MIT World Peace University

Object Oriented Programming with Java and C++
Second Year B. Tech, Semester 1

Developing a Simple Graphical Calculator using Swing in Java

Practical Report
Assignment 8

Prepared By

Krishnaraj Thadesar
Cyber Security and Forensics
Batch A1, PA 20

November 19, 2022

Contents

1 Aim and Objectives

Aim

To Develop a simple calculator using Swing in Java

Objective

1. To understand concept of AWT and Java swings
2. To explore Java Swing containers

2 Problem Statement

Write a Java program to create a simple calculator with the help of java swing.

3 Theory

4 Platform

Operating System: Arch Linux x86-64
IDEs or Text Editors Used: Visual Studio Code
Compilers : g++ and gcc on linux for C++, and javac, with JDK 18.0.2 for Java

5 Input

The numbers and the Operators.

6 Output

The Output of the entered Calculation in the Display Section of the Calculator

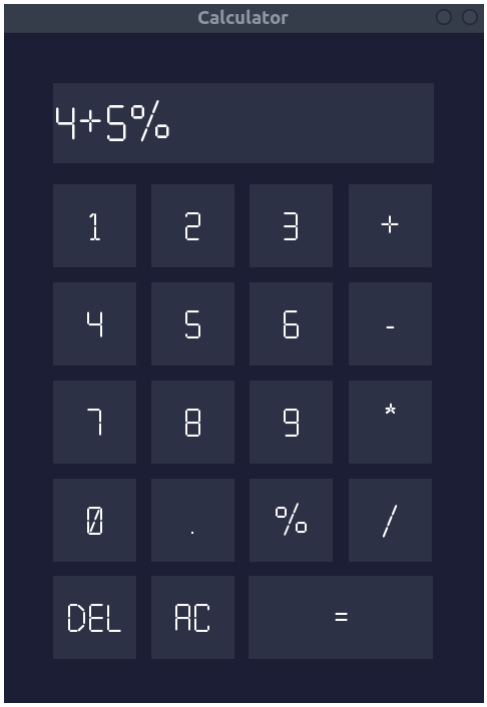

Figure 1: Calculator with Java Swing

7 Code

```
// Krishnaraj Thadesar
// Batch A1, PA20
// OOPCJ Assignment 9
// Making a Calculator in Java using Swing

package org.OOPCJ.Krishnaraj;
import org.mariuszgromada.math.mxparser.*;

import javax.swing.*;
import java.awt.*;
import java.io.*;

class Colors {
    static Color primaryColor = new Color(255, 255, 255); // text color
    static Color bgColor = new Color(27, 30, 52); // background
    static Color secondaryColor = new Color(44, 49, 70); // upper background
    static Color secondaryColorRollover = new Color(53, 59, 80); // upper
    background
    static Color accentColor = new Color(26, 122, 230); // Accent
}

class Numpad extends JPanel {
    JButton[] numbers = new JButton[12];

    Numpad() {
        this.setFocusable(true);
        this.setVisible(true);
        this.setBorder(null);
```

```
28          this.setBounds(50, 150, 280, 380);
29          this.setBackground(Colors.bgColor);
30          this.setLayout(new GridLayout(4, 3, 15, 15));
31          createButtons();
32          for (int i = 0; i < numbers.length; i++) {
33              this.add(numbers[i]);
34          }
35      }
36
37      public void createButtons() {
38          for (int i = 0; i < 12; i++) {
39              numbers[i] = new JButton();
40              numbers[i].setText(String.valueOf(i + 1));
41              numbers[i].setFocusPainted(false);
42              numbers[i].setContentAreaFilled(false);
43              numbers[i].setOpaque(true);
44              numbers[i].setBorder(null);
45              numbers[i].setBackground(Colors.secondaryColor);
46              numbers[i].setForeground(Colors.primaryColor);
47              numbers[i].setFont(Calculator.buttonFont);
48              final JButton temp = numbers[i];
49              temp.addChangeListener(evt -> {
50                  if (temp.getModel().isPressed()) {
51                      temp.setForeground(Colors.primaryColor);
52                      temp.setBackground(Colors.secondaryColorRollover);
53                  } else if (temp.getModel().isRollover()) {
54                      temp.setForeground(Colors.accentColor);
55                      temp.setBackground(Colors.secondaryColorRollover);
56                  } else {
57                      temp.setForeground(Colors.primaryColor);
58                      temp.setBackground(Colors.secondaryColor);
59                  }
60              });
61              temp.addActionListener(e -> {
62                  Calculator.display.setText(Calculator.display.getText() + ((
    JButton) e.getSource()).getText());
63              });
64          }
65          numbers[9].setText("0");
66          numbers[10].setText(".");
67          numbers[11].setText("%");
68      }
69  }
70
71  class Operators_pnl extends JPanel {
72      static JButton[] operators = new JButton[4];
73      static String currentOperator;
74
75      Operators_pnl() {
76          this.setFocusable(true);
77          this.setVisible(true);
78          this.setBorder(null);
79          this.setBounds(345, 150, (int) 250 / 3, 380);
80          this.setBackground(Colors.bgColor);
81          this.setLayout(new GridLayout(4, 1, 0, 15));
82          createButtons();
83          for (int i = 0; i < operators.length; i++) {
84              this.add(operators[i]);
85          }
```

```
86      }
87
88      public void createButtons() {
89          for (int i = 0; i < 4; i++) {
90              operators[i] = new JButton();
91              operators[i].setFocusPainted(false);
92              operators[i].setContentAreaFilled(false);
93              operators[i].setOpaque(true);
94              operators[i].setBorder(null);
95              operators[i].setBackground(Colors.secondaryColor);
96              operators[i].setForeground(Colors.primaryColor);
97              operators[i].setFont(Calculator.buttonFont);
98              final JButton temp = operators[i];
99              temp.addChangeListener(evt -> {
100                 if (temp.getModel().isPressed()) {
101                     temp.setForeground(Colors.primaryColor);
102                     temp.setBackground(Colors.secondaryColorRollover);
103                 } else if (temp.getModel().isRollover()) {
104                     temp.setForeground(Colors.accentColor);
105                     temp.setBackground(Colors.secondaryColorRollover);
106                 } else {
107                     temp.setForeground(Colors.primaryColor);
108                     temp.setBackground(Colors.secondaryColor);
109                 }
110             });
111             temp.addActionListener(e -> {
112                 if (!Calculator.operator_used) {
113                     Calculator.display.setText(Calculator.display.getText() + ((
    JButton) e.getSource()).getText());
114                 }
115             });
116         }
117         operators[0].setText("+");
118         operators[1].setText("-");
119         operators[2].setText("*");
120         operators[3].setText("/");
121     }
122 }
123
124 // Main Calculator Frame no panels
125 class Calculator extends JFrame {
126     static double number_1, number_2;
127     static boolean operator_used = false;
128     JButton clearBtn, backspaceBtn, resultBtn;
129     static JTextField display;
130     Numpad numpad;
131     Operators_pnl operators_pnl;
132     static Font buttonFont;
133
134     Calculator() {
135         this.setTitle("Calculator");
136         this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
137         this.setResizable(false);
138         this.setUndecorated(false);
