Разработка Android приложений для мобильных устройств

Лекция 2.

Элементы управления

Разработка Android приложений для мобильных устройств

Лекция 2. Элементы управления

Основные элементы управления

Разработка Android приложений для мобильных устройств

Лекция 2. Элементы управления

Создание приложения «Тест» (часть 1)

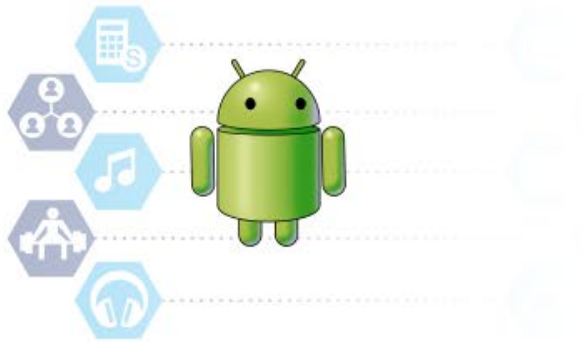
Разработка Android приложений для мобильных устройств

Лекция 2. Элементы управления

Создание приложения «Тест» (часть 2)

Создание приложения «Список сериалов» (часть 1)

Создание приложения «Список сериалов» (часть 2)



Надпись (TextView)

Используется для вывода текста.

- Для изменения размера используется атрибут android:textSize:

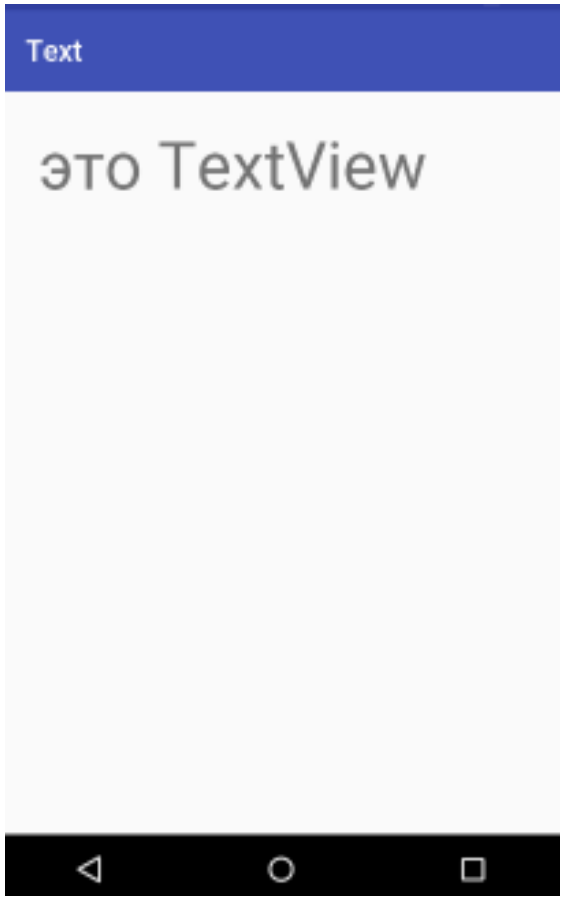
```
android:textSize="14sp"
```

- Определение в XML

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

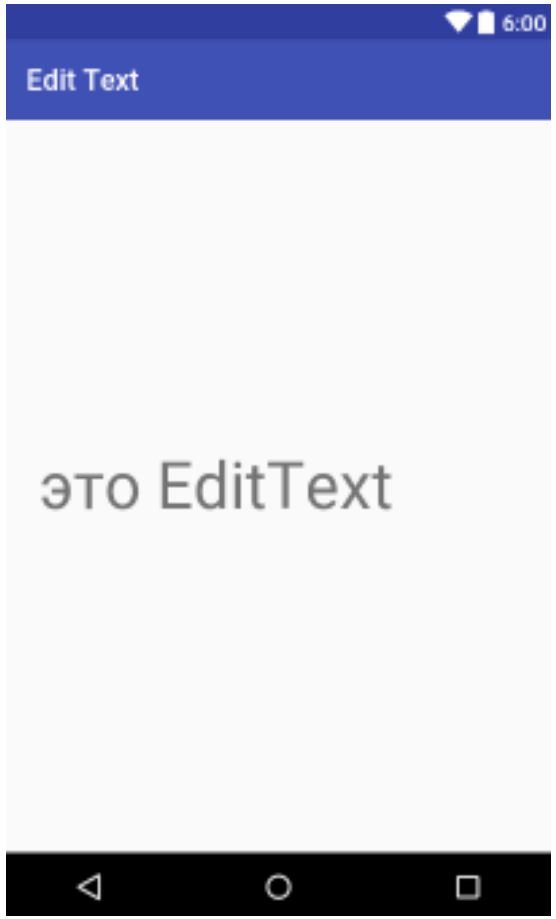
- Использование надписи в коде активности

```
TextView tV = (TextView) findViewById(R.id.textview);
textView.setText("Some other string");
```



