Shri Vile Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC Accredited with "A" Grade (CGPA : 3.18)

Department of Computer Science and Engineering (Data Science)

NAME: **Bhuvi Ghosh** BA **D22** SAP ID: **60009210191**

COURSE CODE: **DJ19DSL504** DATE: **31 / 10 / 2023**

COURSE NAME: **Java and Scala Laboratory** CLASS **T.Y.B.Tech**

Experiment No 6

Aim: - Write a GUI programming in JAVA using Swing components, Containers, JLabel, JButton, JCheckBox, JRadio Buttons, JTextField etc

Theory: -

Java Swing

Java Swing is a GUI Framework that contains a set of classes to provide more powerful and flexible GUI components than AWT. Swing provides the look and feel of modern Java GUI. Swing library is an official Java GUI tool kit released by Sun Microsystems. It is used to create graphical user interface with Java.Swing classes are defined in javax. swing package and its sub-packages.

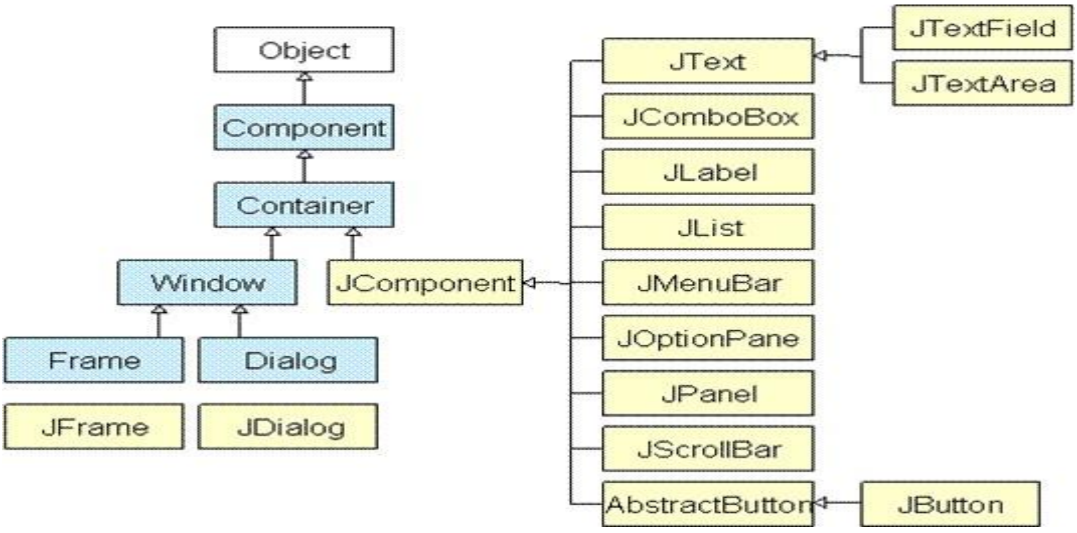
Swing and JFC

JFC is an abbreviation for Java Foundation classes which encompass a group of features for building Graphical User Interfaces(GUI) and adding rich graphical functionalities and interactivity to Java applications. Java Swing is a part of Java Foundation Classes (JFC).

Features of JFC

- Swing GUI components.
- Look and Feel support.
- Java 2D.

AWT and Swing Hierarchy



Introduction to Swing Classes

JPanel: JPanel is Swing's version of AWT class Panel and uses the same default layout, FlowLayout. JPanel is descended directly from JComponent.

JFrame: JFrame is Swing's version of Frame and is descended directly from Frame class. The component which is added to the Frame, is referred as its Content.

JWindow: This is Swing's version of Window and has descended directly from
Window class. Like Window it uses BorderLayout by default.

JLabel: JLabel has descended from JComponent, and is used to create text labels.

JButton: JButton class provides the functioning of push button. JButton allows an icon, string or both associated with a button.

JTextField: JTextField allow editing of a single line of text.

Creating a JFrame

There are two ways to create a JFrame Window.

By instantiating JFrame class.

```
import javax.swing.*; //importing swing package
import javax.swing.*; //importing swing package
import java.awt.*;           //importing awt package
public class First
{
        JFrame   jf;   public
        First() {
        jf = new JFrame("MyWindow");          //Creating a JFrame with name MyWindow
        JButton btn = new JButton("Say Hello");//Creating a Button named Say Hello
        jf.add(btn);          //adding button to frame
        jf.setLayout(new    FlowLayout());    //setting    layout    using    FlowLayout    object
        jf.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE); //setting close operation.
                jf.setSize(400, 400);      //setting      size      jf.setVisible(true);
                    //setting frame visibility
        }
        public static void main(String[] args)
        { new First();
        }
}
```

By extending JFrame class.

```
import javax.swing.*; //importing swing package
import java.awt.*; //importing awt package public
class Second extends JFrame
{
        public Second()
        {
                setTitle("MyWindow"); //setting title of frame as MyWindow
                JLabel lb = new JLabel("Welcome to My Second Window");//Creating a label named
Welcome to My Second Window
                add(lb);      //adding label to frame. setLayout(new FlowLayout()); //setting layout
                using FlowLayout object. setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
                //setting close operation.
                setSize(400, 400); //setting size setVisible(true); //setting
                frame visibility
        }

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

Points To Remember

Import the javax.swing and java.awt package to use the classes and methods of Swing.

While creating a frame (either by instantiating or extending Frame class), following two attributes are must for visibility of the frame:

setSize(int width, int height); setVisible(true);

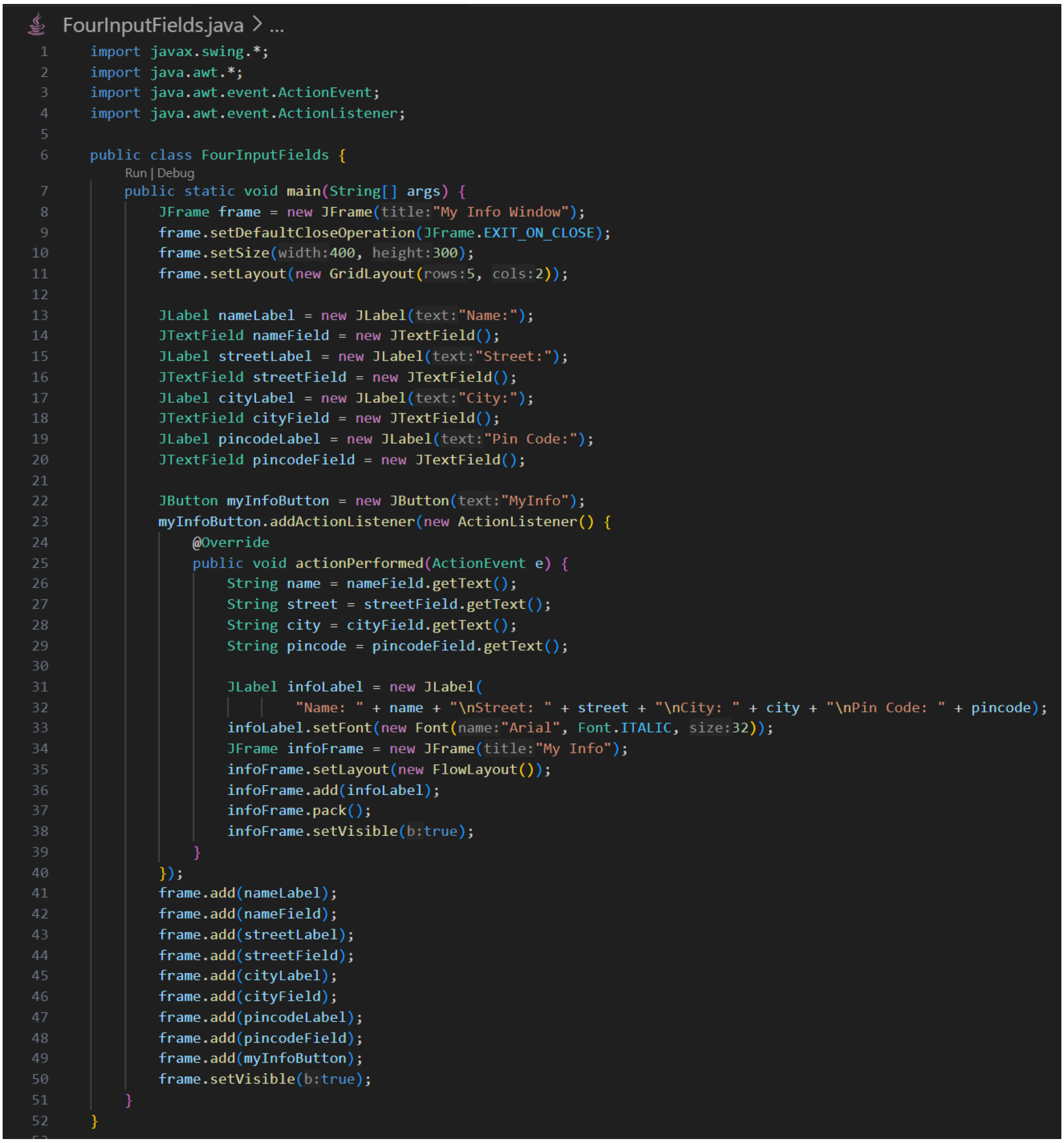
Copy

When you create objects of other components like Buttons, TextFields, etc. Then you need to add it to the frame by using the method - add(Component's Object);

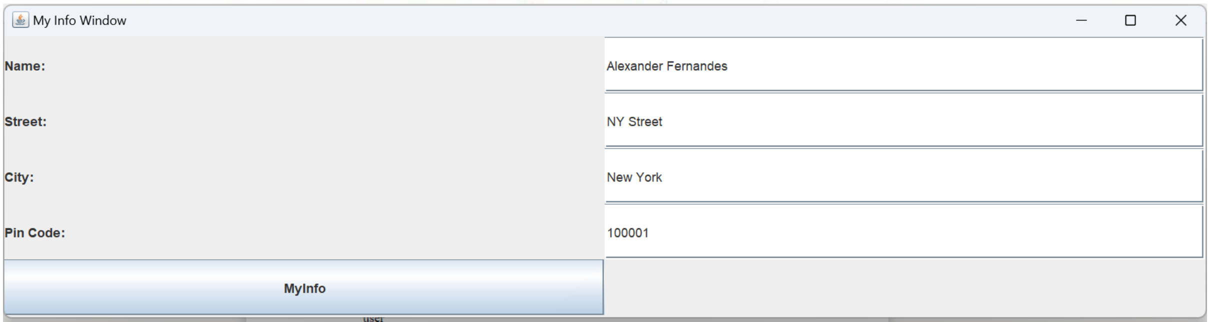
You can add the following method also for resizing the frame - setResizable(true);

Lab Assignments to complete in this session

i. Write a program to create a window with four text fields for the name, street, city and pin code with suitable labels. Also windows contain a button MyInfo. When the user types the name, his street, city and pincode and then clicks the button, the types details must appear in Arial Font with Size 32, Italics.



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

