



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«МИРЭА – Российский технологический университет»
РТУ МИРЭА

Институт информационных технологий (ИИТ)
Кафедра Математического обеспечения и стандартизации информационных технологий

Отчет по выполнению практического задания №4

Тема:

Поддержка различных устройств

Дисциплина:

Разработка мобильных приложений

Выполнил:

Студент группы ИКБО-03-20

Цемкало А.Р.

Принял:

Доцент кафедры МОСИТ ИИТ

Чернов Е. А.

Москва 2022 г.

СОДЕРЖАНИЕ

Ход работы.....	3
Вывод.....	13

Ход работы

1. Поддержка различных языков. Создать директории для поддержки различных языков.

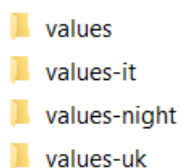


Рисунок 1 – Директории для файлов строковых ресурсов

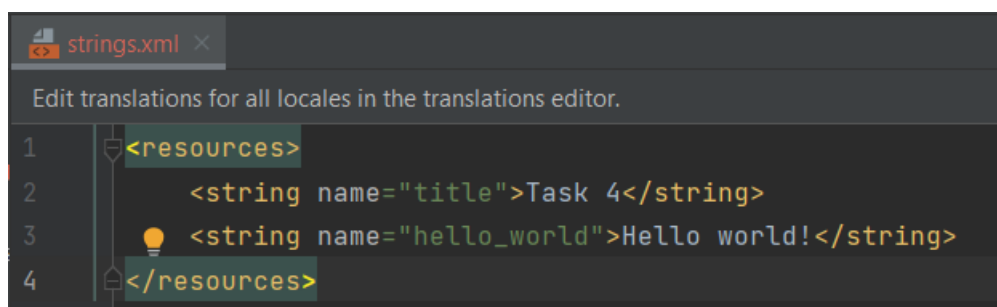


Рисунок 2 – Строковые константы для английского языка

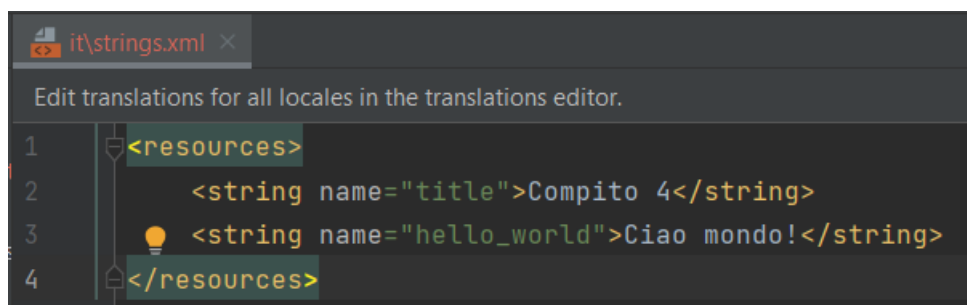


Рисунок 3 – Строковые константы для итальянского языка

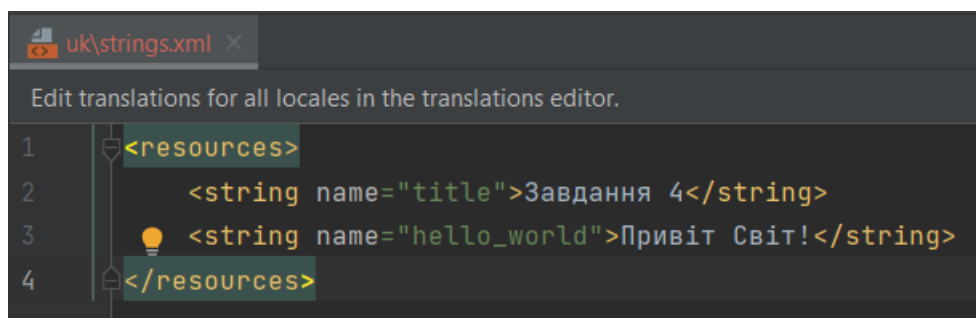


Рисунок 4 – Строковые константы для украинского языка

Обеспечить использование строковых ресурсов.

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/hello_world"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

Рисунок 5 – Использование строковых ресурсов в файле activity_main.xml

```
<application
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="@string/title"
    android:roundIcon="@mipmap/ic_launcher_round"
    android:supportsRtl="true"
```

Рисунок 6 – Использование строковых ресурсов в файле AndroidManifest.xml

2. Поддержка устройств с различными экранами. Обеспечить создание различной разметки. Обеспечить использование различных изображений.

