

Курс:

Программирование под Android с нуля

Тренер:

Пономарев Алексей

oleksii_android@web-academy.com.ua



Лекция 5

Структура приложения, компоненты

Немного истории

- 2003 г. - создание Android Inc.
- 2005 г. - покупка Android Inc. компанией Google
- 2008 г. - первый телефон, Android 1.x
- 2010 г. - действительно массовый Android 2.x
- Начало 2011 г. - Android 3.x для планшетов
- Конец 2011 г. – Android 4.x
- Конец 2014 г. – Android 5.x
- Конец 2015 г. – Android M (6?)

Немного статистики

Статистика мобильных ОС

- Android – 65,58%
- Apple iOS – 27,24%
- Windows Phone – 3,26%
- Nokia Symbian – 1,08%
- Другие ОС – 2,84%

Статистика Android

- 2.2 – 0,1%
- 2.3 – 1,9%
- 4.0 – 1,7%
- 4.1 – 6,4%
- 4.2 – 8,8%
- 4.3 – 2,6%
- 4.4 – 30,1%
- 5.0 – 14,3%
- 5.1 – 20,8%
- 6,0 – 13,3%

Немного теории

- Мобильные и немобильные ОС – есть ли разница?
- Файлоориентированные ОС
- Работа с памятью в разных ОС

Создание программы, часть 1

Create New Project

New Project
Android Studio

Configure your new project

Название программы

Application name: My First Application

Company Domain: web.academy.com.ua

Package name: ua.com.academy.web.myfirstapplication [Edit](#)

Уникальное название программы

Project location: D:\Android\Projects\MyFirstApplication

Previous Next Cancel Finish

Создание программы, часть 2

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 14: Android 4.0 (IceCreamSandwich)

☐ Wear
Minimum SDK: API 20: Android 4.4 (KitKat Wear)

☐ TV
Minimum SDK: L: Android L (Preview)

☐ Android Auto
Minimum SDK: L: Android L (Preview)

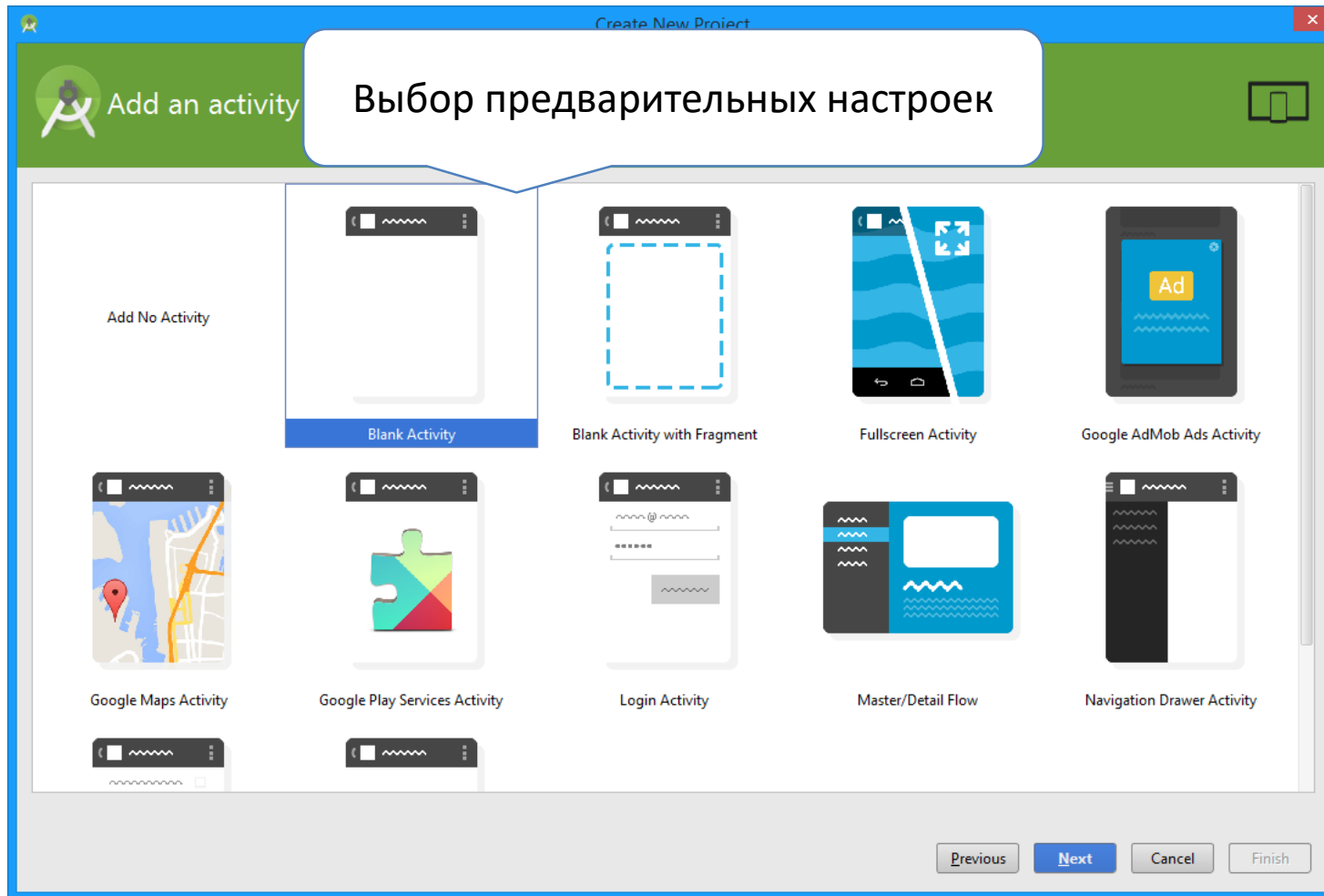
☐ Glass
Minimum SDK: L: Android L (Preview)

