

```
1 import java.awt.*;
2 import java.lang.reflect.InvocationTargetException;
3 import java.util.Scanner;
4 import java.awt.event.ActionEvent;
5 import java.awt.event.ActionListener;
6 import javax.swing.*;
7
8 public class Calculator {
9     private JFrame frame;
10    private static JTextField textField1;
11    private static JTextField textField2;
12    private static JTextField textFieldResult;
13    private JButton buttonAdd, buttonSubtract, buttonMultiply, buttonDivide;
14    private JPanel panel;
15    private Scanner scanner;
16
17    public Calculator() {
18        frame = new JFrame("Calculator");
19
20        textField1 = new JTextField(0);
21        textField1.setBackground(Color.DARK_GRAY);
22        textField1.setFont(new Font("Arial", Font.BOLD, 20));
23        textField1.setForeground(Color.WHITE);
24
25        textField2 = new JTextField(0);
26        textField2.setBackground(Color.DARK_GRAY);
27        textField2.setFont(new Font("Arial", Font.BOLD, 20));
28        textField2.setForeground(Color.WHITE);
29
30        buttonAdd = new JButton("+");
31        buttonAdd.setBackground(Color.DARK_GRAY);
32        buttonAdd.setFont(new Font("Arial", Font.BOLD, 20));
33        buttonAdd.setForeground(Color.WHITE);
34
35        buttonSubtract = new JButton("-");
36        buttonSubtract.setBackground(Color.DARK_GRAY);
37        buttonSubtract.setFont(new Font("Arial", Font.BOLD, 20));
38        buttonSubtract.setForeground(Color.WHITE);
39
40        buttonMultiply = new JButton("*");
41        buttonMultiply.setBackground(Color.DARK_GRAY);
42        buttonMultiply.setFont(new Font("Arial", Font.BOLD, 20));
43        buttonMultiply.setForeground(Color.WHITE);
44
45        buttonDivide = new JButton("/");
46        buttonDivide.setBackground(Color.DARK_GRAY);
47        buttonDivide.setFont(new Font("Arial", Font.BOLD, 20));
48        buttonDivide.setForeground(Color.WHITE);
49
50        textFieldResult = new JTextField(0);
51        textFieldResult.setBackground(Color.DARK_GRAY);
52        textFieldResult.setFont(new Font("Arial", Font.BOLD, 20));
53        textFieldResult.setForeground(Color.WHITE);
54
55
56        panel = new JPanel();
57        panel.setLayout(new GridLayout(5, 4));
58        panel.add(textField1);
59        panel.add(textField2);
60        panel.add(buttonAdd);
```

```

61     panel.add(buttonSubtract);
62     panel.add(buttonMultiply);
63     panel.add(buttonDivide);
64     panel.add(textFieldResult);
65     frame.add(panel, BorderLayout.CENTER);
66
67     frame.setSize(640, 480);
68     frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
69
70     scanner = new Scanner(System.in);
71
72     buttonAdd.addActionListener(new ActionListener() {
73         @Override
74         public void actionPerformed(ActionEvent e) {
75             double number1 = Double.parseDouble(textField1.getText());
76             double number2 = Double.parseDouble(textField2.getText());
77             double result = calculate(number1, number2, "+");
78             textFieldResult.setText("REZULTAT: " + String.valueOf(result));
79         }
80     });
81
82     buttonSubtract.addActionListener(new ActionListener() {
83         @Override
84         public void actionPerformed(ActionEvent e) {
85             double number1 = Double.parseDouble(textField1.getText());
86             double number2 = Double.parseDouble(textField2.getText());
87             double result = calculate(number1, number2, "-");
88             textFieldResult.setText("REZULTAT: " + String.valueOf(result));
89         }
90     });
91
92     buttonMultiply.addActionListener(new ActionListener() {
93         @Override
94         public void actionPerformed(ActionEvent e) {
95             double number1 = Double.parseDouble(textField1.getText());
96             double number2 = Double.parseDouble(textField2.getText());
97             double result = calculate(number1, number2, "*");
98             textFieldResult.setText("REZULTAT: " + String.valueOf(result));
99         }
100    });
101
102    buttonDivide.addActionListener(new ActionListener() {
103        @Override
104        public void actionPerformed(ActionEvent e) {
105            double number1 = Double.parseDouble(textField1.getText());
106            double number2 = Double.parseDouble(textField2.getText());
107            double result = calculate(number1, number2, "/");
108            textFieldResult.setText("REZULTAT: " + String.valueOf(result));
109        }
110    });
111 }
112
113 public double calculate(double num1, double num2, String operation) {
114     // Check the value of the "operation" argument and perform the corresponding
calculation
115     if (operation.equals("+")) {
116         return num1 + num2;
117     } else if (operation.equals("-")) {
118         return num1 - num2;
119     } else if (operation.equals("*")) {

```

```
120         return num1 * num2;
121     } else if (operation.equals("/")) {
122         return num1 / num2;
123     } else {
124         // If the operation is invalid, return 0
125         return 0;
126     }
127 }
128
129 public static void main(String[] args) {
130     try {
131         // Create a new Calculator object
132         Calculator calculator = new Calculator();
