

Object Oriented Programming Project Presentation

A Simple Calculator Application
A Java Swing Project



Team Member

Sushmoy Nandi CS-2203070

Ayan Saha CS-2203072

Mohsina Akter CS-2203061

Md Mydul Islam CS-2203062



Introduction

Project approach study is the natural way to present technology in an attractive manner. Initially, young students have limited knowledge and experience in design or construction of a product. We develop a calculator, this calculator is helpful to do simple arithmetic operations. It is easy to develop and easy to use. A calculator is an electronic device which is used to solve mathematics problems which are being faced by us in our daily life. Most of the calculators perform addition, subtraction, multiplication and division. Some also do square root moreover complex calculator can help us to draw functional graph. There many different ways to solve mathematical problems using the calculator.



Project Motivation and Objectives

- To implement basic mathematical operations.
- To create a user-friendly interface.
- To understand event handling in Java Swing.
- Simple yet practical application.
- Good for learning basic GUI and event handling.



Project Goals

- Learn Java Swing.
- Implement basic arithmetic operations.
- Develop a clean and functional user interface.



Demonstration

Process

The screenshot shows the IntelliJ IDEA IDE with a project named "Calculator". The main editor displays the source code for `Calculator.java`. The code includes package declarations, imports, and a class definition for `Calculator` extending `JFrame`. A small calculator window is visible in the foreground, showing the result 9.0. The bottom status bar indicates the current file path and encoding settings.

```
package in.calculator;

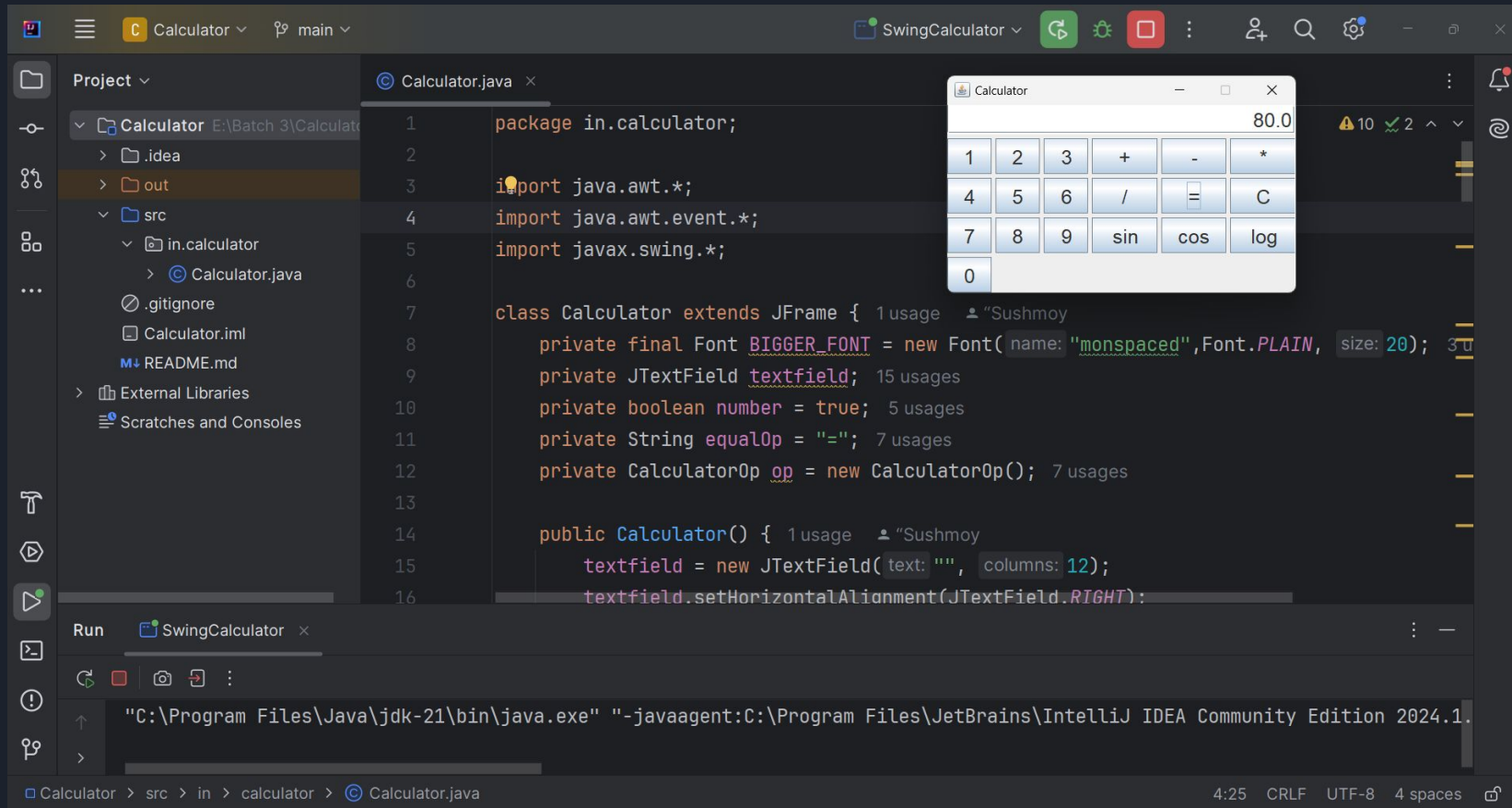
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

class Calculator extends JFrame {
    private final Font BIGGER_FONT = new Font("monospaced", Font.PLAIN, 20);
    private JTextField textfield;
    private boolean number = true;
    private String equalOp = "=";
    private CalculatorOp op = new CalculatorOp();

    public Calculator() {
        textfield = new JTextField("", 12);
        textfield.setHorizontalAlignment(JTextField.RIGHT);
    }
}
```

