

Labs

Lab 6.1: Using CardLayout

What is the purpose?

In this lab, you will write an applet that performs arithmetic calculations on integers and rationals. The program uses two panels in a CardLayout manager—one for integer arithmetic and the other for rational arithmetic.

The program provides a combo box with two items, Integer and Rational. When the user chooses the Integer item, the integer panel is activated. When the user chooses the Rational item, the rational panel is activated.

Here is a sample of the output:

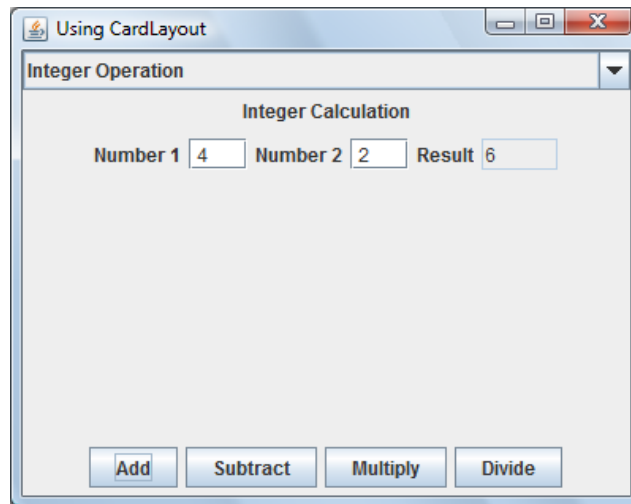


