

## Assignment-10

### Problem Statement:

Develop a GUI-based calculator using Java Swing components such as buttons, text fields, and labels. The calculator should perform basic arithmetic operations (addition, subtraction, multiplication, and division) and display results appropriately.

### Objectives:

To understand the usage of Java Swing components.

To implement event-driven programming using ActionListener.

To create a user-friendly GUI for arithmetic operations.

### Software & Hardware Requirements:

**Software:** JDK 8 or later, IDE (Eclipse/NetBeans/IntelliJ IDEA)

**Hardware:** Standard PC/Laptop with Windows/Linux/macOS

### Theory:

Swing is a part of Java Foundation Classes (JFC) used to create GUI applications. It provides various components like JButton, JTextField, JLabel, etc., for designing user interfaces. Event handling in Swing is done using event listeners, such as ActionListener for button clicks.

### Procedure:

1. Create a Java class and import required Swing libraries.
2. Design the GUI layout using JFrame, JTextField, and JButton components.
3. Implement ActionListener to handle button click events.
4. Retrieve input values from JTextField and perform the corresponding arithmetic operation.
5. Display the output in the text field.
6. Handle exceptions such as division by zero.
7. Compile and execute the program.

### Code Implementation:

```

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class Calculator implements ActionListener {
    JFrame frame;
    JTextField textField;
    JButton[] numberButtons = new JButton[10];
    JButton[] functionButtons = new JButton[6];
    JButton addButton, subButton, mulButton, divButton, clrButton, eqButton;
    JPanel panel;

    double num1, num2, result;
    char operator;

    public Calculator() {
        frame = new JFrame("Calculator");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(350, 500);
        frame.setLayout(null);

        textField = new JTextField();
        textField.setBounds(30, 40, 280, 40);
        frame.add(textField);

        addButton = new JButton("+");
        subButton = new JButton("-");
        mulButton = new JButton("*");
        divButton = new JButton("/");
        clrButton = new JButton("C");
        eqButton = new JButton("=");

        functionButtons[0] = addButton;
        functionButtons[1] = subButton;
        functionButtons[2] = mulButton;
        functionButtons[3] = divButton;
        functionButtons[4] = clrButton;
        functionButtons[5] = eqButton;

        for (int i = 0; i < 6; i++) {
            functionButtons[i].addActionListener(this);
            functionButtons[i].setFont(new Font("Arial", Font.BOLD, 18));
        }

        for (int i = 0; i < 10; i++) {
            numberButtons[i] = new JButton(String.valueOf(i));
            numberButtons[i].addActionListener(this);
            numberButtons[i].setFont(new Font("Arial", Font.BOLD, 18));
        }
    }

```

```

panel = new JPanel();
panel.setBounds(30, 100, 280, 300);
panel.setLayout(new GridLayout(4, 4, 10, 10));

for (int i = 1; i < 10; i++) {
    panel.add(numberButtons[i]);
}
panel.add(addButton);
panel.add(numberButtons[0]);
panel.add(subButton);
panel.add(mulButton);
panel.add(divButton);
panel.add(eqButton);
panel.add(clrButton);

frame.add(panel);
frame.setVisible(true);
}

@Override
public void actionPerformed(ActionEvent e) {
    for (int i = 0; i < 10; i++) {
        if (e.getSource() == numberButtons[i]) {
            textField.setText(textField.getText().concat(String.valueOf(i)));
        }
    }
    if (e.getSource() == addButton) {
        num1 = Double.parseDouble(textField.getText());
        operator = '+';
        textField.setText("");
    }
    if (e.getSource() == subButton) {
        num1 = Double.parseDouble(textField.getText());
        operator = '-';
        textField.setText("");
    }
    if (e.getSource() == mulButton) {
        num1 = Double.parseDouble(textField.getText());
        operator = '*';
        textField.setText("");
    }
    if (e.getSource() == divButton) {
        num1 = Double.parseDouble(textField.getText());
        operator = '/';
        textField.setText("");
    }
    if (e.getSource() == eqButton) {
        num2 = Double.parseDouble(textField.getText());
        switch (operator) {

```

```

        case '+': result = num1 + num2; break;
        case '-': result = num1 - num2; break;
        case '*': result = num1 * num2; break;
        case '/': result = num2 != 0 ? num1 / num2 : 0; break;
    }
    textField.setText(String.valueOf(result));
}
if (e.getSource() == clrButton) {
    textField.setText("");
}
}
}

public static void main(String[] args) {
    new Calculator();
}
}

```

### **Expected Output:**

The calculator should allow users to input numbers, perform basic operations, and display results.

The clear button should reset the display.

Division by zero should be handled.

### **Conclusion:**

Implementing a GUI-based calculator using Swing improves understanding of Java GUI development and event handling.

This project helps in creating interactive applications with Swing components.