CS102A Introduction to Computer Programming

Fall 2020

Lab 13

Objectives

1. Learn basic GUI programming

1 Exercises

1.1 Exercise 1

The following is a simple example of displaying a `.jpg` file with swing API.

DisplayJpg.java

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

public class DisplayJpg
{
  public static void main(String[] args)
  {
    JFrame window=new JFrame(); //create a Frame
    ImageIcon picture=new ImageIcon("C:\\Users\\todd\\Desktop\\a.
       jpg"); //load a picture from computer
    JLabel label=new JLabel(picture); //add the picture to a
       label

```

```
12      window.add(label); //add the label to the frame
13      window.setVisible(true); //Set the window to visible
14      window.setSize(400,400); //set the size of the window
15      window.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE); //let
            the window can be close by click "x"
16    }
17  }
```

Save a `.jpg` file to your PC and modify the path to your `.jpg` file in the above code. Compile and run the program.

Now if your `.jpg` file is too large, the window cannot display it in full size. It was because we hardcode the size when we set the window: `window.setSize(400,400);`

Can you modify the code so that it can set the window size to the size of your image?

> **Hint**
> Look for the functions from class `ImageIcon` to get back the size of the image.

Next, can you rescale your image to 50% of its size and display it?

> **Hint**
> Obtain an object of class `Image` from your existing `ImageIcon` object and use `getScaledInstance()` from class `Image`.

1.2 Exercise 2

Fill in the code below to implement the following functions:

1. Draw a circle in the center of the canvas (画布).
2. Increase the radius of the circle by 10% with a click of the `Enlarge` button.
3. Decrease the radius of the circle by 10% with a click of the `Shrink` button.

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

public class ControlCircle extends JFrame {
  private JButton jbtEnlarge = new JButton("Enlarge");
  private JButton jbtShrink = new JButton("Shrink");
  private CirclePanel canvas = new CirclePanel();

  public ControlCircle() {
    JPanel panel = new JPanel(); // Use the panel to group
       buttons
    panel.add(jbtEnlarge);
    panel.add(jbtShrink);

    this.add(canvas, BorderLayout.CENTER); // Add canvas to
       center
    this.add(panel, BorderLayout.SOUTH); // Add buttons to the
       frame

// Fill in the code to listen to the action event
//
//

  }

  /** Main method */
  public static void main(String[] args) {
    JFrame frame = new ControlCircle();
    frame.setTitle("ControlCircle2");
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.setSize(400, 400);
```

```
30
31      frame.setVisible(true);
32    }
33
34    class Listener implements ActionListener {
35      public void actionPerformed(ActionEvent e) {
36
37  // Fill in the code to response the enlarge or shrink event
38  //
39  //
40
41      }
42    }
43  }
44
45
46  class CirclePanel extends JPanel {
47    private int radius = 50; // Default circle radius
48
49    /** Enlarge the circle */
50    public void enlarge() {
51      radius = (int)(radius * 1.1);
52      this.repaint();
53    }
54
55    /** Enlarge the circle */
56    public void shrink() {
57      radius = (int)(radius * 0.9);
58      this.repaint();
59    }
60
61    /** Repaint the circle */
62    protected void paintComponent(Graphics g) {
```

```
63      super.paintComponent(g);
64
65  // Fill in the code to draw a circle in the center of the canvas
        with the radius of this class
66  //
67  //
68
69    }
70  }
```

