МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

КАФЕДРА ПРОГРАМНОЇ ІНЖЕНЕРІЇ

Звіт

з лабораторної роботи № 3 з дисципліни

Програмування під Android

Виконав: Перевірив:

ст. гр. ПЗПІ-23-5 ст. викладач кафедри ПІ

Даниленко Назар Сергійович Сокорчук Ігор Петрович

Харків 2024

ЗАВДАННЯ

Розробити калькулятор на 4 базові арифметичні операції з кнопками для введення аргументів і дій.

Калькулятор повинен мати адаптивний користувацький інтерфейс у різних орієнтаціях екрану та на екранах різної роздільної здатності.

1. Хід роботи

В рамках лабораторної роботи було розроблено калькулятор, який виконує чотири базові арифметичні операції: додавання, віднімання, множення та ділення. Калькулятор має інтуїтивно зрозумілий інтерфейс, який дозволяє користувачу вводити числа та операції за допомогою кнопок. Основна мета — створити адаптивний користувацький інтерфейс, який підходить для різних роздільних здатностей екрана та орієнтацій.

1. Розробка інтерфейсу Для створення графічного інтерфейсу використано LinearLayout та GridLayout. Основні компоненти калькулятора представлені у вигляді кнопок, розміщених у сітці для зручності доступу. (Див додаток А)
2. Адаптивний дизайн Калькулятор використовує LinearLayout та GridLayout для організації елементів на екрані. Цей підхід дозволяє автоматично адаптуватися до різних розмірів екрана, оскільки компоненти займатимуть доступний простір.
3. Реалізація функціональності Основна логіка калькулятора реалізована в класі MainActivity, де обробляються події натискань кнопок, а також виконуються арифметичні операції. (Див додаток B) Обробка натискань кнопок: Для кнопок чисел і операторів використано OnClickListener, який додає натиснуте значення до рядка вводу і оновлює відображення результату.

private void setButtonListeners() {

int[] numberButtons = {

R.id.btn0, R.id.btn1, R.id.btn2, R.id.btn3,

R.id.btn4, R.id.btn5, R.id.btn6, R.id.btn7,

R.id.btn8, R.id.btn9

};

View.OnClickListener numberClickListener = view -> {

if (isNewOperation) {

input.setLength(0);

isNewOperation = false;

}

Button button = (Button) view;

input.append(button.getText());

resultTextView.setText(input.toString());

};

for (int id : numberButtons) {

findViewById(id).setOnClickListener(numberClickListener);

}

findViewById(R.id.btnPlus).setOnClickListener(view -> setOperator("+"));

findViewById(R.id.btnMinus).setOnClickListener(view -> setOperator("-"));

findViewById(R.id.btnMultiply).setOnClickListener(view -> setOperator("×"));

findViewById(R.id.btnDivide).setOnClickListener(view -> setOperator("÷"));

findViewById(R.id.btnEqual).setOnClickListener(view -> calculateResult());

findViewById(R.id.btnClear).setOnClickListener(view -> {

input.setLength(0);

firstOperand = 0;

operator = "";

isNewOperation = true;

resultTextView.setText("0");

});

}

Визначення операцій: Метод setOperator зберігає перший операнд і оператор, а також переключає стан для нової операції.

private void setOperator(String op) {

if (!input.toString().isEmpty()) {

firstOperand = Double.parseDouble(input.toString());

operator = op;

isNewOperation = true;

}

}

Обчислення результату: Метод calculateResult виконує арифметичну операцію на основі вибраного оператора і відображає результат на екрані. Оброблено ситуацію ділення на нуль.

private void calculateResult() {

if (!input.toString().isEmpty() && !operator.isEmpty()) {

double secondOperand = Double.parseDouble(input.toString());

double result = 0;

switch (operator) {

case "+":

result = firstOperand + secondOperand;

break;

case "-":

result = firstOperand - secondOperand;

break;

case "×":

result = firstOperand \* secondOperand;

break;

case "÷":

if (secondOperand != 0) {

result = firstOperand / secondOperand;

} else {

resultTextView.setText("Error");

return;

}

break;

}

resultTextView.setText(String.valueOf(result));

input.setLength(0);

input.append(result);

operator = "";

isNewOperation = true;

}

}

Очистка вводу: Метод clearInput скидає всі введені значення та оператори, відновлюючи початковий стан калькулятора.

private void clearInput() {

input.setLength(0);

firstOperand = 0;

operator = "";

isNewOperation = true;

resultTextView.setText("0");

}

1. Висновки

Реалізований калькулятор забезпечує базову функціональність для виконання арифметичних операцій (додавання, віднімання, множення та ділення). Завдяки розділенню логіки на окремі методи, код залишається структурованим і легко підтримуваним. Обробка натискань кнопок здійснюється через слухачі подій (OnClickListener), що спрощує керування введенням користувача та оновленням інтерфейсу в реальному часі.

Метод setOperator коректно зберігає перший операнд і вибраний оператор, дозволяючи виконувати операції. Функція calculateResult обробляє арифметичні обчислення, включаючи перевірку на ділення на нуль для уникнення помилок. Функціонал очищення введених даних, реалізований у clearInput, забезпечує повернення додатка до початкового стану, що робить взаємодію з калькулятором інтуїтивною.

