**Министерство науки и высшего образования Российской Федерации**

Федеральное государственное автономное образовательное учреждение

высшего образования «Уральский федеральный университет

имени первого Президента России Б.Н. Ельцина»

Институт радиоэлектроники и информационных технологий - РТФ

Центр ускоренного обучения

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ 19**

по дисциплине «Разработка мобильных приложений»

**Тема: Unit 2 Пользовательский интерфейс “Ассинхроность”**

Студент группы РИЗ-300028у: А.С. Севостьянов

Преподаватель: С.И.Тимошенко,

доц., к.т.н.

**Екатеринбург 2022**

содержание

[1 Постановка задачи 3](#_Toc118041527)

[2 Результаты проделанной работы 4](#_Toc118041528)

[3 Анализ полученных результатов 5](#_Toc118041529)

## 1 Постановка задачи

Создайте простое приложение, которое выполняет фоновую задачу с помощью AsyncTask. Запустите приложение и посмотрите, что происходит при повороте устройства. Реализуйте состояние экземпляра действия, чтобы сохранить состояние TextView сообщения.

.

## Результаты проделанной работы

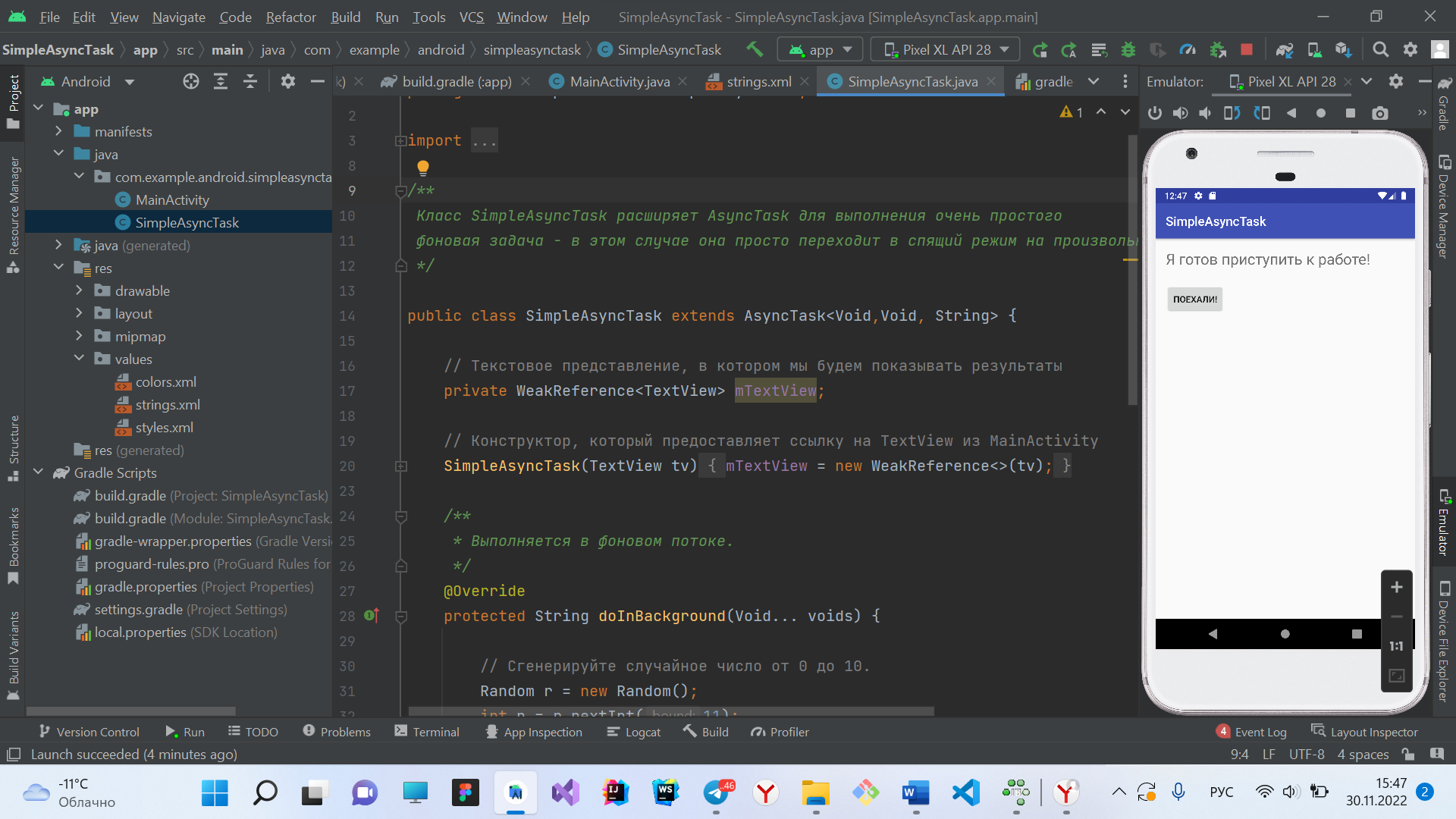
Программный код проекта, MainActivity

package com.example.android.simpleasynctask;  
  
import android.support.v7.app.AppCompatActivity;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.TextView;  
  
*/\*\*  
 Приложение SimpleAsyncTask содержит кнопку, которая запускает AsyncTask  
 \*/*public class MainActivity extends AppCompatActivity {  
  
 // Клавиша для сохранения состояния TextView  
 private static final String *TEXT\_STATE* = "currentText";  
  
 // Текстовое представление, в котором мы будем показывать результаты  
 private TextView mTextView;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 mTextView = findViewById(R.id.*textView1*);  
  
 // Восстановите TextView, если есть пакет savedInstanceState.  
 if (savedInstanceState != null) {  
 mTextView.setText(savedInstanceState.getString(*TEXT\_STATE*));  
 }  
 }  
  
 */\*\*  
 \* Обрабатывает нажатие кнопки "Запустить задачу". Запускает AsyncTask  
 \*/* public void startTask(View view) {  
 // Поместите сообщение в текстовое представление.  