public class FourInputFields {
    public static void main(String[] args) {
        JFrame frame = new JFrame(title:"My Info Window");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(width:400, height:300);
        frame.setLayout(new GridLayout(rows:5, cols:2));

        JLabel nameLabel = new JLabel(text:"Name:");
        JTextField nameField = new JTextField();
        JLabel streetLabel = new JLabel(text:"Street:");
        JTextField streetField = new JTextField();
        JLabel cityLabel = new JLabel(text:"City:");
        JTextField cityField = new JTextField();
        JLabel pincodeLabel = new JLabel(text:"Pin Code:");
        JTextField pincodeField = new JTextField();

        JButton myInfoButton = new JButton(text:"MyInfo");
        myInfoButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                String name = nameField.getText();
                String street = streetField.getText();
                String city = cityField.getText();
                String pincode = pincodeField.getText();

                JLabel infoLabel = new JLabel(
                        "Name: " + name + "\nStreet: " + street + "\nCity: " + city + "\nPin Code: " + pincode);
                infoLabel.setFont(new Font(name:"Arial", Font.ITALIC, size:32));
                JFrame infoFrame = new JFrame(title:"My Info");
                infoFrame.setLayout(new FlowLayout());
                infoFrame.add(infoLabel);
                infoFrame.pack();
                infoFrame.setVisible(b:true);
            }
        });
        frame.add(nameLabel);
        frame.add(nameField);
        frame.add(streetLabel);
        frame.add(streetField);
        frame.add(cityLabel);
        frame.add(cityField);
        frame.add(pincodeLabel);
        frame.add(pincodeField);
        frame.add(myInfoButton);
        frame.setVisible(b:true);
    }
}
```