Текстовое поле (EditText)

Аналог надписи, но с возможностью редактирования.



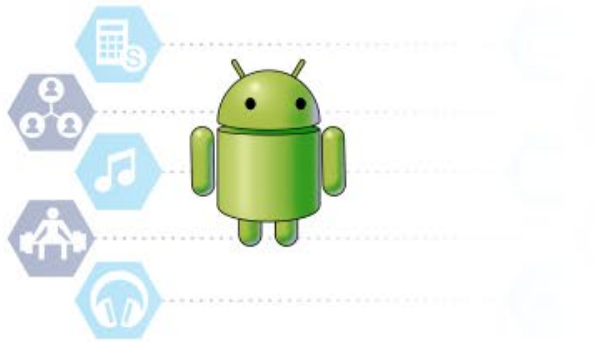
- Определение в XML

```
<EditText
    android:id="@+id/edit_text"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:hint="@string/edit_text" />
```

- Атрибут android:inputType="number"
  - phone - предоставляет клавиатуру для ввода номеров.
  - textPassword - предоставляет клавиатуру для ввода пароля.
  - textCapSentences - первое слово начинается с прописной буквы.
  - textAutoCorrect - автоматически исправляет вводимый текст.

- Использование в коде активности

```
EditText editText = (EditText) findViewById(R.id.edit_text);
String text = editText.getText().toString();
```



Кнопка (Button)

- Определение в XML

```
<Button
   android:id="@+id/button"
   android:layout_width="wrap_content"
   android:layout_height="wrap_content"
   android:text="@string/button_text" />
```

- Использование в коде активности

```
android:onClick="onButtonClicked"
```

Затем в активности определяется метод следующего вида:

```
    /** Вызывается при щелчке на кнопке */
public void onButtonClicked(View view) {
   // Сделать что-то по щелчку на кнопке
}
```

Вывод изображений на кнопках (Button)

- Вывод текста и изображения на кнопке

Вывести графический ресурс android в правой части кнопки.

```
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:drawableRight="@drawable/android"
    android:text="@string/click_me" />
```

Чтобы изображение располагалось слева от текста, воспользуйтесь атрибутом android:drawableLeft:

```
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:drawableLeft="@drawable/android"
    android:text="@string/click_me" />
```

Двухпозиционная кнопка (ToggleButton)

Щелкая на двухпозиционной кнопке, пользователь выбирает одно из двух состояний.

- Определение в XML

```
<ToggleButton android:layout_width="wrap_content"
   android:layout_height="wrap_content"
   android:textOn="@string/on"
   android:textOff="@string/off" />
```

- Использование в коде активности

```
   android:onClick="onToggleButtonClicked"
```

Затем в активности определяется метод следующего вида:

```
      /** Вызывается при щелчке на двухпозиционной кнопке*/
 public void onToggleClicked(View view) {
    // Получить состояние двухпозиционной кнопки.
   boolean on = ((ToggleButton) view).isChecked();
   if (on) {
// Вкл
   } else {
// Выкл
   }
 }
```

Переключатель(Switch)

Выключатель представляет собой рычажок, который работает по тому же принципу, что и двухпозиционная кнопка.