Figure 6-1-1: Sample Output 1

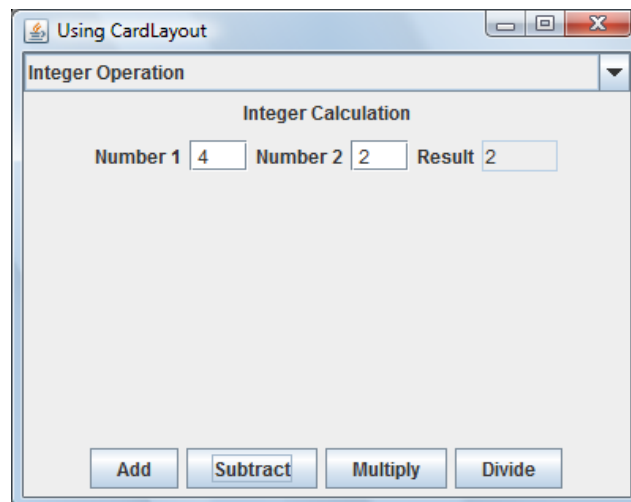


Figure 6-1-2: Sample Output 2

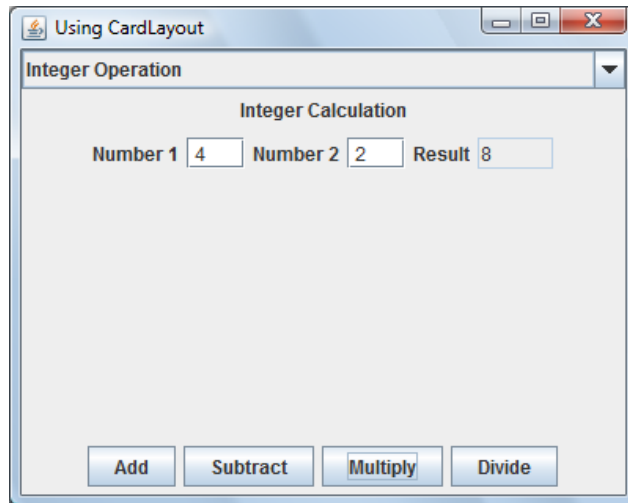


Figure 6-1-3: Sample Output 3

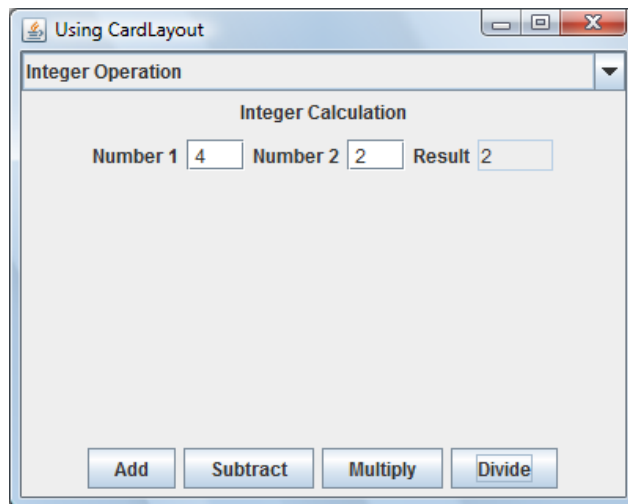


Figure 6-1-4: Sample Output 4

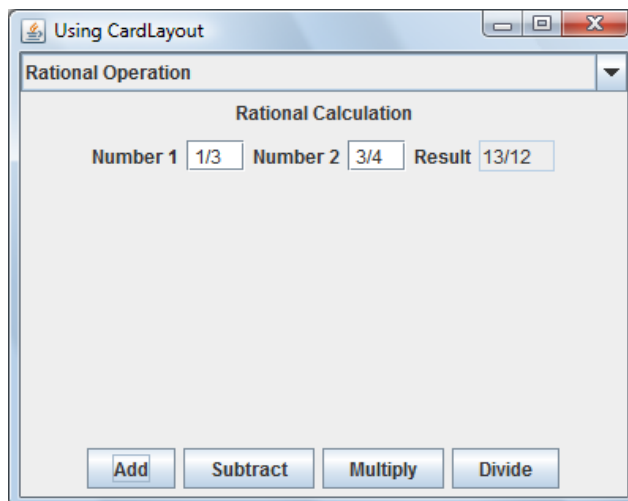
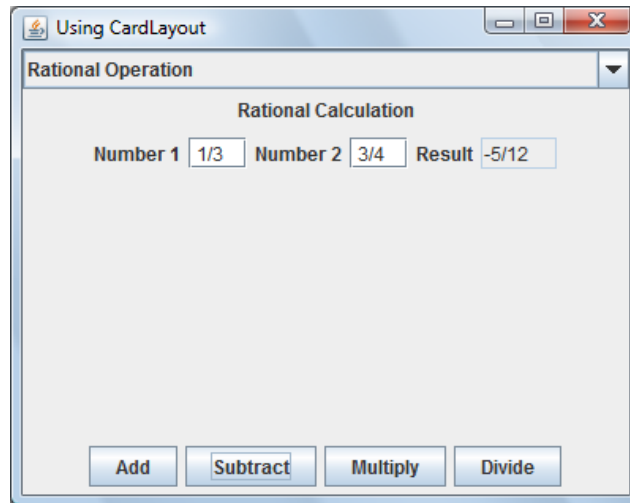
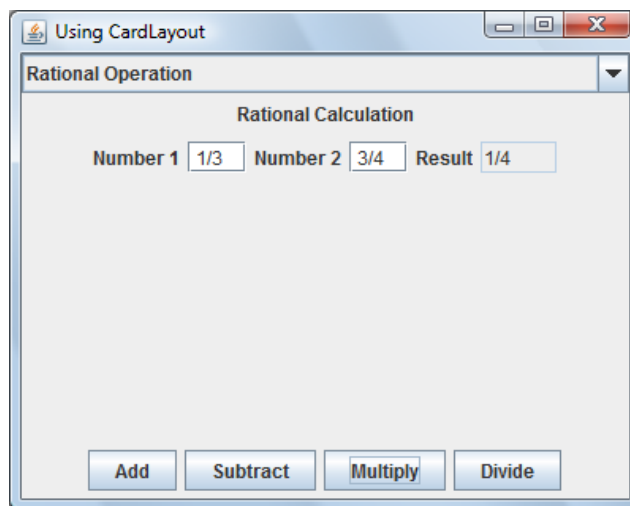
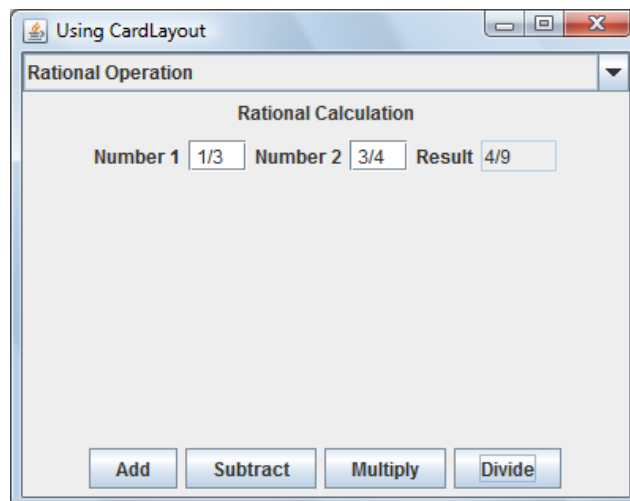


Figure 6-1-5: Sample Output 5

**Figure 6-1-6: Sample Output 6****Figure 6-1-7: Sample Output 7****Figure 6-1-8: Sample Output 8**

What are the steps?

