

Разработка мобильных приложений

Android

# Версии Android

- С 2008 года — 40 обновлений системы
- 13 версий + вышла последняя: Oreo
- Начиная с версии 1.5 каждой версии присваивается кодовое слово:

Nougat



Oreo

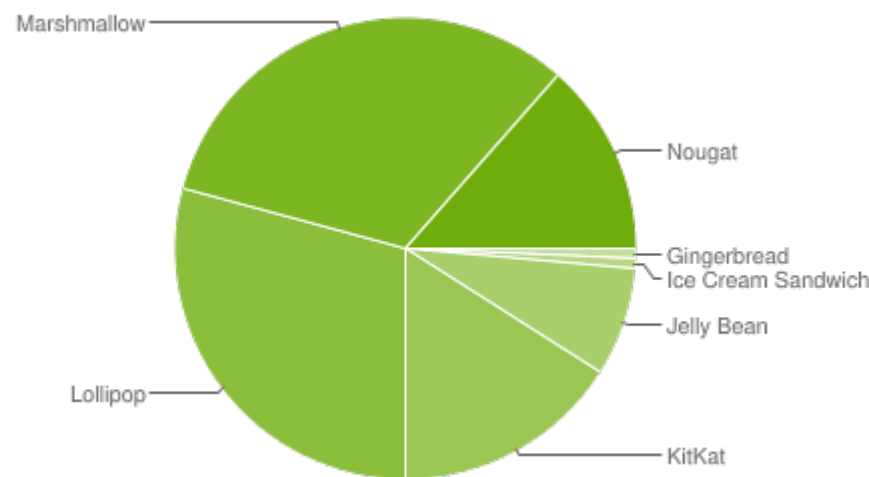


KitKat



# Версии Android

Version	Codename	API	Distribution
2.3.3 -2	Gingerbread	10	0.7%
4.0.3 -4	Ice Cream Sandwich	15	0.7%
4.1.x	Jelly Bean	16	2.7%
4.2.x		17	3.8%
4.3		18	1.1%
4.4	KitKat	19	16.0%
5.0	Lollipop	21	7.4%
5.1		22	21.8%
6.0	Marshmallow	23	32.3%
7.0	Nougat	24	12.3%
7.1		25	1.2%



# Версии Android

- В 2008 году — первая официальная версия ОС с пакетом разработчика, а самая первая — 12 ноября 2007 года
  - В 2005 году Google купил Android Inc.
  - В 2007 году Google объявил о создании альянса Open Handset Alliance
  - В 2007 году Open Handset Alliance объявил Android открытой системой

Первый смартфон на базе Android — 2008 год



# Android Studio

IDE для разработки приложений Android

<https://developer.android.com/studio/index.html>



Поддерживается Google



Содержит эмулятор Android


Состоит из SDK и IDE на базе IntelliJ Idea

# План

- Чему я сегодня научусь?
  - Создавать и запускать приложение
- Про что я сегодня узнаю?
  - Android emulator
  - Android SDK (minSdk, targetSdk)
  - Activity, Service, Broadcast Receiver, Content Provider
  - Android Manifest
  - Layout (FrameLayout, LinearLayout, ConstraintLayout)
  - Views
  - R class

# Создание проекта

✕ □ Create New Project

 New Project  
Android Studio

**Configure your new project**

Application name:  Любой текст

Company domain:

Package name:  Глобально уникальный текст во всем интернете

☐ Include C++ support

Project location:  ...


Желательно, чтобы путь не содержал русских символов

Your project location contains non-ASCII characters, which can cause problems. Proceed with caution.

Previous Next Cancel Finish

# Создание проекта

Create New Project

 Target Android Devices

## Select the form factors your app will run on

Different platforms may require separate SDKs

☒ Phone and Tablet

Minimum SDK 

API 20: Android 4.4W (KitKat Wear)

Lower API levels target more devices, but have fewer features available.  
By targeting API 20 and later, your app will run on approximately **71,3%** of the devices that are active on the Google Play Store.  
[Help me choose](#)

☐ Wear

Minimum SDK 

API 21: Android 5.0 (Lollipop)

☐ TV

Minimum SDK 

API 21: Android 5.0 (Lollipop)

☐ Android Auto

Тип приложения

Минимальная версия Android, на которую можно будет установить приложение

Previous

Next

Cancel

Finish




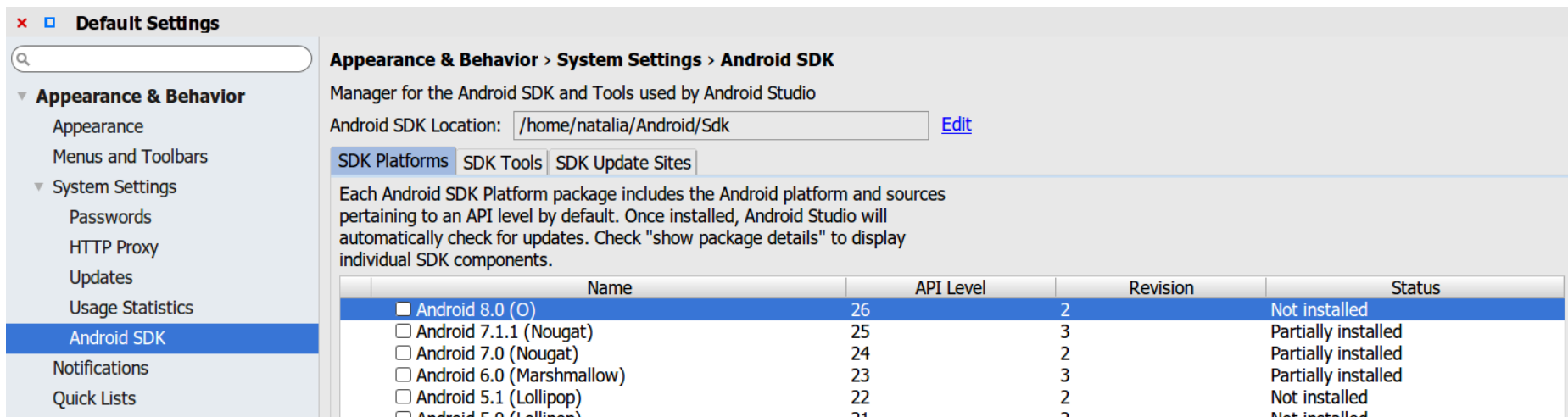
# Android SDK

## Software Development Kit

- Компилирует код, файлы данных и ресурсов в исполняемый **файл APK** (Android Package)
- Управление версиями и указание пути к SDK осуществляется через **SDK Manager**:

