Лабораторная работа № 5

«Работа с ресурсами в Android-приложении»

Продолжительность выполнения лабораторной работы: 2 ак. часа.

Целью лабораторной работы является знакомство с различными типами ресурсов и их использованием в Android-приложениях.

Теоретическая часть

Ресурсы — это неотъемлемая часть Android-приложений. Это внешние элементы, которые включаются в приложение: изображения, аудио, видео, строки, компоновки, темы и т. д. Ресурсы хранятся в отдельных файлах во вложенных папках в каталоге res. Доступ к ресурсам осуществляется через класс R, который автоматически создается инструментом aapt. Ресурсы легко доступны для использования в коде программы. Во время компиляции приложения инструмент aapt создает класс R, в котором находятся идентификаторы для всех ресурсов в каталоге res/. Для каждого типа ресурсов предусмотрен подкласс R (например, класс R.drawable для изображений), а для каждого ресурса указанного типа создаётся статическая целочисленная переменная. Эта переменная как раз и служит идентификатором ресурса, который можно использовать для его получения.

В таблице 1 представлены типы ресурсов, которые могут использоваться в Android-приложениях. Каталог res/ содержит все ресурсы в подкаталогах, названия которых приведены в левом столбце.

Таблица 1. Типы ресурсов, использующиеся в Android-приложениях.

Папка	**Описание ресурса**
animator	Файлы XML, которые определяют анимации свойств, например, анимацию изменения цвета.
anim	Файлы XML, которые определяют анимации преобразований (<alpha>, <scale>, <translate>, <rotate>). (Анимации свойств также можно сохранять в этом каталоге, но для них предпочтительнее использовать каталог animator/, чтобы различать эти два типа).
color	Файлы XML, которые определяют список состояний цветов.
drawable	Файлы растровых изображений (.png, .9.png, .jpg, .gif) или файлы XML, например, Shape Drawable.
mipmap	Графические файлы для иконок приложения для экранов с различной плотностью.
layout	Файлы XML, которые определяют макет пользовательского интерфейса (компоновки).

Продолжение таблицы 1.

menu	Файлы XML, которые определяют меню приложения, такие как меню параметров, контекстные меню или вложенные меню.
raw	Произвольные файлы для сохранения в исходной форме. Открыть эти ресурсы можно с помощью InputStream, вызвав Resources.openRawResource() и передав в данный метод идентификатор ресурса (R.raw.filename). В том случае, если требуется получить доступ к исходным именам файлов и иерархии файлов, можно сохранять ресурсы в каталоге assets/ (вместо каталога res/raw/). Файлы в каталоге assets/ не получают идентификатора ресурса.
values	Файлы XML, которые содержат простые значения, такие как строки, целые числа и цвета. Тогда как XML-файлы ресурсов в других подкаталогах каталога res/ определяют отдельные ресурсы на базе имени файла XML, файлы в каталоге values/ описывают несколько ресурсов. Для файла в этом каталоге каждый дочерний элемент элемента <resources> определяет один ресурс. Например, элемент <string> создает ресурс R.string, а элемент <color> создает ресурс R.color. Так как каждый ресурс определяется с помощью своего собственного элемента XML, можно назначать имя файла по своему усмотрению и помещать ресурсы разных типов в один файл. Тем не менее, существуют соглашения для имен файлов ресурсов: • arrays.xml для ресурсов-массивов (массивы с указанием типа) • colors.xml для значений цветов • dimens.xml для значений единиц измерений • strings.xml для строковых значений • styles.xml для стилей.
xml	Произвольные XML-файлы, которые можно читать в режиме выполнения вызовом метода Resources.getXML().
font	Ресурс шрифта определяет пользовательский шрифт, который вы можете использовать в своем приложении. Шрифты могут быть отдельными файлами шрифтов (.ttf, .ttc, .otf) или набором файлов шрифтов, известным как семейство шрифтов и определенным в XML.

Доступ к ресурсу можно получить из java-кода и из xml. Из кода доступ получается с помощью статической целочисленной переменной из подкласса вашего класса R, например:

R.string.name

string — это тип ресурса, а name — это имя ресурса.

Из XML доступ получается с помощью особого синтаксиса, который также соответствует идентификатору ресурса, заданному в классе R, например:

@string/name

string — это тип ресурса, a name — это его имя. Этот синтаксис можно использовать в любом фрагменте ресурса XML, где ожидается значение, указанное вами в ресурсе.

