МИНИCTEPCTBO НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

ИНСТИТУТ ПЕРСПЕКТИВНОЙ ИНЖЕНЕРИИ

**Кафедра прикладной информатики**

Лабораторная работа №11

ПО ДИСЦИПЛИНЕ «Программирование мобильных устройств»

НА ТЕМУ:

«Работа с потоками»

**Выполнил:**

студент группы ПИН-Б-З-22-1

Гадиян Сергей Гариевич

Ставрополь, 2025 г.

Цель работы: научиться распределять задачи между потоками.

Формируемые компетенции: ПК-7

Ход работы.

Задания: для выполнения лабораторной работы необходимо выполнить следующее:

1. Рассчитать результат и отобразить его в интерфейсе.

2. Загрузить несколько любых изображений из интернета с отображением строки состояния (ProgressBar).

1. Рассчитать результат и отобразить его в интерфейсе.

В одномерном массиве, состоящем из n вещественных элементов вычислить:

a. Сумму элементов с нечетными индексами.

b. Сумму элементов массива, расположенных между первым и

последним отрицательными элементами.

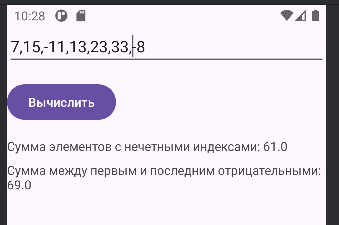


Рисунок 1 – Результат выполнения

package com.example.lab11\_new;  
  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import androidx.activity.EdgeToEdge;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.core.graphics.Insets;  
import androidx.core.view.ViewCompat;  
import androidx.core.view.WindowInsetsCompat;  
  
public class MainActivity extends AppCompatActivity {  
  
 private EditText editTextArray;  
 private Button calculateButton;  
 private TextView textViewResultA;  
 private TextView textViewResultB;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 EdgeToEdge.*enable*(this);  
 setContentView(R.layout.*activity\_main*);  
 ViewCompat.*setOnApplyWindowInsetsListener*(findViewById(R.id.*main*), (v, insets) -> {  
 Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.*systemBars*());  
 v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);  
 return insets;  
 });  
  
 editTextArray = findViewById(R.id.*editTextArray*);  
 calculateButton = findViewById(R.id.*calculateButton*);  
 textViewResultA = findViewById(R.id.*textViewResultA*);  
 textViewResultB = findViewById(R.id.*textViewResultB*);  
  
 calculateButton.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 calculateResults();  
 }  
 });  
 }  
  
 private void calculateResults() {  
 String arrayInput = editTextArray.getText().toString();  
 if (arrayInput.isEmpty()) {  
 Toast.*makeText*(this, "Введите элементы массива", Toast.*LENGTH\_SHORT*).show();  
 return;  
 }  
  
 String[] stringElements = arrayInput.split(",");  
 double[] array;  
 try {  
 array = new double[stringElements.length];  
 for (int i = 0; i < stringElements.length; i++) {  
 array[i] = Double.*parseDouble*(stringElements[i].trim());  
 }  
 } catch (NumberFormatException e) {  
 Toast.*makeText*(this, "Некорректный ввод. Введите числа через запятую.", Toast.*LENGTH\_SHORT*).show();  
 return;  
 }  
  
 // a. сумма элементов с нечетными индексами  
 double sumOddIndex = 0;  
 for (int i = 1; i < array.length; i += 2) {  
 sumOddIndex += array[i];  
 }  
 textViewResultA.setText("Сумма элементов с нечетными индексами: " + sumOddIndex);  
  
 // b. сумма элементов массива, расположенных между первым и последним отрицательными элементами  
 double sumBetweenNegative = 0;  
 int firstNegativeIndex = -1;  
 int lastNegativeIndex = -1;  
  
 for (int i = 0; i < array.length; i++) {  
 if (array[i] < 0) {  
 if (firstNegativeIndex == -1) {  
 firstNegativeIndex = i;  
 }  
 lastNegativeIndex = i;  
 }  
 }  
  
 if (firstNegativeIndex != -1 && lastNegativeIndex != -1 && firstNegativeIndex < lastNegativeIndex) {  
 for (int i = firstNegativeIndex + 1; i < lastNegativeIndex; i++) {  
 sumBetweenNegative += array[i];  
 }  
 textViewResultB.setText("Сумма между первым и последним отрицательными: " + sumBetweenNegative);  
 } else {  
 textViewResultB.setText("Нет двух отрицательных элементов для расчета суммы между ними.");  
 }  
 }  
}