- Определение в XML

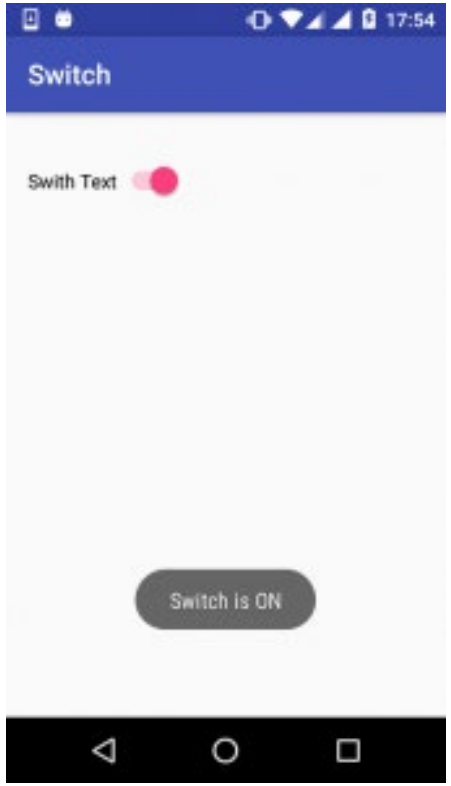
```
<Switch
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:textOn="@string/on"
      android:textOff="@string/off" />
```

- Использование в коде активности

```
android:onClick="onSwitchClicked"
```

Затем в активности определяется метод следующего вида:

```
    /** Вызывается при щелчке на выключателе. */
public void onSwitchClicked(View view) {
    // Включенное состояние?
boolean on = ((Switch) view).isChecked();
   if (on) {
            // Вкл
   } else {
            // Выкл
   }
}
```



Флажки (CheckBox)

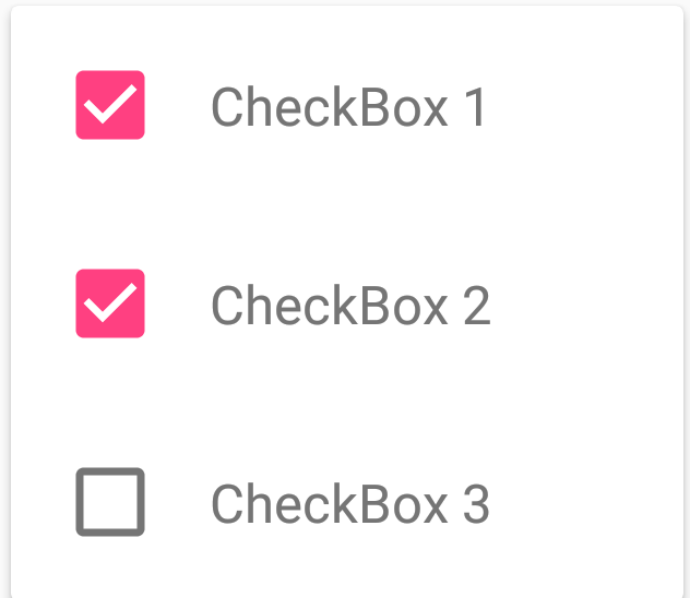
Флажки (check boxes) предоставляют пользователю набор независимых вариантов. Каждый флажок может устанавливаться или сниматься независимо от всех остальных флажков.

- Определение в XML

```
<CheckBox
...
android:text="@string/content1" />

<CheckBox
...
android:text="@string/content2" />
```

- Использование в коде активности

```
CheckBox cb = (CheckBox) findViewById(R.id.check);
boolean checked = checkbox.isChecked();
if (checked) {
        //Действия для установленного флажка
}
```

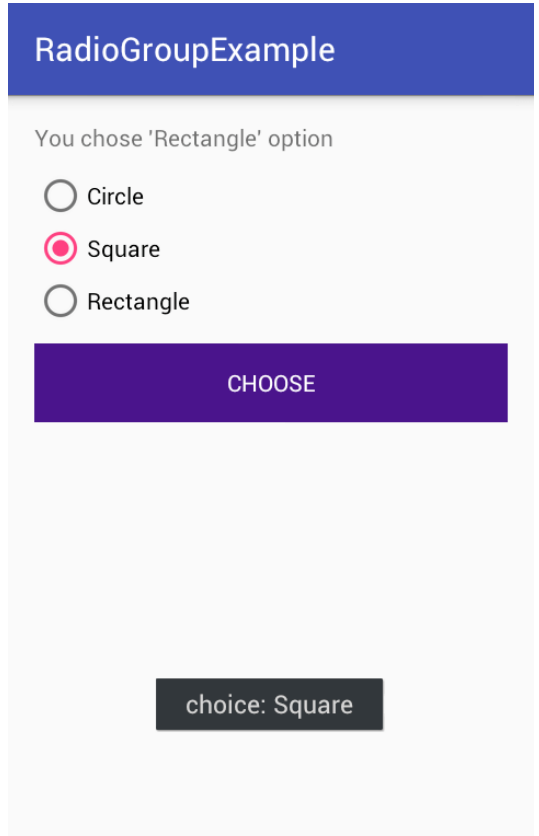


Переключатели (RadioButton)