Альтернативные ресурсы

Для ресурсов любого типа можно указать ресурс *по умолчанию* и несколько *альтернативных* ресурсов для приложения:

- Ресурсы по умолчанию должны использоваться независимо от конфигурации устройства или в том случае, когда отсутствуют альтернативные ресурсы, соответствующие текущей конфигурации.
- Альтернативные ресурсы предназначены для работы с определенными конфигурациями. Чтобы указать, что группа ресурсов предназначена для определенной конфигурации, необходимо добавить соответствующий квалификатор к имени каталога.

Например, в качестве квалификатора может быть указана **плотность экрана устройства** (ldpi, mdpi, hdpi, xhdpi и т.д.) Такие квалификаторы часто указываются для папок drawable и mipmap. Так в папки drawable-mdpi, drawable-hdpi, drawable-xhdpi размещают одинаковые изображения с одинаковыми именами, но с разным разрешением для качественного отображения на устройствах с разной плотностью экрана. В зависимости от плотности экрана реального устройства изображения автоматически подгружаются из соответствующей папки.

Другим примером квалификаторов могут служить квалификаторы, указывающие **язык и регион**, для которых предназначены данные ресурсы. С помощью этих квалификаторов можно легко создавать мультиязычные приложения.

Язык задается двухбуквенным кодом языка ISO 639-1 (например, ru, en, fr), к которому можно добавить двухбуквенный код региона ISO 3166-1-alpha-2 с предшествующей строчной буквой "r" (например, en-rUS). Коды не зависят от регистра, нельзя указывать только код региона.

В случае использования ресурсов с указанными квалификаторами языка и региона, эти ресурсу будут подгружаться в соответствии с тем, какой язык и регион задан пользователем в системных настройках устройства.

Также в качестве квалификатора может быть указана **ориентация экрана устройства**: port (устройство в портретной (вертикальной) ориентации) и land (устройство в книжной (горизонтальной) ориентации). Ориентация может измениться за время работы приложения, если пользователь поворачивает экран. Таким образом, можно создавать компоновки с различным расположением элементов пользовательского интерфейса в зависимости от ориентации экрана.

Например, несмотря на то, что макет пользовательского интерфейса по умолчанию сохранен в каталоге res/layout/, можно указать другой макет для использования на экране с альбомной ориентацией, сохранив его в каталоге res/layout-land/. Android автоматически применяет соответствующие ресурсы, сопоставляя текущую конфигурацию устройства с именами каталогов ресурсов.

Практическая часть

Разработаем приложение, использующее различные типы ресурсов (графические, строковые, меню, цвета, стили и темы).

Создайте новый проект с именем Resources. При создании проекта выберите создание пустого активити (англ. Empty Activity). Откройте папку ресурсов (рис. 1).

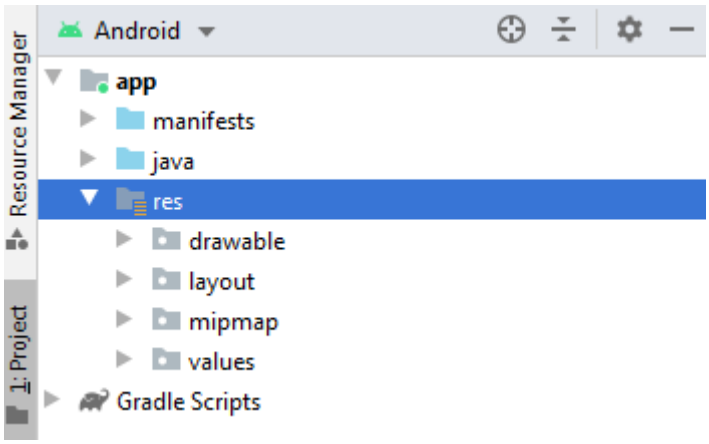

Рисунок 1. Папка ресурсов.

Скопируйте в папку drawable графический ресурс с именем background (например, background.jpg или background.png), который будет использоваться как фоновое изображение для активити.

В файле strings.xml, расположенном в папке values, создайте следующие строковые ресурсы (листинг 1).

Листинг 1. Файл strings.xml

```
<resources>
    <string name="app_name">Resources</string>
    <string name="text">Hello world!</string>
    <string name="red">Red</string>
    <string name="green">Green</string>
    <string name="blue">Blue</string>
    <string name="color">Color</string>
    <string name="exit">Exit</string>
</resources>
```

Затем создайте файл для переведённых на русский язык строковых ресурсов (рис. 2, рис. 3).

