



VIGNANA BHARATHI
Institute of Technology

Name ; lasya Priya

Roll No : 22P61A67J2

Subject : Java

Template for converting applet to frame

Step 1: Save a copy of the original program.

Step 2: Remove all references to the `java.awt` package.

Step 3: If your program is an applet, remove the `java.applet.*` import statement (if present) and any references to `java.applet.Applet`.

Step 4: Import the main Swing package.

Step 5: Be aware of thread-safety issues!

Step 6: Change each AWT component to its closest Swing equivalent.

Step 7: Change calls to the `add` and `setLayout` methods.

Step 8: Move painting code out of the `paint` and `update` methods.

Step 9: Use the compiler to find any other needed changes.

Step 10: Run the Swing program.

Step 11: Compare the Swing version to the AWT version, and make any improvements that Swing enables.

Step 12: Clean up!

CALCULATOR:

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

import java.awt.*;

import javax.swing.*;

public class Calculator extends JFrame implements
ActionListener

{

    JButton b10,b11,b12,b13,b14,b15;

    JButton b[]=new JButton[10];

    int i,r,n1,n2;

    JTextField res;

    char op;

    public Calculator()
    {
        super("Calulator");
        setLayout(new BorderLayout());
        JPanel p=new JPanel();
        p.setLayout(new GridLayout(4,4));
        for(int i=0;i<=9;i++)
        {
            b[i]=new JButton(i+"");
            p.add(b[i]);
```

```
        b[i].addActionListener(this);  
    }
```

```
    b10=new JButton("+");  
    p.add(b10);  
    b10.addActionListener(this);
```

```
    b11=new JButton("-");  
    p.add(b11);  
    b11.addActionListener(this);
```

```
    b12=new JButton("*");  
    p.add(b12);  
    b12.addActionListener(this);
```

```
    b13=new JButton("/");  
    p.add(b13);  
    b13.addActionListener(this);
```

```
    b14=new JButton("=");  
    p.add(b14);  
    b14.addActionListener(this);
```

```
    b15=new JButton("C");  
    p.add(b15);  
    b15.addActionListener(this);
```

```

        res=new JTextField(10);
        add(p, BorderLayout.CENTER);
        add(res, BorderLayout.NORTH);
        setVisible(true);
        setSize(200,200);
    }

    public void actionPerformed(ActionEvent ae)
    {
        JButton pb=(JButton)ae.getSource();
        if(pb==b15)
        {
            r=n1=n2=0;
            res.setText("");
        }
        else
        {
            if(pb==b14)
            {
                n2=Integer.parseInt(res.getText());
                eval();
                res.setText(""+r);
            }

            else
            {

```

```
boolean opf=false;
if(pb==b10)
    { op='+';
      opf=true;
    }
if(pb==b11)
    { op='-';opf=true;}
if(pb==b12)
    { op='*';opf=true;}
if(pb==b13)
    { op='/';opf=true;}

if(opf==false)
{
    for(i=0;i<10;i++)
    {
        if(pb==b[i])
        {
            String t=res.getText();
            t+=i;
            res.setText(t);
        }
    }
}
else
```

```

        {
            n1=Integer.parseInt(res.getText());
            res.setText("");
        }
    }
}

int eval()
{
    switch(op)
    {
        case '+': r=n1+n2; break;
        case '-': r=n1-n2; break;
        case '*': r=n1*n2; break;
        case '/': r=n1/n2; break;

    }
    return 0;
}

public static void main(String arg[])
{
    new Calculator();
}
}

```

FIBONUCCI SERIES

```
import java.awt.Color;  
import java.awt.GridLayout;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;
```

```
import javax.swing.JButton;  
import javax.swing.JFrame;  
import javax.swing.JLabel;  
import javax.swing.JPanel;  
import javax.swing.JTextField;  
import javax.swing.border.LineBorder;  
import javax.swing.border.TitledBorder;
```

```
public class FibonacciNumbers extends JFrame {  
    // components for calculating the Fibonacci of a user-entered  
    number  
    private final JPanel workerJPanel = new JPanel(new  
    GridLayout(2, 2, 5, 5));  
    private final JTextField numberJTextField = new  
    JTextField();  
    private final JButton goJButton = new JButton("Go");  
    private final JLabel fibonacciJLabel = new JLabel();  
    // components and variables for getting the next Fibonacci  
    number  
    private final JPanel eventThreadJPanel = new JPanel(new  
    GridLayout(2, 2, 5,
```



```

        5));

        private long n1 = 0; // initialize with first Fibonacci number
        private long n2 = 1; // initialize with second Fibonacci
number
        private int count = 1; // current Fibonacci number to display
        private final JLabel nJLabel = new JLabel("Fibonacci of 1:
");

        private final JLabel nFibonacciJLabel = new
JLabel(String.valueOf(n2));

        private final JButton nextNumberJButton = new
JButton("Next Number");