- **Task 1**

Procedures:

1. Create a Java file named Rational.java to define a rational number and its associated operations, such as add, subtract, multiply, and divide.
2. Copy the following code or create your own:

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

public class cardLayout extends JApplet implements ActionListener
{
    private CardLayout queue = new CardLayout();
    private JPanel cardPanel = new JPanel();
    private JMenuItem jmiInt, jmiRat, jmiClose;
    private JComboBox jcboIntRational = new JComboBox(new Object[]{
        "Integer Operation", "Rational Operation"});

    public cardLayout() {

        // Create JMenuBar jmb
        JMenuBar jmb = new JMenuBar();

        //add a menu "Operation" in jmb
        JMenu operationMenu = new JMenu("Operation", false);
        jmb.add(operationMenu);

        //add a menu "Exit" in jmb
        JMenu exitMenu = new JMenu("Exit", true);
        jmb.add(exitMenu);

        //add JMenuItem
        operationMenu.add(jmiInt = new JMenuItem("Integer"));
        operationMenu.add(jmiRat = new JMenuItem("Rational"));
        exitMenu.add(jmiClose = new JMenuItem("Close"));

        //create intPanel for integer arithmetic
        JPanel intPanel = new JPanel();

        //create rationalPanel for rational arithmetic
        JPanel rationalPanel = new JPanel();

        cardPanel.setLayout(queue);
        cardPanel.add(intPanel, "Integer");
        cardPanel.add(rationalPanel, "Rational");

        //set FlowLayout in the frame
        getContentPane().setLayout(new BorderLayout());
        getContentPane().add(jcboIntRational, BorderLayout.NORTH);
        getContentPane().add(cardPanel, BorderLayout.CENTER);

        jcboIntRational.addActionListener(this);
    }
}
```

```

    }

    //handling menu selection
    public void actionPerformed(ActionEvent e) {
        if (jcboIntRational.getSelectedItem().equals("Integer
Operation"))
            queue.first(cardPanel);
        else if (jcboIntRational.getSelectedItem().equals("Rational
Operation"))
            queue.last(cardPanel);
    }

    public static void main(String[] args) {
        cardLayout applet = new cardLayout();
        JFrame frame = new JFrame();
        //EXIT_ON_CLOSE == 3
        frame.setDefaultCloseOperation(3);
        frame.setTitle("Using CardLayout");
        frame.getContentPane().add(applet, BorderLayout.CENTER);
        applet.init();
        applet.start();
        frame.setSize(400,320);
        Dimension d = Toolkit.getDefaultToolkit().getScreenSize();
        frame.setLocation((d.width - frame.getSize().width) / 2,
(d.height - frame.getSize().height) / 2);
        frame.setVisible(true);
    }
}

class IntPanel extends CalculationPanel {
    IntPanel() {
        super("Integer Calculation");
    }

    void add() {
        int result = getNum1() + getNum2();
        //set result in JTextField tf3
        tfResult.setText(String.valueOf(result));
    }

    void subtract() {
        int result = getNum1() - getNum2();
        //set result in JTextField tf3
        tfResult.setText(String.valueOf(result));
    }

    void multiply() {
        int result = getNum1() * getNum2();
        //set result in JTextField tfResult
        tfResult.setText(String.valueOf(result));
    }

    void divide() {
        int result = getNum1() / getNum2();
        //set result in JTextField tfResult
        tfResult.setText(String.valueOf(result));
    }
}

```

```
private int getNum1() {
    //use trim() to trim eztraneous space in the text field
    int num1 = Integer.parseInt(tfNum1.getText().trim());
    return num1;
}

private int getNum2() {
    //use trim() to trim eztraneous space in the text field
    int num2 = Integer.parseInt(tfNum2.getText().trim());
    return num2;
}
}

class RationalPanel extends CalculationPanel {
    RationalPanel() {
        super("Rational Calculation");
    }

    void add() {
        Rational num1 = getNum1();
        Rational num2 = getNum2();
        Rational result = num1.add(num2);