Run: `C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.1.`

Here, we calculate $5+4=9$ which can perform in addition function.



The screenshot displays an IDE with a Java project named 'Calculator'. The code in 'Calculator.java' defines a class that extends 'JFrame' and includes a text field and a grid of buttons for digits, operations, and trigonometric functions. A separate window titled 'Calculator' shows the result '80.0'.

```
package in.calculator;

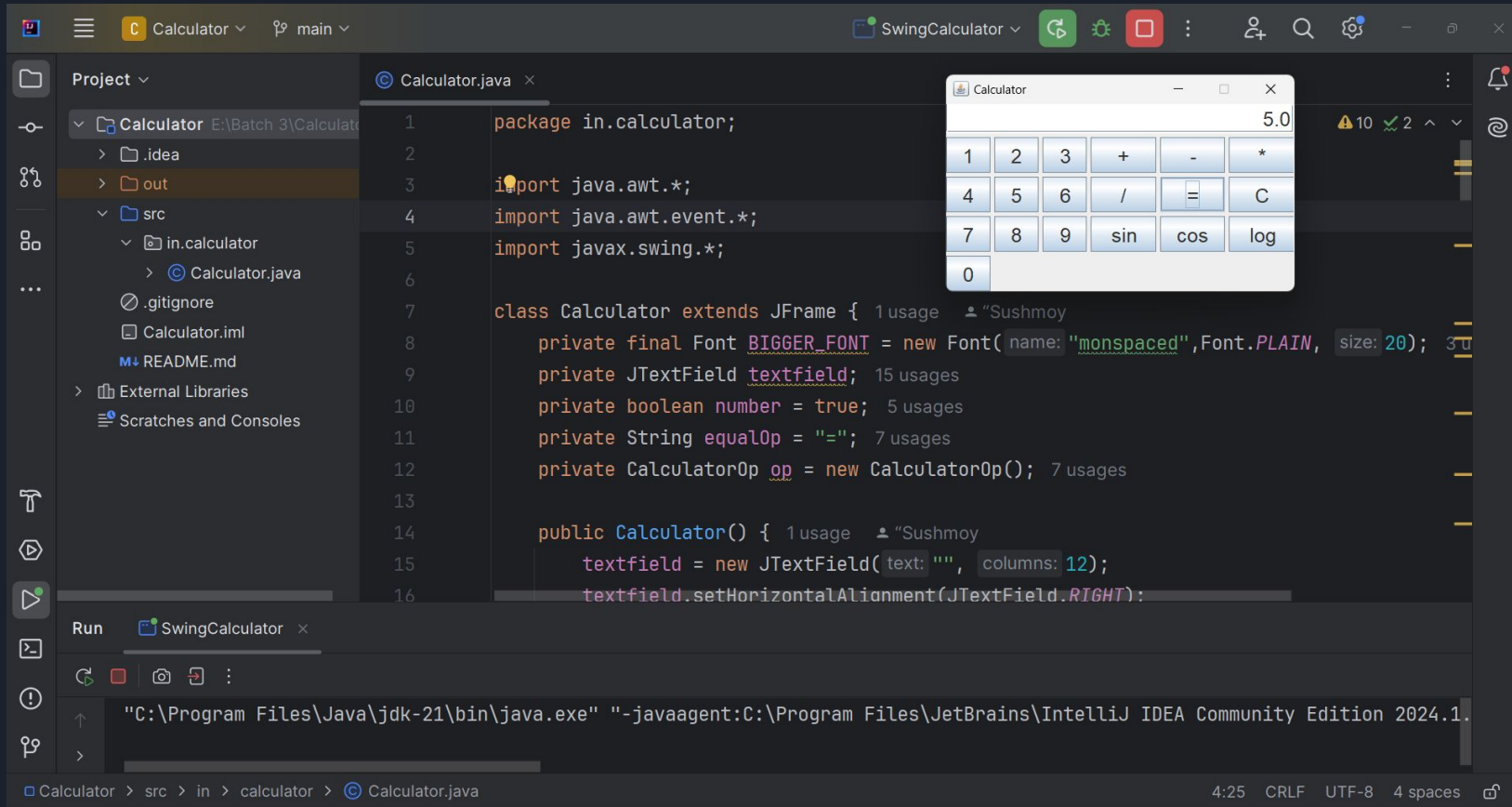
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

class Calculator extends JFrame {
    private final Font BIGGER_FONT = new Font("monospaced", Font.PLAIN, 20);
    private JTextField textfield;
    private boolean number = true;
    private String equalOp = "=";
    private CalculatorOp op = new CalculatorOp();

    public Calculator() {
        textfield = new JTextField("", 12);
        textfield.setHorizontalAlignment(JTextField.RIGHT);
```

