|  |
| --- |
| 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](#_Toc99395910)

[Вывод 13](#_Toc99395911)

Ход работы

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

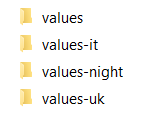


Рисунок 1 – Директории для файлов строковых ресурсов

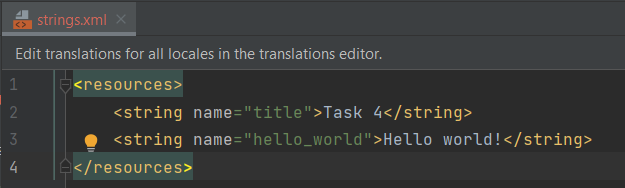


Рисунок 2 – Строковые константы для английского языка

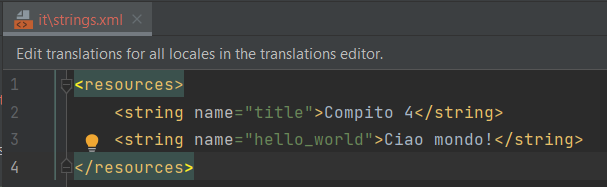


Рисунок 3 – Строковые константы для итальянского языка

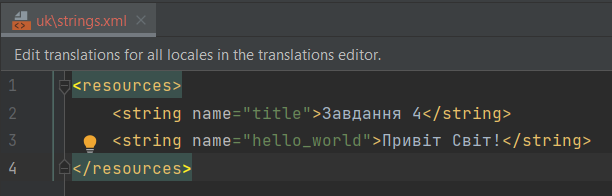


Рисунок 4 – Строковые константы для украинского языка

Обеспечить использование строковых ресурсов.

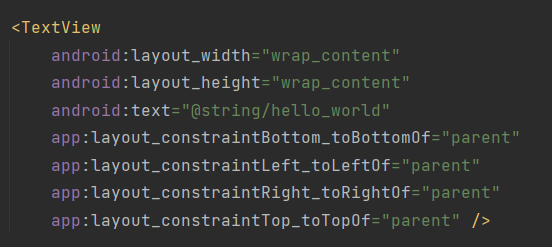


Рисунок 5 – Использование строковых ресурсов в файле activity\_main.xml

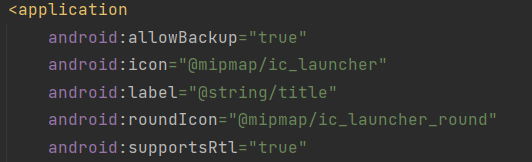


Рисунок 6 – Использование строковых ресурсов в файле AndroidManifest.xml

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

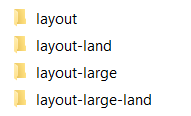


Рисунок 7 – Поддержка разных размеров и ориентации экрана

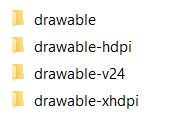


Рисунок 8 – Поддержка разных разрешений экрана

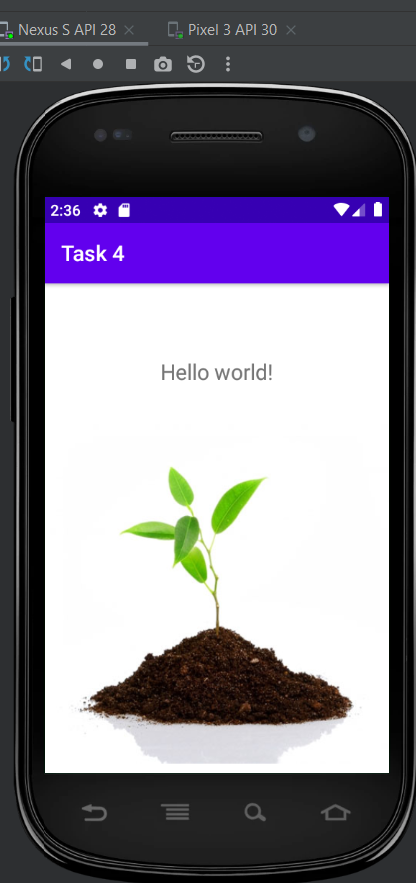


Рисунок 9 – Вид открытого приложения в виртуальном девайсе с разрешением экрана hdpi