        //set result in JTextField tfResult
        tfResult.setText(result.toString());
    }

    void subtract() {
        Rational num1 = getNum1();
        Rational num2 = getNum2();
        Rational result = num1.subtract(num2);

        //set result in JTextField tfResult
        tfResult.setText(result.toString());
    }

    void multiply() {
        Rational num1 = getNum1();
        Rational num2 = getNum2();
        Rational result = num1.multiply(num2);

        //set result in JTextField tfResult
        tfResult.setText(result.toString());
    }

    void divide() {
        Rational num1 = getNum1();
        Rational num2 = getNum2();
        Rational result = num1.divide(num2);

        //set result in JTextField tfResult
        tfResult.setText(result.toString());
    }

    Rational getNum1() {
        StringTokenizer st1 = new
```

```

        StringTokenizer(stfNum1.getText().trim(), "/");
        int numer1 = Integer.parseInt(st1.nextToken());
        int denom1 = Integer.parseInt(st1.nextToken());
        return new Rational(numer1,denom1);
    }

    Rational getNum2() {
        StringTokenizer st2 = new
            StringTokenizer(stfNum2.getText().trim(), "/");
        int numer2 = Integer.parseInt(st2.nextToken());
        int denom2 = Integer.parseInt(st2.nextToken());
        return new Rational(numer2,denom2);
    }
}

/*design a generic calculation user interface for int and
rational arithmetic*/
abstract class CalculationPanel extends JPanel
implements ActionListener {
    private JPanel p0 = new JPanel();
    private JPanel p1 = new JPanel();
    private JPanel p2 = new JPanel();
    JTextField tfNum1, tfNum2, tfResult;
    private JButton jbtAdd, jbtSub, jbtMul, jbtDiv;

    public CalculationPanel(String title) {
        p0.add(new JLabel(title));

        //add labels and text fields
        p1.setLayout(new FlowLayout());
        p1.add(new JLabel("Number 1"));
        p1.add(tfNum1 = new JTextField(" ", 3));
        p1.add(new JLabel("Number 2"));
        p1.add(tfNum2 = new JTextField(" ", 3));
        p1.add(new JLabel("Result"));
        p1.add(tfResult = new JTextField(" ", 4));
        tfResult.setEditable(false);

        //set FlowLayout for p2
        JPanel p2 = new JPanel();
        p2.setLayout(new FlowLayout());
        p2.add(jbtAdd = new JButton("Add"));
        p2.add(jbtSub = new JButton("Subtract"));
        p2.add(jbtMul = new JButton("Multiply"));
        p2.add(jbtDiv = new JButton("Divide"));

        //add panels into CalculationPanel
        setLayout(new BorderLayout());
        add("North",p0);
        add("Center",p1);
        add("South",p2);

        //register listener for source objects
        jbtAdd.addActionListener(this);
        jbtSub.addActionListener(this);
        jbtMul.addActionListener(this);
        jbtDiv.addActionListener(this);
    }
}

```

```

    }

    public void actionPerformed(ActionEvent e) {
        String actionCommand = e.getActionCommand();
        if (e.getSource() instanceof JButton) {
            if ("Add".equals(actionCommand))
                add();
            else if ("Subtract".equals(actionCommand))
                subtract();
            else if ("Multiply".equals(actionCommand))
                multiply();
            else if ("Divide".equals(actionCommand))
                divide();
        }
    }

    abstract void add();

    abstract void subtract();

    abstract void multiply();

    abstract void divide();
}

```

Figure 6-1-9

3. Create an applet file named `cardLayout.java` as a GUI to implement arithmetic operations on integers and rationals. `CardLayout` is used to select panels that perform integer operations and rational number operations.
4. Complete the following code for `cardLayout.java` or create your own:

```

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

public class cardLayout extends JApplet implements ActionListener {
    private CardLayout queue = new CardLayout();
    private JPanel cardPanel = new JPanel();
    private JMenuItem jmiInt, jmiRat, jmiClose;
    private JComboBox jcboIntRational = new JComboBox(new Object[]{
        "Integer Operation", "Rational Operation"});

    public cardLayout() {

        // Create JMenuBar jmb
        JMenuBar jmb = new JMenuBar();

        //add a menu "Operation" in jmb
        JMenu operationMenu = new JMenu("Operation", false);
        jmb.add(operationMenu);

        //add a menu "Exit" in jmb
        JMenu exitMenu = new JMenu("Exit", true);
    }
}

```



