

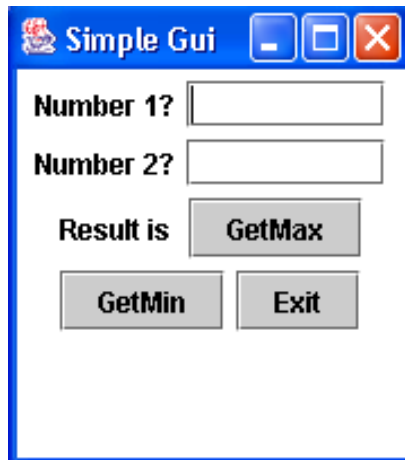
Lab Assignment 8

In this lab, we learn to create a simple gui project which involves

1. Creation of a JFrame with many components (ex: JButtons, JLabels and JTextFields) on it.
2. Event Handling: Usage of ActionListener interface having actionPerformed method.
3. Usage of debugger for removing the bugs found.

Instructions:

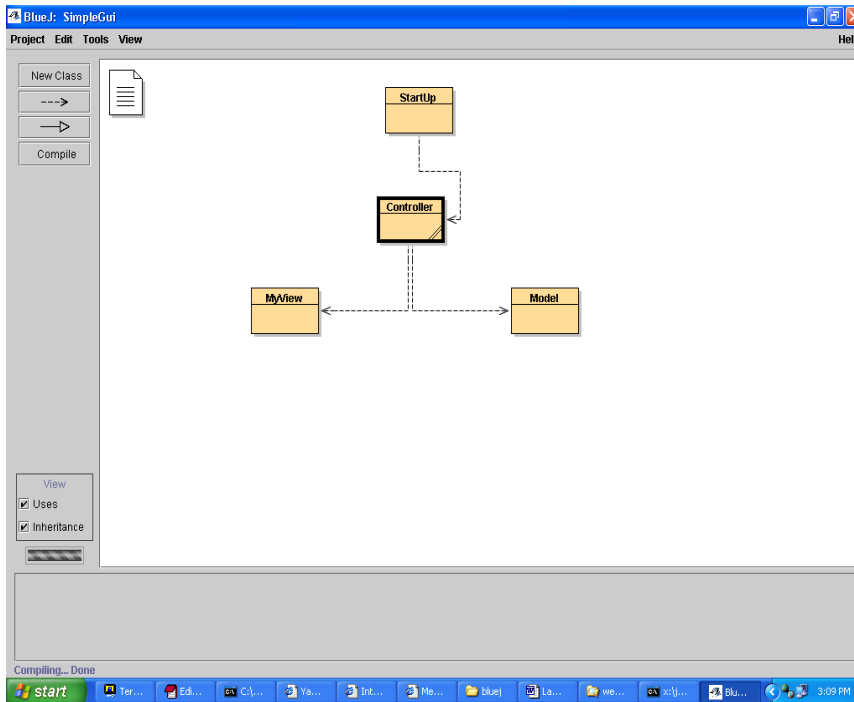
1. In this Lab assignment, you will write a program which displays a JFrame which looks something like this.



2. The JFrame contains two JLabels (**Number 1?** and **Number 2?**), three JButtons (**GetMax**, **GetMin** and **Exit**) and another JLabel(**Result is**) which displays the result.
3. Open BlueJ and create a new project named Lab8.

Create the following four classes as shown in Figure 1.

Note: Only the **StartUp** class will be created with the main method.



4. In the StartUp class, add the following code.

```
class StartUp {
    public static void main(String[] args) {
        new Controller(); //the Constructor of Controller gets called automatically
    }
}
```

5. Open MyView class and add the following code.

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

class MyView extends JFrame implements ActionListener {
    private JButton GetMax = new JButton("GetMax");
    private JButton GetMin = new JButton("GetMin");
    private JButton Exit = new JButton("Exit");
    private JLabel Result = new JLabel("Result is ");
    private JTextField textfield1 = new JTextField("", 8);
    private JTextField textfield2 = new JTextField("", 8);
    private JLabel Label1 = new JLabel("Number 1?");
    private JLabel Label2 = new JLabel("Number 2?");
    private String StrX;
    private String StrY;
    private String Command;
    private Container c;
    private Controller control;
```

```

public MyView(Controller cont) {           //constructor of the View
    control = cont;
    c = getContentPane();
    c.setLayout(new FlowLayout());
    c.setBackground(Color.white);
    // add ActionListener to each of the three JButtons created
    Exit.addActionListener(this);
    GetMax.addActionListener(this);
    GetMin.addActionListener(this);
    //Add the Components created on to the JFrame
    c.add(Label1);
    c.add(textfield1);
    c.add(Label2);
    c.add(textfield2);
    c.add(Result);
    c.add(GetMax);
    c.add(GetMin);
    c.add(Exit);

    setSize(180,200);
    setVisible(true);
    setTitle("Simple Gui ");
}

public void actionPerformed(ActionEvent ae) {
    //to identify the Component that generated this event
    JButton EventSource = (JButton) ae.getSource();
    // Extract the texts filled in both the TextFields into local variables
    StrX = textfield1.getText();
    StrY = textfield2.getText();
    //Remember control is the object reference of the Controller
    control.str1 = StrX;
    control.str2 = StrY;
    // See how an if-else condition is used
    if (EventSource == GetMax){
        control.setCommand("GETMAX");
        Result.setText("The Maximum value is " + control.getResult());
    }
    else if (EventSource == GetMin){
        control.setCommand("GETMIN");
        Result.setText("The Minimum value is " + control.getResult());
    }
    else if (EventSource == Exit){
        System.exit(0); //to exit out of the program
    }
}
}

```

6. In the Model class, add the following code

```
import java.lang.*;

class Model {
    int operand1,operand2;

    public Model() {
        }
        //Empty Constructor

    public int getMax(String x, String y) {
        operand1 = Integer.parseInt(x);
        operand2 = Integer.parseInt(y);

        if (operand1 > operand2){
            return operand2;
        }
        else if (operand1 < operand2) {
            return operand1;
        }
        else
            return 0;
    }

    public int getMin(String x, String y) {
        operand1 = Integer.parseInt(x);
        operand2 = Integer.parseInt(y);

        if (operand1 > operand2){
            return operand2;
        }
        else if (operand1 < operand2){
            return operand1;
        }
        else
            return 0;
    }
}
```

7. In the Controller, add the following code

```
class Controller {
    private int result;
    public String str1;
    public String str2;
    public String command;
    Model m = new Model();

    public Controller() {
        MyView v = new MyView(this);
    }
    public void setCommand(String c) {
        command = c;
        if (c == "GETMAX") {
            result = m.getMax(str1, str2);
        }
        else if (c == "GETMIN"){
            result = m.getMin(str1, str2);
        }
    }

    public int getResult() {
        return result;
    }
}
```

8. Compile all the classes.

9. Run the program.

10. Test the program by giving different values for Number 1 and Number 2 in the JTextFields.

- Do you find anything going wrong? If yes, **use the debugger** to remove the bug.
- Which class do you think has got the bug? Is it the Controller? or Is it the Model? Find the bug and report it to your instructor.

11. After debugging, show the instructor, the correct execution of the program and let him know where the bug was found and what it is.

12. Also, explain to the instructor if there is any communication between the Model and the View components.