МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Звіт з лабораторної роботи

«Калькулятор»

Виконала:

ст. групи ПI-15-1

Михневич Т. К.

Харків 2018

**Завдання**

Розробити калькулятор на 4 базові арифметичні операції з кнопками для введення аргументів і дій. Калькулятор повинен мати адаптивний користувацький інтерфейс в різних орієнтаціях екрана та на різної розподільної здатності.

**Опис реалізації**

У результаті виконання практичного заняття був розроблений калькулятор.

На рисунку 1 зображена схема розташування елементів Activity.



Рисунок 1 – Схема розташування елементів Activity

Як можна побачити зі схеми, інтерфейс додатку реалізований з використанням LinearLayout. Для пропорційного розміщення елементів на екрані використовується атрибути weightSum та layout\_weight.

Калькулятор дозволяє виконувати дії на базові 4 операції з цілими та дробовими числами. При натисненні на будь-яку кнопку на інтерфейсі спрацьовує обробник події onClick в залежності від того, яка кнопка буда натиснута – кнопка з цифрою, кнопка з оператором тощо.

Даний калькулятор виконує дії над двома числами, які зберігаються у стеку. Для виведення результату та даних використовується TextView з максимальною довжиною 16 символів.

При повороті екрану для збереження введених даних та інших змінних переопреділяємо методи onSaveInstanceState та onSaveRestoreState. У першому мі зберігаємо усі потрібні нам дані для коректної роботи калькулятора, а у другому зчитуємо їх у змінні.

На рисунку 2, 3 зображений інтерфейс калькулятора.

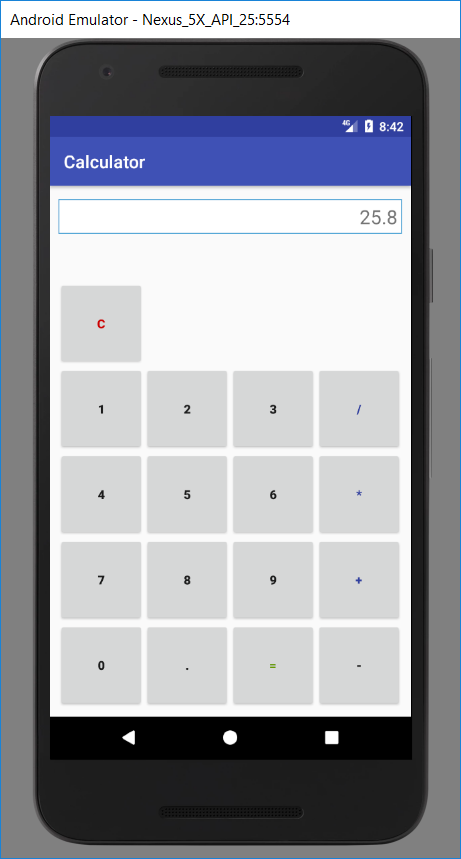


Рисунок 2 – Скріншот роботи калькулятора у portrait режимі

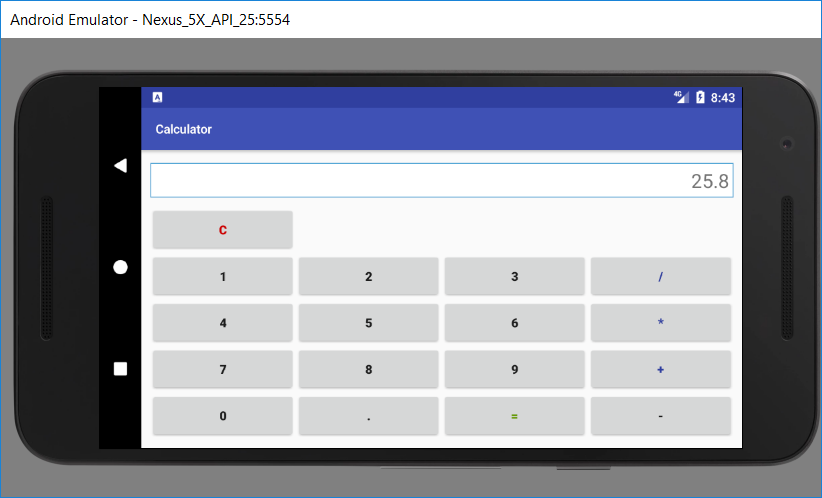


Рисунок 3 – Скріншот роботи калькулятора у landscape режимі

**Додаток 1. Фрагменту коду MainActivity**

**package** ua.nure.calculator;  
  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.os.Bundle;  
**import** android.view.View;  
**import** android.widget.Button;  
**import** android.widget.TextView;  
  
**import** java.util.Stack;  
  
**public class** MainActivity **extends** AppCompatActivity {  
  
 **private** String \_operator = Operators.EMPTY;  
 **private** Stack<Double> \_numbers = **new** Stack<>();  
 **private** TextView \_input;  
 **private boolean** \_needToClear = **false**;  
 **private boolean** \_numberWasEntered = **false**;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.main);  
 \_input = (TextView) findViewById(R.id.inputTv);  
 }  
  