- layout
- layout-land
- layout-large
- layout-large-land

Рисунок 7 – Поддержка разных размеров и ориентации экрана

- drawable
- drawable-hdpi
- drawable-v24
- drawable-xhdpi

Рисунок 8 – Поддержка разных разрешений экрана

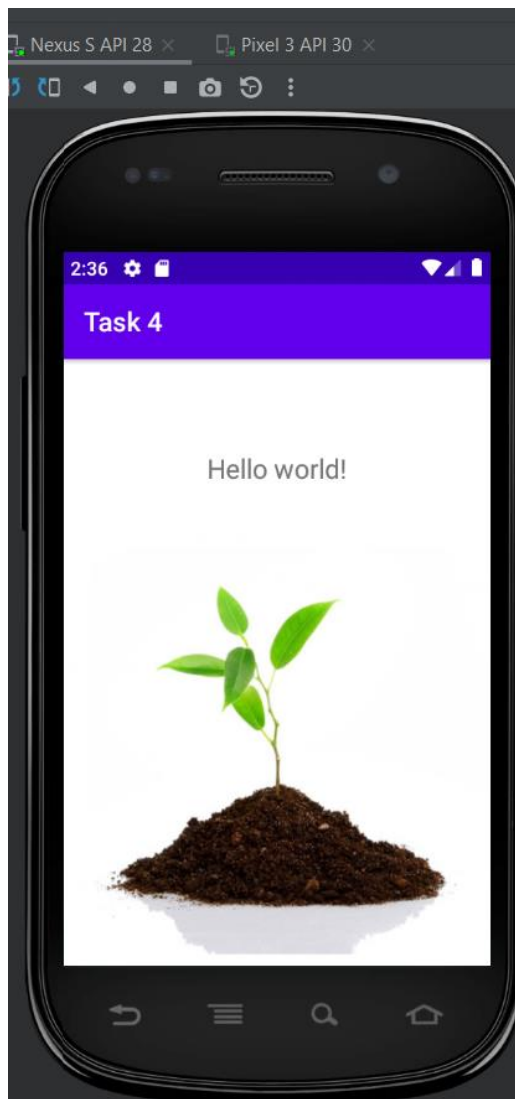


Рисунок 9 – Вид открытого приложения в виртуальном девайсе с разрешением экрана hdpi

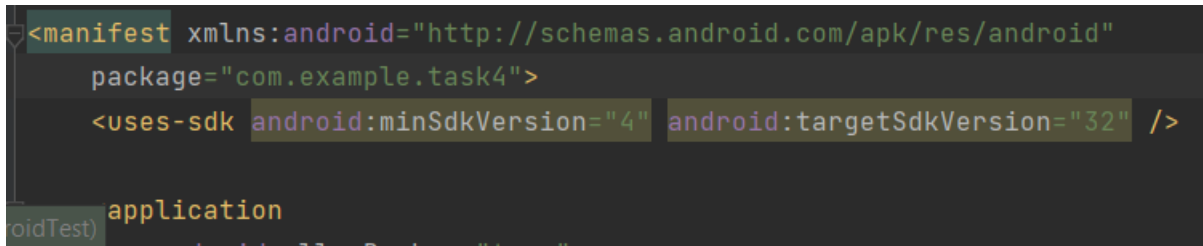


Рисунок 10 – Вид открытого приложения в виртуальном девайсе с разрешением экрана xhdpі в вертикальном положении



Рисунок 10 – Вид открытого приложения в виртуальном девайсе с разрешением экрана xhdpі в горизонтальном положении

3. Поддержка различных версий Android. Указать минимальную и целевую версии API. Обеспечить получение версии Android во время выполнения приложения.



```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.task4">
    <uses-sdk android:minSdkVersion="4" android:targetSdkVersion="32" />
    <application
        android:label="@string/app_name"
        android:icon="@mipmap/ic_launcher"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:theme="@style/Theme.AppCompat.NoActionBar">
```

Рисунок 11 – Указание минимальной и целевой версий API

Код файла MainActivity.java

```
package com.example.task4;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Build;
import android.os.Bundle;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {
    private TextView textView;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        textView = (TextView) findViewById(R.id.textView2);
        textView.setText(textView.getText() + " " + Build.VERSION.SDK_INT);
    }
}
```



Рисунок 12 – Вид открытого приложения в виртуальном девайсе с отображением версии API

4. Обеспечить использование встроенных тем и стилей.