```

jmb.add(exitMenu);

//add JMenuItem
operationMenu.add(jmiInt = new JMenuItem("Integer"));
operationMenu.add(jmiRat = new JMenuItem("Rational"));
exitMenu.add(jmiClose = new JMenuItem("Close"));

//create intPanel for integer arithmetic
JPanel intPanel = new IntPanel();

//create rationalPanel for rational arithmetic
JPanel rationalPanel = new RationalPanel();

cardPanel.setLayout(queue);
cardPanel.add(intPanel, "Integer");
cardPanel.add(rationalPanel, "Rational");

//set FlowLayout in the frame
getContentPane().setLayout(new BorderLayout());
getContentPane().add(jcboIntRational, BorderLayout.NORTH);
getContentPane().add(cardPanel, BorderLayout.CENTER);

jcboIntRational.addActionListener(this);
}

//handling menu selection
public void actionPerformed(ActionEvent e) {
    if (jcboIntRational.getSelectedItem().equals("Integer Operation"))
        queue.first(cardPanel);
    else if (jcboIntRational.getSelectedItem().equals("Rational
Operation"))
        queue.last(cardPanel);
}

public static void main(String[] args) {
    cardLayout applet = new cardLayout();
    JFrame frame = new JFrame();
    //EXIT_ON_CLOSE == 3
    frame.setDefaultCloseOperation(3);
    frame.setTitle("Using CardLayout");
    frame.getContentPane().add(applet, BorderLayout.CENTER);
    applet.init();
    applet.start();
    frame.setSize(400,320);
    Dimension d = Toolkit.getDefaultToolkit().getScreenSize();
    frame.setLocation((d.width - frame.getSize().width) / 2,
        (d.height - frame.getSize().height) / 2);
    frame.setVisible(true);
}
}

class IntPanel extends CalculationPanel {
    IntPanel() {
        super("Integer Calculation");
    }

    void add() {

```

```
int result = getNum1() + getNum2();
//set result in JTextField tf3
tfResult.setText(String.valueOf(result));
}

void subtract() {
    int result = getNum1() - getNum2();
    //set result in JTextField tf3
    tfResult.setText(String.valueOf(result));
}

void multiply() {
    int result = getNum1() * getNum2();
    //set result in JTextField tfResult
    tfResult.setText(String.valueOf(result));
}

void divide() {
    int result = getNum1() / getNum2();
    //set result in JTextField tfResult
    tfResult.setText(String.valueOf(result));
}

private int getNum1() {
    //use trim() to trim eztraneous space in the text field
    int num1 = Integer.parseInt(tfNum1.getText().trim());
    return num1;
}

private int getNum2() {
    //use trim() to trim eztraneous space in the text field
    int num2 = Integer.parseInt(tfNum2.getText().trim());
    return num2;
}
}

class RationalPanel extends CalculationPanel {
    RationalPanel() {
        super("Rational Calculation");
    }

    void add() {
        Rational num1 = getNum1();
        Rational num2 = getNum2();
        Rational result = num1.add(num2);

        //set result in JTextField tfResult
        tfResult.setText(result.toString());
    }

    void subtract() {
        Rational num1 = getNum1();
        Rational num2 = getNum2();
        Rational result = num1.subtract(num2);

        //set result in JTextField tfResult
        tfResult.setText(result.toString());
    }
}
```

```

    }

    void multiply() {
        Rational num1 = getNum1();
        Rational num2 = getNum2();
        Rational result = num1.multiply(num2);

        //set result in JTextField tfResult
        tfResult.setText(result.toString());
    }

    void divide() {
        Rational num1 = getNum1();
        Rational num2 = getNum2();
        Rational result = num1.divide(num2);

        //set result in JTextField tfResult
        tfResult.setText(result.toString());
    }

    Rational getNum1() {
        StringTokenizer st1 = new
            StringTokenizer(tfNum1.getText().trim(), "/");
        int numer1 = Integer.parseInt(st1.nextToken());
        int denom1 = Integer.parseInt(st1.nextToken());
        return new Rational(numer1,denom1);
    }

    Rational getNum2() {
        StringTokenizer st2 = new
            StringTokenizer(tfNum2.getText().trim(), "/");
        int numer2 = Integer.parseInt(st2.nextToken());
        int denom2 = Integer.parseInt(st2.nextToken());
        return new Rational(numer2,denom2);
    }
}