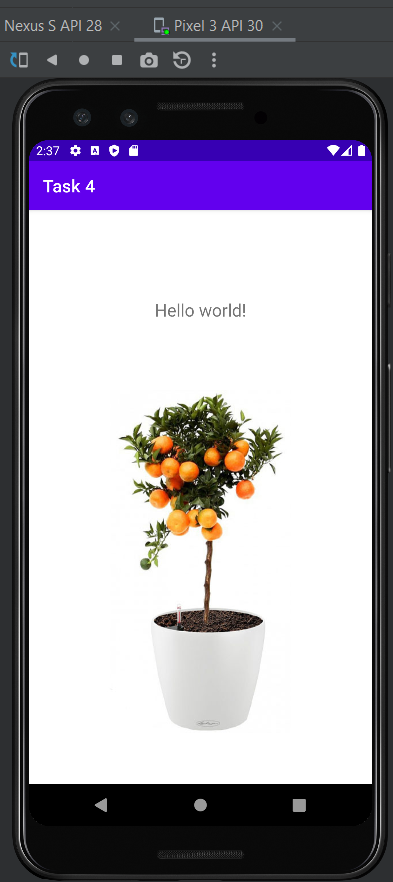


Рисунок 10 – Вид открытого приложения в виртуальном девайсе с разрешением экрана xhdpi в вертикальном положении

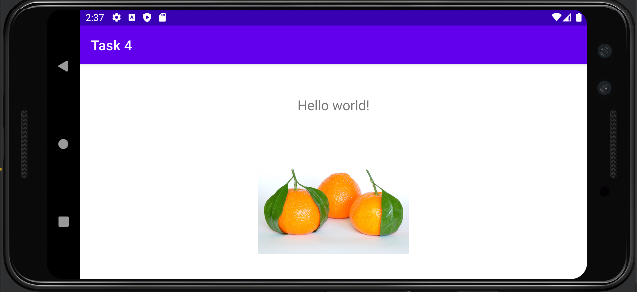


Рисунок 10 – Вид открытого приложения в виртуальном девайсе с разрешением экрана xhdpi в горизонтальном положении

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

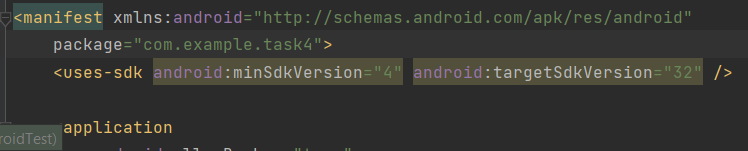


Рисунок 11 – Указание минимальной и целевой версий API

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);

}

}

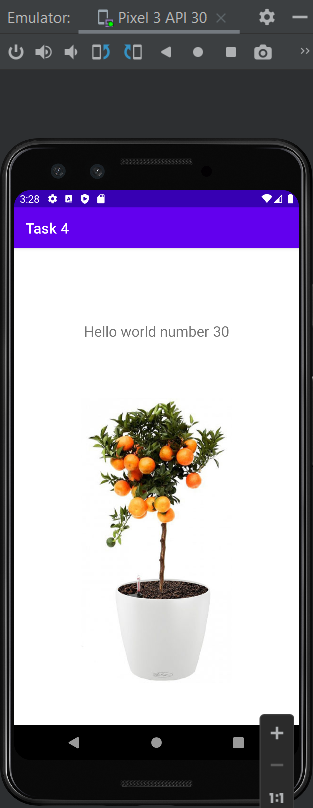


Рисунок 12 – Вид открытого приложения в виртуальном девайсе с отображением версии API

1. Обеспечить использование встроенных тем и стилей.