Tools → Android → SDK Manager

или нажать в панели на значок SDK Manager 



The screenshot shows the 'Default Settings' dialog in Android Studio, with the 'Android SDK' tab selected under 'System Settings'. The 'Android SDK Location' is set to '/home/natalia/Android/Sdk'. The 'SDK Platforms' tab is active, displaying a table of installed and available SDK platforms.

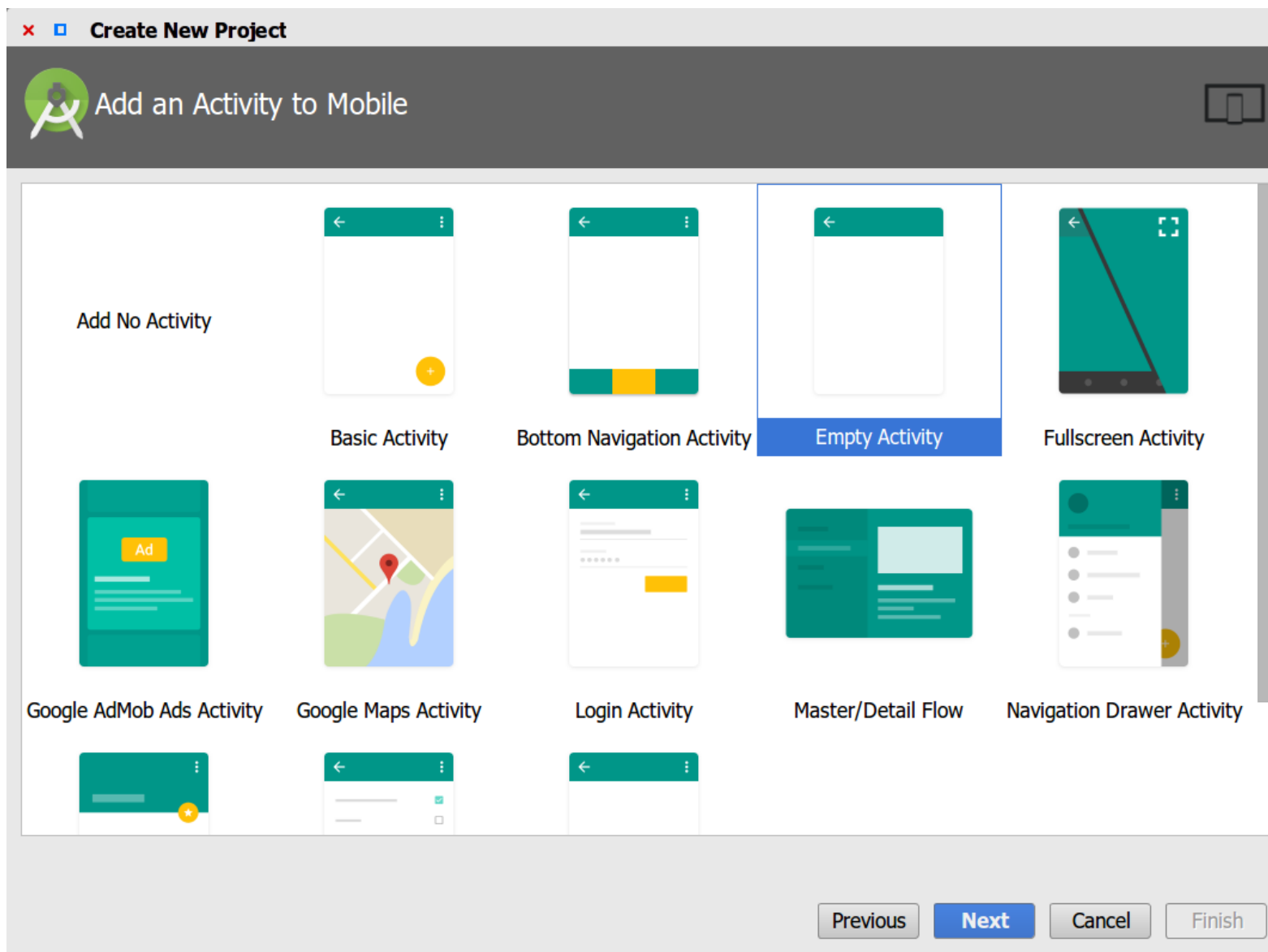
	Name	API Level	Revision	Status
<input checked="" type="checkbox"/>	Android 8.0 (O)	26	2	Not installed
<input type="checkbox"/>	Android 7.1.1 (Nougat)	25	3	Partially installed
<input type="checkbox"/>	Android 7.0 (Nougat)	24	2	Partially installed
<input type="checkbox"/>	Android 6.0 (Marshmallow)	23	3	Partially installed
<input type="checkbox"/>	Android 5.1 (Lollipop)	22	2	Not installed
<input type="checkbox"/>	Android 5.0 (Lollipop)	21	2	Not installed

# minSDK, targetSDK

- **minSDK** — минимальная версия SDK, на которой может запускаться приложение
  - Распознается и Google Play, и платформой телефона
  - Невозможно безопасно вернуться к предыдущим версиям
  - Необходимо найти баланс между возможностью установить приложение на большинство устройств и ценой поддержки всех пользователей
- **targetSDK** — версия SDK, на которой тестируется приложение
  - НЕ является максимальной допустимой версией
  - По умолчанию совпадает с последней доступной версией

