем нажмите на “Download”, если образ еще не скачан, или просто выберите его и нажмите «Next». (Android Studio будет предлагать вам установить образы Android x86 так как при запуске этих образов будет работать аппаратное ускорение, что позволит эмулятору работать быстрее). (Рис. 29)

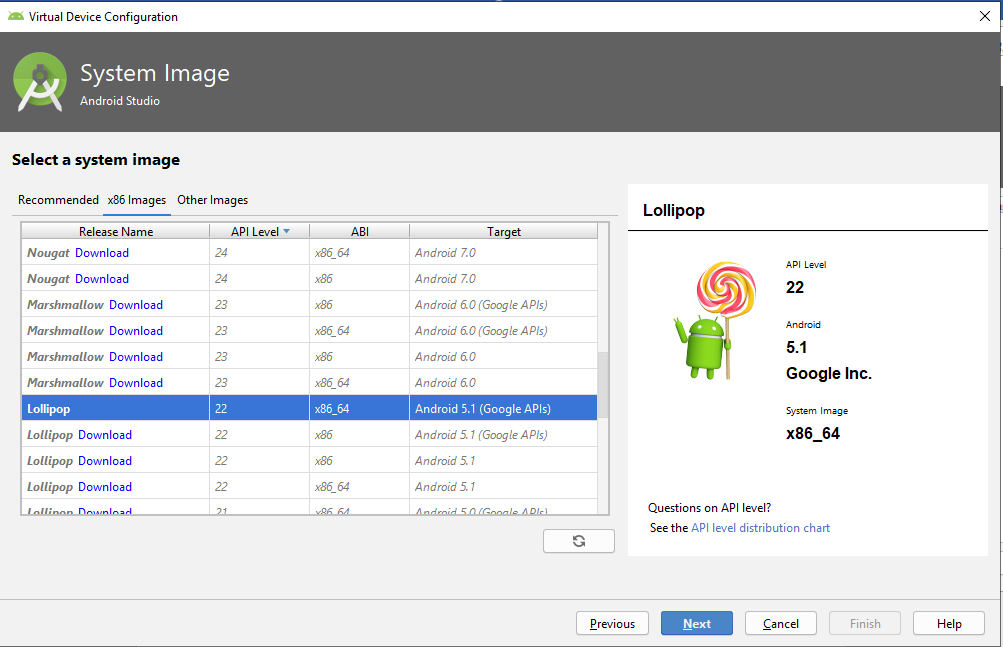


Рис. 29.

В следующем окне вводим имя устройства, выбираем ориентацию экрана при запуске Портретная или Ландшафтная, выбираем тип отображения Software – программный рендеринг, Hardware – рендеринг с помощью вашей физической видеокарты (Рекомендуется выбирать аппаратный рендеринг, так как он должен работать быстрее. Если у вас возникли проблемы с аппаратным рендерингом, то можно изменить его на программный), а также выбираем нужно ли отображать рамку эмулируемого устройства (Рис. 30).