/*design a generic calculation user interface for int and
rational arithmetic*/
abstract class CalculationPanel extends JPanel
implements ActionListener {
    private JPanel p0 = new JPanel();
    private JPanel p1 = new JPanel();
    private JPanel p2 = new JPanel();
    JTextField tfNum1, tfNum2, tfResult;
    private JButton jbtAdd, jbtSub, jbtMul, jbtDiv;

    public CalculationPanel(String title) {
        p0.add(new JLabel(title));

        //add labels and text fields
        p1.setLayout(new FlowLayout());
        p1.add(new JLabel("Number 1"));
        p1.add(tfNum1 = new JTextField(" ", 3));
        p1.add(new JLabel("Number 2"));
        p1.add(tfNum2 = new JTextField(" ", 3));
        p1.add(new JLabel("Result"));
    }
}

```

```

p1.add(tfResult = new JTextField(" ", 4));
tfResult.setEditable(false);

//set FlowLayout for p2
JPanel p2 = new JPanel();
p2.setLayout(new FlowLayout());
p2.add(jbtAdd = new JButton("Add"));
p2.add(jbtSub = new JButton("Subtract"));
p2.add(jbtMul = new JButton("Multiply"));
p2.add(jbtDiv = new JButton("Divide"));

//add panels into CalculationPanel
setLayout(new BorderLayout());
add("North",p0);
add("Center",p1);
add("South",p2);

//register listener for source objects
jbtAdd.addActionListener(this);
jbtSub.addActionListener(this);
jbtMul.addActionListener(this);
jbtDiv.addActionListener(this);
}

public void actionPerformed(ActionEvent e) {
    String actionCommand = e.getActionCommand();
    if (e.getSource() instanceof JButton) {
        if ("Add".equals(actionCommand))
            add();
        else if ("Subtract".equals(actionCommand))
            subtract();
        else if ("Multiply".equals(actionCommand))
            multiply();
        else if ("Divide".equals(actionCommand))
            divide();
    }
}

abstract void add();

abstract void subtract();

abstract void multiply();

abstract void divide();
}

```

Figure 6-1-10

5. Submit your Java code and sample output to your instructor.

Did it work?

Were you able to:

- Create a panel in a CardLayout manager called integer panel that can perform integer calculations?

- Create a panel in a CardLayout manager called ration panel that can perform rational calculations?
- Switch the integer panel and ration panel in a combo box?

Lab 6.2: Using a Tabbed Pane

What is the purpose?

In this lab, you will reuse all the classes in Lab 6.1, using tabbed panes instead of the CardLayout manager.

Here is a sample of the output:

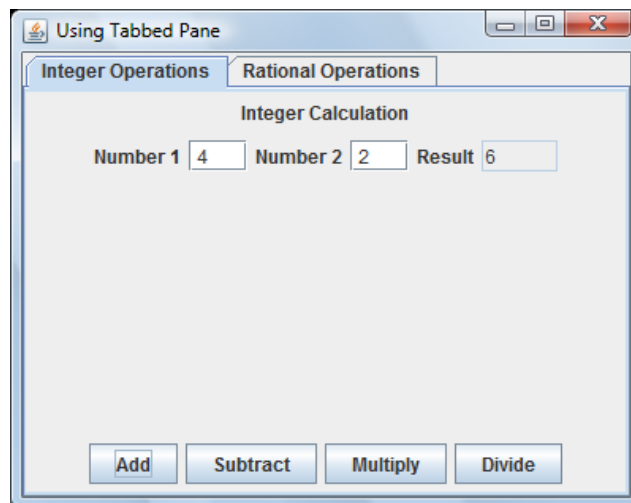


Figure 6-2-1: Sample Output 1

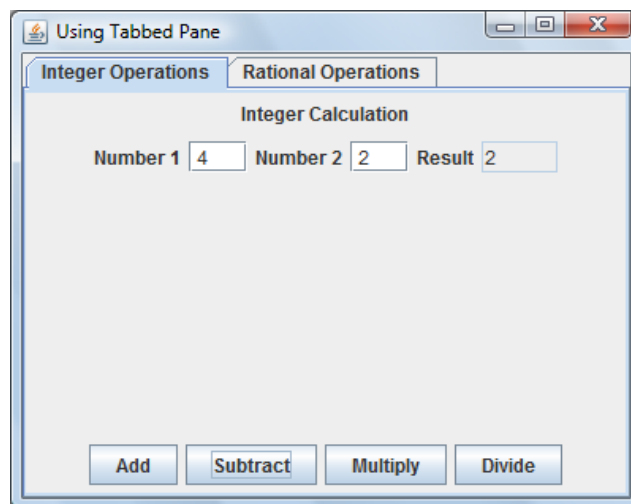


Figure 6-2-2: Sample Output 2

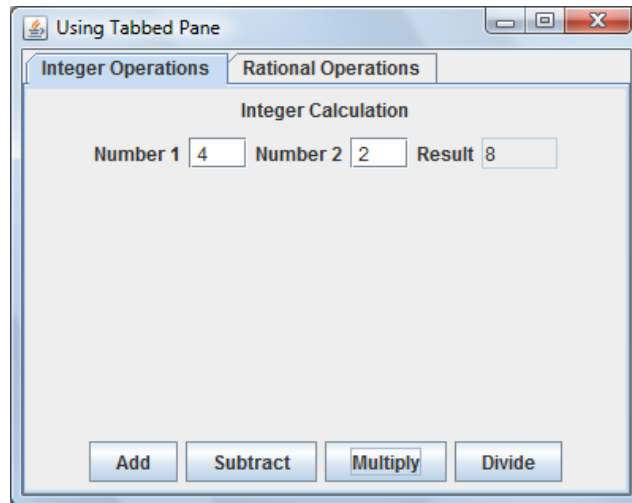


Figure 6-2-3: Sample Output 3

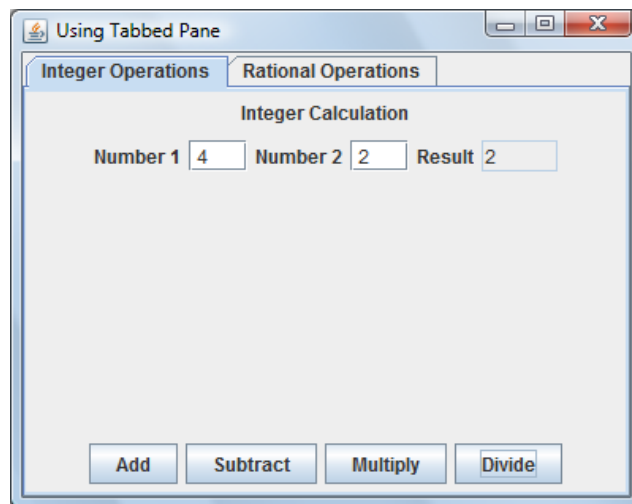


Figure 6-2-4: Sample Output 4

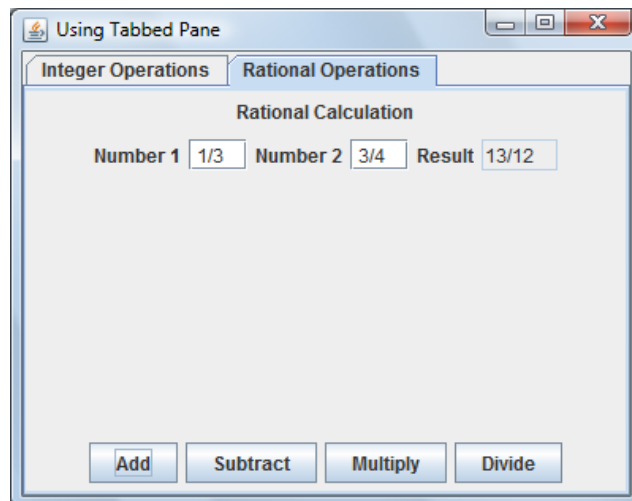


Figure 6-2-5: Sample Output 5

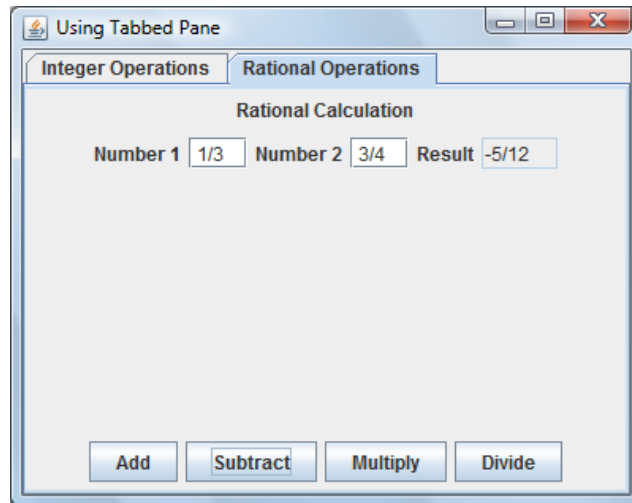


Figure 6-2-6: Sample Output 6

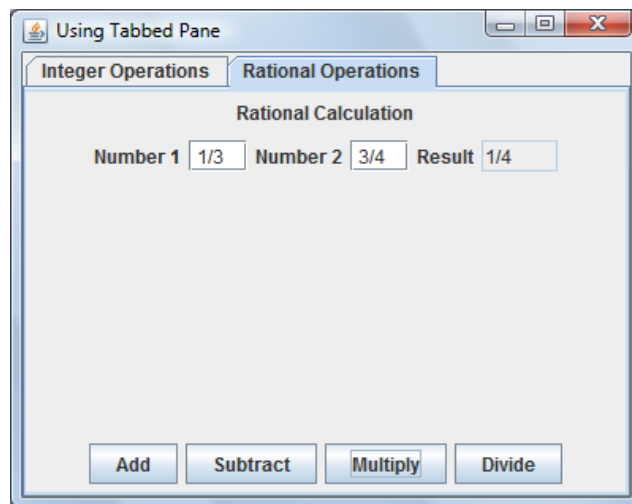


Figure 6-2-7: Sample Output 7

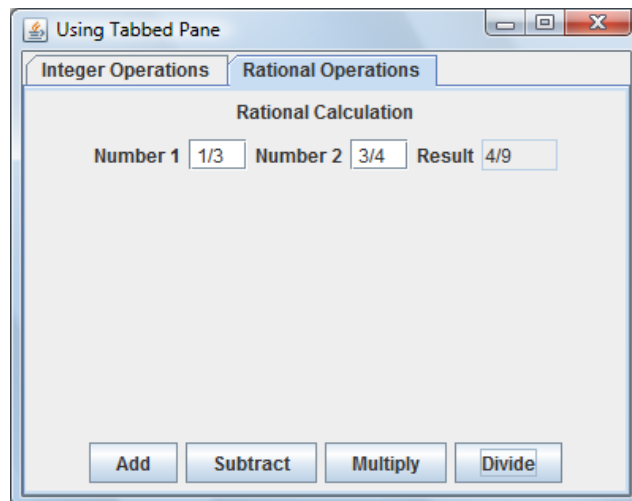


Figure 6-2-8: Sample Output 8

What are the steps?

- **Task 1**

Procedure:

1. Reuse the Rational and CardLayout classes in Lab 6.1.
2. Create an applet file named tabbedPane.java to use tabbed panes instead of the CardLayout manager. A tabbed pane is used to select panels that perform integer operations and rational number operations.
3. Complete the following code or create your own code:

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

public class tabbedPane extends JApplet {