Для установки: **File** → **Project Structure** → **app** → **Flavors**

# Создание проекта



<https://developer.android.com/studio/projects/templates.html>

# Создание проекта

Create New Project

Customize the Activity

Creates a new empty activity

Имя java-класса

Activity Name: MainActivity

☒ Generate Layout File

Layout Name: activity\_main

☒ Backward Compatibility (AppCompat)

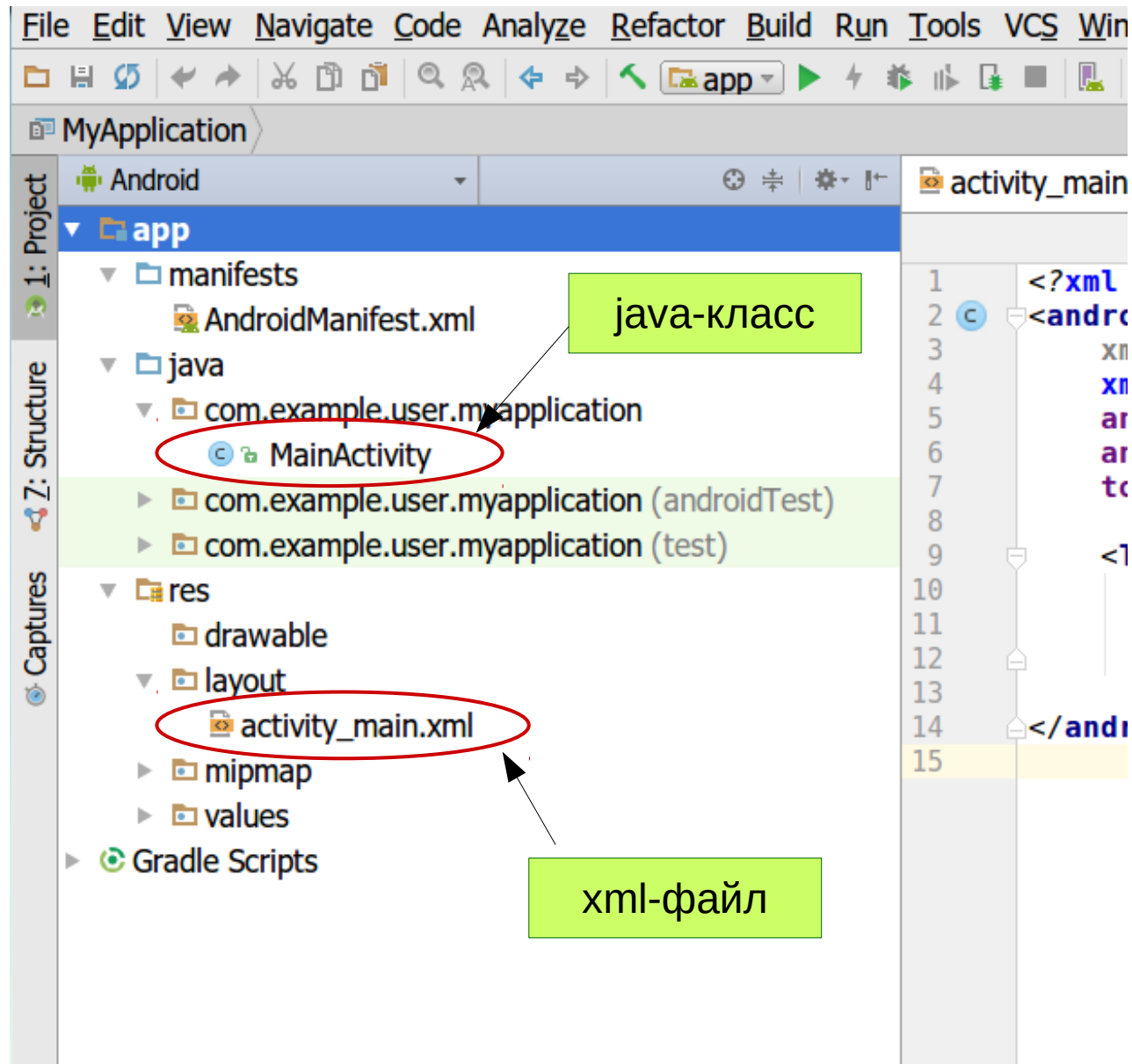
Имя xml-файла

Empty Activity

The name of the activity class to create

Previous Next Cancel Finish

# Создание проекта



# Эмулятор Android | Устройство

- Запуск на эмуляторе
  - <https://developer.android.com/studio/run/emulator.html>
- Запуск на устройстве
  - <https://developer.android.com/studio/run/emulator.html>