139         this.setPreferredSize(new Dimension(480, 670));
140         this.getContentPane().setBackground(Colors.bgColor);
141         this.setLayout(null);
142         createFonts();
143         createPanels();
```

```
144         createButtons();
145         this.add(display);
146         this.add(numpad);
147         this.add(operators_pnl);
148         this.add(clearBtn);
149         this.add(backspaceBtn);
150         this.add(resultBtn);
151         this.pack();
152         this.setVisible(true);
153         this.setLocationRelativeTo(null);
154     }
155 
156     public void createPanels() {
157         numpad = new Numpad();
158         operators_pnl = new Operators_pnl();
159     }
160 
161     public void createButtons() {
162         display = new JTextField();
163         display.setBounds(50, 50, 380, 80);
164         display.setOpaque(true);
165         display.setAlignmentX(RIGHT_ALIGNMENT);
166         display.setBorder(null);
167         display.setBackground(Colors.secondaryColor);
168         display.setForeground(Colors.primaryColor);
169         display.setFont(Calculator.buttonFont.deriveFont(55f));
170         display.addActionListener(e -> {
171         });
172 
173         clearBtn = new JButton();
174         clearBtn.setText("AC");
175         clearBtn.setBounds(50 + 15 + (int) 250 / 3, 542, (int) 250 / 3, (int) 250
    / 3);
176         clearBtn.setFocusPainted(false);
177         clearBtn.setContentAreaFilled(false);
178         clearBtn.setOpaque(true);
179         clearBtn.setBorder(null);
180         clearBtn.setBackground(Colors.secondaryColor);
181         clearBtn.setForeground(Colors.primaryColor);
182         clearBtn.setFont(Calculator.buttonFont);
183         clearBtn.addChangeListener(evt -> {
184             if (clearBtn.getModel().isPressed()) {
185                 clearBtn.setForeground(Colors.primaryColor);
186                 clearBtn.setBackground(Colors.secondaryColorRollover);
187             } else if (clearBtn.getModel().isRollover()) {
188                 clearBtn.setForeground(Colors.accentColor);
189                 clearBtn.setBackground(Colors.secondaryColorRollover);
190             } else {
191                 clearBtn.setForeground(Colors.primaryColor);
192                 clearBtn.setBackground(Colors.secondaryColor);
193             }
194         });
195         clearBtn.addActionListener(e -> {
196             display.setText("");
197             operator_used = false;
198             Operators_pnl.operators[0].setEnabled(true);
199             Operators_pnl.operators[1].setEnabled(true);
200             Operators_pnl.operators[2].setEnabled(true);
201             Operators_pnl.operators[3].setEnabled(true);
```

```
202         });
203 
204         backspaceBtn = new JButton();
205         backspaceBtn.setText("DEL");
206         backspaceBtn.setBounds(50, 542, (int) 250 / 3, (int) 250 / 3);
207         backspaceBtn.setFocusPainted(false);
208         backspaceBtn.setContentAreaFilled(false);
209         backspaceBtn.setOpaque(true);
210         backspaceBtn.setBorder(null);
211         backspaceBtn.setBackground(Colors.secondaryColor);
212         backspaceBtn.setForeground(Colors.primaryColor);
213         backspaceBtn.setFont(Calculator.buttonFont);
214         backspaceBtn.addChangeListener(evt -> {
215             if (backspaceBtn.getModel().isPressed()) {
216                 backspaceBtn.setForeground(Colors.primaryColor);
217                 backspaceBtn.setBackground(Colors.secondaryColorRollover);
218             } else if (backspaceBtn.getModel().isRollover()) {
219                 backspaceBtn.setForeground(Colors.accentColor);
220                 backspaceBtn.setBackground(Colors.secondaryColorRollover);
221             } else {
222                 backspaceBtn.setForeground(Colors.primaryColor);
223                 backspaceBtn.setBackground(Colors.secondaryColor);
224             }
225         });
226         backspaceBtn.addActionListener(e -> {
227             try {
228                 display.setText(display.getText().substring(0, display.getText().
    length() - 1));
229             } catch (Exception f) {
230                 System.out.println("You got nothing on screen then how can you
    delete? ");
231             }
232         });
233 
234         resultBtn = new JButton();
235         resultBtn.setText("=");
236         resultBtn.setBounds(245, 542, (int) 184, (int) 250 / 3);
237         resultBtn.setFocusPainted(false);
238         resultBtn.setContentAreaFilled(false);
239         resultBtn.setOpaque(true);
240         resultBtn.setBorder(null);
241         resultBtn.setBackground(Colors.secondaryColor);
242         resultBtn.setForeground(Colors.primaryColor);
243         resultBtn.setFont(Calculator.buttonFont);
244         resultBtn.addChangeListener(evt -> {
245             if (resultBtn.getModel().isPressed()) {
246                 resultBtn.setForeground(Colors.primaryColor);
247                 resultBtn.setBackground(Colors.secondaryColorRollover);
248             } else if (resultBtn.getModel().isRollover()) {
249                 resultBtn.setForeground(Colors.accentColor);
250                 resultBtn.setBackground(Colors.secondaryColorRollover);
251             } else {
252                 resultBtn.setForeground(Colors.primaryColor);
253                 resultBtn.setBackground(Colors.secondaryColor);
254             }
255         });
256         resultBtn.addActionListener(e -> {
257             String currentString = display.getText();
258             Expression expr = new Expression(currentString);
```

```
259             display.setText(String.valueOf(expr.calculate()));
260         });
261     }
262 
263     public static void createFonts() {
264         try {
265             buttonFont = Font.createFont(Font.TRUETYPE_FONT, new File("/run/media/
    krishnaraj/Classes/University/Second Year/First Semister/OOPJC/Programs/
    java_implementations/assignment_8/Calculator/src/main/resources/Calculator.ttf"
    )).deriveFont(45f);
266             GraphicsEnvironment ge = GraphicsEnvironment.
    getLocalGraphicsEnvironment();
267             // register the font
268             ge.registerFont(buttonFont);
269 
270         } catch (FontFormatException | IOException e) {
271             e.printStackTrace();
272         }
273     }
274 }
275 
276 public class Main {
277     static Calculator calc;
278 
279     public static void main(String[] args) {
280         calc = new Calculator();
281     }
282 }
```