 @Override  
 **public void** onSaveInstanceState(Bundle savedInstanceState) {  
 **super**.onSaveInstanceState(savedInstanceState);  
  
 savedInstanceState.putBoolean(**"needToClear"**, \_needToClear);  
 savedInstanceState.putBoolean(**"numberWasEntered"**, \_numberWasEntered);  
 savedInstanceState.putString(**"operator"**, \_operator);  
 savedInstanceState.putString(**"inputText"**, \_input.getText().toString());  
  
 **if**(\_numbers.size() >= 1)  
 {  
 savedInstanceState.putDouble(**"stackNumber1"**, \_numbers.pop());  
 }  
 **if**(\_numbers.size() >= 1)  
 {  
 savedInstanceState.putDouble(**"stackNumber2"**, \_numbers.pop());  
 }  
 }  
  
 @Override  
 **public void** onRestoreInstanceState(Bundle savedInstanceState) {  
 **super**.onRestoreInstanceState(savedInstanceState);  
  
 \_operator = savedInstanceState.getString(**"operator"**);  
 \_needToClear = savedInstanceState.getBoolean(**"needToClear"**);  
 \_numberWasEntered = savedInstanceState.getBoolean(**"numberWasEntered"**);  
 \_input.setText(savedInstanceState.getString(**"inputText"**));  
  
 Double num2 = savedInstanceState.getDouble(**"stackNumber2"**);  
 **if** (num2 != 0)  
 {  
 \_numbers.push(num2);  
 }  
  
 Double num1 = savedInstanceState.getDouble(**"stackNumber1"**);  
 **if** (num1 != 0)  
 {  
 \_numbers.push(num1);  
 }  
 }  
  
 **public void** clearInput(View view) {  
 \_input.setText(ValuesConstants.ZERO);  
 \_numbers.clear();  
 \_operator = Operators.EMPTY;  
 \_numberWasEntered = **false**;  
 \_needToClear = **false**;  
 }  
  
 **public void** onNumberClick(View view) {  
 Button number = (Button) view;  
  
 **if** (\_input.getText().toString().equals(ValuesConstants.ZERO) || \_needToClear) {  
 \_input.setText(ValuesConstants.EMPTY);  
 }  
  
 **if** (number.getText().toString().equals(ValuesConstants.DOT)) {  
 onDotClick();  
 } **else** {  
 \_input.append(number.getText());  
 }  
  
 \_needToClear = **false**;  
 \_numberWasEntered = **true**;  
 }  
  
 **public void** onDotClick() {  
 **if** (\_input.getText().toString().isEmpty()) {  
 \_input.append(**"0."**);  
 }  
  
 **if** (!\_input.getText().toString().contains(ValuesConstants.DOT))  
 \_input.append(ValuesConstants.DOT);  
 }  
  
 **public void** onEqualClick(View view) {  
 **if** (!\_numberWasEntered) **return**;  
 \_needToClear = **true**;  
 \_numbers.push(getInputValue());  
 processOperatorClick();  
  
 \_operator = Operators.EMPTY;  
 }  
  
 **public void** onOperatorClick(View view) {  
 **if** (!\_numberWasEntered) **return**;  
 \_needToClear = **true**;  
 \_numbers.push(getInputValue());  
  
 Button operator = (Button) view;  
 String currentOperator = operator.getText().toString();  
  
 **if** (\_operator.isEmpty() || \_numbers.size() < 2) {  
 \_operator = currentOperator;  
 **return**;  
 } **else** {  
 processOperatorClick();  
 \_operator = currentOperator;  
 }  
 }  
  
 **private void** processOperatorClick() {  
 Double res = 0.0;  
 **switch** (\_operator) {  
 **case** Operators.PLUS:  
 res = (\_numbers.pop() + \_numbers.pop());  
 **break**;  
 **case** Operators.MINUS:  
 res = (-\_numbers.pop() + \_numbers.pop());  
 **break**;  
 **case** Operators.MULTIPLICATION:  
 res = (\_numbers.pop() \* \_numbers.pop());  
 **break**;  
 **case** Operators.DIVISION:  
 Double divider = \_numbers.pop();  
 res = \_numbers.pop() / divider;  
 res = Double.isNaN(res) ? 0 : res;  
 **break**;  
 **default**:  
 **break**;  
 }  
  
 **if** (res % 1 == 0) {  
 Integer result = (**int**) Math.round(res);  
 \_input.setText(result.toString());  
 } **else** {  
 \_input.setText(res.toString());  
 }  
  
 \_numbers.push(res);  
 }  
  
 **private** Double getInputValue() {  
 **return** Double.parseDouble(\_input.getText().toString());  
 }  
}