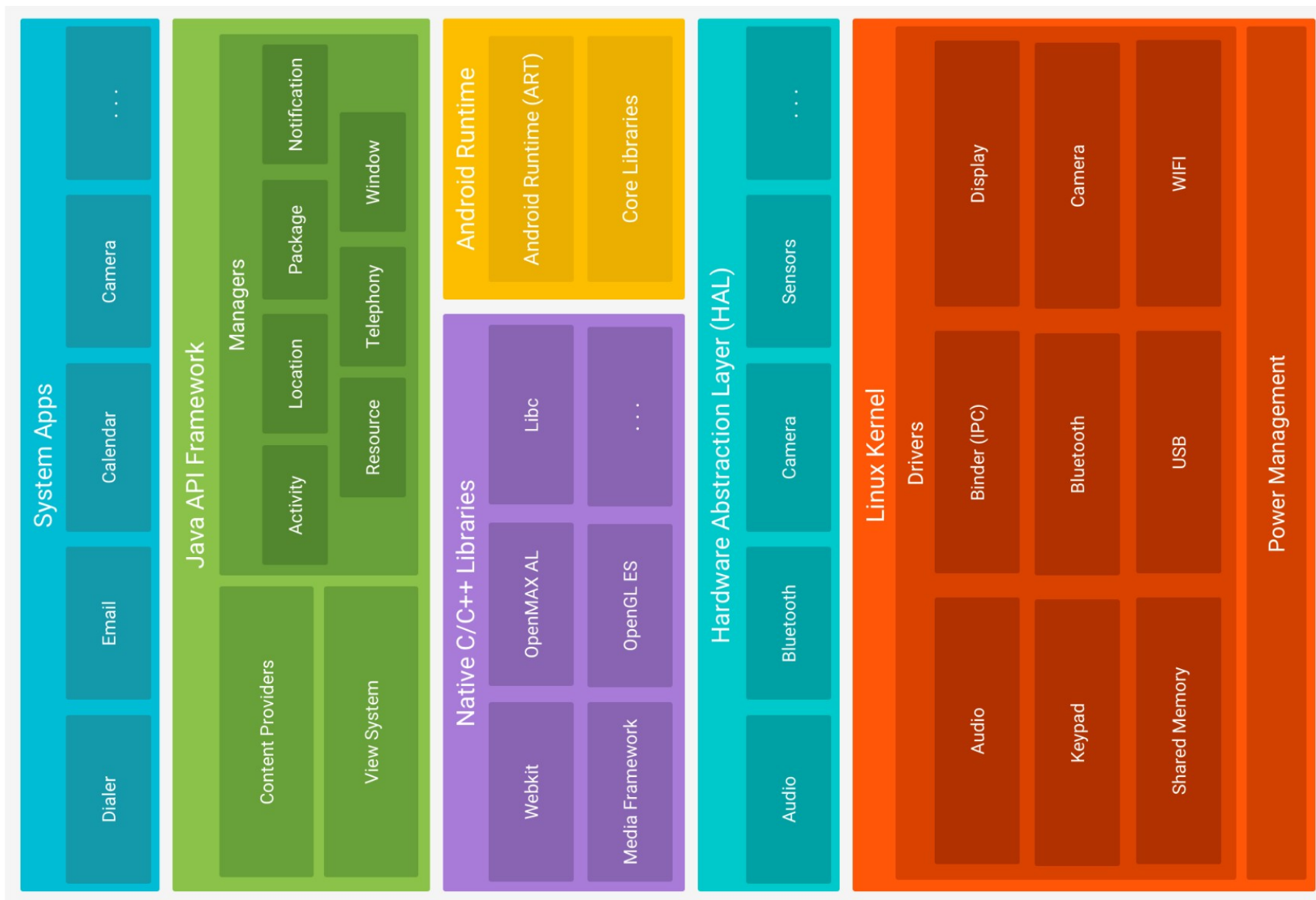


# Android Software Stack

- Application Layer
  - Установленные приложения
- Application Framework
  - Классы и сервисы для запуска приложения в собственном окружении
- C/C++ Libs, Android Runtime
  - LibsC и SQLite, базовые библиотеки Android
- Linux Kernel
  - низкоуровневые задачи: аппаратные средства, драйвера, управление питанием

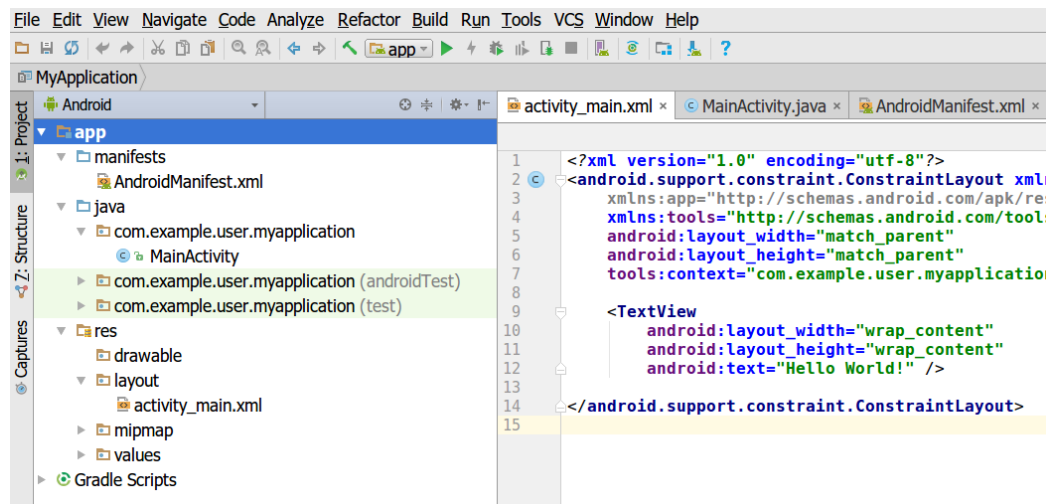
<https://developer.android.com/guide/platform/index.html>

