**Работа с ресурсами**

Ресурс в приложении Android представляет собой файл, например, файл разметки интерфейса или некоторое значение, например, простую строку. То есть ресурсы представляют собой и файлы разметки, и отдельные строки, и звуковые файлы, файлы изображений и т.д. Все ресурсы находятся в проекте в каталоге *res*. Для различных типов ресурсов, определенных в проекте, в каталоге *res* создаются подкаталоги. Поддерживаемые подкаталоги:

* **animator/**: xml-файлы, определяющие анимацию свойств
* **anim/**: xml-файлы, определяющие tween-анимацию
* **color/**: xml-файлы, определяющие список цветов
* **drawable/**: Графические файлы (*.png*, *.jpg*, *.gif*)
* **mipmap/**: Графические файлы, используемые для иконок приложения под различные разрешения экранов
* **layout/**: xml-файлы, определяющие пользовательский интерфейс приложения
* **menu/**: xml-файлы, определяющие меню приложения
* **raw/**: различные файлы, которые сохраняются в исходном виде
* **values/**: xml-файлы, которые содержат различные используемые в приложении значения, например, ресурсы строк
* **xml/**: Произвольные xml-файлы

В общей сложности мы можем определить следующие типы ресурсов:

|  |  |  |  |
| --- | --- | --- | --- |
| **Ресурс** | **Каталог проекта** | **Файл** | **элемент в файле** |
| Строки | /res/values/ | strings.xml | <string> |
| Plurals | /res/values/ | strings.xml | <plurals> |
| Массивы строк | /res/values/ | strings.xml или arrays.xml | <string-array> |
| Логические значения Boolean | /res/values/ | bools.xml | <bool> |
| Цвета | /res/values/ | colors.xml | <color> |
| Список цветов | /res/color/ | Произвольное название | <selector> |
| Размеры (Dimensions) | /res/values/ | dimens.xml | <dimen> |
| Идентификаторы ID | /res/values/ | ids.xml | <item> |
| Целые числа | /res/values/ | integers.xml | <integer> |
| Массив целых чисел | /res/values/ | integers.xml | <integer-array> |
| Графические файлы | /res/drawable/ | Файлы с расширением jpg и png | - |
| Tween-анимация | /res/anim/ | Файл xml с произвольным названием | <set>, <alpha>, <rotate>, <scale>, <translate> |
| Покадровая анимация | /res/drawable/ | Файл xml с произвольным названием | <animation-list> |
| Анимация свойств | /res/animator/ | Файл xml с произвольным названием | <set>, <objectAnimator>, <valueAnimator> |
| Меню | /res/menu/ | Файл xml с произвольным названием | <menu> |
| XML-файлы | /res/xml/ | Файл xml с произвольным названием |  |
| Бинарные и текстовые ресурсы | /res/raw/ | Файлы мультимедиа (mp3, mp4), текстовые и другие файлы |  |
| Разметка графического интерфейса | /res/layout/ | Файл xml с произвольным названием |  |
| Стили и темы | /res/values/ | styles.xml, themes.xml | <style> |

К примеру, если мы возьмем стандартный проект Android Studio, который создается по умолчанию, то там можем заметить наличие уже нескольких папок для различных ресурсов в каталоге *res*:

По умолчанию здесь есть каталоги не для всех типов ресурсов, которые использоваться в Android, однако при необходимости мы можем добавить в папку *res* нужный каталог, а в него затем поместить ресурс.

Когда происходит компиляция проекта сведения обо всех ресурсах добавляются в специальный файл **R.java**, который можно найти в проекте по пути *app\build\generated\source\r\debug\[пакет\_приложения]*. Для его просмотра надо переключиться в полноценный вид проекта:

В этом файле будут храниться все определения ресурсов в виде числовых констант. К примеру, в проекте по умолчанию есть ресурс разметки интерфейса - файл *activity\_main.xml* в папке *res/layout*. Для этого ресурса в классе R будет создаваться примерно следующая константа:

|  |
| --- |
| public final class R {  ................................  public static final class layout {          public static final int activity\_main=0x7f030001;      }  ........................................ |

**Применение ресурсов**

Существует два способа доступа к ресурсам: в файле исходного кода и в файле xml.

**Ссылка на ресурсы в коде**

Тип ресурса в данной записи ссылается на одно из пространств (внутренних классов), определенных в файле R.java, которые имеют соответствующие им типы в xml:

* R.drawable (ему соответствует тип в xml drawable)
* R.id (id)
* R.layout (layout)
* R.string (string)
* R.attr (attr)
* R.plural (plurals)
* R.array (string-array)

Например, для установки ресурса activity\_main.xml в качестве графического интерфейса в коде MainActivity в методе onCreate() есть такая строка:

|  |
| --- |
| setContentView(R.layout.activity\_main); |

Через выражение R.layout.activity\_main мы и ссылаемся на ресурс activity\_main.xml, где layout - тип ресурса, а activity\_main - имя ресурса.

Подобным образом мы можем получать другие ресурсы. Например, в файле *res/values/strings.xml* определен ресурс app\_name:

|  |
| --- |
| <resources>      <string name="app\_name">ViewsApplication</string>  </resources> |

Этот ресурс ссылается на строку. Чтобы получить ссылку на данный ресурс в коде java, мы можем использовать выражение R.string.app\_name.

**Доступ в файле xml**

Нередко возникает необходимость ссылаться на ресурс в файле xml, например, в файле, который определяет визуальный интерфейс, к примеру, в activity\_main.xml. Ссылки на ресурсы в файлах xml имеют следующую формализованную форму: @[имя\_пакета:]тип\_ресурса/имя\_ресурса

* имя\_пакета представляет имя пакета, в котором ресурс находится (указывать необязательно, если ресурс находится в том же пакете)
* тип\_ресурса представляет подкласс, определенный в классе R для типа ресурса
* имя\_ресурса имя файла ресурса без расширения или значение атрибута android:name в XML-элементе (для простых значений).

Например, мы хотим вывести в элемент TextView строку, которая определена в виде ресурса в файле strings.xml:

|  |  |
| --- | --- |
| 1 |  |

**Метод getResources**

Для получения ресурсов в классе Activity мы можем использовать метод **getResources()**, который возвращает объект**android.content.res.Resources**. Но чтобы получить сам ресурс, нам надо у полученного объекта Resources вызвать один из методов:

* **getString()**: получает строку из файла strings.xml по числовому идентификатору
* **getDimension()**: получает числовое значение - ресурс dimen
* **getDrawable()**: получает графический файл
* **getBoolean()**: получает значение boolean

Это только некоторые методы. Но вкратце рассмотрим их применение. Возьмем тот же файл *res/values/strings.xml* в качестве источника ресурсов. Пусть файл **strings.xml** выглядит так:

|  |
| --- |
| <resources>      <string name="app\_name">ViewsApplication</string>  </resources> |

И изменим код MainActivity:

|  |
| --- |
| package com.example.eugene.viewsapplication;    import android.support.v7.app.AppCompatActivity;  import android.os.Bundle;  import android.widget.TextView;    public class MainActivity extends AppCompatActivity {        protected void onCreate(Bundle savedInstanceState) {          super.onCreate(savedInstanceState);            // получение ресурсов из файла values/strings.xml          String app\_name = getResources().getString(R.string.app\_name);            TextView textView = new TextView(this);          textView.setText(app\_name);            setContentView(textView);      }  } |

Здесь, используя метод getResources() получаем все ресурсы. Числовые ресурсы устанавливаем из в качестве отступов внутри элемента TextView, а строковый ресурс - в качестве текста. При запуске приложения мы увидим применение этого ресурса: