Tutorial of GUI and Java Swing

> Based on the tutorial of "2020S-Java-A" designed by teaching group in SUSTech
>
> Minor changes by Yida Tao, Dec. 1st 2022

Objective

- Learn basic GUI programming using Java Swing

Demo

Download `DisplayJpg.java`, save a `.jpg` file on your PC and modify the path to your `.jpg` file in the code. Then compile and run the program.

```
JFrame window = new JFrame();  //create a Frame
ImageIcon picture = new ImageIcon("src/lab13/swing.jpg");  //load a picture from
computer
JLabel label = new JLabel(picture);  //add the picture to a label

window.add(label);  //add the label to the frame
window.setVisible(true);  //Set the window to visible
//set the size of the window (try modifying the width and height)
window.setSize(400, 400);
window.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  // program exits when
window is closed by clicking "x"
```

Since your `.jpg` file is too large, the window cannot display it in full size. It was because we **hardcoded** the size when we set the window:

```
window.setSize(400,400);
```

Instead of hardcoding window size, you probably want to adapt the window size based on the image:

- Can you modify the code so that it can set the window size based on the size of your image? (Hint: Look for methods from class `ImageIcon` to get back the size of the image)
- 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`. Check `display2()`)

Exercise 1

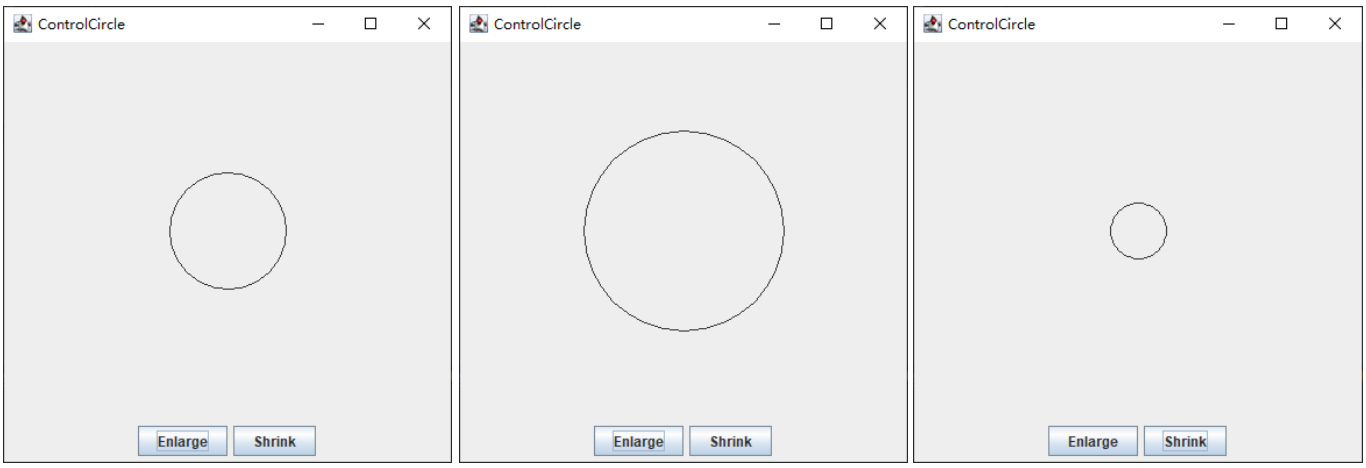
Fill in `\\TODO` in the code below to implement the following functions:

- Draw a circle in the center of the canvas.
- Increase the radius of the circle by 10% by clicking the `Enlarge` button
- Decreasing the radius of the circle by 10% by clicking the `Shrink` button.

```
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

        // TODO: register listeners to the buttons

    }

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

        frame.setVisible(true);
    }

    class Listener implements ActionListener {
        public void actionPerformed(ActionEvent e) {
            // TODO: respond to enlarge or shrink button click events
        }
    }
}

class CirclePanel extends JPanel {
    private int radius = 50; // Default circle radius

    /** Enlarge the circle */
    public void enlarge() {
        radius = (int)(radius * 1.1);
        this.repaint();

    }

    /** Enlarge the circle */
    public void shrink() {
        radius = (int)(radius * 0.9);
        this.repaint();
    }

    /** Repaint the circle */
    protected void paintComponent(Graphics g) {
```