# Android Software Stack

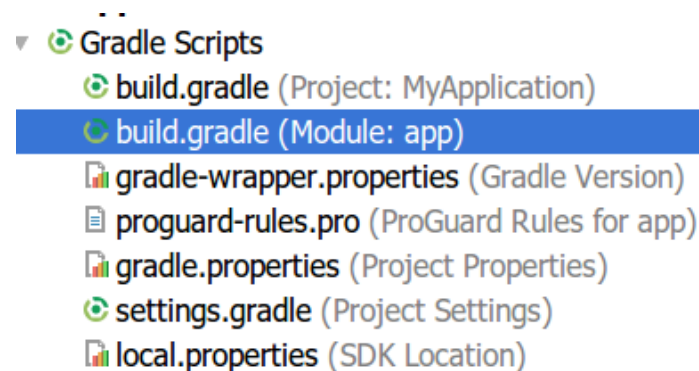




## 1. Создание проекта Android, написание кода приложения



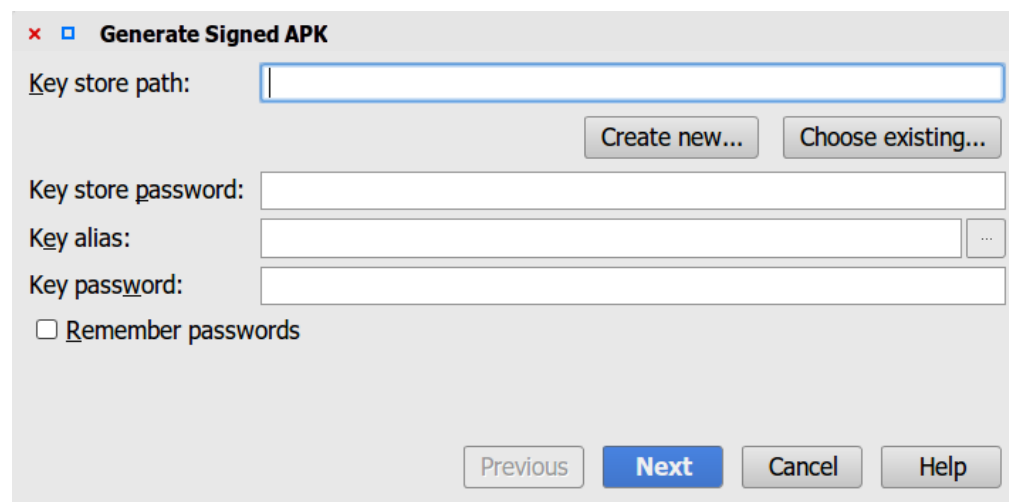
## 2. Компиляция с помощью Gradle



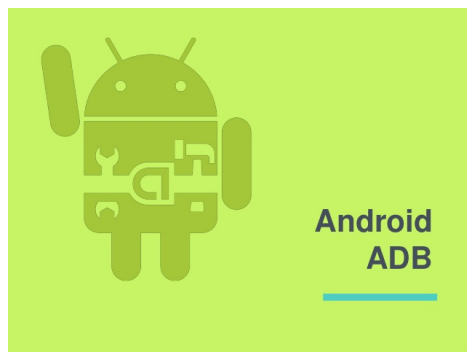
## 3. APK-файл, содержащий код, ресурсы и манифест



## 4. Подпись приложения с помощью Jar Signer



## 5. Установка на устройство с помощью ADB



# Компоненты Android-приложения:

- Activity (Операция)  
один экран с пользовательским интерфейсом, например:  
экран со списком контактов, экран с настройками
- Service (Служба)  
не имеет пользовательского интерфейса, работает в фоновом режиме неограниченное время, например:  
воспроизведение музыки, получение данных по сети
- Broadcast Receiver (Приемник широковестьательных сообщений)  
нужен для реакции на события, происходящие в системе вне приложения, например:  
экран выключился, аккумулятор разряжен, сделан фотоснимок
- Content Provider (Поставщик контента)  
Посредством поставщика контента другие приложения могут запрашивать или даже изменять данные, например:  
для запроса списка контактов

# Компоненты Android-приложения:

- AndroidManifest.xml

В этом файле, который должен находиться в корневой папке приложения, **должны быть объявлены все компоненты приложения**

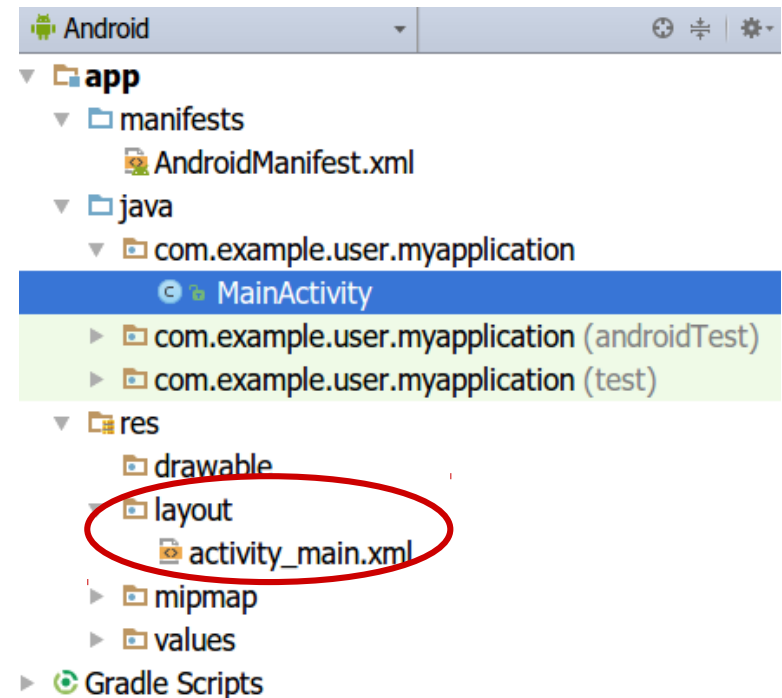
```
1 |<?xml version="1.0" encoding="utf-8"?>
2 |<manifest xmlns:android="http://schemas.android.com/apk/res/android"
3 |    package="com.example.user.myapplication">
4 |
5 |    <application
6 |        android:allowBackup="true"
7 |        android:icon="@mipmap/ic_launcher"
8 |        android:label="My Application"
9 |        android:roundIcon="@mipmap/ic_launcher_round"
10 |        android:supportRtl="true"
11 |        android:theme="@style/AppTheme">
12 |        <activity android:name=".MainActivity">
13 |            <intent-filter>
14 |                <action android:name="android.intent.action.MAIN" />
15 |
16 |                <category android:name="android.intent.category.LAUNCHER" />
17 |            </intent-filter>
18 |        </activity>
19 |    </application>
20 |
21 |</manifest>
```

