```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
public class Calculator extends JFrame implements ActionListener {
private JTextField display;
private JPanel panel;
private String[] buttonLabels = {
"7", "8", "9", "/",
"4", "5", "6", "*",
"1", "2", "3", "-",
"0", ".", "=", "+",
"C", "←"
};
private JButton[] buttons = new JButton[buttonLabels.length];
private String currentOperation = "";
private double currentResult = 0;
private boolean newOperation = true;
public Calculator() {
display = new JTextField();
display.setEditable(false);
display.setFont(new Font("Arial", Font.PLAIN, 24));
display.setHorizontalAlignment(SwingConstants.RIGHT);
display.setBackground(Color.WHITE);
```

```
panel = new JPanel();
panel.setLayout(new GridLayout(5, 4, 10, 10));
for (int i = 0; i < buttonLabels.length; i++) {
buttons[i] = new JButton(buttonLabels[i]);
buttons[i].setFont(new Font("Arial", Font.PLAIN, 20));
buttons[i].addActionListener(this);
panel.add(buttons[i]);
}
setLayout(new BorderLayout());
add(display, BorderLayout.NORTH);
add(panel, BorderLayout.CENTER);
setTitle("Calculator");
setSize(400, 700);
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
setVisible(true);
}
@Override
public void actionPerformed(ActionEvent e) {
String command = e.getActionCommand();
try {
switch (command) {
case "C":
display.setText("");
currentOperation = "";
```

```
currentResult = 0;
newOperation = true;
break;
case "←":
if (display.getText().length() > 0) {
display.setText(display.getText().substring(0, display.getText().length() - 1));
}
break;
case "=":
computeResult();
currentOperation = "";
newOperation = true;
break;
case "/":
case "*":
case "-":
case "+":
if (!newOperation) {
computeResult();
}
currentOperation = command;
currentResult = Double.parseDouble(display.getText());
display.setText("");
newOperation = true;
```

```
break;
default:
if (newOperation) {
display.setText("");
newOperation = false;
}
display.setText(display.getText() + command);
}
} catch (NumberFormatException ex) {
display.setText("Error");
newOperation = true;
} catch (ArithmeticException ex) {
display.setText("Math Error");
newOperation = true;
} catch (Exception ex) {
display.setText("Error");
newOperation = true;
}
}
private void computeResult() {
double operand = Double.parseDouble(display.getText());
switch (currentOperation) {
case "/":
if (operand == 0) {
```

```
throw new ArithmeticException("Cannot divide by zero");
}
currentResult /= operand;
break;
case "*":
currentResult *= operand;
break;
case "-":
currentResult -= operand;
break;
case "+":
currentResult += operand;
break;
default:
currentResult = operand;
break;
}
display.setText(String.valueOf(currentResult));
}
public static void main(String[] args) {
SwingUtilities.invokeLater(Calculator::new);
}
}
```