**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 8**

по дисциплине «Разработка мобильных приложений»

**Тема: Unit 1 Начало работы “Отладчик”**

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

Создать приложение SimpleCalc. Установить и просмотреть точки останова в коде для SimpleCalc. Просматривать свой код по мере его выполнения. Изучить переменные и оценивать выражения. Определить и устранить проблемы в образце приложения.

.

## Результаты проделанной работы

Программный код, класс MainActivity.

package com.example.android.SimpleCalcTest;  
  
import android.app.Activity;  
import android.os.Bundle;  
import android.text.TextUtils;  
import android.util.Log;  
import android.view.View;  
import android.widget.EditText;  
import android.widget.TextView;  
  
  
public class MainActivity extends Activity {  
  
 private static final String TAG = "CalculatorActivity";  
 private Calculator mCalculator;  
 private EditText mOperandOneEditText;  
 private EditText mOperandTwoEditText;  
 private TextView mResultTextView;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_main);  
  
 // Инициализируйте класс calculator и все представления.  
 mCalculator = new Calculator();  
 mResultTextView = findViewById(R.id.operation\_result\_text\_view);  
 mOperandOneEditText = findViewById(R.id.operand\_one\_edit\_text);  
 mOperandTwoEditText = findViewById(R.id.operand\_two\_edit\_text);  
 }  
  
 */\*\*  
 \* Метод OnClick вызывается при нажатии кнопки добавить.  
 \*/* public void onAdd(View view) {  
 compute(Calculator.Operator.ADD);  
 }  
  
 */\*\*  
 \* Метод OnClick вызывается при нажатии кнопки вычитания.  
 \*/* public void onSub(View view) {  
 compute(Calculator.Operator.SUB);  
 }  
  
 */\*\*  
 \* Метод OnClick вызывается при нажатии кнопки разделения.  
 \*/* public void onDiv(View view) {  
 try {  
 compute(Calculator.Operator.DIV);  
 } catch (IllegalArgumentException iae) {  
 Log.e(TAG, "IllegalArgumentException", iae);  
 mResultTextView.setText(getString(R.string.computationError));  
 }  
 }  
  
 */\*\*  
 \* Метод OnClick вызывается при нажатии кнопки умножения.  
 \*/* public void onMul(View view) {  
 compute(Calculator.Operator.MUL);  
 }  
  
 private void compute(Calculator.Operator operator) {  
 double operandOne;  
 double operandTwo;  
 try {  
 operandOne = getOperand(mOperandOneEditText);  
 operandTwo = getOperand(mOperandTwoEditText);  
 } catch (NumberFormatException nfe) {  
 Log.e(TAG, "NumberFormatException", nfe);  
 mResultTextView.setText(getString(R.string.computationError));  
 return;  
 }  
  
 String result;  
 switch (operator) {  
 case ADD:  
 result = String.valueOf(  
 mCalculator.add(operandOne, operandTwo));  
 break;  
 case SUB:  
 result = String.valueOf(  
 mCalculator.sub(operandOne, operandTwo));  
 break;  
 case DIV:  
 result = String.valueOf(  
 mCalculator.div(operandOne, operandTwo));  
 break;  
 case MUL:  
 result = String.valueOf(  
 mCalculator.mul(operandOne, operandTwo));  
 break;  
 default:  
 result = getString(R.string.computationError);  
 break;  
 }  
 mResultTextView.setText(result);  
 }  
  
 private static Double getOperand(EditText operandEditText) {  
 String operandText = getOperandText(operandEditText);  
 return Double.valueOf(operandText);  
 }  
  
 private static String getOperandText(EditText operandEditText) {  
 String operandText = operandEditText.getText().toString();  
 if (TextUtils.isEmpty(operandText)) {  
 throw new NumberFormatException("Operand cannot be empty!");  
 }  
 return operandText;  
 }  
}

Программный код, класс Calculator.

package com.example.android.SimpleCalcTest;

*/\*\*  
 \* \* Класс утилиты для SimpleCalc для выполнения фактических вычислений.  
 \*/*public class Calculator {  
  