 mTextView.setText(R.string.*napping*);  
 //AsyncTask есть обратный вызов, который обновит текстовое представление.  
 new SimpleAsyncTask(mTextView).execute();  
 }  
  
 */\*\*  
 \* Сохраняет содержимое TextView для восстановления при изменении конфигурации.  
 \*/* @Override  
 protected void onSaveInstanceState(Bundle outState) {  
 super.onSaveInstanceState(outState);  
 // Сохранить состояние текстового представления  
 outState.putString(*TEXT\_STATE*, mTextView.getText().toString());  
 }  
}

Программный код, файл SimpleAsyncTask.

package com.example.android.simpleasynctask;  
  
import android.os.AsyncTask;  
import android.widget.TextView;  
  
import java.lang.ref.WeakReference;  
import java.util.Random;  
  
*/\*\*  
 Класс SimpleAsyncTask расширяет AsyncTask для выполнения очень простого  
 фоновая задача - в этом случае она просто переходит в спящий режим на произвольное количество времени.  
 \*/*public class SimpleAsyncTask extends AsyncTask<Void,Void, String> {  
  
 // Текстовое представление, в котором мы будем показывать результаты  
 private WeakReference<TextView> mTextView;  
  
 // Конструктор, который предоставляет ссылку на TextView из MainActivity  
 SimpleAsyncTask(TextView tv) {  
 mTextView = new WeakReference<>(tv);  
 }  
  
 */\*\*  
 \* Выполняется в фоновом потоке.  
 \*/* @Override  
 protected String doInBackground(Void... voids) {  
  
 // Сгенерируйте случайное число от 0 до 10.  
 Random r = new Random();  
 int n = r.nextInt(11);  
  
 // Сделайте так, чтобы задача занимала достаточно много времени, чтобы у нас было  
 // время поворота телефона во время его работы.  
 int s = n \* 200;  
  
 // Спящий режим на произвольное количество времени.  
 try {  
 Thread.*sleep*(s);  
 } catch (InterruptedException e) {  
 e.printStackTrace();  
 }  
  
 // Возвращает строковый результат.  
 return "Проснувшись наконец после долгого сна" + s + "миллисекунд!";  
 }  
  
 */\*\*  
 \* Делает что-то с результатом в потоке пользовательского интерфейса;  
 \*/* protected void onPostExecute(String result) {  
 mTextView.get().setText(result);  
 }  
}

На рисунке 1 показано начальное состояние программы.