Listing 1: Calculator.java

8 Dependencies

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <project xmlns="http://maven.apache.org/POM/4.0.0"
3           xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4           xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache
    .org/xsd/maven-4.0.0.xsd">
5      <modelVersion>4.0.0</modelVersion>
6 
7      <groupId>org.example</groupId>
8      <artifactId>org.OOPCJ.Krishnaraj.Calculator</artifactId>
9      <version>1.0-SNAPSHOT</version>
10 
11     <properties>
12         <maven.compiler.source>18</maven.compiler.source>
13         <maven.compiler.target>18</maven.compiler.target>
14         <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
15     </properties>
16     <dependencies>
17         <dependency>
18             <groupId>org.mariuszgromada.math</groupId>
19             <artifactId>MathParser.org-mXparser</artifactId>
20             <version>5.0.7</version>
21         </dependency>
22 
23     </dependencies>
24 
```

```
25 </project>
```

Listing 2: pom.xml

9 Conclusion

Thus, implemented simple calculator with the help of java swing and performed various operations.

10 FAQs

1. *What are the methods of component class in Java Swing?*
2. *How many ways to create a frame in Java Swing? Explain with examples*
3. *What are the methods of JLabel class in Java Swing?*
4. *What are the methods of AbstractButton class in Java Swing?*
5. *Write a simple Java Swing program of displaying image on the button?*