 // Доступные операции  
 public enum Operator {ADD, SUB, DIV, MUL}  
  
 */\*\*  
 \* Операция сложения  
 \*/* public double add(double firstOperand, double secondOperand) {  
 return firstOperand + secondOperand;  
 }  
  
 */\*\*  
 \* Операция вычитания  
 \*/* public double sub(double firstOperand, double secondOperand) {  
 return firstOperand - secondOperand;  
 }  
  
 */\*\*  
 \* Операция разделения  
 \*/* public double div(double firstOperand, double secondOperand) {  
 if (secondOperand == 0 ) {  
 throw new IllegalArgumentException("You cannot divide by zero");  
 }  
 return firstOperand / secondOperand;  
 }  
  
 */\*\*  
 \* Операция умножения  
 \*/* public double mul(double firstOperand, double secondOperand) {  
 return firstOperand \* secondOperand;  
 }  
}

Собрали проект и включаем режим отладки.

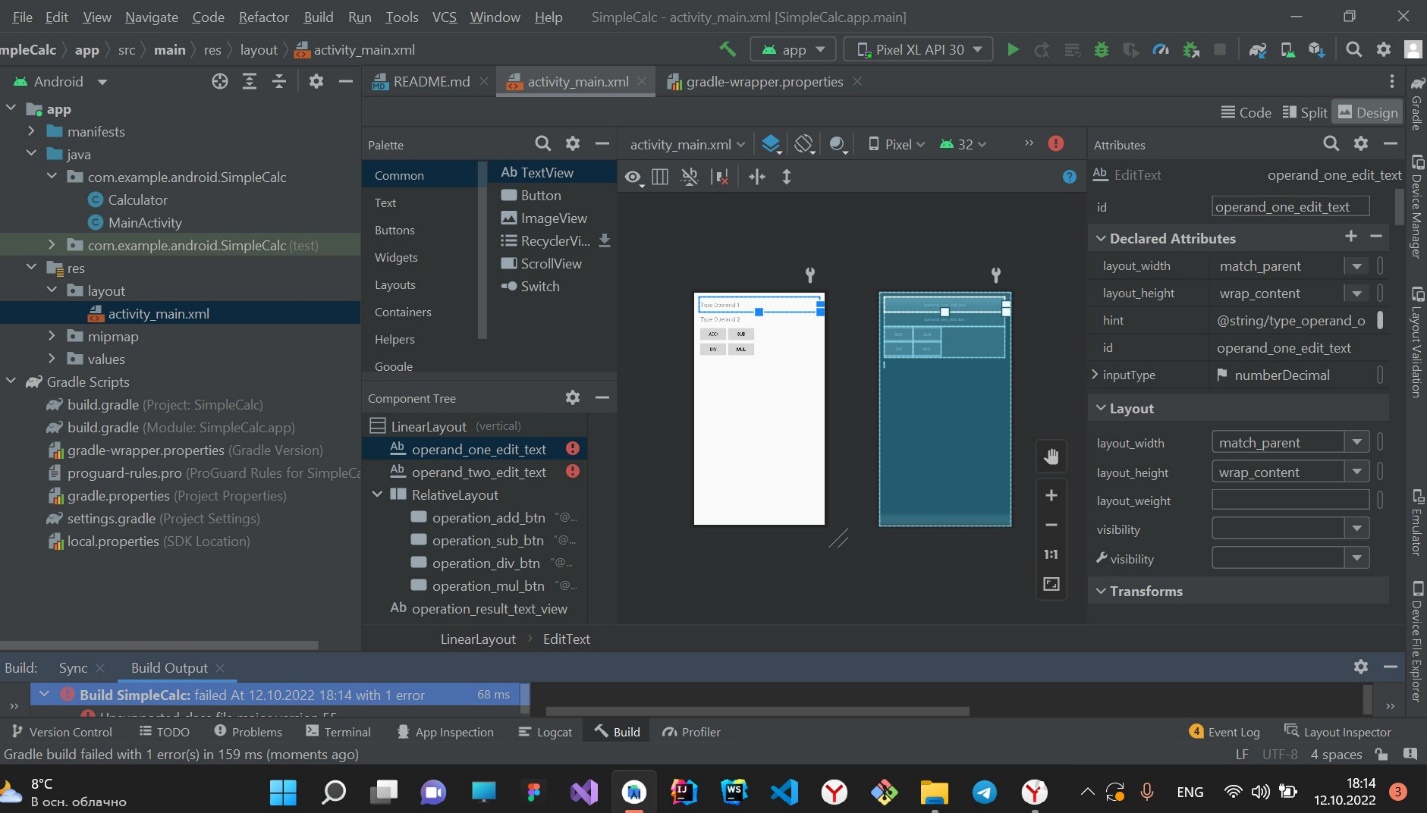


Рисунок 1 – Собранный проект в режиме отладки.

Основной класс проекта. Данным рисунком 2 показываю работоспособность проекта.

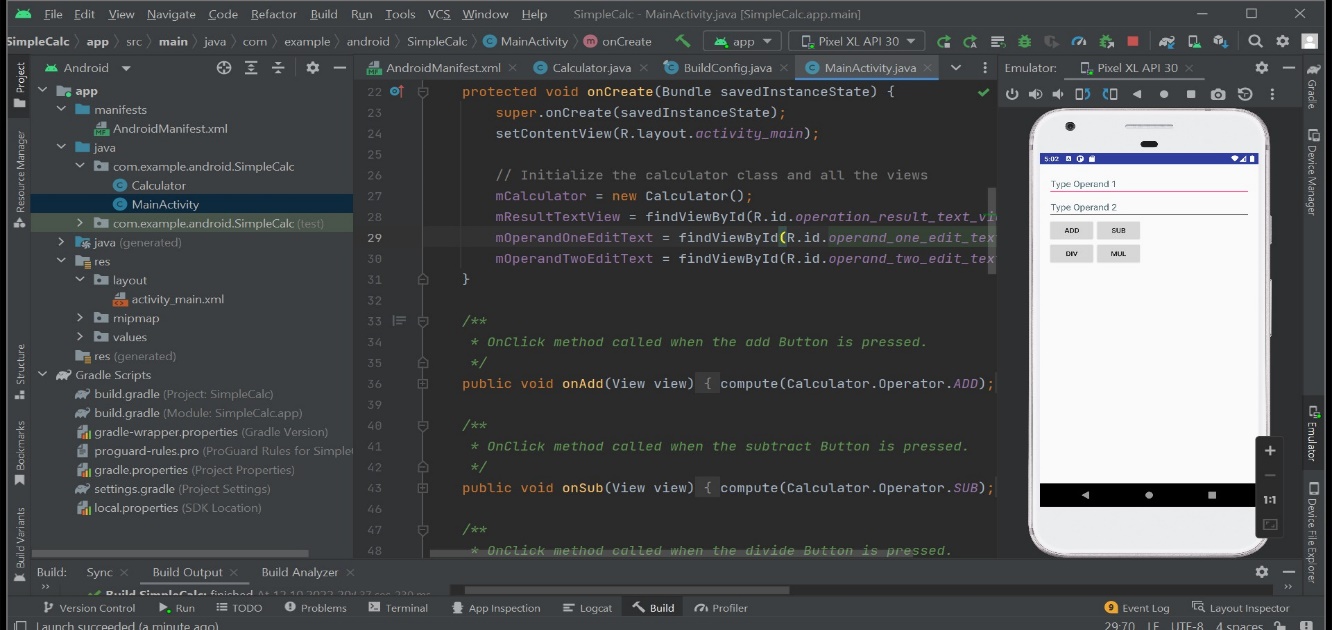


Рисунок 2 – правильное функционирование проекта.

## 

## **Анализ полученных результатов**

В результате выполнения данной лабораторной работы мной был разработан проект, в котором я научился запускать приложение в режиме отладки в эмуляторе или на устройстве. Как пошагово выполнить ваше приложение. Как установить и упорядочить точки останова. Как проверять и изменять переменные в отладчике.