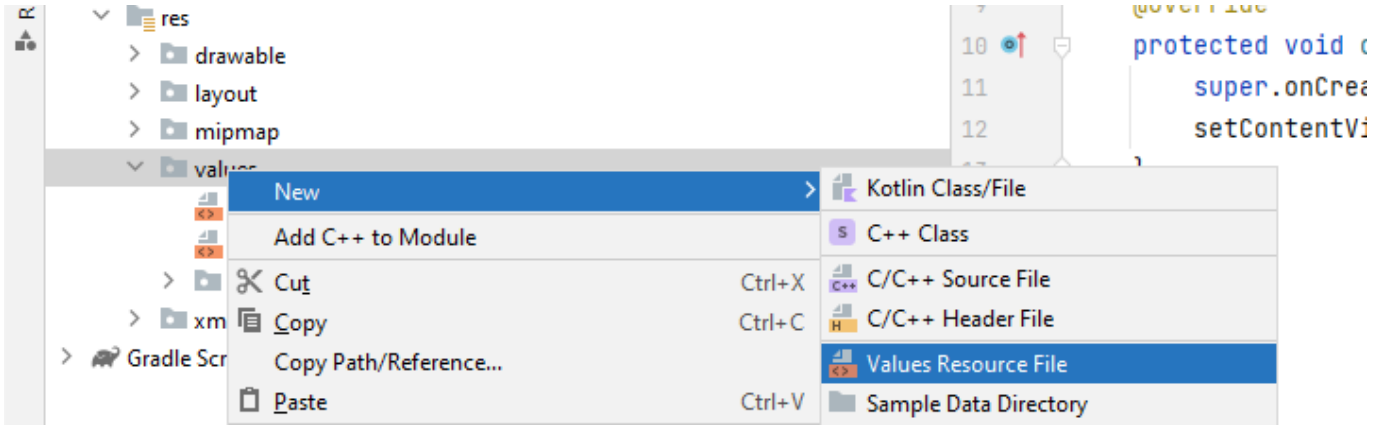

Рисунок 2. Создание нового файла ресурсов.