Листинг 1 – Код MainActivity.java для задания 1

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

<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:id="@+id/main"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:padding="16dp"

tools:context=".MainActivity">

<EditText

android:id="@+id/editTextArray"

android:layout\_width="0dp"

android:layout\_height="wrap\_content"

android:hint="Введите элементы массива через запятую (например, 1.0, 2.5, -3.0)"

android:inputType="text"

app:layout\_constraintTop\_toTopOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintEnd\_toEndOf="parent" />

<Button

android:id="@+id/calculateButton"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Вычислить"

app:layout\_constraintTop\_toBottomOf="@+id/editTextArray"

app:layout\_constraintStart\_toStartOf="parent"

android:layout\_marginTop="16dp" />

<TextView

android:id="@+id/textViewResultA"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginTop="16dp"

android:text="Сумма элементов с нечетными индексами: "

app:layout\_constraintTop\_toBottomOf="@+id/calculateButton"

app:layout\_constraintStart\_toStartOf="parent" />

<TextView

android:id="@+id/textViewResultB"

android:layout\_width="0dp"

android:layout\_height="wrap\_content"

android:layout\_marginTop="8dp"

android:text="Сумма между первым и последним отрицательными: "

app:layout\_constraintTop\_toBottomOf="@+id/textViewResultA"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintEnd\_toEndOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>

Листинг 2 – activity\_main.xml для задания 1

Задание 2.

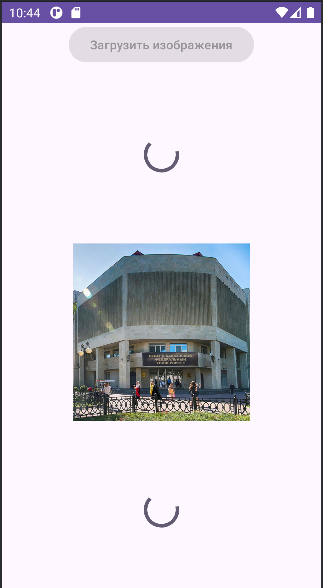


Рисунок 2 – Загрузка изображений

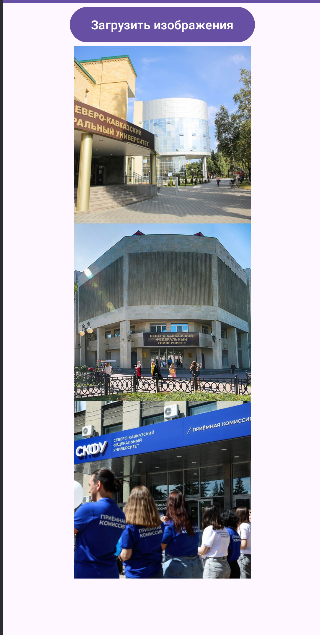


Рисунок 3 – Изображение загружены

package com.example.lab11\_new;  
  
import android.graphics.Bitmap;  
import android.graphics.BitmapFactory;  
import android.os.Bundle;  
import android.os.Handler;  
import android.os.Looper;  
import android.view.View;  
import android.widget.Button;  
import android.widget.ImageView;  
import android.widget.ProgressBar;  
import android.widget.Toast;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import java.io.IOException;  
import java.io.InputStream;  
import java.net.URL;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.concurrent.ExecutorService;  
import java.util.concurrent.Executors;  
  
public class MainActivity extends AppCompatActivity {  
  
 private ImageView imageView1, imageView2, imageView3;  
 private ProgressBar progressBar1, progressBar2, progressBar3;  
 private Button buttonLoad;  
 private final List<String> imageUrls = new ArrayList<>();  
 private ExecutorService executorService;  
 private final Handler handler = new Handler(Looper.*getMainLooper*());  
  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 imageView1 = findViewById(R.id.*imageView1*);  
 imageView2 = findViewById(R.id.*imageView2*);  
 imageView3 = findViewById(R.id.*imageView3*);  
 progressBar1 = findViewById(R.id.*progressBar1*);  
 progressBar2 = findViewById(R.id.*progressBar2*);  
 progressBar3 = findViewById(R.id.*progressBar3*);  
 buttonLoad = findViewById(R.id.*buttonLoad*);  
  