Lower API levels target more devices, but have fewer features available. By targeting API 14 and later, your app will run on approximately **94,0%** of the devices that are active on the Google Play Store.
[Help me choose](#)

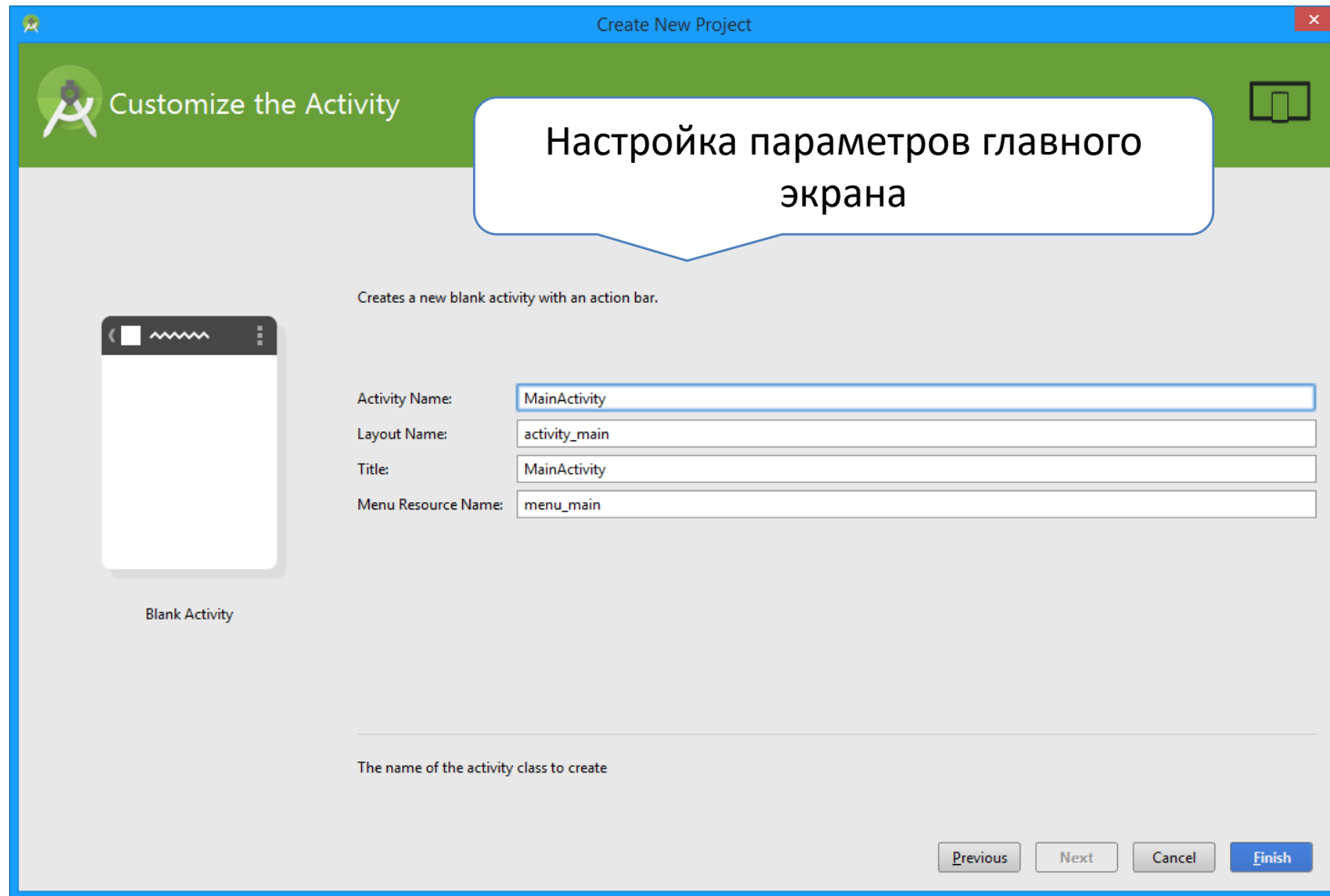
Минимальная версия Android, на которой будет работать программа