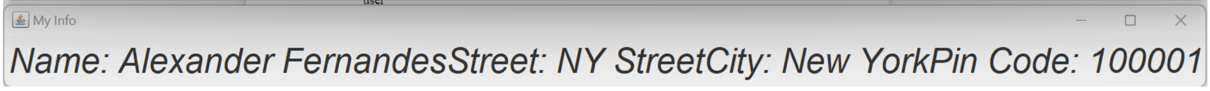


Name: Alexander Fernandes

Street: NY Street

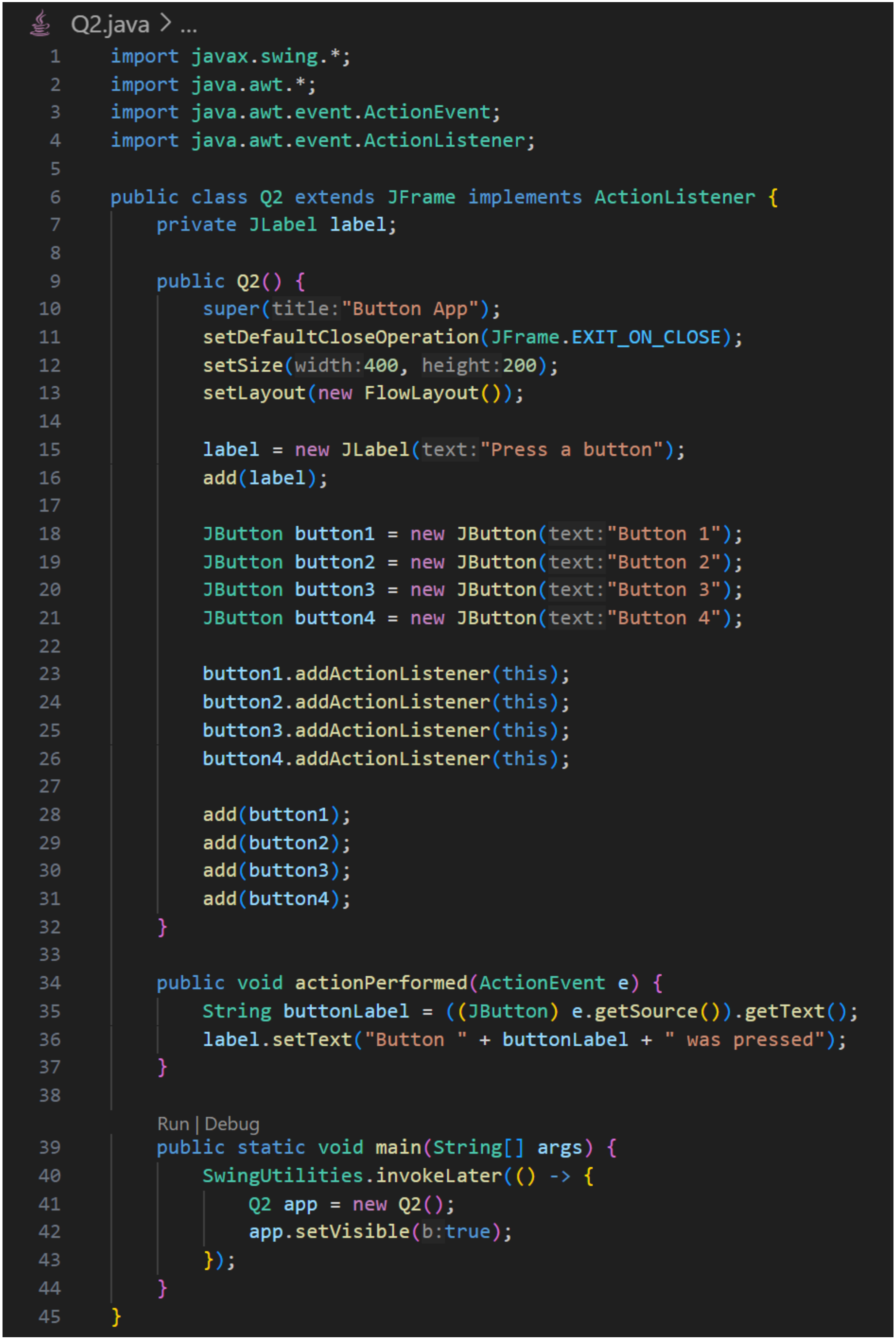
City: New York

Pin Code: 100001

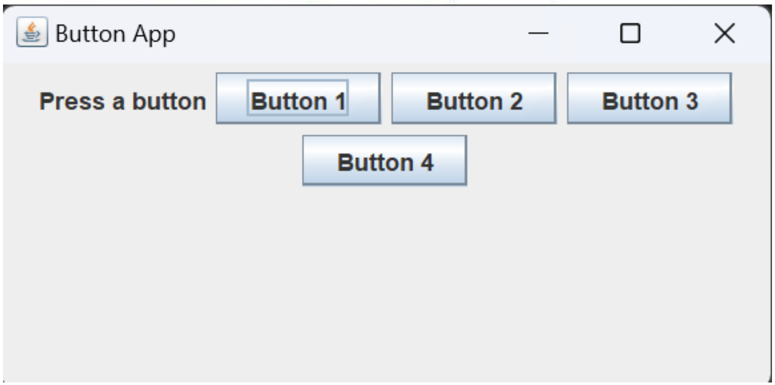


```
Name: Alexander FernandesStreet: NY StreetCity: New YorkPin Code: 100001
```

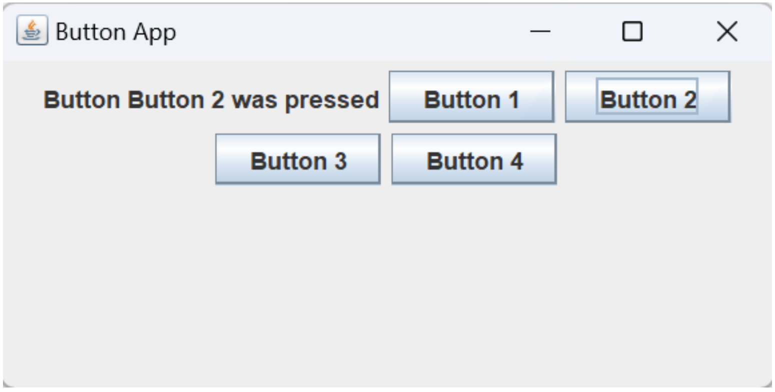
ii. WA applet with 4 swing buttons with suitable texts on them. When the user presses a button a message should appear in the label as to which button was pressed by the user



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

public class Q2 extends JFrame implements ActionListener {
    private JLabel label;

    public Q2() {
        super(title:"Button App");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setSize(width:400, height:200);
        setLayout(new FlowLayout());

        label = new JLabel(text:"Press a button");
        add(label);

        JButton button1 = new JButton(text:"Button 1");
        JButton button2 = new JButton(text:"Button 2");
        JButton button3 = new JButton(text:"Button 3");
        JButton button4 = new JButton(text:"Button 4");

        button1.addActionListener(this);
        button2.addActionListener(this);
        button3.addActionListener(this);
        button4.addActionListener(this);

        add(button1);
        add(button2);
        add(button3);
        add(button4);
    }

    public void actionPerformed(ActionEvent e) {
        String buttonLabel = ((JButton) e.getSource()).getText();
        label.setText("Button " + buttonLabel + " was pressed");
    }

    public static void main(String[] args) {
        SwingUtilities.invokeLater(() -> {
            Q2 app = new Q2();
            app.setVisible(b:true);
        });
    }
}
```

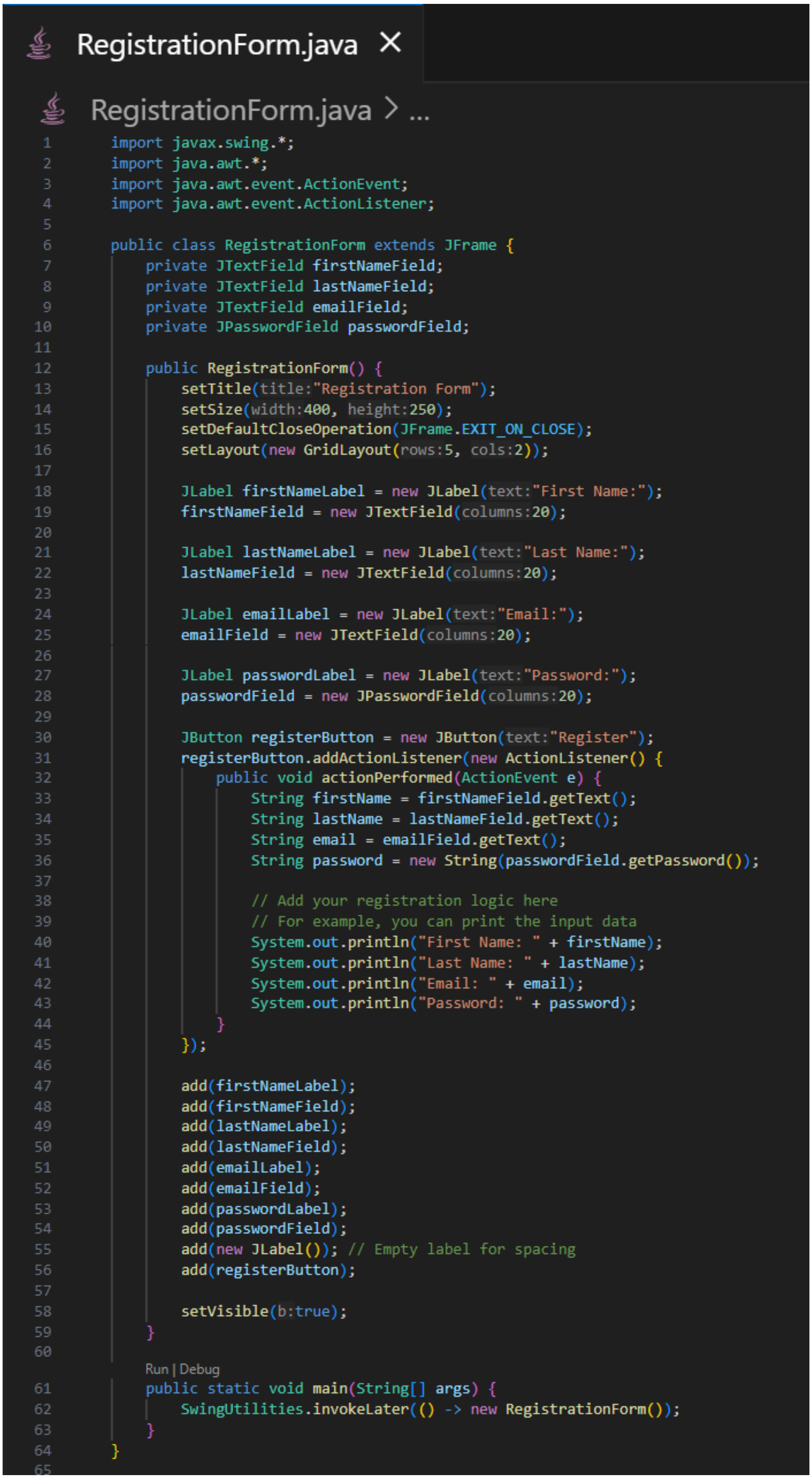


Press a button

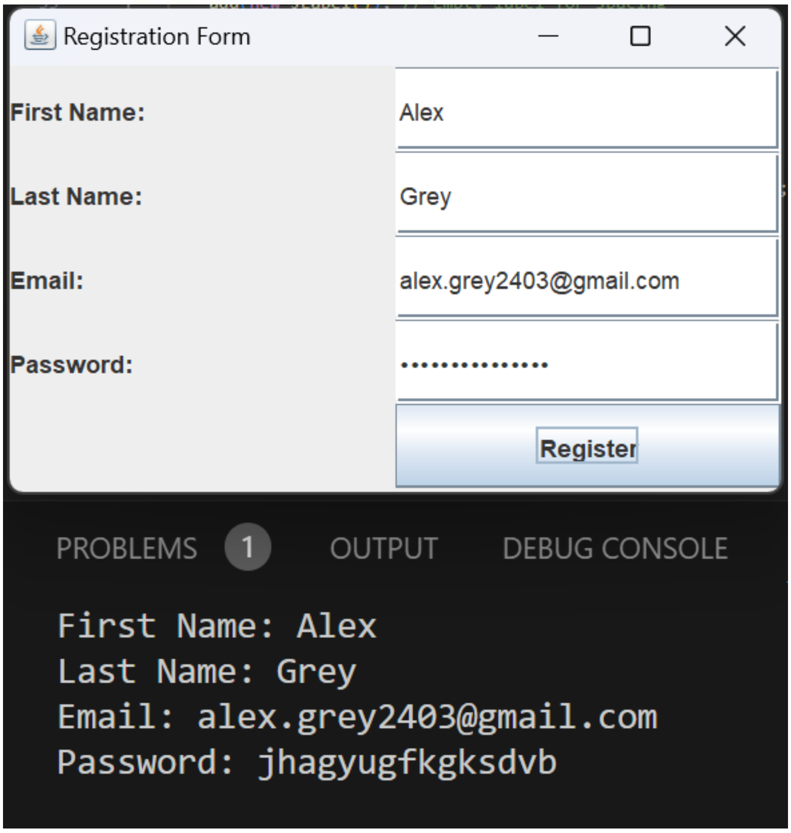


Button Button 2 was pressed

