**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**



**МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**ВЫСШАЯ ШКОЛА ПЕЧАТИ И МЕДИАИНДУСТРИИ**

***Институт Принтмедиа и информационных технологий***

***Кафедра Информатики и информационных технологий***

**направление подготовки**

**09.03.02 «Информационные системы и технологии»**

**ЛАБОРАТОРНАЯ РАБОТА № 5**

**Дисциплина: Программирование для мобильных устройств**

**Тема: «**Работа с ресурсами в Android-приложении»

**Выполнил(а): студент(ка) группы \_\_211-722\_**

Дерендяев Дмитрий Сергеевич\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Фамилия И.О.)

**Дата, подпись** 24.10.2023\_\_\_\_  ***\_\_\_\_\_\_\_\_\_\_\_***

(Дата) (Подпись)

**Проверил: \_\_*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_***

(Фамилия И.О., степень, звание) **(Оценка)**

**Дата, подпись** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  ***\_\_\_\_\_\_\_\_\_\_\_***

(Дата) (Подпись)

**Замечания: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Москва**

**2023**

# Введение

**Целью** лабораторной работы является знакомство c различными типами

ресурсов и их использованием в Android-приложениях.Разработка MainActivity.java

# Ход работы

## string.xml

Реализуем файл string.xml для отображения строковых переменных по умолчанию и string.xml(ru) для отображения при языковых настройках на русском.

Листинг 1.

1. <resources>
2. <string name="app\_name">Resources</string>
3. <string name="text">Hello world!</string>
4. <string name="red">Red</string>
5. <string name="green">Green</string>
6. <string name="blue">Blue</string>
7. <string name="color">Color</string>
8. <string name="exit">Exit</string>
9. </resources>

Листинг 2.

1. <resources>
2. <string name="app\_name">Ресурсы</string>
3. <string name="text">Привет, мир!</string>
4. <string name="red">Красный</string>
5. <string name="green">Зелёный</string>
6. <string name="blue">Синий</string>
7. <string name="color">Цвет</string>
8. <string name="exit">Выход</string>
9. </

## color.xml

Реализуем color.xml для доступа к цветам по ID

Листинг 3.

1. <?xml version="1.0" encoding="utf-8"?>  
   <resources>  
    <color name="black">#FF000000</color>  
    <color name="white">#FFFFFFFF</color>  
     
    <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>

## menu.xml

Реализуем menu.xml для отображения меня приложения.

Листинг 4.

1. <?xml version="1.0" encoding="utf-8"?>  
   <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/add\_text\_field"  
    android:title="Добавить текстовое поле" />  
    <item  
    android:id="@+id/add\_image"  
    android:title="Добавить картинку" />  
    <item  
    android:id="@+id/change\_font"  
    android:title="Изменить шрифт" />  
     
    <item android:id="@+id/exit"  
    android:title="@string/exit"/>  
   </menu>

## AndroidManifest.xml

А также настроим файл AndroidManifest.xml для того, чтобы связать основные сущности и позволить приложению работать как в вертикальном, так и в горизонтальном режимах.

Листинг 5.

1. <?xml version="1.0" encoding="utf-8"?>  
   <manifest xmlns:android="http://schemas.android.com/apk/res/android"  
    xmlns:tools="http://schemas.android.com/tools">  
     
    <application  
    android:allowBackup="true"  
    android:dataExtractionRules="@xml/data\_extraction\_rules"  
    android:fullBackupContent="@xml/backup\_rules"  
    android:icon="@mipmap/ic\_launcher"  
    android:label="@string/app\_name"  
    android:roundIcon="@mipmap/ic\_launcher\_round"  
    android:supportsRtl="true"  
    android:theme="@style/AppTheme"  
    tools:targetApi="31">  
    <activity  
    android:name=".MainActivity"  
    android:configChanges="orientation|screenSize"  
    android:exported="true">  
    <intent-filter>  
    <action android:name="android.intent.action.MAIN" />  
     
    <category android:name="android.intent.category.LAUNCHER" />  
    </intent-filter>  
    </activity>  
    </application>  
     
   </manifest>

## Activity\_main.xml

В этом приложении не будет изначально сложного activity\_main.xml, все элементы будут добавляться по кнопкам.