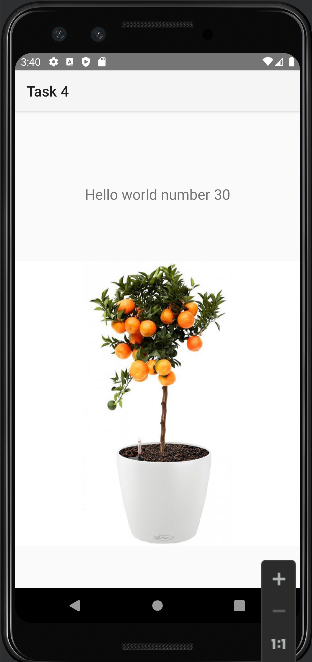


Рисунок 13 – Вид открытого приложения в виртуальном девайсе с использованием встроенной темы

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>

1. 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();

imageView.setImageResource(R.drawable.dry\_tangerine);

}

@Override

protected void onResume() {

super.onResume();

textView.setText(greeting + "!\nYou killed this tangerine for " + killingTangerineNumber + " times.\n" +

"Then you raised the plant for " + raisingTangerineNumber + " times.");

}

}

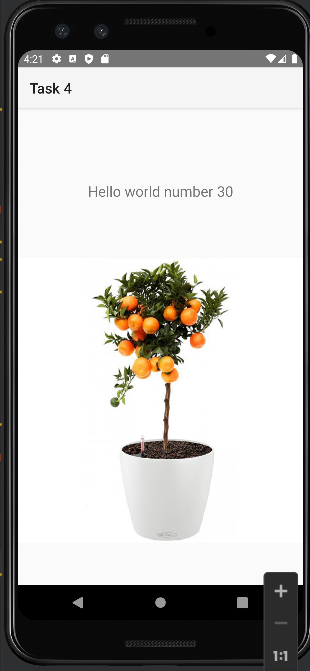
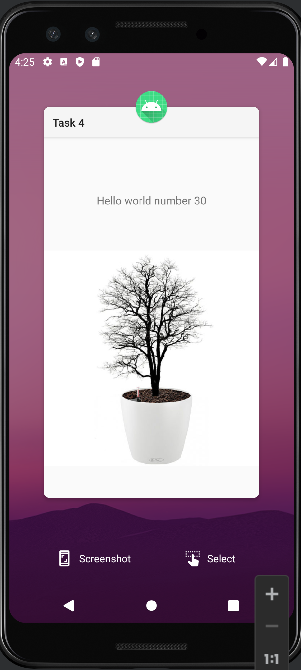
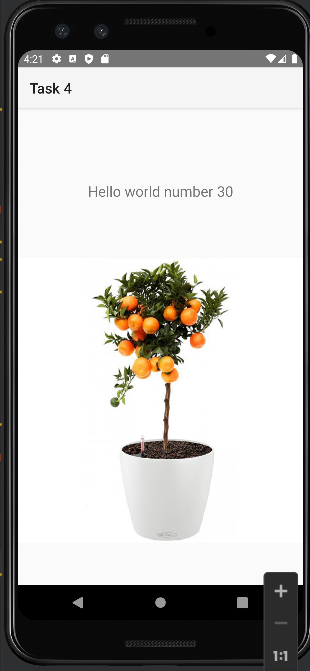
  

Рисунок 14 – Вид приложения после открытия, приостановки, возобновления явления (слева направо)

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

@Override

protected void onStop() {

super.onStop();

killingTangerineNumber++;

}

@Override

protected void onRestart() {

super.onRestart();

raisingTangerineNumber++;

imageView.setImageResource(R.drawable.tangerine);

}

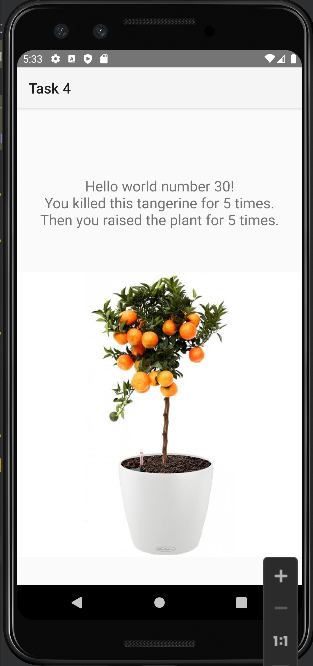


Рисунок 14 – Вид приложения с подсчётом количества остановок и перезапусков явлений

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.