Объявление activity

# Activity и Layout

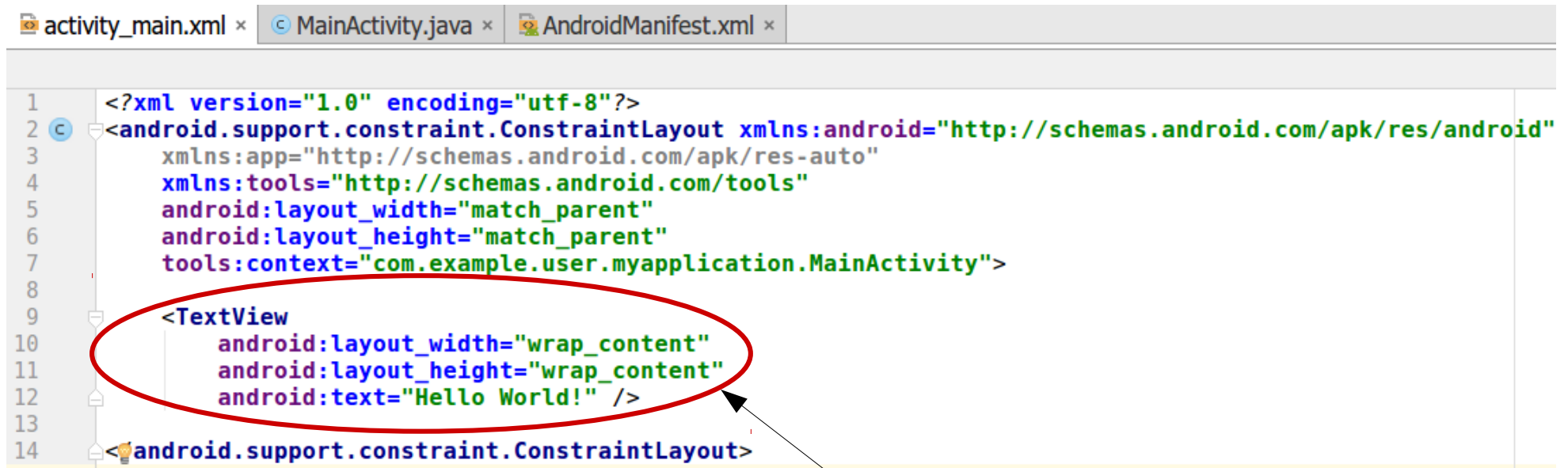
- Activity

- отвечает за окно, которое приложение использует для отрисовки и получения событий от системы
- Создает **views** для отображения информации и обеспечения интерактивности. (View — класс в платформе Android; занимает прямоугольную область и отвечает за отрисовку и обработку событий)
- Activity понимает, где и какие views отображать по **xml layout файлу**, в котором задается описание создаваемых объектов views
- xml файл с описанием лежит в папке **res/layouts**



# Activity и Layout

- Layout: activity\_main.xml



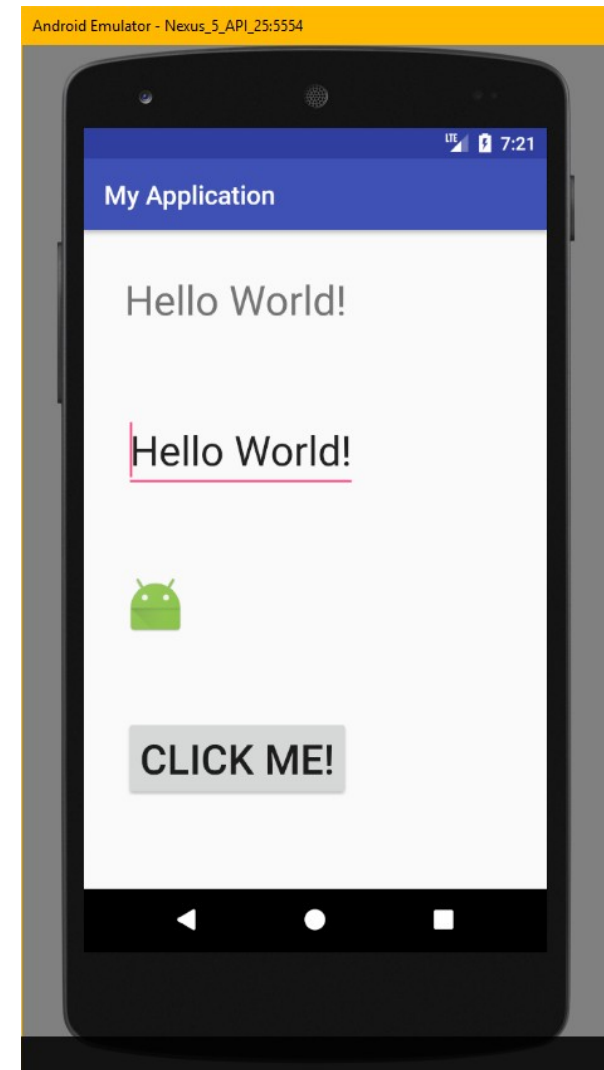
```
1 <?xml version="1.0" encoding="utf-8"?>
2 <android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     tools:context="com.example.user.myapplication.MainActivity">
8
9     <TextView
10         android:layout_width="wrap_content"
11         android:layout_height="wrap_content"
12         android:text="Hello World!" />
13
14 </android.support.constraint.ConstraintLayout>
```

Пример View

# Типы View

## 1. UI Components

- TextView
  - фиксированный текст
- EditText
  - окно для ввода текста
- ImageView
  - изображение на экране
- Button
  - кнопка на экране



# Типы View

## 2. Container Views

- `LinearLayout`
  - Объекты располагаются в одну колонку или строку
- `RelativeLayout`
  - Можно указать расположение объектов относительно друг друга
- `FrameLayout`
  - Содержит единственный дочерний объект `View`
- `ScrollView`
  - Позволяет прокручивать объекты, находящиеся внутри
- `ConstraintLayout`
  - Позволяет более гибко располагать объекты