Листинг 6.

1. <?xml version="1.0" encoding="utf-8"?>  
   <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>

## Activity\_main.xml

Activity\_main.xml позволит добавлять элементы в компоновку приложения, менять шрифт, обновлять фон приложения и позволять из него выйти.

1. package com.example.resources;  
     
   import android.content.res.Configuration;  
   import android.graphics.Typeface;  
   import android.os.Bundle;  
   import android.view.Menu;  
   import android.view.MenuItem;  
   import android.view.View;  
   import android.view.ViewGroup;  
   import android.widget.EditText;  
   import android.widget.ImageView;  
   import android.widget.LinearLayout;  
   import android.widget.TextView;  
     
   import androidx.appcompat.app.AppCompatActivity;  
   import androidx.core.content.res.ResourcesCompat;  
     
   public class MainActivity extends AppCompatActivity {  
     
    private LinearLayout myLayout;  
     
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.*activity\_main*);  
    myLayout = findViewById(R.id.*root*);  
    }  
     
    @Override  
    public void onConfigurationChanged(Configuration newConfig) {  
    super.onConfigurationChanged(newConfig);  
     
    if (newConfig.orientation == Configuration.*ORIENTATION\_LANDSCAPE*) {  
    myLayout.setBackgroundResource(R.drawable.*back\_landscape*);  
    } else if (newConfig.orientation == Configuration.*ORIENTATION\_PORTRAIT*) {  
    myLayout.setBackgroundResource(R.drawable.*back\_portrait*);  
    }  
    }  
     
    @Override  
    public boolean onCreateOptionsMenu(Menu menu) {  
    getMenuInflater().inflate(R.menu.*menu*, menu);  
    return true;  
    }  
     
    @Override  
    public boolean onOptionsItemSelected(MenuItem item) {  
    final LinearLayout myLayout = findViewById(R.id.*root*);  
    int viewId = item.getItemId();  
     
    if (viewId == R.id.*red*) {  
    myLayout.setBackgroundColor(getResources().getColor(R.color.*red*));  
    return true;  
    } else if (viewId == R.id.*green*) {  
    myLayout.setBackgroundColor(getResources().getColor(R.color.*green*));  
    return true;  
    } else if (viewId == R.id.*blue*) {  
    myLayout.setBackgroundColor(getResources().getColor(R.color.*blue*));  
    return true;  
    } else if (viewId == R.id.*exit*) {  
    finish();  
    return true;  
    } else if (viewId == R.id.*add\_text\_field*) {  
    EditText editText = new EditText(this);  
    myLayout.addView(editText, new ViewGroup.LayoutParams(  
    ViewGroup.LayoutParams.*MATCH\_PARENT*,  
    ViewGroup.LayoutParams.*WRAP\_CONTENT* ));  
    return true;  
    } else if (viewId == R.id.*add\_image*) {  
    ImageView imageView = new ImageView(this);  
    imageView.setImageResource(R.drawable.*cat*);  
    myLayout.addView(imageView, new ViewGroup.LayoutParams(  
    ViewGroup.LayoutParams.*WRAP\_CONTENT*,  
    ViewGroup.LayoutParams.*WRAP\_CONTENT* ));  
    return true;  
    } else if (viewId == R.id.*change\_font*) {  
    Typeface newTypeface = ResourcesCompat.*getFont*(this, R.font.*fonts*);  
    setFontForViewGroup(myLayout, newTypeface);  
    return true;  
    }  
    else {  
    return super.onOptionsItemSelected(item);  
    }  
    }  
     
    private void setFontForViewGroup(ViewGroup viewGroup, Typeface typeface) {  
    for (int i = 0; i < viewGroup.getChildCount(); i++) {  
    View child = viewGroup.getChildAt(i);  
    if (child instanceof TextView) {  
    ((TextView) child).setTypeface(typeface);  
    } else if (child instanceof ViewGroup) {  
    setFontForViewGroup((ViewGroup) child, typeface);  
    }  
    }  
    }  
   }