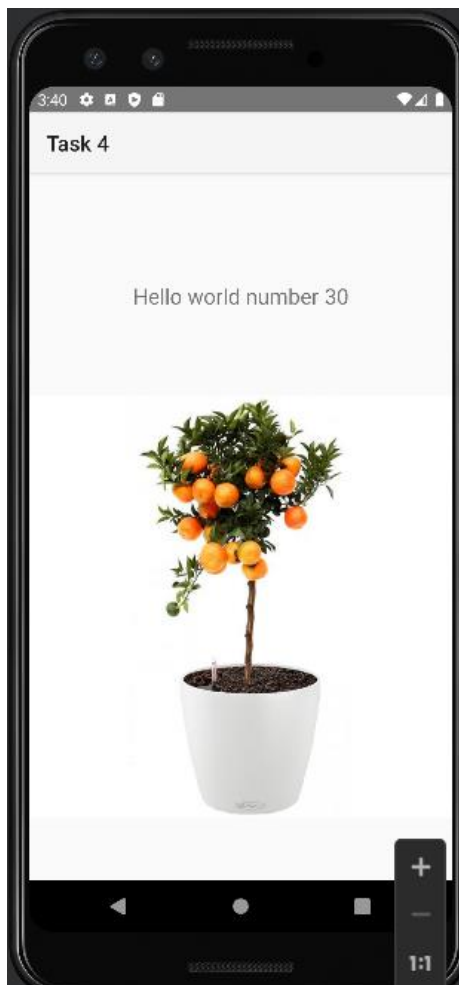


Рисунок 13 – Вид открытого приложения в виртуальном девайсе с использованием встроенной темы

Часть кода файла AndroidManifest.xml

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.task4">
    <uses-sdk android:minSdkVersion="4" android:targetSdkVersion="32" />
    <application
        ...
        android:theme="@style/Theme.AppCompat.DayNight">
        ...
    </application>
</manifest>
```

5. Жизненный цикл явлений. Обеспечить запуск явлений. Обеспечить функции обратного вызова жизненного цикла. Обеспечить объявление главного явления. Обеспечить создание экземпляра явления. Обеспечить приостановку и

возобновление явлений. Обеспечить остановку, запуск и перезапуск явлений. Обеспечить пересоздание явлений. Сохранить состояния экземпляра явлений. Восстановить состояния экземпляра явлений.

Часть кода файла MainActivity.java, обеспечивающего приостановку и возобновление явления

```
@Override
protected void onPause() {
    super.onPause();
    String uri = "@drawable/dry_tangerine";
    int    imageResource    =    getResources().getIdentifier(uri,    null,
getPackageName());
    Drawable src = getResources().getDrawable(imageResource);
    imageView.setImageDrawable(src);
}