Previous Next Cancel Finish

Создание программы, часть 3



Создание программы, часть 4



Create New Project

Customize the Activity

Настройка параметров главного экрана

Creates a new blank activity with an action bar.

Blank Activity

Activity Name: MainActivity

Layout Name: activity_main

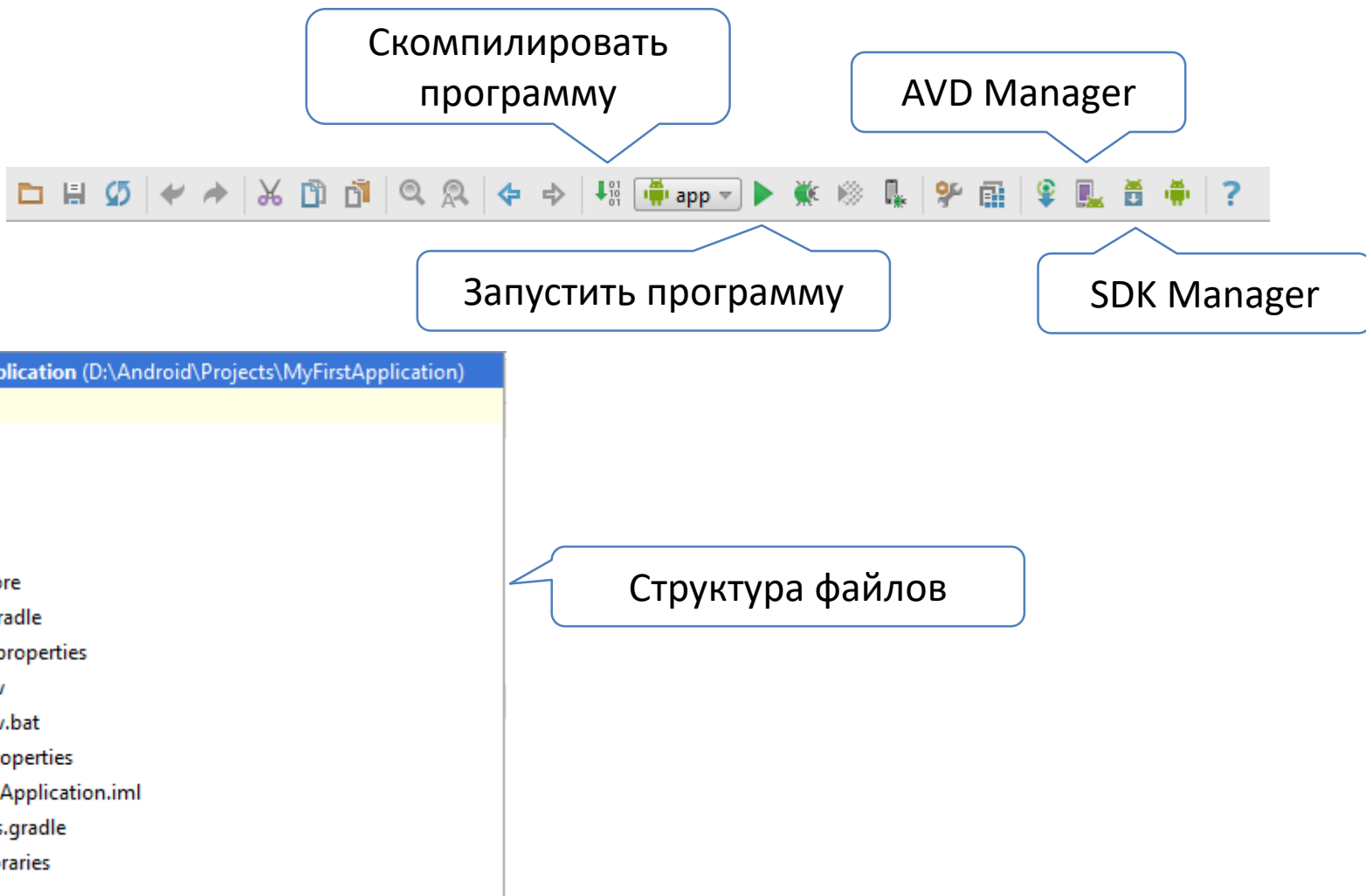
Title: MainActivity

Menu Resource Name: menu_main

The name of the activity class to create

Previous Next Cancel Finish

Android Studio, главные кнопки



Из чего это сделано

- **Resources** – ресурсы приложения, такие как тексты, картинки и т.д.
- **Activity** – это один «экран». В приложении может быть много «экранов», как например в одном сайте может быть много страниц.
- **BroadcastReceiver** – приемник сообщений. Например сообщение о приходе смс, загрузке телефона и т.д.
- **Service** – код, который выполняется без интерфейса.

Структура ресурсов

Каталог	Файл	Описание
/res/values	colors.xml	Ресурсы для цвета
	strings.xml	Тексты, надписи, названия
	dimens.xml	Числа, такие как отступы, размеры и т.д.
/res/drawable	*.png *.jpg	Картинки, логотипы и т.д.