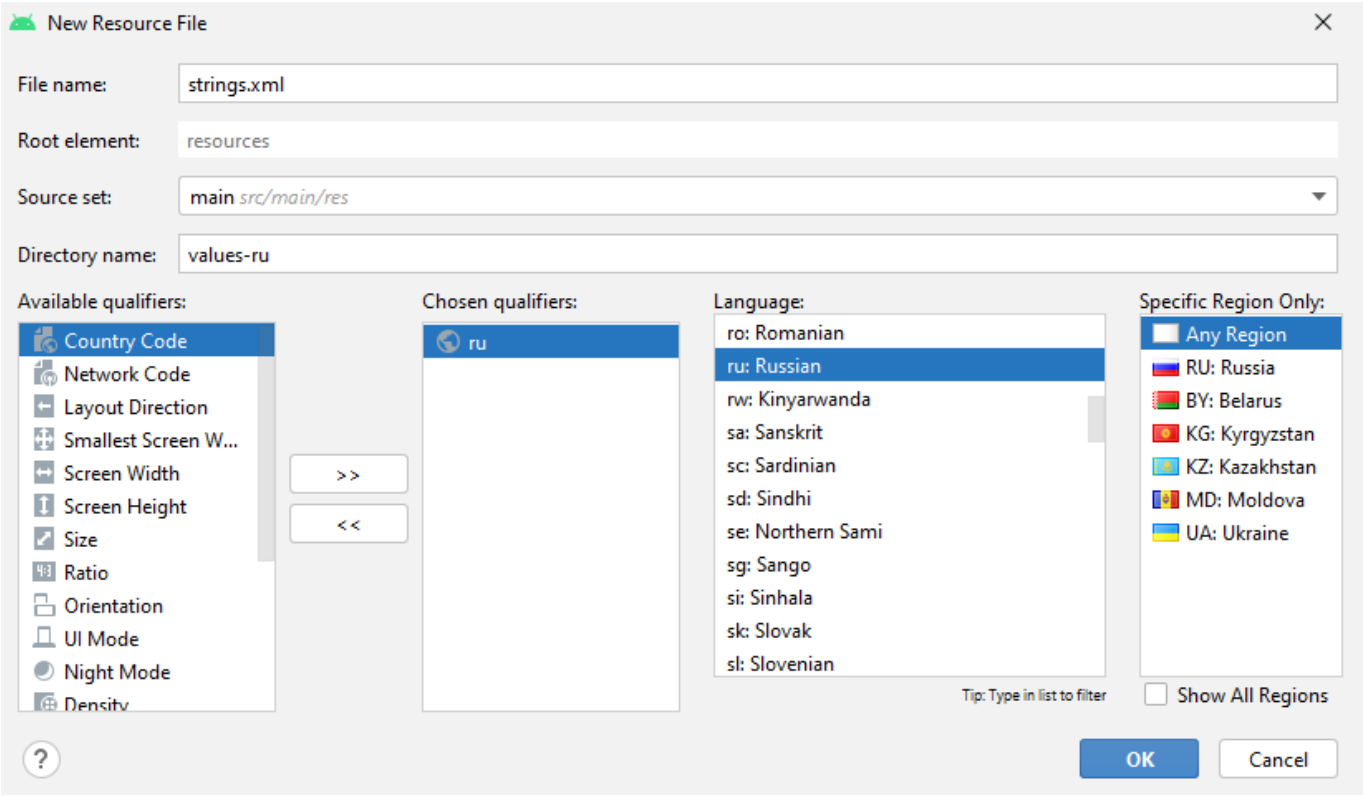

Рисунок 3. Создание строковых ресурсов для русского языка.

В результате будет создана папка values-ru с файлом stings.xml (рис. 4), в который нужно поместить переведённые строковые ресурсы (листинг 2).

Листинг 2. Файл strings.xml (ru)

```
<resources>
    <string name="app_name">Ресурсы</string>
    <string name="text">Привет, мир!</string>
    <string name="red">Красный</string>
    <string name="green">Зелёный</string>
    <string name="blue">Синий</string>
    <string name="color">Цвет</string>
    <string name="exit">Выход</string>
</resources>
```

