



Лекция #15. Отображение графики (ImageView)

Добавление изображений в проект

Папка **drawable** , расположенная по адресу **app/src/main/res/drawable**, используется по умолчанию для хранения ресурсов изображений. Чтобы добавить файл с изображением, просто перетащите его в папку.

Определение в XML макета

Графическое представление определяется в **XML** элементом **<ImageView>**. Атрибут **android:src** указывает, какое изображение должно выводиться. Атрибут **android:contentDescription** позволяет добавить текстовое описание изображения, чтобы сделать приложение более доступным:

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

Значение атрибута **android:src** задается в форме **"@drawable/имя_изображения"**, где имя_изображения — имя файла изображения (без расширения).

Префикс **@drawable** сообщает **Android**, что ресурс изображения хранится в одной или нескольких папках **drawable**.

Использование в коде активности

Исходное изображение и его текстовое описание задаются в коде активности методами

- **setImageResource(int resId)**

- **setContentDescription(String description)**

Кроме того, в классе **ImageView** определены методы для установки размеров изображения:

- **setMaxHeight(int maxHeight)**
- **setMaxWidth(int maxWidth)**
- **getMinimunHeight()**
- **getMinimunWidth()**

и его масштабирования:

- **getScaleType ()**
- **setScaleType (ImageView.ScaleType scaleType scaleType)**

Пример:

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

Этот фрагмент кода ищет ресурс изображения с именем **mggtk_logo** в папках **drawable*** и назначает его источником данных для графического представления с идентификатором **photo**.

ImageButton

Виджет **ImageButton** представляет собой кнопку с изображением (вместо текста). При смене состояния **ImageButton**, происходит смена текущего изображения. Для установки изображения на поверхность кнопки используется атрибут: **android:src**, в программном коде мы используем метод: **setImageResource(int);**

ToggleButton

Виджет **ToggleButton**— это двухпозиционная кнопка с двумя состояниями: "включено" и "выключено". По умолчанию на кнопке определены надписи ON/OFF и LED - индикатор, изменяющий цвет на зеленый при переходе в состояние ON.

Определение в XML

Кнопка определяется в **XML** элементом **<ToggleButton>**. Атрибуты **android:textOn** и **android:textOff** определяют текст, который должен выводиться на двухпозиционной кнопке в зависимости от ее состояния:

```
<ToggleButton
    android:id="@+id/toggleButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textOn="@string/on"        //включено
    android:textOff="@string/off"     //выключено
```

В программном коде им соответствуют методы **setTextOff()** и **setTextOn()**, устанавливающие надписи на кнопке в разных состояниях.

Использование в коде активности

Чтобы кнопка реагировала на щелчки, необходимо включить атрибут **android:onClick** и присвоить имя вызываемого метода из кода в **XML** макете.

- **android:onClick="onToggleButtonClicked"**

Далее в активности определяется метод следующего вида

```
fun toggleClick(view: View) {
    val on: Boolean = (view as ToggleButton).isChecked
    if (on) { //включено
        //пишем здесь действия
    } else { // выключено
        //пишем здесь действия
    }
}
```

Метод **setChecked(boolean checked)** позволяет программно менять состояние кнопки.