/res/layout	*.xml	Ресурсы разметки
/res/menu	*.xml	Ресурсы меню

Взгляд с другой стороны

При создании или обновлении файла со ресурсами среда разработки автоматически создает или обновляет класс **R.java**, сообщая уникальные ID для определенных в файлах ресурсов.

То есть, для каждого ресурса, и для каждого элемента в ресурсе (цвета, текста) создается свой уникальный int ID.

```
public final class R {  
    public static final class string {  
        public static final int app_name=0x7f0a0011;  
        public static final int hello_world=0x7f0a0012;  
    }  
  
    public static final class menu {  
        public static final int menu_main=0x7f0c0000;  
    }  
    ...  
}
```

XML

- **XML** (*eXtensible Markup Language*) — расширяемый язык разметки

```
<resources>
  <string name="app_name">My Application</string>
  <string name="hello_world">Hello world!</string>
</resources>
```

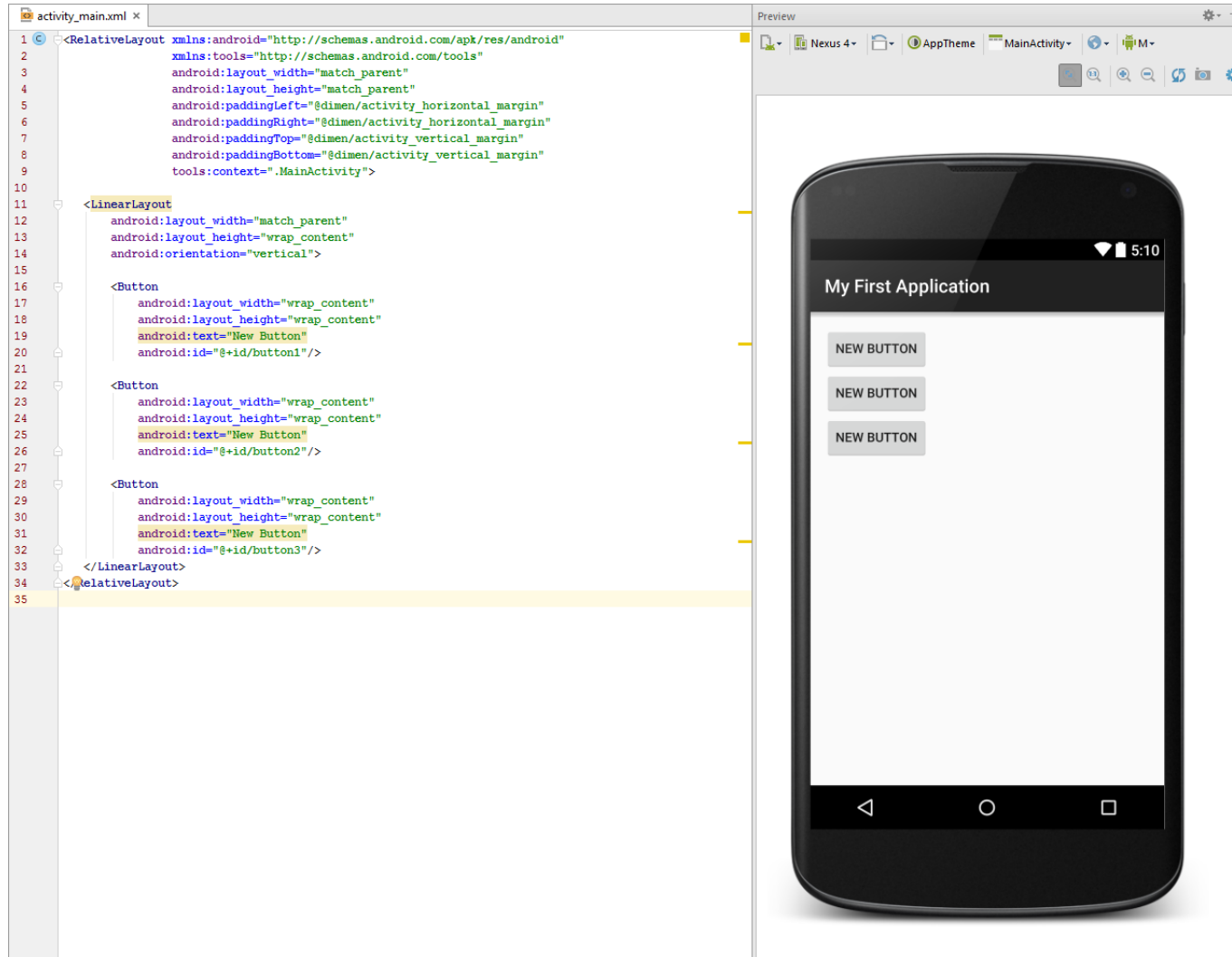
- `<resources>` - открывающий **тег**
- `</resources>` - закрывающий **тег**
- `name="app_name"` - **аттрибут**