@Override
protected void onResume() {
    super.onResume();
    String uri = "@drawable/tangerine";
    int    imageResource    =    getResources().getIdentifier(uri,    null,
getPackageName());
    Drawable src = getResources().getDrawable(imageResource);
    imageView.setImageDrawable(src);
}
}
```

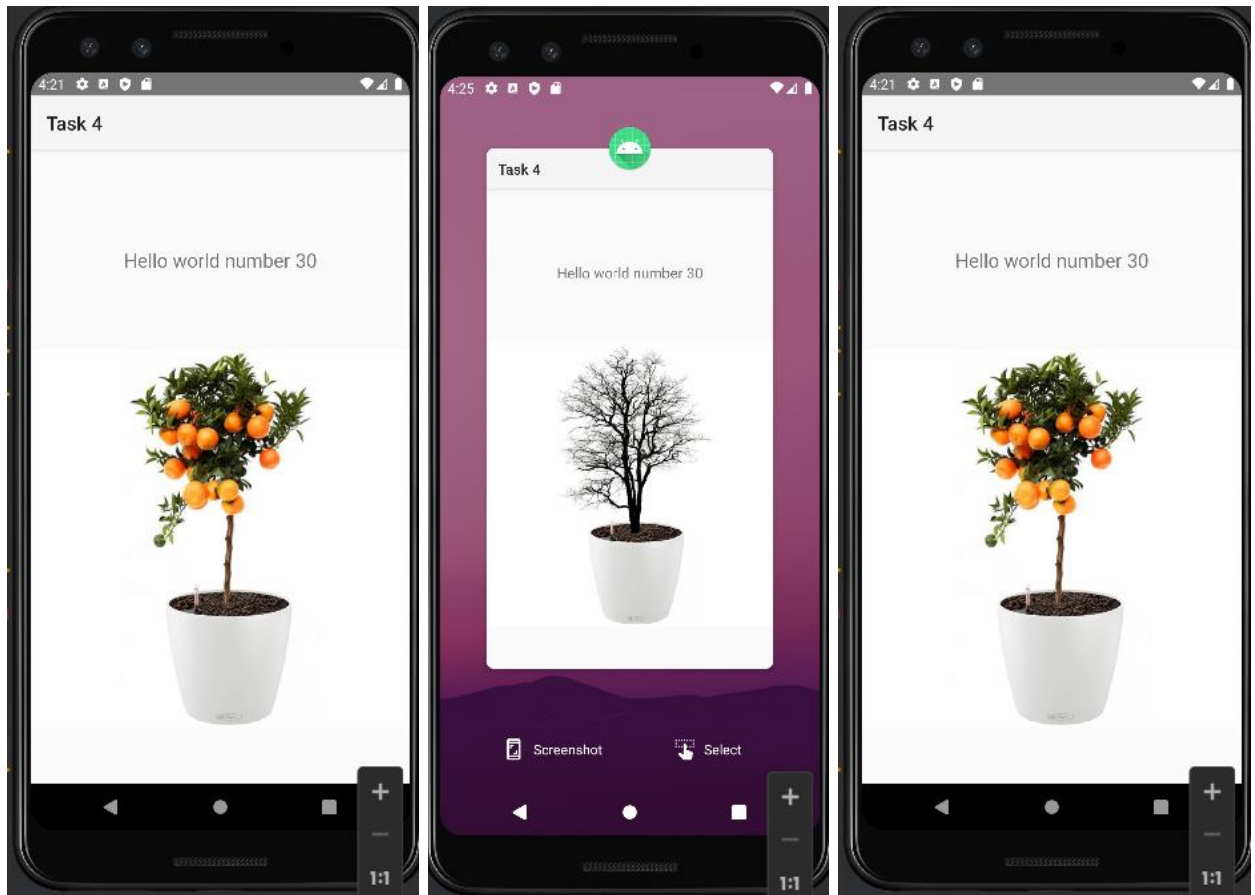


Рисунок 14 – Вид приложения после открытия, приостановки, возобновления явления (слева направо)

Часть кода файла MainActivity.java, обеспечивающего остановку и перезапуск явления.

```
@Override
protected void onStop() {
    super.onStop();
    killingTangerineNumber++;
}

@Override
protected void onRestart() {
    super.onRestart();
    raisingTangerineNumber++;
}
```



Рисунок 14 – Вид приложения с подсчётом количества остановок и перезапусков явлений

Часть файла ActivityMain.java с кодом, восстанавливающим состояние экземпляра явлений

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    if (savedInstanceState != null) {
        killingTangerineNumber = savedInstanceState.getInt(KILLING_NUMBER);
        raisingTangerineNumber = savedInstanceState.getInt(RAISING_NUMBER);
    }
    else {
```

```

        killingTangerineNumber = 0;
        raisingTangerineNumber = 0;
    }
    textView = (TextView) findViewById(R.id.textView2);
    greeting = textView.getText() + " " + Build.VERSION.SDK_INT;
    textView.setText(greeting);

    imageView = (ImageView) findViewById(R.id.imageView);
}

```

Часть файла ActivityMain.java с кодом, сохраняющим состояние экземпляра явлений

```

@Override
protected void onSaveInstanceState(Bundle savedInstanceState) {
    savedInstanceState.putInt(KILLING_NUMBER, killingTangerineNumber);
    savedInstanceState.putInt(RAISING_NUMBER, raisingTangerineNumber);
    super.onSaveInstanceState(savedInstanceState);
}

```

Вывод

Создано приложение, поддерживающее различные языки, различные экраны, версии Android, использующее встроенные темы с обеспечением жизненного цикла явлений.