133         // Set the frame to be visible
134         calculator.frame.setVisible(true);
135         SwingUtilities.invokeLater(new Runnable() {
136             @Override
137             public void run() {
138                 calculator.frame.setVisible(true);
139                 textField1.setText("");
140                 textField2.setText("");
141             }
142         });
143     } catch (InterruptedException e) {
144         e.printStackTrace();
145     } catch (InvocationTargetException e) {
146         e.printStackTrace();
147     }
148 }
149 }
150
151
```

```
1 import java.awt.*;
2 import java.lang.reflect.InvocationTargetException;
3 import java.util.Scanner;
4 import java.awt.event.ActionEvent;
5 import java.awt.event.ActionListener;
6 import java.util.concurrent.ExecutionException;
7 import javax.swing.*;
8
9 public class Calculator {
10     private JFrame frame;
11     private static JTextField textField1;
12     private static JTextField textField2;
13     private static JTextField textFieldResult;
14     private JButton buttonAdd, buttonSubtract, buttonMultiply, buttonDivide;
15     private JPanel panel;
16     private Scanner scanner;
17
18     public Calculator() {
19         frame = new JFrame("Calculator");
20
21         textField1 = new JTextField(1);
22         textField1.setBackground(Color.DARK_GRAY);
23         textField1.setFont(new Font("Arial", Font.BOLD, 20));
24         textField1.setForeground(Color.WHITE);
25
26         textField2 = new JTextField(1);
27         textField2.setBackground(Color.DARK_GRAY);
28         textField2.setFont(new Font("Arial", Font.BOLD, 20));
29         textField2.setForeground(Color.WHITE);
30
31         buttonAdd = new JButton("+");
32         buttonAdd.setBackground(Color.DARK_GRAY);
33         buttonAdd.setFont(new Font("Arial", Font.BOLD, 20));
34         buttonAdd.setForeground(Color.WHITE);
35
36         buttonSubtract = new JButton("-");
37         buttonSubtract.setBackground(Color.DARK_GRAY);
38         buttonSubtract.setFont(new Font("Arial", Font.BOLD, 20));
39         buttonSubtract.setForeground(Color.WHITE);
40
41         buttonMultiply = new JButton("*");
42         buttonMultiply.setBackground(Color.DARK_GRAY);
43         buttonMultiply.setFont(new Font("Arial", Font.BOLD, 20));
44         buttonMultiply.setForeground(Color.WHITE);
45
46         buttonDivide = new JButton("/");
47         buttonDivide.setBackground(Color.DARK_GRAY);
48         buttonDivide.setFont(new Font("Arial", Font.BOLD, 20));
49         buttonDivide.setForeground(Color.WHITE);
50
51         textFieldResult = new JTextField("Nakon odabira operatora ovdje će se pojaviti
rezultat.", 2);
52         textFieldResult.setBackground(Color.DARK_GRAY);
53         textFieldResult.setFont(new Font("Arial", Font.ITALIC, 12));
54         textFieldResult.setForeground(Color.WHITE);
55         textFieldResult.setEditable(false); // make the result field non-editable
56
57         panel = new JPanel();
58         panel.setLayout(new GridLayout(4, 2));
59         panel.add(textField1);
```

```

60     panel.add(textField2);
61     panel.add(buttonAdd);
62     panel.add(buttonSubtract);
63     panel.add(buttonMultiply);
64     panel.add(buttonDivide);
65     panel.add(textFieldResult);
66     frame.add(panel, BorderLayout.CENTER);
67
68     frame.setSize(640, 480);
69     frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
70
71     scanner = new Scanner(System.in);
72
73     buttonAdd.addActionListener(new ActionListener() {
74         public void actionPerformed(ActionEvent event) {
75             // get the numbers from the text fields
76             double number1 = Double.parseDouble(textField1.getText());
77             double number2 = Double.parseDouble(textField2.getText());
78
79             // perform the addition in a separate thread using a SwingWorker
80             SwingWorker<Double, Void> worker = new SwingWorker<Double, Void>() {
81                 @Override
82                 protected Double doInBackground() throws Exception {
83                     return number1 + number2;
84                 }
85
86                 @Override
87                 protected void done() {
88                     try {
89                         // get the result of the calculation and update the result
90                         // in the event dispatch thread using the invokeLater() method
91                         final double result = get();
92                         SwingUtilities.invokeLater(new Runnable() {
93                             public void run() {
94                                 textFieldResult.setText(String.valueOf(result));
95                             }
96                         });
97                     } catch (InterruptedException e) {
98                         e.printStackTrace();
99                     } catch (ExecutionException e) {
100                         throw new RuntimeException(e);
101                     }
102                 }
103             };
104             worker.execute();
105         }
106     });
107
108     buttonSubtract.addActionListener(new ActionListener() {
109         public void actionPerformed(ActionEvent event) {
110             // get the numbers from the text fields