Виды View

Widgets	
TextView	Текст
EditText	Поле ввода
Button	Кнопка
ToggleButton	Кнопка – переключатель
CheckBox	Галочка
Spinner	Выпадающий список

Layouts	
LinearLayout	Список
TableLayout	Таблица
RelativeLayout	Относительное позиционирование
FrameLayout	Один на другом
GridLayout	Продвинутая таблица

LinearLayout



TableLayout

The image displays the Android Studio interface with the XML code for `activity_main.xml` on the left and a preview of the application on a Nexus 4 device on the right.

XML Code (activity_main.xml):

```
1 <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
2   xmlns:tools="http://schemas.android.com/tools"
3   android:layout_width="match_parent"
4   android:layout_height="match_parent"
5   android:paddingBottom="@dimen/activity_vertical_margin"
6   android:paddingLeft="@dimen/activity_horizontal_margin"
7   android:paddingRight="@dimen/activity_horizontal_margin"
8   android:paddingTop="@dimen/activity_vertical_margin"
9   tools:context=".MainActivity">
10
11   <TableLayout
12     android:layout_width="match_parent"
13     android:layout_height="wrap_content">
14
15     <TableRow>
16
17       <Button
18         android:id="@+id/button1"
19         android:layout_width="wrap_content"
20         android:layout_height="wrap_content"
21         android:text="New Button"/>
22
23       <Button
24         android:id="@+id/button2"
25         android:layout_width="wrap_content"
26         android:layout_height="wrap_content"
27         android:text="New Button"/>
28     </TableRow>
29
30     <TableRow>
31
32       <Button
33         android:id="@+id/button3"
34         android:layout_width="wrap_content"
35         android:layout_height="wrap_content"
36         android:text="New Button"/>
37
38       <Button
39         android:id="@+id/button4"
40         android:layout_width="wrap_content"
41         android:layout_height="wrap_content"
42         android:text="New Button"/>
43     </TableRow>
44   </TableLayout>
45 </RelativeLayout>
46
```

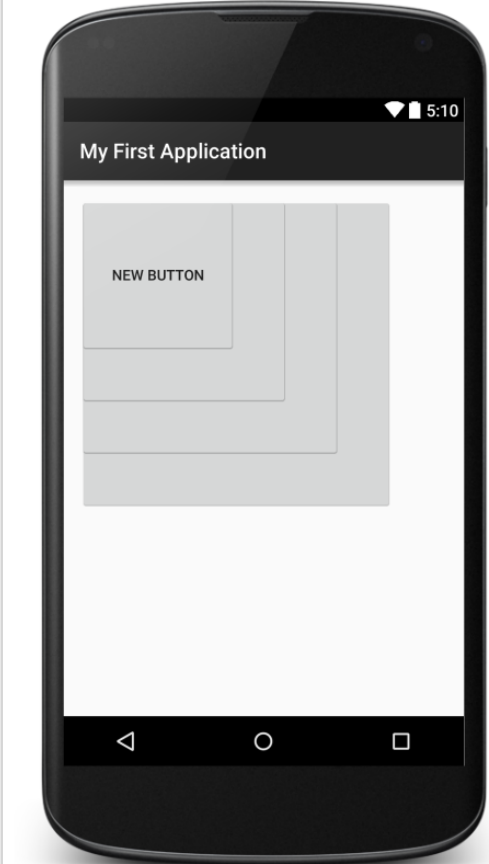
Preview: The application is running on a Nexus 4 device. The title bar shows "My First Application". The screen displays four buttons arranged in a 2x2 grid, each labeled "NEW BUTTON". The status bar at the top shows the time as 5:10 and the battery level.