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3      android:layout_width="match_parent"
4      android:layout_height="match_parent"
5      android:orientation="vertical">
6      <TextView
7          android:layout_width="wrap_content"
8          android:layout_height="wrap_content"
9          android:padding="32dp"
10         android:text="Hello World!"
11         android:textSize="32sp"/>
12      <EditText
13          android:layout_width="wrap_content"
14          android:layout_height="wrap_content"
15          android:layout_margin="32dp"
16          android:text="Hello World!"
17          android:textSize="32sp"/>
18      <ImageView
19          android:layout_width="wrap_content"
20          android:layout_height="wrap_content"
21          android:padding="32dp"
22          android:src="@mipmap/ic_launcher"/>
23      <Button
24          android:layout_width="wrap_content"
25          android:layout_height="wrap_content"
26          android:layout_margin="32dp"
27          android:text="Click me!"
28          android:textSize="32sp"/>
29  </LinearLayout>
```



# Атрибуты XML

- Атрибуты объектов `views` позволяют определить положение, размер и другие свойства объектов `view`, например:
  - `textSize` задает размер текста
  - `padding` задает отступ от границы
- У каждого объекта `view` много свойств, все они перечислены в документации
  - Для `TextView`:  
<https://developer.android.com/reference/android/widget/TextView.html>

## TextView

```
public class TextView  
extends View implements ViewTreeObserver.OnPreDrawListener
```

...

added in API level 1  
Summary: Nested Classes | XML Attrs | Inherited XML  
Attrs | Constants | Inherited Constants | Inherited Fields |  
Ctors | Methods | Protected Methods | Inherited  
Methods | [Expand All]

XML attributes	
<code>android:autoLink</code>	Controls whether links such as urls and email addresses are automatically found and converted to clickable links.
<code>android:autoSizeMaxTextSize</code>	The maximum text size constraint to be used when auto-sizing text.
<code>android:autoSizeMinTextSize</code>	The minimum text size constraint to be used when auto-sizing text.
<code>android:autoSizePresetSizes</code>	Resource array of dimensions to be used in conjunction with <code>autoSizeTextType</code> set to <code>uniform</code> .

# Атрибуты XML

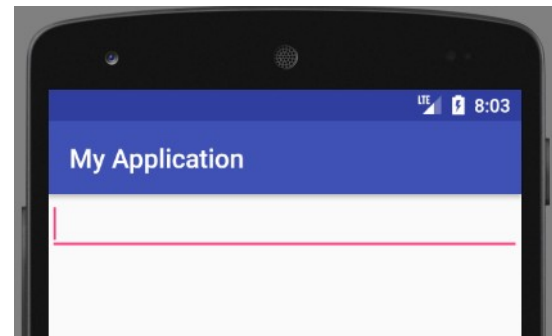
## layout\_width, layout\_height

У **каждого объекта** view должны быть заданы ширина и высота

например:

- android:layout\_width = «200dp»
- android:layout\_height = «300dp»
- Стандартные значения этих атрибутов
  - wrap\_content — объект будет занимать столько места, сколько занимает его содержимое
  - match\_parent — объект будет занимать все пространство, которое занимает родительский объект