Ця реалізація є ефективною для простих арифметичних задач та може бути розширена додатковими функціями, такими як підтримка більше операцій (наприклад, відсотків чи піднесення до степеня) або введення десяткових чисел. Структурований код забезпечує легкість у подальшій розробці та вдосконаленні.

Посилання на Ютуб<https://youtu.be/P1Dx3afgfms>

1. Додатки

Додаток А:

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

<LinearLayout

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

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical"

android:gravity="center"

android:padding="16dp">

<TextView

android:id="@+id/resultTextView"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="0"

android:gravity="end"

android:textSize="32sp"

android:padding="16dp"

android:background="#E0E0E0" />

<GridLayout

android:layout\_width="match\_parent"

android:layout\_height="0dp"

android:layout\_weight="1"

android:columnCount="4"

android:rowCount="5"

android:padding="8dp">

<Button

android:text="7"

android:id="@+id/btn7"

style="@style/CalcButton"/>

<Button

android:text="8"

android:id="@+id/btn8"

style="@style/CalcButton"/>

<Button

android:text="9"

android:id="@+id/btn9"

style="@style/CalcButton"/>

<Button

android:text="÷"

android:id="@+id/btnDivide"

style="@style/CalcButton"/>

<Button

android:text="4"

android:id="@+id/btn4"

style="@style/CalcButton"/>

<Button

android:text="5"

android:id="@+id/btn5"

style="@style/CalcButton"/>

<Button

android:text="6"

android:id="@+id/btn6"

style="@style/CalcButton"/>

<Button

android:text="×"

android:id="@+id/btnMultiply"

style="@style/CalcButton"/>

<Button

android:text="1"

android:id="@+id/btn1"

style="@style/CalcButton"/>

<Button

android:text="2"

android:id="@+id/btn2"

style="@style/CalcButton"/>

<Button

android:text="3"

android:id="@+id/btn3"

style="@style/CalcButton"/>

<Button

android:text="-"

android:id="@+id/btnMinus"

style="@style/CalcButton"/>

<Button

android:text="0"

android:id="@+id/btn0"

style="@style/CalcButton"/>

<Button

android:text="C"

android:id="@+id/btnClear"

style="@style/CalcButton"/>

<Button

android:text="="

android:id="@+id/btnEqual"

style="@style/CalcButton"/>

<Button

android:text="+"

android:id="@+id/btnPlus"

style="@style/CalcButton"/>

</GridLayout>

</LinearLayout>

Додаток B:

public class MainActivity extends AppCompatActivity {

private TextView resultTextView;

private StringBuilder input = new StringBuilder();

private double firstOperand = 0;

private String operator = "";

private boolean isNewOperation = true;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

resultTextView = findViewById(R.id.resultTextView);

setButtonListeners();

}

private void clearInput() {

input.setLength(0);

firstOperand = 0;

operator = "";

isNewOperation = true;

resultTextView.setText("0");

}

private void setButtonListeners() {

int[] numberButtons = {

R.id.btn0, R.id.btn1, R.id.btn2, R.id.btn3,

R.id.btn4, R.id.btn5, R.id.btn6, R.id.btn7,

R.id.btn8, R.id.btn9

};

View.OnClickListener numberClickListener = view -> {

if (isNewOperation) {

input.setLength(0);

isNewOperation = false;

}

Button button = (Button) view;

input.append(button.getText());

resultTextView.setText(input.toString());

};

for (int id : numberButtons) {

findViewById(id).setOnClickListener(numberClickListener);

}

findViewById(R.id.btnPlus).setOnClickListener(view -> setOperator("+"));

findViewById(R.id.btnMinus).setOnClickListener(view -> setOperator("-"));

findViewById(R.id.btnMultiply).setOnClickListener(view -> setOperator("×"));

findViewById(R.id.btnDivide).setOnClickListener(view -> setOperator("÷"));

findViewById(R.id.btnEqual).setOnClickListener(view -> calculateResult());

findViewById(R.id.btnClear).setOnClickListener(view -> {

input.setLength(0);

firstOperand = 0;

operator = "";

isNewOperation = true;

resultTextView.setText("0");

});

}

private void setOperator(String op) {

if (!input.toString().isEmpty()) {

firstOperand = Double.parseDouble(input.toString());

operator = op;

isNewOperation = true;

}

}

private void calculateResult() {

if (!input.toString().isEmpty() && !operator.isEmpty()) {

double secondOperand = Double.parseDouble(input.toString());

double result = 0;

switch (operator) {

case "+":

result = firstOperand + secondOperand;

break;

case "-":

result = firstOperand - secondOperand;

break;

case "×":

result = firstOperand \* secondOperand;

break;

case "÷":

if (secondOperand != 0) {

result = firstOperand / secondOperand;

} else {

resultTextView.setText("Error");

return;

}

break;

}

resultTextView.setText(String.valueOf(result));

input.setLength(0);

input.append(result);

operator = "";

isNewOperation = true;

}

}

}