На рисунке 1 – Начальное состояние программы

На рисунке 2 показано результат работы программы после того, как мы использовали кнопку поехали, которая вернула нас после сна.

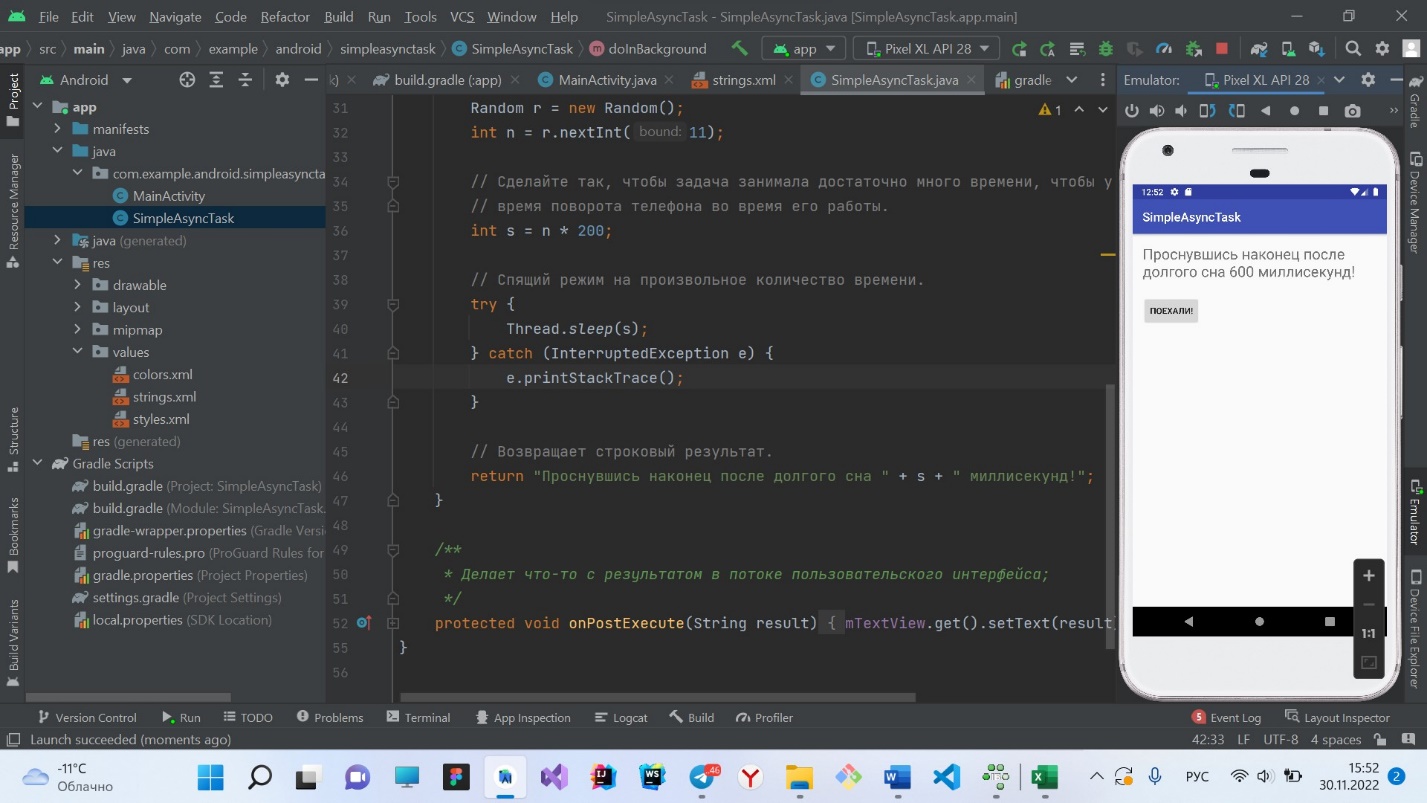


Рисунок 2 – Результат работы программы

## **Анализ полученных результатов**

В результате выполнения данной лабораторной работы мной был разработан проект, в котором я научился, как добавить AsyncTask в свое приложение, чтобы запустить задачу в фоновом режиме приложения.