// constructor
public FibonacciNumbers() {
    super("Fibonacci Numbers");
    setLayout(new GridLayout(2, 1, 10, 10));
    // add GUI components to the SwingWorker panel
    workerJPanel.setBorder(new TitledBorder(new
LineBorder(Color.BLACK),
        "With SwingWorker"));
    workerJPanel.add(new JLabel("Get Fibonacci of:"));
    workerJPanel.add(numberJTextField);
    goJButton.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent event) {
            int n;
            try {

```



```

        // retrieve user's input as an integer
        n =
Integer.parseInt(numberJTextField.getText());
    } // end try
    catch (NumberFormatException ex) {
        // display an error message if the user
did not
        // enter an integer
        fibonacciJLabel.setText("Enter an
integer.");

        return;
    } // end catch
    // indicate that the calculation has
begun
    fibonacciJLabel.setText("Calculating...");
    // create a task to perform calculation in
background
    BackgroundCalculator task = new
BackgroundCalculator(n,
        fibonacciJLabel);
    task.execute(); // execute the task
    } // end method actionPerformed
} // end anonymous inner class
); // end call to addActionListener
workerJPanel.add(goJButton);
workerJPanel.add(fibonacciJLabel);

```

```

        // add GUI components to the event-dispatching thread
panel
        eventThreadJPanel.setBorder(new TitledBorder(
            new LineBorder(Color.BLACK), "Without
SwingWorker"));
        eventThreadJPanel.add(nJLabel);
        eventThreadJPanel.add(nFibonacciJLabel);
        nextNumberJButton.addActionListener(new
ActionListener() {
            public void actionPerformed(ActionEvent event) {
                // calculate the Fibonacci number after n2
                long temp = n1 + n2;
                n1 = n2;
                n2 = temp;
                System.out.println("temp = n1+n2 ~ " +
temp + " " + n1 + " "
                    + n2);
                ++count;
                // display the next Fibonacci number
                nJLabel.setText("Fibonacci of " + count + ":
");

                nFibonacciJLabel.setText(String.valueOf(n2));
            } // end method actionPerformed
        } // end anonymous inner class
    ); // end call to addActionListener
    eventThreadJPanel.add(nextNumberJButton);

```

```

        add(workerJPanel);
        add(eventThreadJPanel);
        setSize(275, 200);
        setVisible(true);
    } // end constructor

    // main method begins program execution

    public static void main(String[] args) {
        FibonacciNumbers application = new
FibonacciNumbers();

        application.setDefaultCloseOperation(EXIT_ON_CLOSE);
    } // end main
}

```

DIVISION

The user enters two numbers in the textfields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a NumberFormatException. If Num2 were Zero, the program would throw an ArithmeticException. Display the exception in a message dialog box.

```

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

public class Division extends JFrame implements ActionListener
{

```

```
Container c;  
JButton btn;  
JLabel lbl1, lbl2, lbl3;  
JTextField tf1, tf2, tf3;  
JPanel;
```

```
Division()  
{  
    super("Exception Handler");  
    c=getContentPane();  
    c.setBackground(Color.red);  
    btn=new JButton("DIVIDE");  
    btn.addActionListener(this);  
    tf1=new JTextField(30);  
    tf2=new JTextField(30);  
    tf3=new JTextField(30);  
    lbl1=new JLabel("NUM 1");  
    lbl2=new JLabel("NUM 2");  
    lbl3=new JLabel("RESULT");  
    p=new JPanel();  
    p.setLayout(new GridLayout(3,2));  
    p.add(lbl1);  
    p.add(tf1);  
    p.add(lbl2); p.add(tf2);  
    p.add(lbl3); p.add(tf3);  
    c.add(new JLabel("Division"),"North");  
    c.add(p,"Center");  
    c.add(btn,"South");  
}
```

```
public void actionPerformed(ActionEvent e)  
{  
    if(e.getSource()==btn)  
    {
```

```
try
{
    int a=Integer.parseInt(tf1.getText());
    int b=Integer.parseInt(tf2.getText());
    int c=a/b;
    tf3.setText(""+c);
}
catch(NumberFormatException ex)
{
    tf3.setText("-");
    JOptionPane.showMessageDialog(this,"NumberFormatException");
}
catch(ArithmeticException ex)
{
    tf3.setText("-");
    JOptionPane.showMessageDialog(this,"Division by zero");
}
catch(Exception ex)
{
    tf3.setText("-");
    JOptionPane.showMessageDialog(this,"Other Err "+ex.getMessage());
}
}
}

//actionPerformed

public static void main(String args[])
{
    Division b=new Division();
    b.setSize(200,200);
    b.setVisible(true);
}
}
```