Переключатели предоставляют набор вариантов, из которого пользователь может выбрать ровно один вариант:

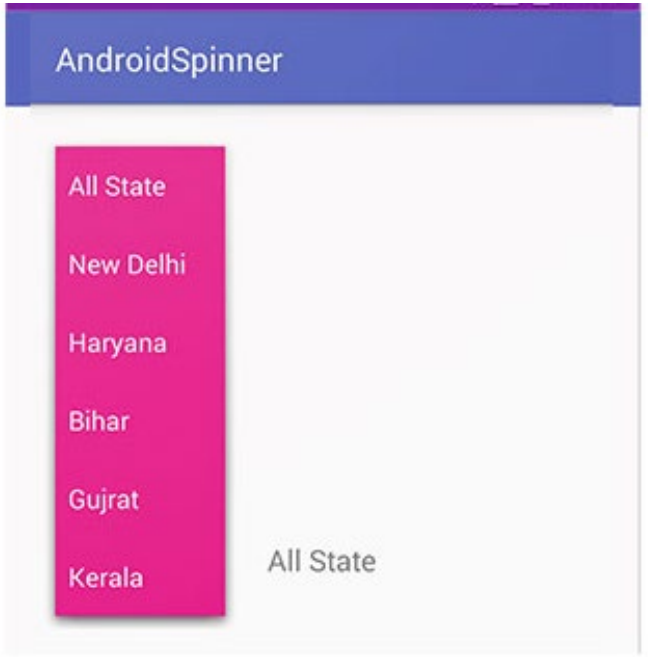
- Определение в XML

```
<RadioGroup
    android:id="@+id/radio_group"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical">

    <RadioButton
        android:id="@+id/radio_circle"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/circle" />
    <RadioButton
        android:id="@+id/radio_square"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/square" />

</RadioGroup>
```



Раскрывающийся список (Spinner)

Раскрывающийся список содержит набор значений, из которых
пользователь может выбрать только одно.



Определение в XML

```
<Spinner
    android:id="@+id/spinner"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:entries="@array/spinner_values" />
```

Массив строк добавляется в файл *strings.xml :*

```
<string-array name="spinner_values">
    <item>New Delphi</item>
    <item>Haryana</item>
    <item>Bihar/item>
    <item>Guijrat</item>
</string-array>
```

Использование в коде активности

```
Spinner spinner = (Spinner) findViewById(R.id.spinner);
String string = String.valueOf(spinner.getSelectedItem());
```

Графическое представление (ImageView)

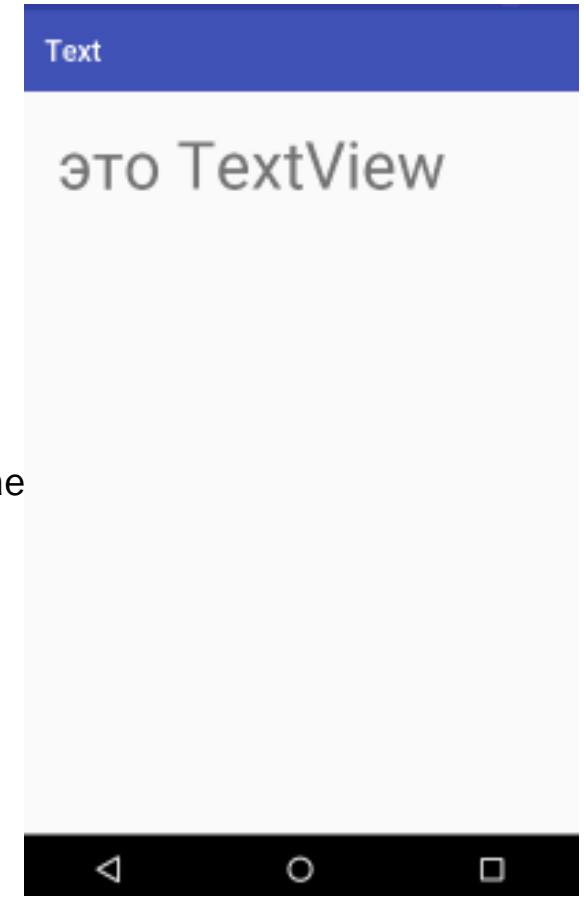
- Определение в XML макета

```
<ImageView
android:layout_width="200dp"
android:layout_height="100dp"
android:src="@drawable/logo"
android:contentDescription="@string/logo" />
```