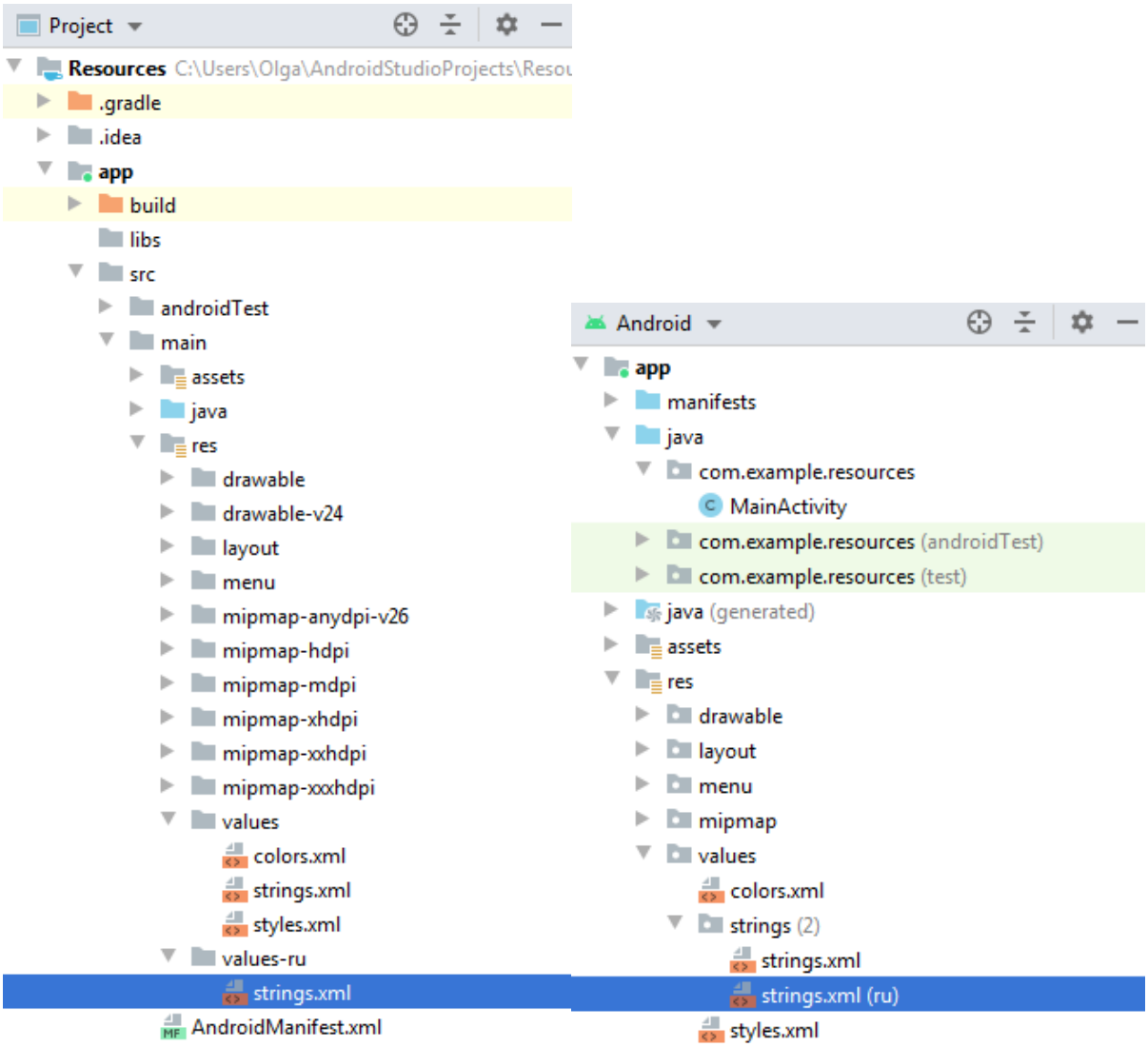

Рисунок 4. Файл strings.xml (ru)

Теперь создадим цветовые ресурсы в файле colors.xml (листинг 3), созданные по умолчанию не удаляем.

Листинг 3. Файл colors.xml

```
<resources>
...
    <color name="colorPrimary">#6200EE</color>
    <color name="colorPrimaryDark">#3700B3</color>
    <color name="colorAccent">#03DAC5</color>
    <color name="red">#CC1B30</color>
    <color name="green">#009900</color>
    <color name="blue">#0099FF</color>
</resources>
```

Создадим папке menu для ресурсов меню (рис. 5).

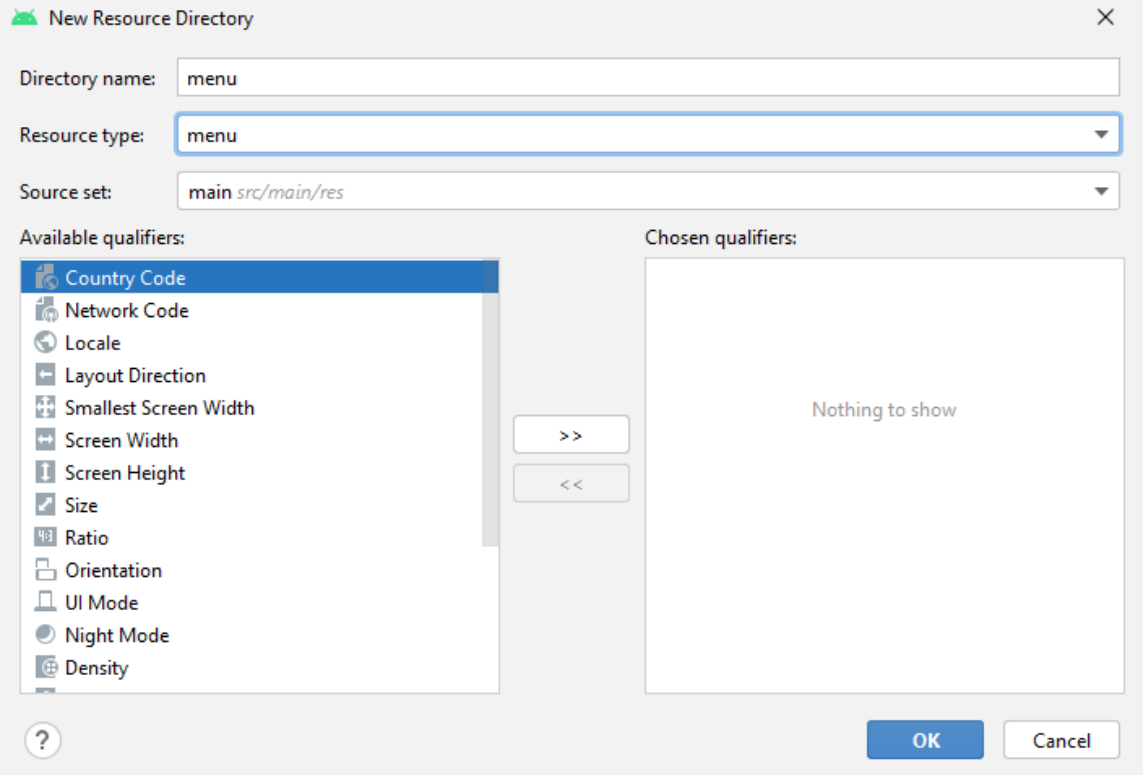

Рисунок 5. Создание папки menu

В папке menu создадим файл menu.xml и опишем пункты, которые будут в меню параметров: будет кнопка выхода из активити, а также подменю с выбором цвета фона активити (листинг 4).