iii. Write java program to create a registration form using Swing



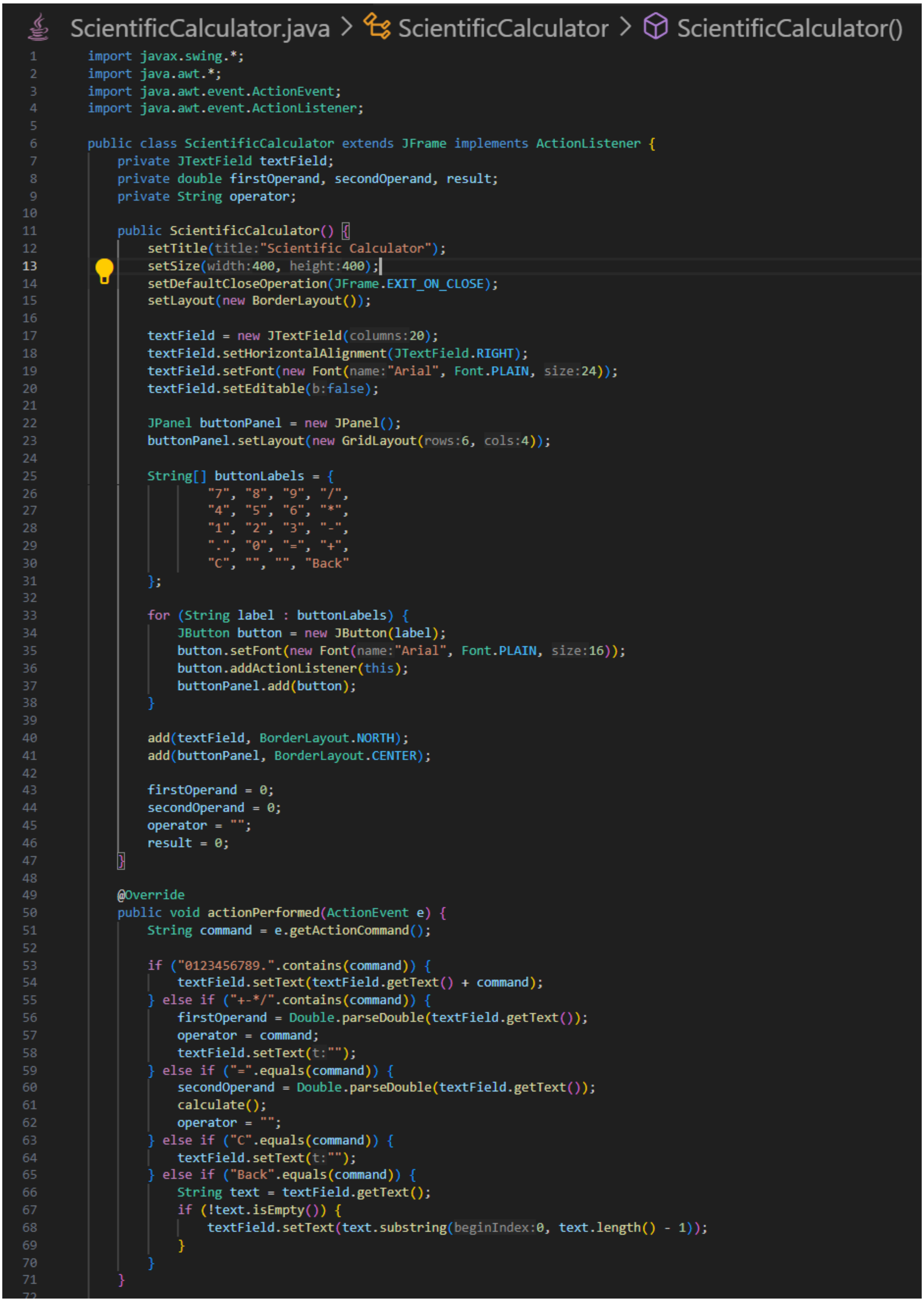
```java
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class RegistrationForm extends JFrame {
    private JTextField firstNameField;
    private JTextField lastNameField;
    private JTextField emailField;
    private JPasswordField passwordField;