    // Create a tabbed pane to hold figure panels
    private JTabbedPane jtpPanels = new JTabbedPane();

    public tabbedPane() {
        // Create intPanel for integer arithmetic
        JPanel intPanel = new JPanel();

        // Create rationalPanel for rational arithmetic
        JPanel rationalPanel = new JPanel();

        jtpPanels.add(intPanel, "Integer Operations");
        jtpPanels.add(rationalPanel, "Rational Operations");

        //set FlowLayout in the frame
        getContentPane().setLayout(new BorderLayout());
        getContentPane().add(jtpPanels, BorderLayout.CENTER);
    }

    public static void main(String[] args) {
        tabbedPane applet = new tabbedPane();
        JFrame frame = new JFrame();
        //EXIT_ON_CLOSE == 3
        frame.setDefaultCloseOperation(3);
        frame.setTitle("Using Tabbed Pane");
        frame.getContentPane().add(applet, BorderLayout.CENTER);
        applet.init();
        applet.start();
        frame.setSize(400,320);
        Dimension d = Toolkit.getDefaultToolkit().getScreenSize();
        frame.setLocation((d.width - frame.getSize().width) / 2,
            (d.height - frame.getSize().height) / 2);
        frame.setVisible(true);
    }
}
```

Figure 6-2-9

4. Submit your Java code and sample output to your instructor.

Did it work?

Were you able to:

- Create a tabbed panel in a CardLayout manager called integer panel that can perform integer calculations?
- Create a tabbed panel in a CardLayout manager called ration panel that can perform rational calculations?
- Switch the integer panel and ration panel in two different tabs?