Листинг 4. Файл menu.xml

```
<menu  xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:id="@+id/color"
          android:title="@string/color">
        <menu>
            <item android:id="@+id/blue"
                android:title="@string/blue"/>
            <item android:id="@+id/red"
                android:title="@string/red"/>
            <item android:id="@+id/green"
                android:title="@string/green"/>
        </menu>
    </item>
    <item android:id="@+id/exit"
          android:title="@string/exit"/>
</menu>
```

Теперь зададим стили и темы в файле styles.xml (листинг 5) или themes.xml. В файле манифеста приложения должна быть объявлена ссылка на тему для приложения (листинг 6).

Листинг 5. Файл styles.xml (themes.xml)

```
<resources>
    <style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">
        <item name="colorPrimary">@color/colorPrimary</item>
        <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
        <item name="colorAccent">@color/colorAccent</item>
        <item name="android:windowBackground">@drawable/background</item>
        <item name="android:windowNoTitle">true</item>
    </style>
    <style name="SpecialText" parent="TextAppearance.AppCompat.Title">
        <item name="android:textSize">36sp</item>
        <item name="android:textColor">#548f00</item>
        <item name="android:background">#b7e4fb</item>
    </style>
</resources>
```

Листинг 6. Элемент application из файла AndroidManifest.xml

```
<application
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="@string/app_name"
    android:roundIcon="@mipmap/ic_launcher_round"
    android:supportsRtl="true"
    android:theme="@style/AppTheme">
```

В файле компоновки определим единственный элемент TextView, и для этого элемента зададим созданный в файле styles.xml стиль, используя атрибут style (листинг 7).

Листинг 7. Файл activity_main.xml

```
<LinearLayout xmlns:android=http://schemas.android.com/apk/res/android
    android:id="@+id/root"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center">
    <TextView
        style="@style/SpecialText"
        android:id="@+id/my_text"
        android:text="@string/text"
        android:layout_height="wrap_content"
        android:layout_width="wrap_content"
        android:gravity="center"
        android:padding="5dp"/>
</LinearLayout>
```

И наконец, отредактируем файл MainActivity.java, добавим в него код загрузки меню из файла menu.xml и опишем действия при выборе пунктов меню (листинг 8).

Листинг 8. Файл MainActivity.java

```
package com.example.resources;
import androidx.appcompat.app.AppCompatActivity;
import android.graphics.Color;
import android.graphics.Typeface;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.widget.LinearLayout;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.menu, menu);
        return true;
    }
    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        final LinearLayout mylayout = (LinearLayout)findViewById(R.id.root);
        switch (item.getItemId()) {
            case R.id.red:

mylayout.setBackgroundColor(getResources().getColor(R.color.red));
                return true;
            case R.id.green:

mylayout.setBackgroundColor(getResources().getColor(R.color.green));
                return true;
            case R.id.blue:

mylayout.setBackgroundColor(getResources().getColor(R.color.blue));
                return true;
            case R.id.exit:
                finish();
                return true;
            default:
                return super.onOptionsItemSelected(item);
        }
    }
}
```

Запустите приложение на виртуальном или физическом устройстве Android. Посмотрите, как изменение языка по умолчанию (английский, русский) в настройках устройства влияет на приложение.

Задание

Самостоятельно дополните приложение, разработанное в практической части, добавьте различные фоновые картинки для горизонтальной и вертикальной ориентации экрана, добавьте новые пункты меню, создающие на компоновке дополнительные элементы (например, текстовые поля). По желанию, добавьте ресурсы шрифтов в папке res/font и используйте их в приложении.

Контрольные вопросы:

1. Какие типы ресурсов хранятся в файле arrays.xml?
2. Какие типы ресурсов хранятся в файле colors.xml?
3. Какие типы ресурсов хранятся в файле dimens.xml?
4. Какие типы ресурсов хранятся в файле drawables.xml?
5. Какие типы ресурсов хранятся в файле strings.xml?
6. Чем стили отличаются от тем?
7. В каком файле задаются стили?
8. В каком файле задаются пункты меню?
9. Как объявляется ссылка на тему в файле манифеста приложения?
10. Как возможно осуществить локализацию ресурсов в приложении?