 imageUrls.add("https://egm.fondsmena.ru/media/projects/photo\_location/history\_14.png");  
 imageUrls.add("https://static.tildacdn.com/tild3838-6235-4131-b837-323961623337/image.png");  
 imageUrls.add("https://www.nevworker.ru/upload/iblock/6a5/6a54d8c39b482a9efbf1384c580dd710.png");  
  
 executorService = Executors.*newFixedThreadPool*(3);  
  
 buttonLoad.setOnClickListener(v -> loadImages());  
 }  
  
 private void loadImages() {  
 buttonLoad.setEnabled(false);  
  
 loadImage(imageUrls.get(0), imageView1, progressBar1);  
 loadImage(imageUrls.get(1), imageView2, progressBar2);  
 loadImage(imageUrls.get(2), imageView3, progressBar3);  
 }  
  
  
 private void loadImage(final String url, final ImageView imageView, final ProgressBar progressBar) {  
 progressBar.setVisibility(View.*VISIBLE*);  
 imageView.setImageBitmap(null);  
  
 executorService.execute(() -> {  
 try {  
 InputStream inputStream = (InputStream) new URL(url).getContent();  
 final Bitmap bitmap = BitmapFactory.*decodeStream*(inputStream);  
 inputStream.close();  
  
  
 handler.post(() -> {  
 imageView.setImageBitmap(bitmap);  
 progressBar.setVisibility(View.*GONE*);  
  
  
 if (areAllImagesLoaded()) {  
 buttonLoad.setEnabled(true);  
 }  
 });  
  
  
 } catch (IOException e) {  
 handler.post(()-> {  
 Toast.*makeText*(MainActivity.this, "Ошибка загрузки: " + e.getMessage(), Toast.*LENGTH\_LONG*).show();  
 progressBar.setVisibility(View.*GONE*);  
 if (areAllImagesLoaded()) {  
 buttonLoad.setEnabled(true);  
 }  
 });  
  
 }  
 });  
 }  
 private boolean areAllImagesLoaded() {  
 return progressBar1.getVisibility() == View.*GONE* &&  
 progressBar2.getVisibility() == View.*GONE* &&  
 progressBar3.getVisibility() == View.*GONE*;  
 }  
  
  
 @Override  
 protected void onDestroy() {  
 super.onDestroy();  
 executorService.shutdown();  
 }  
}

Листинг 3 – Код MainActivity.java для Задания 2

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

<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".MainActivity">

<ScrollView

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="vertical">

<Button

android:id="@+id/buttonLoad"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Загрузить изображения"

android:layout\_gravity="center\_horizontal"/>

<FrameLayout

android:layout\_width="200dp"

android:layout\_height="200dp"

android:layout\_gravity="center\_horizontal">

<ImageView

android:id="@+id/imageView1"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:scaleType="centerCrop" />

<ProgressBar

android:id="@+id/progressBar1"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center"

android:visibility="gone" />

</FrameLayout>

<FrameLayout

android:layout\_width="200dp"

android:layout\_height="200dp"

android:layout\_gravity="center\_horizontal">

<ImageView

android:id="@+id/imageView2"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:scaleType="centerCrop" />

<ProgressBar

android:id="@+id/progressBar2"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center"

android:visibility="gone" />

</FrameLayout>

<FrameLayout

android:layout\_width="200dp"

android:layout\_height="200dp"

android:layout\_gravity="center\_horizontal">

<ImageView

android:id="@+id/imageView3"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:scaleType="centerCrop"/>

<ProgressBar

android:id="@+id/progressBar3"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center"

android:visibility="gone"/>

</FrameLayout>

</LinearLayout>

</ScrollView>

</androidx.constraintlayout.widget.ConstraintLayout>

Листинг 4 – Код activity\_main.xml для задания 2

Вывод: В ходе выполнения лабораторной работы мы изучили работу потоков.