111             double number1 = Double.parseDouble(textField1.getText());
112             double number2 = Double.parseDouble(textField2.getText());
113
114             // perform the subtraction in a separate thread using a SwingWorker
115             SwingWorker<Double, Void> worker = new SwingWorker<Double, Void>() {
116                 @Override
117                 protected Double doInBackground() throws Exception {
118                     return number1 - number2;

```

```

119         }
120
121         @Override
122         protected void done() {
123             try {
124                 // get the result of the calculation and update the result
125                 // in the event dispatch thread using the invokeLater() method
126                 final double result = get();
127                 SwingUtilities.invokeLater(new Runnable() {
128                     public void run() {
129                         textFieldResult.setText(String.valueOf(result));
130                     }
131                 });
132             } catch (InterruptedException e) {
133                 e.printStackTrace();
134             } catch (ExecutionException e) {
135                 throw new RuntimeException(e);
136             }
137         }
138     };
139     worker.execute();
140 }
141 });
142
143 buttonMultiply.addActionListener(new ActionListener() {
144     public void actionPerformed(ActionEvent event) {
145         // get the numbers from the text fields
146         double number1 = Double.parseDouble(textField1.getText());
147         double number2 = Double.parseDouble(textField2.getText());
148
149         // perform the multiplication in a separate thread using a SwingWorker
150         SwingWorker<Double, Void> worker = new SwingWorker<Double, Void>() {
151             @Override
152             protected Double doInBackground() throws Exception {
153                 return number1 * number2;
154             }
155
156             @Override
157             protected void done() {
158                 try {
159                     // get the result of the calculation and update the result
160                     // in the event dispatch thread using the invokeLater() method
161                     final double result = get();
162                     SwingUtilities.invokeLater(new Runnable() {
163                         public void run() {
164                             textFieldResult.setText(String.valueOf(result));
165                         }
166                     });
167                 } catch (InterruptedException e) {
168                     e.printStackTrace();
169                 } catch (ExecutionException e) {
170                     throw new RuntimeException(e);
171                 }
172             }
173         };
174         worker.execute();
175     }
176 });

```

```

177
178     buttonDivide.addActionListener(new ActionListener() {
179         public void actionPerformed(ActionEvent event) {
180             // get the numbers from the text fields
181             double number1 = Double.parseDouble(textField1.getText());
182             double number2 = Double.parseDouble(textField2.getText());
183
184             // perform the division in a separate thread using a SwingWorker
185             SwingWorker<Double, Void> worker = new SwingWorker<Double, Void>() {
186                 @Override
187                 protected Double doInBackground() throws Exception {
188                     return number1 / number2;
189                 }
190
191                 @Override
192                 protected void done() {
193                     try {
194                         // get the result of the calculation and update the result
195                         // in the event dispatch thread using the invokeLater() method
196                         final double result = get();
197                         SwingUtilities.invokeLater(new Runnable() {
198                             public void run() {
199                                 textFieldResult.setText(String.valueOf(result));
200                             }
201                         });
202                     } catch (InterruptedException e) {
203                         e.printStackTrace();
204                     } catch (ExecutionException e) {
205                         throw new RuntimeException(e);
206                     }
207                 }
208             };
209             worker.execute();
210         }
211     });
212 }
213
214 public static void main(String[] args) {
215     try {
216         // Create a new Calculator object
217         Calculator calculator = new Calculator();
218         // Set the frame to be visible
219         calculator.frame.setVisible(true);
220         SwingUtilities.invokeAndWait(new Runnable() {
221             @Override
222             public void run() {
223                 calculator.frame.setVisible(true);
224                 textField1.setText("");
225                 textField2.setText("");
226             }
227         });
228     } catch (InterruptedException e) {
229         e.printStackTrace();
230     } catch (InvocationTargetException e) {
231         e.printStackTrace();
232     }
233 }
234 }
235

```