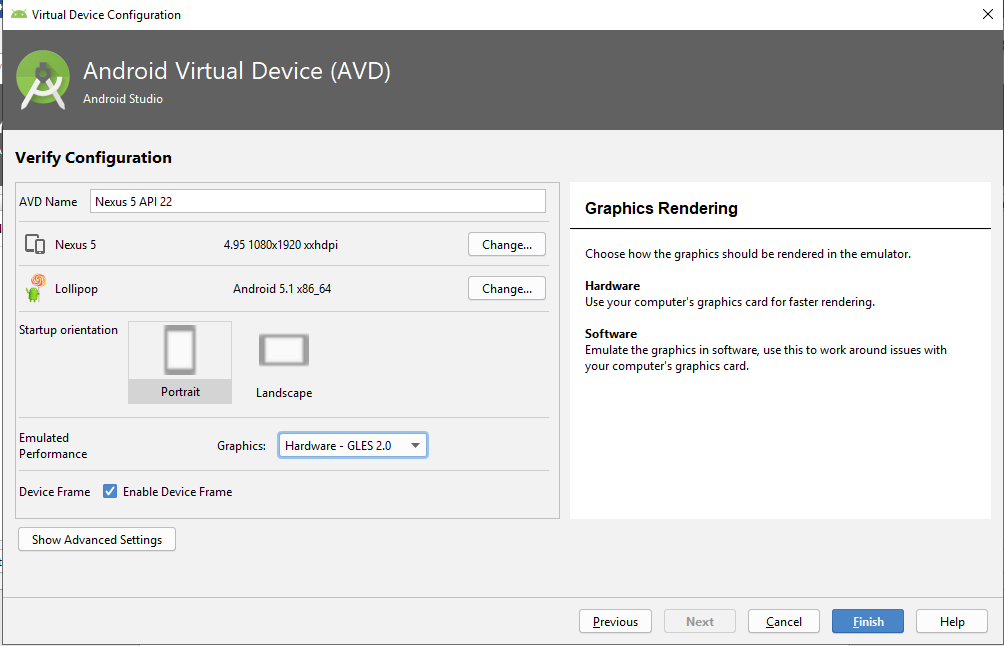


Рис. 30

Жмем “Finish” и ждем пока создастся устройство, после чего закрываем окно AVD, жмем «Run» и выбираем наше эмулируемое устройство в списке «Available Virtual Devices» (Рис. 31).

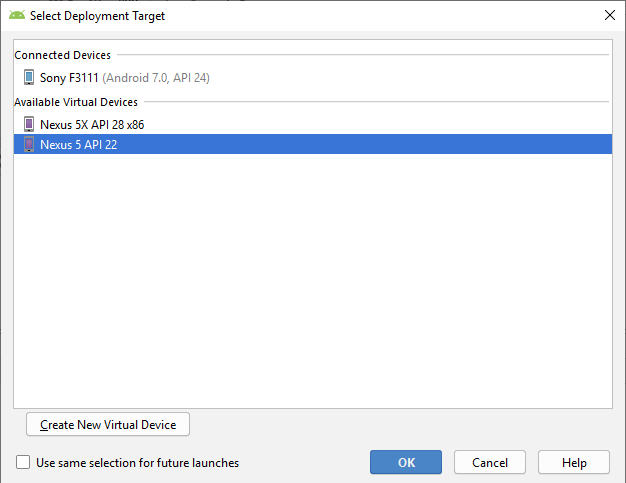
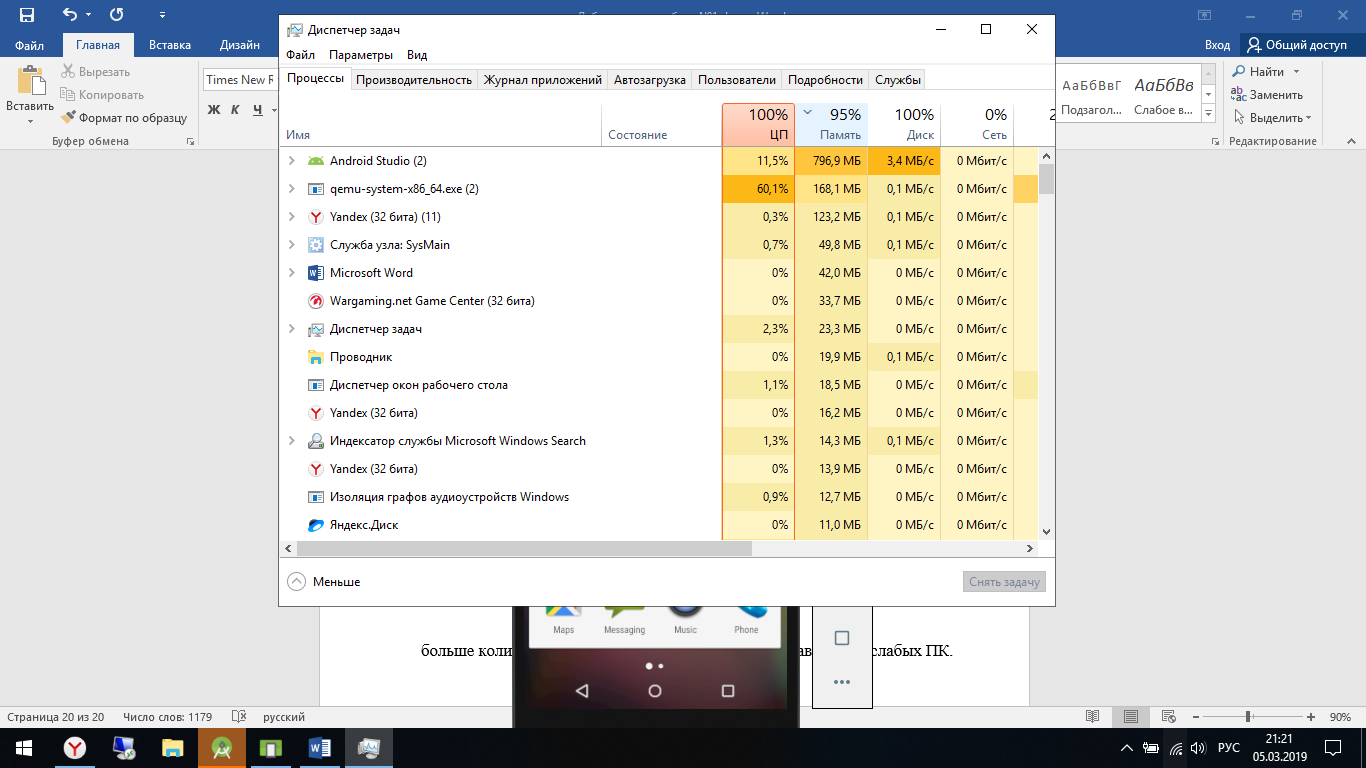