1.3 Exercise 3

Understand the following code and fill in the `actionPerformed()` method to implement the plus and minus operation.

```
1  import java.awt.BorderLayout;
2  import java.awt.GridLayout;
3  import java.awt.event.ActionEvent;
4  import java.awt.event.ActionListener;
5
6  import javax.swing.JButton;
7  import javax.swing.JFrame;
8  import javax.swing.JPanel;
9  import javax.swing.JTextField;
10
11 public class Calculation extends JFrame {
12   private JButton plainJButton1;
13   private JButton plainJButton2;
14   private JButton plainJButton3;
15   private JButton plainJButton4;
16   private JButton plainJButton5;
17   private JButton plainJButton6;
18   private JButton plainJButton7;
19   private JButton plainJButton8;
```

```
20    private JButton plainJButton9;
21    private JButton plainJButton0;
22    private JButton plainJButtonAdd;
23    private JButton plainJButtonSub;
24    private JButton plainJButtonEq;
25
26    private JTextField answer;
27
28    private String operation1 = "";
29    private String operation2 = "";
30    private String operator = "";
31
32    // ButtonFrame adds JButtons to JFrame
33    public Calculation() {
34      super( "Calculator" );
35      JPanel jp = new JPanel();
36      jp.setLayout( new GridLayout(4,4) );
37
38      plainJButton1 = new JButton( "1" );
39      jp.add( plainJButton1 );
40
41      plainJButton2 = new JButton( "2" );
42      jp.add( plainJButton2 );
43
44      plainJButton3 = new JButton( "3" );
45      jp.add( plainJButton3 );
46
47      plainJButton4 = new JButton( "4" );
48      jp.add( plainJButton4 );
49
50      plainJButton5 = new JButton( "5" );
51      jp.add( plainJButton5 );
52
```

```
53      plainJButton6 = new JButton( "6" );
54      jp.add( plainJButton6 );
55
56      plainJButton7 = new JButton( "7" );
57      jp.add( plainJButton7 );
58
59      plainJButton8 = new JButton( "8" );
60      jp.add( plainJButton8 );
61
62      plainJButton9 = new JButton( "9" );
63      jp.add( plainJButton9 );
64
65      plainJButton0 = new JButton( "0" );
66      jp.add( plainJButton0 );
67
68      plainJButtonAdd = new JButton( "+" );
69      jp.add( plainJButtonAdd );
70
71      plainJButtonSub = new JButton( "-" );
72      jp.add( plainJButtonSub );
73
74      plainJButtonEq = new JButton( "=" );
75      jp.add( plainJButtonEq );
76
77      add(jp, BorderLayout.SOUTH);
78
79      answer = new JTextField("");
80      answer.setEditable(false);
81      answer.setHorizontalAlignment(JTextField.RIGHT);
82      add(answer, BorderLayout.CENTER);
83
84      // create new ButtonHandler for button event handling
85      ButtonHandler handler = new ButtonHandler();
```

```
86      plainJButton1.addActionListener( handler );
87      plainJButton2.addActionListener( handler );
88      plainJButton3.addActionListener( handler );
89      plainJButton4.addActionListener( handler );
90      plainJButton5.addActionListener( handler );
91      plainJButton6.addActionListener( handler );
92      plainJButton7.addActionListener( handler );
93      plainJButton8.addActionListener( handler );
94      plainJButton9.addActionListener( handler );
95      plainJButton0.addActionListener( handler );
96      plainJButtonAdd.addActionListener( handler );
97      plainJButtonSub.addActionListener( handler );
98      plainJButtonEq.addActionListener( handler );
99    } // end ButtonFrame constructor
100
101   public int compute(String operation1, String operation2, String
          operator) {
102     int a = Integer.parseInt(operation1);
103     int b = Integer.parseInt(operation2);
104     if (operator.charAt(0) == '+') {
105       return a + b;
106     } else {
107       return a - b;
108     }
109   }
110
111   public static void main( String[] args ) {
112     Calculation calculationFrame = new Calculation(); // create
          ButtonFrame
113     calculationFrame.setDefaultCloseOperation( JFrame.
          EXIT_ON_CLOSE );
114     calculationFrame.setLocationRelativeTo(null);
115     calculationFrame.pack(); // set frame size
```

```
116     calculationFrame.setVisible( true ); // display frame
117   } // end main
118
119   // inner class for button event handling
120   private class ButtonHandler implements ActionListener {
121     // handle button event
122     public void actionPerformed( ActionEvent event ) {
123
124 // Fill in the code
125 //
126 //
127
128     } // end method actionPerformed
129   } // end private inner class ButtonHandler
130 } // end class ButtonFrame
```