The 'Calculator' window shows the result '80.0'.

Here, we calculate $10 \times 8 = 80$ which can perform in multiply function.



The screenshot displays an IDE environment with a Java project named "Calculator". The code in `Calculator.java` defines a `Calculator` class that extends `JFrame`. It includes a `textfield` for input and a `subtract` method that calculates the difference between two numbers. The IDE's Run tab shows the command to execute the application using `java.exe`. A separate window titled "Calculator" shows the graphical user interface with a numeric keypad and a display showing the result "5.0".

```
package in.calculator;

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

class Calculator extends JFrame {
    private final Font BIGGER_FONT = new Font("monospaced", Font.PLAIN, 20);
    private JTextField textfield;
    private boolean number = true;
    private String equalOp = "=";
    private CalculatorOp op = new CalculatorOp();

    public Calculator() {
        textfield = new JTextField("", 12);
        textfield.setHorizontalAlignment(JTextField.RIGHT);
    }

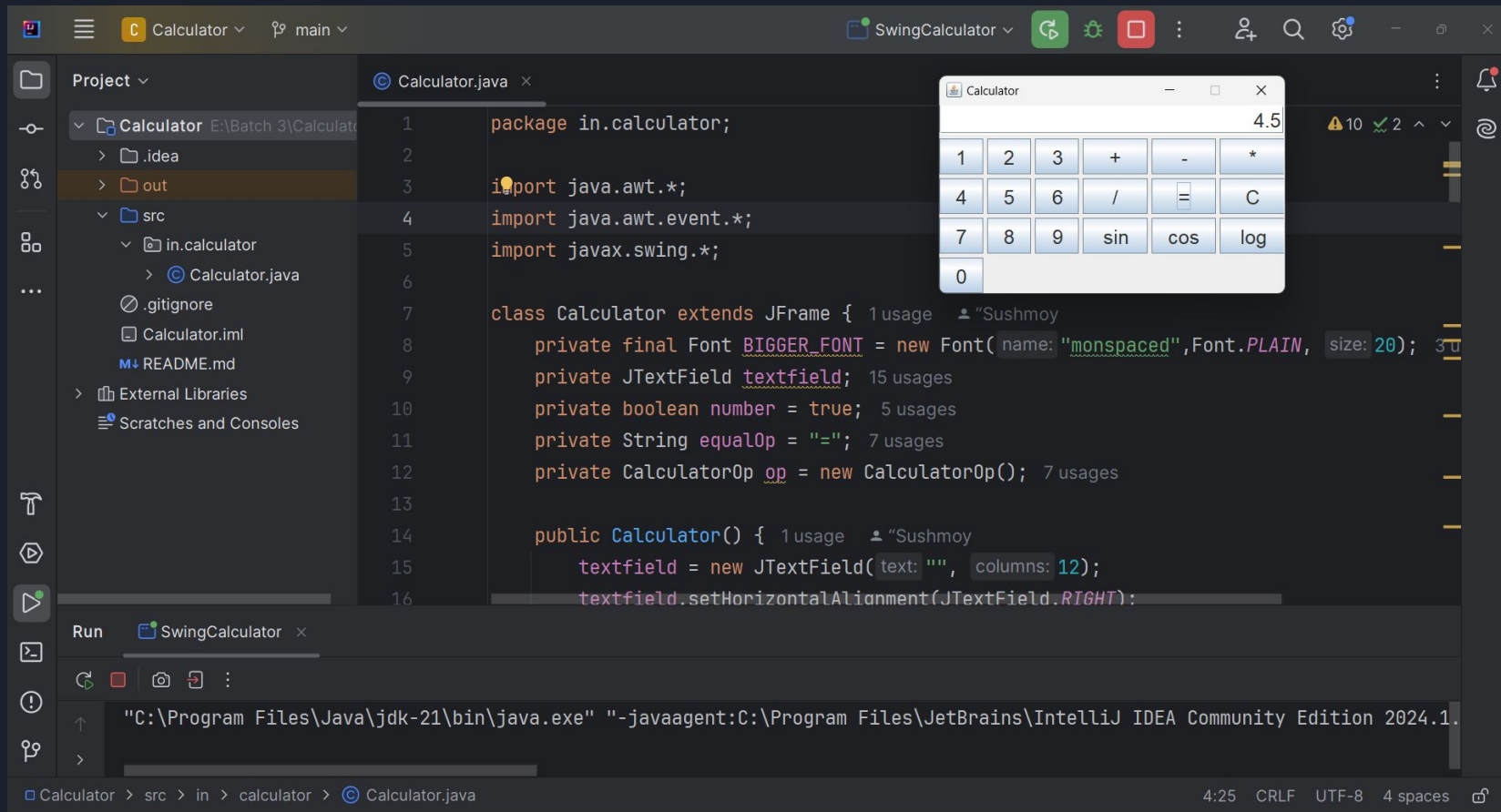
    private void subtract() {
        try {
            double num1 = Double.parseDouble(textfield.getText());
            double num2 = Double.parseDouble(textfield.getText());
            double result = num1 - num2;
            textfield.setText(String.format("%.1f", result));
        } catch (NumberFormatException e) {
            textfield.setText("");
        }
    }
}
```