RelativeLayout

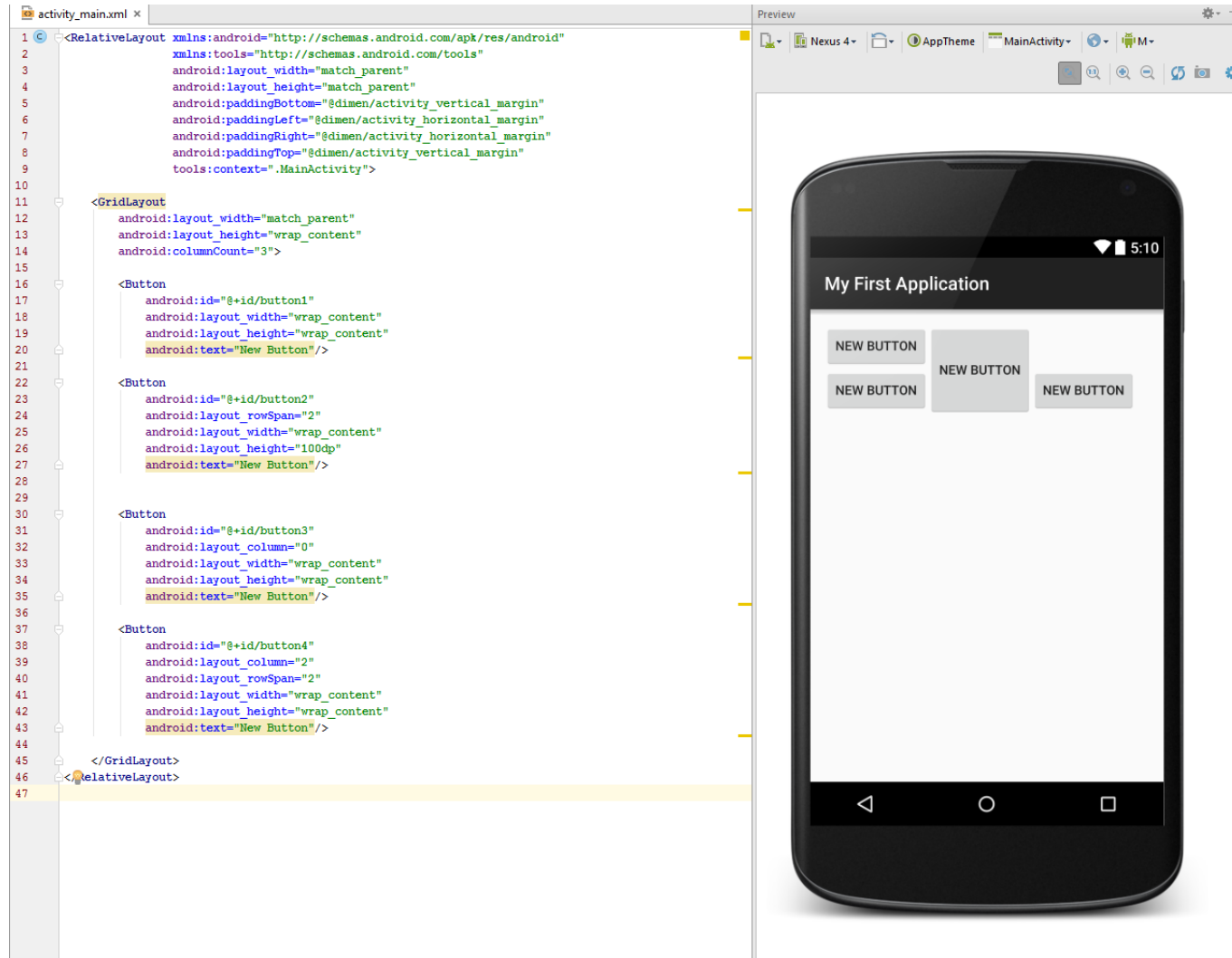


FrameLayout

```
1 <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
2   xmlns:tools="http://schemas.android.com/tools"
3   android:layout_width="match_parent"
4   android:layout_height="match_parent"
5   android:paddingBottom="@dimen/activity_vertical_margin"
6   android:paddingLeft="@dimen/activity_horizontal_margin"
7   android:paddingRight="@dimen/activity_horizontal_margin"
8   android:paddingTop="@dimen/activity_vertical_margin"
9   tools:context=".MainActivity">
10
11   <FrameLayout
12     android:layout_width="match_parent"
13     android:layout_height="match_parent">
14
15     <Button
16       android:id="@+id/button1"
17       android:layout_width="300dp"
18       android:layout_height="300dp"
19       android:text="New Button"/>
20
21     <Button
22       android:id="@+id/button2"
23       android:layout_width="250dp"
24       android:layout_height="250dp"
25       android:text="New Button"
26       android:layout_alignParentTop="true"
27       android:layout_alignParentRight="true"/>
28
29     <Button
30       android:id="@+id/button3"
31       android:layout_width="200dp"
32       android:layout_height="200dp"
33       android:text="New Button"
34       android:layout_alignParentBottom="true"
35       android:layout_alignParentLeft="true"/>
36
37     <Button
38       android:id="@+id/button4"
39       android:layout_width="150dp"