    public RegistrationForm() {
        setTitle(title:"Registration Form");
        setSize(width:400, height:250);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLayout(new GridLayout(rows:5, cols:2));

        JLabel firstNameLabel = new JLabel(text:"First Name:");
        firstNameField = new JTextField(columns:20);

        JLabel lastNameLabel = new JLabel(text:"Last Name:");
        lastNameField = new JTextField(columns:20);

        JLabel emailLabel = new JLabel(text:"Email:");
        emailField = new JTextField(columns:20);

        JLabel passwordLabel = new JLabel(text:"Password:");
        passwordField = new JPasswordField(columns:20);

        JButton registerButton = new JButton(text:"Register");
        registerButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                String firstName = firstNameField.getText();
                String lastName = lastNameField.getText();
                String email = emailField.getText();
                String password = new String(passwordField.getPassword());

                // Add your registration logic here
                // For example, you can print the input data
                System.out.println("First Name: " + firstName);
                System.out.println("Last Name: " + lastName);
                System.out.println("Email: " + email);
                System.out.println("Password: " + password);
            }
        });

        add(firstNameLabel);
        add(firstNameField);
        add(lastNameLabel);
        add(lastNameField);
        add(emailLabel);
        add(emailField);
        add(passwordLabel);
        add(passwordField);
        add(new JLabel()); // Empty label for spacing
        add(registerButton);

        setVisible(b:true);
    }

    Run | Debug
    public static void main(String[] args) {
        SwingUtilities.invokeLater(() -> new RegistrationForm());
    }
}
```



First Name:	Alex
Last Name:	Grey
Email:	alex.grey2403@gmail.com
Password:	•••••••••••••••

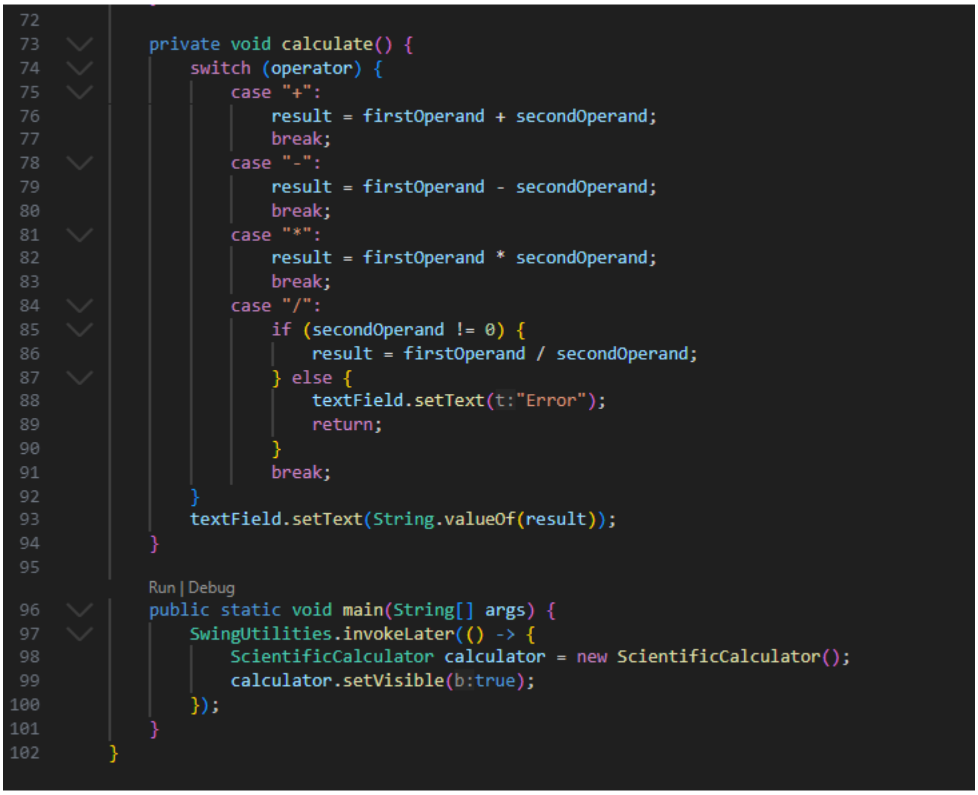
```
First Name: Alex
Last Name: Grey
Email: alex.grey2403@gmail.com
Password: jhagyugfkgksdvb
```

iv. Implement Scientific Calculator Using JAVA Swing

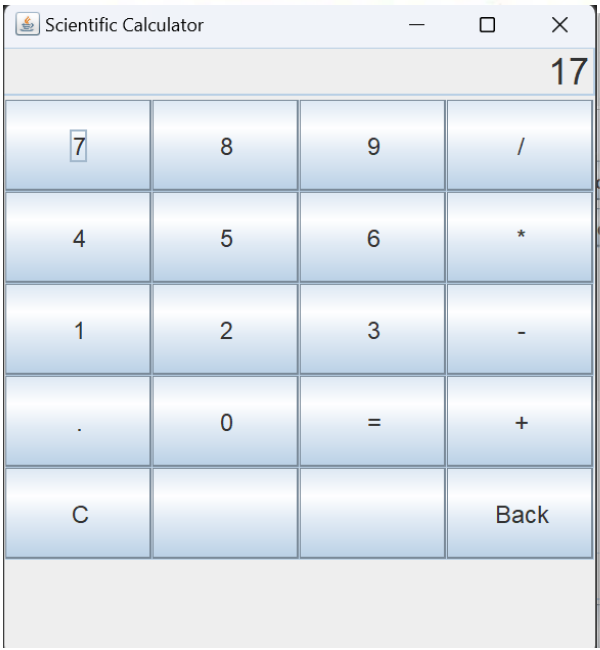


ScientificCalculator.java > ScientificCalculator > ScientificCalculator()

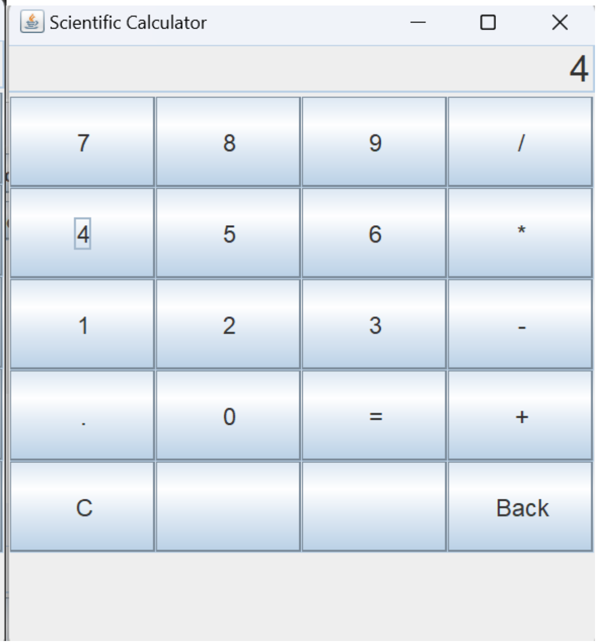
```java
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class ScientificCalculator extends JFrame implements ActionListener {
    private JTextField textField;
    private double firstOperand, secondOperand, result;
    private String operator;

    public ScientificCalculator() {
        setTitle(title:"Scientific Calculator");
        setSize(width:400, height:400);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLayout(new BorderLayout());

        textField = new JTextField(columns:20);
        textField.setHorizontalAlignment(JTextField.RIGHT);
        textField.setFont(new Font(name:"Arial", Font.PLAIN, size:24));
        textField.setEditable(b:false);