Run `SwingCalculator`

`"C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.1."`

Calculator > src > in > calculator > Calculator.java

Here, we calculate $10-5=5$ which can perform in subtraction function.



The screenshot displays the IntelliJ IDEA IDE with a Java Swing calculator application. The code in `Calculator.java` is as follows:

```
1 package in.calculator;
2
3 import java.awt.*;
4 import java.awt.event.*;
5 import javax.swing.*;
6
7 class Calculator extends JFrame { 1 usage  Sushmoy
8     private final Font BIGGER_FONT = new Font( name: "monospaced", Font.PLAIN, size: 20); 3 U
9     private JTextField textfield; 15 usages
10    private boolean number = true; 5 usages
11    private String equalOp = "="; 7 usages
12    private CalculatorOp op = new CalculatorOp(); 7 usages
13
14    public Calculator() { 1 usage  Sushmoy
15        textfield = new JTextField( text: "", columns: 12);
16        textfield.setHorizontalAlignment(JTextField.RIGHT);
```

A small calculator window titled "Calculator" is open in the foreground, showing the calculation $9/2=4.5$. The window includes a numeric keypad, arithmetic operators (+, -, *, /), a decimal point, and trigonometric functions (sin, cos, log). The result "4.5" is displayed in the top right corner.

The IDE interface shows the project structure on the left, the code editor in the center, and the Run configuration and output at the bottom. The Run configuration is set to "SwingCalculator" and the output shows the command to run the application.

Here, we calculate $9/2=4.5$ which can perform in division function.



User Interface Design

GUI Components:

- JTextField for displaying input and results.
- JButton for numerical and operation inputs.
- JPanel for organizing components.

Layout:

- GridLayout for buttons.
- BorderLayout for the main panel.

Content:

- **Components Used:**
 - JTextField
 - JButton
 - JPanel
- **Layout Design:**
 - GridLayout for buttons.
 - BorderLayout for the overall layout.



Technical Implementation

Event Handling:

- ActionListeners for handling button clicks.

Class Structure:

- Calculator: Main GUI class.
- OperatorListener: Handles operator buttons.
- NumberListener: Handles number buttons.
- CalculatorOp: Performs arithmetic operations.

Code Walkthrough

Main Class:

- SwingCalculator: Initializes the calculator.

Constructor:

- Calculator: Sets up GUI components.

Action Listeners:

- OperatorListener and NumberListener: Handle user interactions.

Operations Class:

- CalculatorOp: Performs arithmetic calculations.

Content:

- **Main Class Code Snippet**
 - `public class SwingCalculator { ... }`
- **Constructor Code Snippet**
 - `public Calculator() { ... }`
- **Action Listener Code Snippets**
 - `class OperatorListener implements ActionListener { ... }`
 - `class NumberListener implements ActionListener { ... }`
- **Operations Class Code Snippet**
 - `public class CalculatorOp { ... }`



Conclusion

Summary of Achievements

- Functional calculator with a user-friendly interface.

Key Learnings

- Java Swing, event handling, GUI design.

Future Plans

- Potential features include more advanced functions, history of calculations, better error handling.



[GitHub Repository of our Project](#)