40       android:layout_height="150dp"
41       android:text="New Button"
42       android:layout_alignParentBottom="true"
43       android:layout_alignParentRight="true"/>
44
45   </FrameLayout>
46 </RelativeLayout>
```



GridLayout



Единицы измерения

- **dp или dip - Density-independent Pixels.** Абстрактная ЕИ, позволяющая приложениям выглядеть одинаково на различных экранах и разрешениях.
- **sp - Scale-independent Pixels.** То же, что и dp, только используется для размеров шрифта в View элементах

Тип экрана	Соотношение
low (ldpi)	1 dp = 0.75 пикселя
medium (mdpi)	1 dp = 1 пиксель
high (hdpi)	1 dp = 1.5 пикселя
extra high (xhdpi)	1 dp = 2 пикселя
extra-extra-high (xxhdpi)	1 dp = 3 пикселя

Что-то общее

Свойство	Значение
layout_width (ширина) и layout_height (высота)	<ul style="list-style-type: none">• match_parent – по родителю• fill_parent – то же что и match_parent, устаревшее• wrap_content – по содержимому• Размер в dp
margin (отступ снаружи) padding (отступ внутри)	Размер в dp
gravity (положение внутри) layout_gravity (положение в родителе)	top, bottom, left, etc.

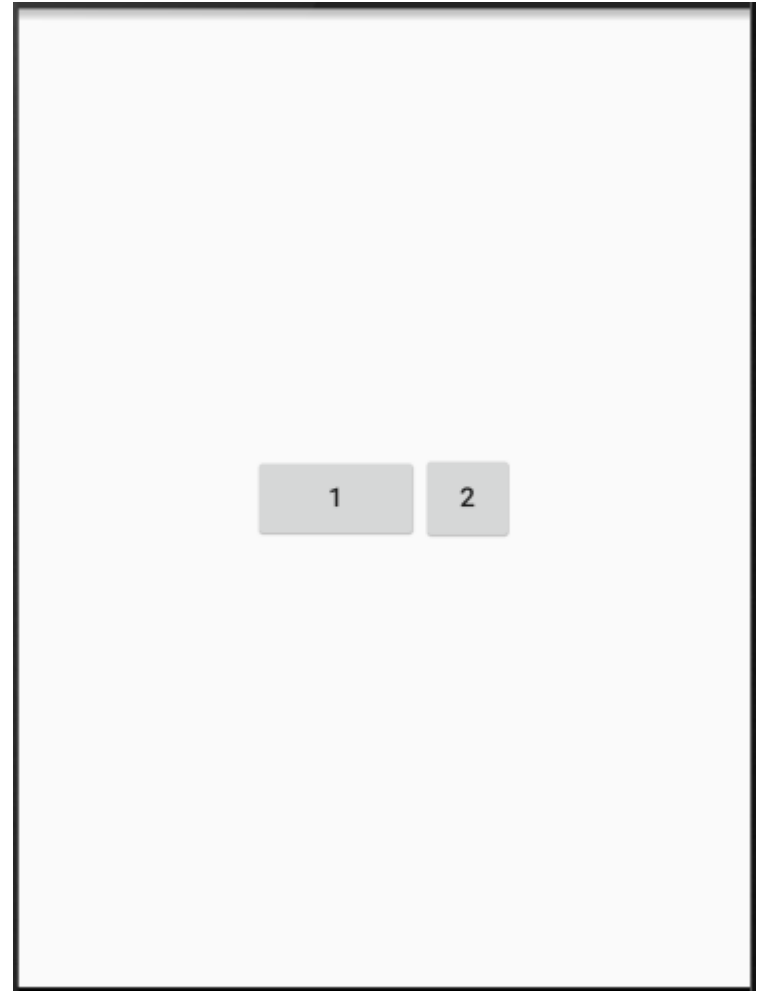
Пример установки свойств

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center">

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="1"/>

    <Button
        android:layout_width="50dp"
        android:layout_height="50dp"
        android:text="2"/>