```
<EditText  
    android:id="@+id/plain_text_input"  
    android:layout_height="wrap_content"  
    android:layout_width="match_parent"  
    android:inputType="text"/>
```



# Атрибуты XML

padding, margin

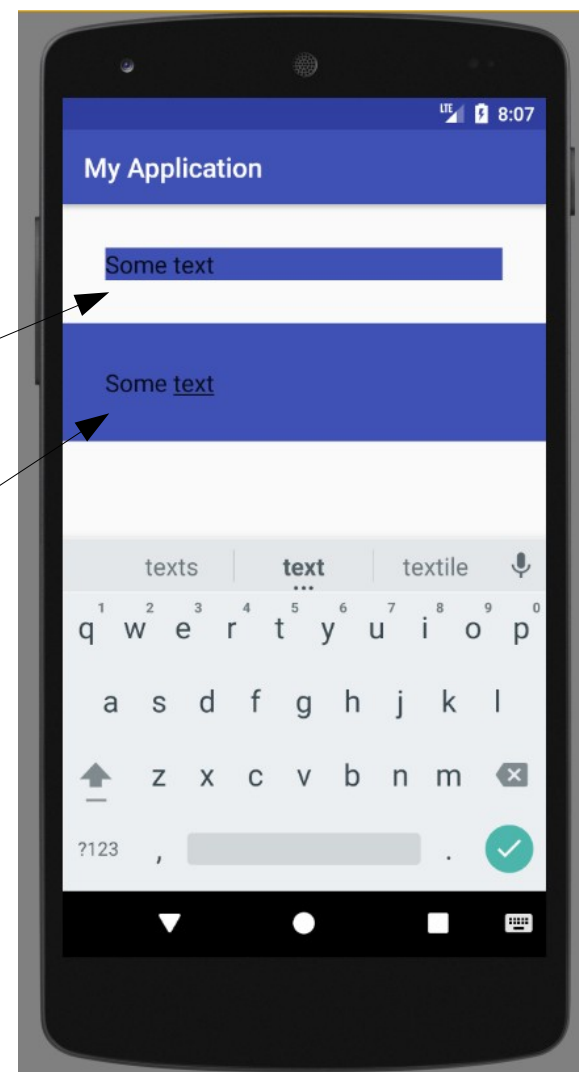
Оба атрибута задают пространство вокруг объекта view

- Padding — задает пространство внутри границ view
- Margin — задает пространство вокруг view за его границами

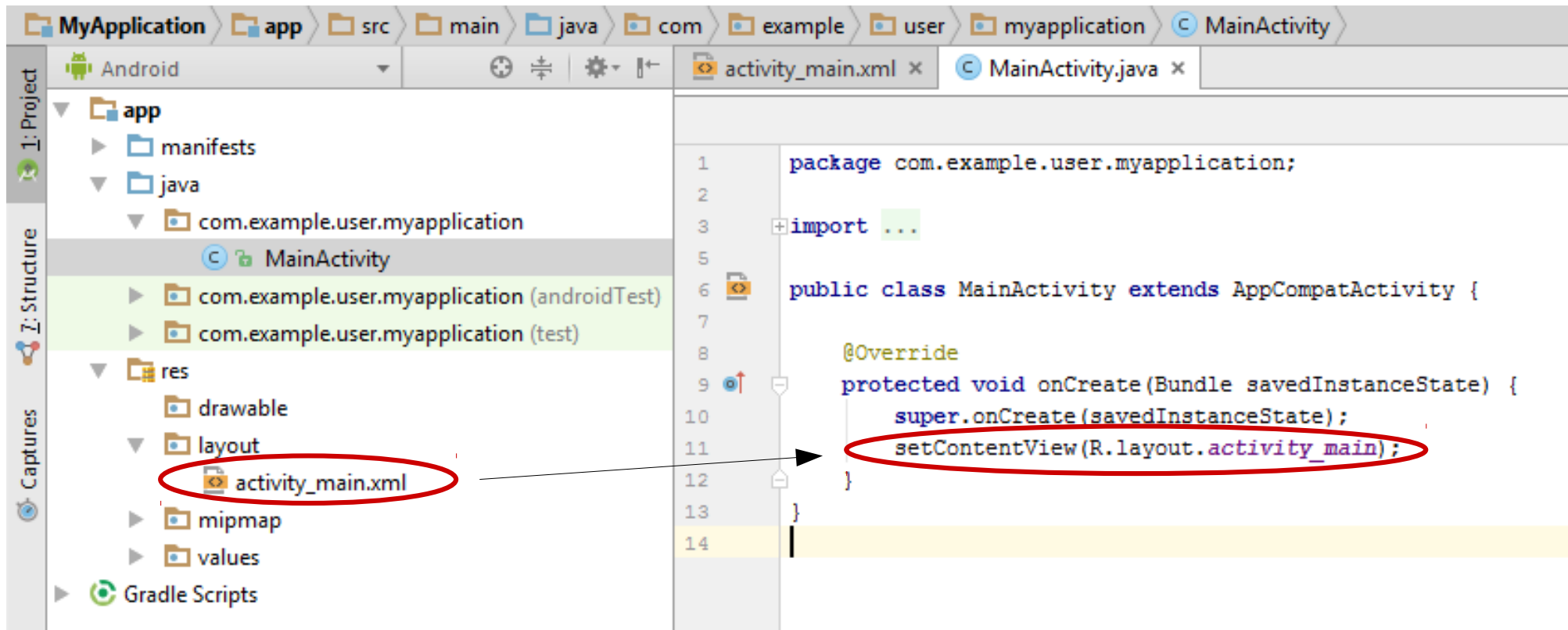
# Атрибуты XML

## padding, margin

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     android:layout_width="match_parent"
4     android:layout_height="match_parent"
5     android:orientation="vertical">
6
7     <EditText
8         android:id="@+id/plain_text_input"
9         android:layout_height="wrap_content"
10        android:layout_width="match_parent"
11        android:inputType="text"
12        android:background="@color/colorPrimary"
13        android:layout_margin="32dp"/>
14
15    <EditText
16        android:id="@+id/plain_text_input2"
17        android:layout_height="wrap_content"
18        android:layout_width="match_parent"
19        android:inputType="text"
20        android:background="@color/colorPrimary"
21        android:padding="32dp"/>
22
23 </LinearLayout>
```



# Как соотносятся Java Activity и XML Layout



# R class

- Класс R генерируется во время компиляции
  - Создает константы для динамической идентификации содержимого папки res
- setContentView
  - размещает объекты на экране приложения в соответствии с описанием в xml-файле
- findViewById
  - находит view по его id, указанному в xml-файле и размещает его в соответствующем объекте java.

# R class

```
activity_main.xml x MainActivity.java x
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     android:layout_width="match_parent"
4     android:layout_height="match_parent"
5     android:orientation="vertical">
6
7     <EditText
8         android:id="@+id/plain_text_input"
9         android:layout_height="wrap_content"
10        android:layout_width="match_parent"
11        android:inputType="text"/>
12
13 </LinearLayout>
```

```
activity_main.xml x MainActivity.java x
1 package com.example.user.myapplication;
2
3 import ...
4
5
6
7 public class MainActivity extends AppCompatActivity {
8
9     @Override
10    protected void onCreate(Bundle savedInstanceState) {
11        super.onCreate(savedInstanceState);
12        setContentView(R.layout.activity_main);
13
14        EditText editText = (EditText) findViewById(R.id.plain_text_input);
15        editText.setText("text changed");
16    }
17 }
```

# Типы ресурсов

<https://developer.android.com/guide/topics/resources/providing-resources.html>

```
MyProject/  
  src/  
    MainActivity.java  
  res/  
    drawable/  
      graphic.png  
    layout/  
      main.xml  
      info.xml  
    mipmap/  
      icon.png  
    values/  
      strings.xml
```



# Лабораторная работа

- Создать проект с EmptyActivity
- Создать TextView, EditView, ImageView, Button и +1 любой view; поменять атрибуты по умолчанию, добавить необходимые атрибуты
  - Рекомендуемые значения: для padding — 16dp, для textSize — 20sp
- Изменить атрибуты созданных view из java-класса
  - указать id в xml-файле
  - создать соответствующие объекты в java-классе, инициализировав их с помощью findViewById
- \* испробовать ScrollView
  - Создать ScrollView в xml-файле, поместить внутрь TextView, добавить в ScrollView список строк через java-класс