Ресурсы изображений снабжаются префиксом @drawable, который сообщае Android, что ресурс изображения хранится в одной или нескольких папках *drawable.*

- Использование в коде активности

```
ImageView photo = (ImageView)findViewById(R.id.photo);
int image = R.drawable.logo;
String description = "This is the logo";
photo.setImageResource(image);
photo.setContentDescription(description);
```



Графическая кнопка (ImageButton)

Графическая кнопка почти не отличается от обычной — просто на ней выводится только изображение, без текста.

- Определение в XML

```
<ImageButton
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:src="@drawable/button_icon />
```

- Использование в коде активности

```
android:onClick="onButtonClicked"
```

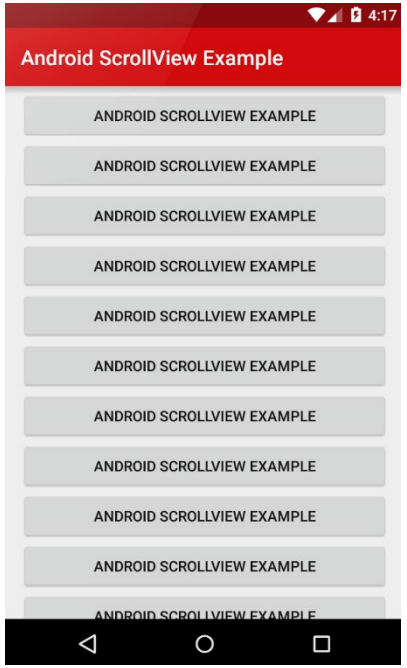
Затем в активности определяется метод следующего вида:

```
    /** Вызывается при щелчке на кнопке */
public void onButtonClicked(View view) {
    // Сделать что-то по щелчку на кнопке
}
```

Прокручиваемые представления (ScrollView)

- Чтобы добавить вертикальную полосу прокрутки, заключите существующий макет в элемент <ScrollView>:

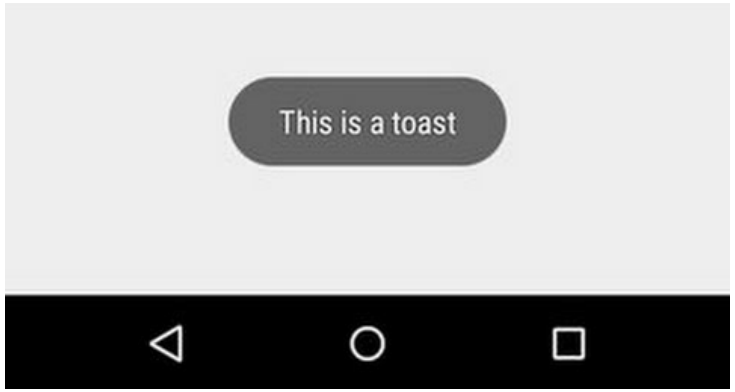
```
<ScrollView
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".MainActivity" >

    <LinearLayout
android:layout_width="match_parent"
android:layout_height="match_parent"
android:paddingBottom="16dp"
android:paddingLeft="16dp"
android:paddingRight="16dp"
android:paddingTop="16dp"
android:orientation="vertical" >
    ...
    </LinearLayout>
<ScrollView
```



- Чтобы добавить в макет горизонтальную полосу прокрутки, заключите существующий макет в элемент <HorizontalScrollView>.

Уведомления (Toast)

Уведомления выполняют чисто информационные функции, пользователь не может с ними взаимодействовать. Пока уведомление находится на экране, активность остается видимой и доступной для взаимодействия с пользователем. Уведомление автоматически закрывается по истечении тайм-аута.



- Использование в коде активности:

1. Уведомления создаются только в коде активности;
2. Определить их в макете невозможно.

```
CharSequence text = "Hello, I'm a Toast!";
int duration = Toast.LENGTH_SHORT;
Toast toast = Toast.makeText(this, text, duration);
toast.show();
```