</LinearLayout>
```



Работа с ресурсами

strings.xml:

```
<resources>
  ...
  <string name="text_from_resource">Текст из ресурса</string>
</resources>
```

<TextView

```
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Просто текст"/>
```

<TextView

```
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="@string/text_from_resource"/>
```

Просто текст

Текст из ресурса

Еще примеры

strings.xml:

```
<resources>
...
<string name="text_from_resource">Текст из ресурса</string>
</resources>
```

color.xml:

```
<resources>
...
<color name="color_from_resource">#FF459832</color>
</resources>
```

dimens.xml:

```
<resources>
...
<dimen name="dimen_from_resource">25sp</dimen>
</resources>
```

Текст из ресурса

<TextView

```
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textColor="@color/color_from_resource"
    android:textSize="@dimen/dimen_from_resource"
    android:text="@string/text_from_resource"/>
```


Activity

- **Класс**, он же код, который находится в каталоге */java*
- **Ресурс** с разметкой, в каталоге */res/layouts*

Связывание кода и разметки происходит в момент создания activity. С одним и тем же кодом можно связывать разную разметку.

Жизненный цикл Activity



Методы Activity

```
public class MainActivity extends Activity {
```

```
    @Override
```

```
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.main);  
    }
```

```
    @Override
```

```
    protected void onStart() {  
        super.onStart();  
    }
```

```
    @Override
```

```
    protected void onResume() {  
        super.onResume();  
    }
```

```
    @Override
```

```
    protected void onPause() {  
        super.onPause();  
    }
```

```
    @Override
```

```
    protected void onStop() {  
        super.onStop();  
    }
```

```
    @Override
```

```
    protected void onDestroy() {  
        super.onDestroy();  
    }  
}
```

Поиск элементов

```
<TextView  
    android:id="@+id/textView1"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:text="@string/text_from_resource"/>
```

android:id="@+id/textView1" – идентификатор элемента

```
View view = findViewById(R.id.textView1);
```

findViewById – метод в activity, который ищет элемент по его идентификатору

```
TextView view = (TextView) findViewById(R.id.textView1);
```

Указание конкретного типа элемента

Работа с элементами

- <http://developer.android.com/reference/android/view/View.html>

void	<code>setSelection(int index)</code> Convenience for <code>setSelection(Spannable, int)</code> .
void	<code>setSelection(int start, int stop)</code> Convenience for <code>setSelection(Spannable, int, int)</code> .
void	<code>setText(CharSequence text, TextView.BufferType type)</code> Sets the text that this TextView is to display (see <code>setText(CharSequence)</code>) and also sets whether it is stored in a styleable/spannable buffer and whether it is editable.

Обработка нажатий, способ 1

```
<TextView
    android:id="@+id/textView1"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="@string/text_from_resource"/>

<Button
    android:id="@+id/button1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Click me"/>
```

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

    final TextView textView = (TextView)
        findViewById(R.id.textView1);

    Button button = (Button) findViewById(R.id.button1);
    button.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            textView.setText("Button clicked!");
        }
    });
}
```

Обработка нажатий, способ 2

```
<TextView
    android:id="@+id/textView1"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="@string/text_from_resource"/>

<Button
    android:id="@+id/button1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:onClick="ButtonClick"
    android:text="Click me"/>
```

```
TextView textView;

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

    textView = (TextView) findViewById(R.id.textView1);
}

public void ButtonClick(View v) {
    textView.setText("Button clicked!");
}
```

Способ 2, продвинутый

```
<TextView
    android:id="@+id/textView1"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="@string/text_from_resource"/>
```

```
<Button
    android:id="@+id/button1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:onClick="ButtonClick1"
    android:text="Click me 1"/>
```

```
<Button
    android:id="@+id/button2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:onClick="ButtonClick1"
    android:text="Click me 2"/>
```

```
<Button
    android:id="@+id/button3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:onClick="ButtonClick2"
    android:text="Click me 3"/>
```

```
TextView textView;
```

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

```
    textView = (TextView) findViewById(R.id.textView1);
}
```

```
public void ButtonClick1(View v) {
    switch (v.getId()){
        case R.id.button1:
            textView.setText("Button 1 clicked!");
            break;
        case R.id.button2:
            textView.setText("Button 2 clicked!");
            break;
    }
}
```

```
public void ButtonClick2(View v) {
    textView.setText("Button 3 clicked!");
}
```


Спасибо за внимание

Тренер:

Пономарев Алексей

oleksii_android@web-academy.com.ua



Лекция 5

Структура приложения, компоненты