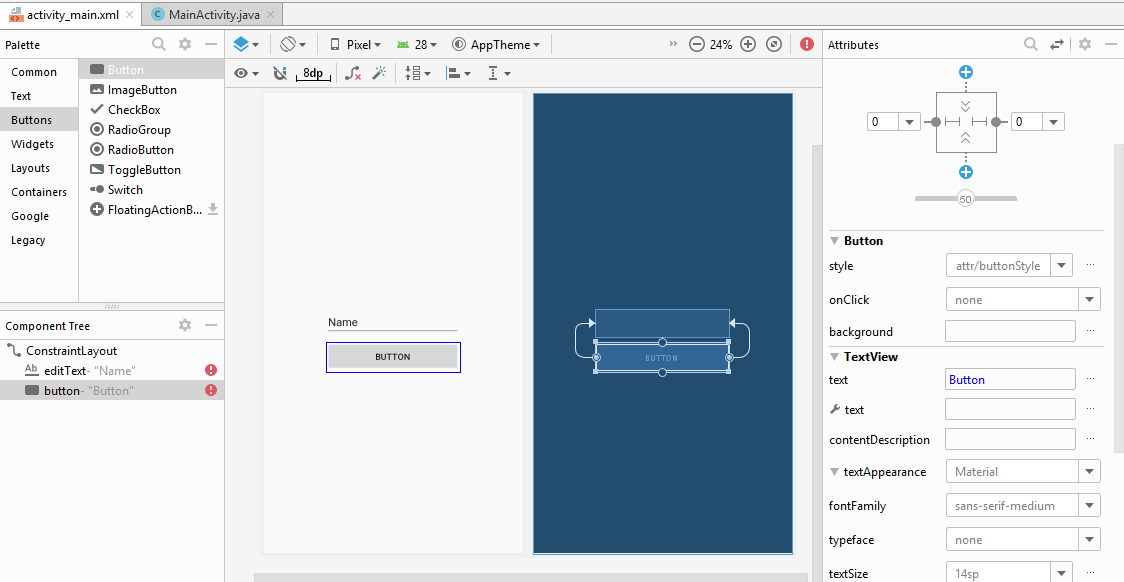
Рис. 31.