        JPanel buttonPanel = new JPanel();
        buttonPanel.setLayout(new GridLayout(rows:6, cols:4));

        String[] buttonLabels = {
                "7", "8", "9", "/",
                "4", "5", "6", "*",
                "1", "2", "3", "-",
                ".", "0", "=", "+",
                "C", "", "", "Back"
        };

        for (String label : buttonLabels) {
            JButton button = new JButton(label);
            button.setFont(new Font(name:"Arial", Font.PLAIN, size:16));
            button.addActionListener(this);
            buttonPanel.add(button);
        }

        add(textField, BorderLayout.NORTH);
        add(buttonPanel, BorderLayout.CENTER);

        firstOperand = 0;
        secondOperand = 0;
        operator = "";
        result = 0;
    }

    @Override
    public void actionPerformed(ActionEvent e) {
        String command = e.getActionCommand();

        if ("0123456789.".contains(command)) {
            textField.setText(textField.getText() + command);
        } else if ("+-*/".contains(command)) {
            firstOperand = Double.parseDouble(textField.getText());
            operator = command;
            textField.setText(t:"");
        } else if ("=".equals(command)) {
            secondOperand = Double.parseDouble(textField.getText());
            calculate();
            operator = "";
        } else if ("C".equals(command)) {
            textField.setText(t:"");
        } else if ("Back".equals(command)) {
            String text = textField.getText();
            if (!text.isEmpty()) {
                textField.setText(text.substring(beginIndex:0, text.length() - 1));
            }
        }
    }
```



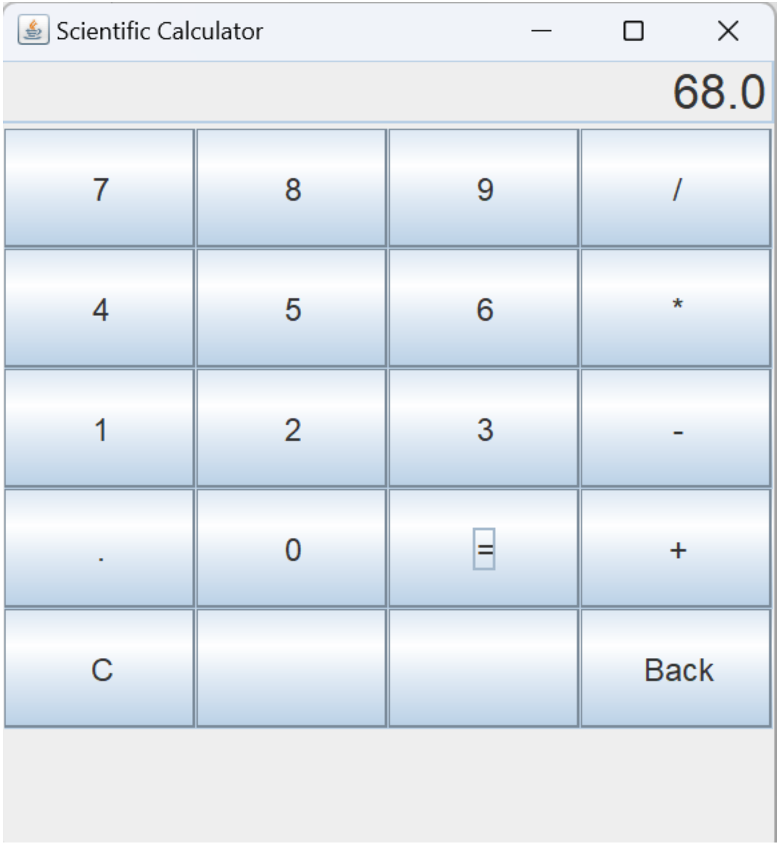
```
    private void calculate() {
        switch (operator) {
            case "+":
                result = firstOperand + secondOperand;
                break;
            case "-":
                result = firstOperand - secondOperand;
                break;
            case "*":
                result = firstOperand * secondOperand;
                break;
            case "/":
                if (secondOperand != 0) {
                    result = firstOperand / secondOperand;
                } else {
                    textField.setText(t:"Error");
                    return;
                }
                break;
        }
        textField.setText(String.valueOf(result));
    }

    public static void main(String[] args) {
        SwingUtilities.invokeLater(() -> {
            ScientificCalculator calculator = new ScientificCalculator();
            calculator.setVisible(b:true);
        });
    }
}
```


Scientific Calculator
7
8
9
/
4
5
6
*
1
2
3
-
.
0
=
+
C
Back

```
17
```


Scientific Calculator
7
8
9
/
4
5
6
*
1
2
3
-
.
0
=
+
C
Back

```
4
```


Scientific Calculator
7
8
9
/
4
5
6
*
1
2
3
-
.
0
=
+
C
Back

```
68.0
```