После всех действий, у вас запустится эмулятор, а в нем запустится ваше приложение.

Необходимо уточнить, что эмуляция потребляет достаточно больше количество ресурсов (в частности ЦП) и может произвести к зависанию слабых ПК.



Теперь добавим функционала нашему приложению. Удалите TextView, а вместо него добавьте на Activity “PlainText” и “Button” (Все доступные элементы интерфейса находятся в списке Palette слева от области Activity). Теперь давайте привяжем наши объекты к определенному расположению на экране. Для начала наведитесь на editText, затем наведитесь на появившейся кружок на верхней границе элемента и зажав левый клик потяните кружок в верхнем краю Activity, чем привяжите editText к верхнему краю формы. Затем тоже самое, только с нижним кружком и с нижней гранью, с левым, с правым. В итоге мы получим, что editText будет в середине. Он будет в середине на любом экране и при любой ориентации. Теперь привяжите левую грань button к левой грани editText, правую к правой, верхнюю к нижней и немного сдвиньте кнопку вниз и получите, что и editText и button теперь всегда будут в середине, а также button всегда будет на одном расстоянии от editText.



Теперь перейдите в MainActivity.java и впишите туда следующий код:

**package** ru.stankin.lab1;  
  
**import** android.app.Activity;  
**import** android.os.Bundle;  
**import** android.view.View;  
**import** android.widget.Button;  
**import** android.widget.EditText;  
**import** android.widget.Toast;  
  
**public class** MainActivity **extends** Activity {  
  
 Button **button**; *//Инициализируем переменную для хранения обхекта кнопки* EditText **editText**; *//Инициализируем переменную для хранения обхекта текстового поля* @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
  
 **editText** = (EditText) findViewById(R.id.***editText***); *//Присваеваем текстовому полю объект, полученный с Activity* **button** = (Button) findViewById(R.id.***button***); *//Присваеваем кнопке объект, полученный с Activity* **button**.setOnClickListener(  
 **new** View.OnClickListener() { *//Здесь добавляется обработчик нажатия на кнопку* @Override  
 **public void** onClick(View v) {  
 Toast makeText = Toast.*makeText*(getApplicationContext(), **editText**.getText(), Toast.***LENGTH\_SHORT***);  
 makeText.show(); *//Выводим сообщение пользователю, содержащее текст из текстового поля* }  
 }  
 );  
 }  
  
  
}

Готово. Теперь если вы запустите приложение, введете какой-либо текст в поле и нажмете на кнопку, то увидите сообщение с этим текстом.

**Варианты персональных заданий:**

1. **Добавьте на Activity TextView и при нажатии на кнопку меняйте текст в этом TextView**
2. **Добавьте на Activity CheckBox и выводите сообщение, только если CheckBox отмечен**
3. **Сделайте так, чтобы EditText был всегда привязан к верхнему краю, а Button к нижнему.**
4. **Сделайте так, чтобы на Activity было несколько кнопок, и они выводили разный текст**
5. **Сделайте несколько полей для ввода Фамилии, Имени и Отчества и при нажатии на кнопку выводите все три поля**
6. **Сделайте два поля для ввода и при нажатии на кнопку меняйте их содержимое местами**
7. **Сделайте цветную кнопку, которая меняет цвет при нажатии**
8. **\*Сделайте подобие простого калькулятора, который считывает два числа и выполняемую операцию из трех полей и при нажатии на кнопку вычисляет и выводит сообщение с результатом.**