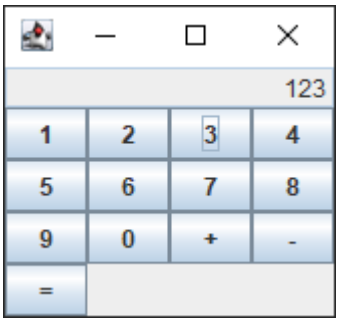
```
            super.paintComponent(g);

            g.drawOval(this.getWidth()/2-radius, this.getHeight()/2-radius, 2 *
radius, 2 * radius);
        }
}
```

A sample run:



Exercise 2

The following code implements a simple calculator. However, you can only enter a number with 1 digit (e.g., 8). Please modify the code so that users can enter a number that has multiple digits (e.g., 123). A sample run:




```
public class CalculationTest
{
   public static void main( String[] args )
   {
       Calculation calculationFrame = new Calculation();
       calculationFrame.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
       calculationFrame.setLocationRelativeTo(null);
       calculationFrame.pack(); // set frame size
       calculationFrame.setVisible( true ); // display frame
   }
}
```

```
public class Calculation extends JFrame
{
   private JButton plainJButton1;
   private JButton plainJButton2;
   private JButton plainJButton3;
   private JButton plainJButton4;
   private JButton plainJButton5;
   private JButton plainJButton6;
   private JButton plainJButton7;
   private JButton plainJButton8;
   private JButton plainJButton9;
   private JButton plainJButton0;
   private JButton plainJButtonAdd;
   private JButton plainJButtonSub;
   private JButton plainJButtonEq;

   private JTextField answer;

   private String operation1 = "";
   private String operation2 = "";
   private String operator = "";

   public Calculation()
   {
      super( "Calculation" );
      JPanel jp = new JPanel();
      jp.setLayout( new GridLayout(4,4) );

      plainJButton1 = new JButton( "1" );
      jp.add( plainJButton1 );

      plainJButton2 = new JButton( "2" );
      jp.add( plainJButton2 );

      plainJButton3 = new JButton( "3" );
      jp.add( plainJButton3 );

      plainJButton4 = new JButton( "4" );
      jp.add( plainJButton4 );

      plainJButton5 = new JButton( "5" );
      jp.add( plainJButton5 );

      plainJButton6 = new JButton( "6" );
      jp.add( plainJButton6 );

      plainJButton7 = new JButton( "7" );
      jp.add( plainJButton7 );

      plainJButton8 = new JButton( "8" );
      jp.add( plainJButton8 );

      plainJButton9 = new JButton( "9" );
```

```
        jp.add( plainJButton9 );

        plainJButton0 = new JButton( "0" );
        jp.add( plainJButton0 );

        plainJButtonAdd = new JButton( "+" );
        jp.add( plainJButtonAdd );

        plainJButtonSub = new JButton( "-" );
        jp.add( plainJButtonSub );

        plainJButtonEq = new JButton( "=" );
        jp.add( plainJButtonEq );

        add(jp, BorderLayout.SOUTH);

        answer = new JTextField("");
        answer.setEditable(false);
        answer.setHorizontalAlignment(JTextField.RIGHT);
        add(answer, BorderLayout.CENTER);


        // create new ButtonHandler for button event handling
        ButtonHandler handler = new ButtonHandler();
        plainJButton1.addActionListener( handler );
        plainJButton2.addActionListener( handler );
        plainJButton3.addActionListener( handler );
        plainJButton4.addActionListener( handler );
        plainJButton5.addActionListener( handler );
        plainJButton6.addActionListener( handler );
        plainJButton7.addActionListener( handler );
        plainJButton8.addActionListener( handler );
        plainJButton9.addActionListener( handler );
        plainJButton0.addActionListener( handler );
        plainJButtonAdd.addActionListener( handler );
        plainJButtonSub.addActionListener( handler );
        plainJButtonEq.addActionListener( handler );
    }

     public int compute(String operation1, String operation2, String operator) {
         int a = Integer.parseInt(operation1);
         int b = Integer.parseInt(operation2);
         if (operator.charAt(0) == '+') {
             return a + b;
         } else {
             return a - b;
         }
     }

    // inner class for button event handling
    private class ButtonHandler implements ActionListener
    {
        // handle button event
        public void actionPerformed( ActionEvent event )
```

```
      {
        if (( event.getSource() == plainJButtonAdd ) || ( event.getSource() ==
plainJButtonSub )) {
            // op1 +
            if (!operation1.equals("") && operation2.equals("")) {
                operator = event.getActionCommand();
            }
            // op1 + op2 +
            else if (!operation1.equals("") && !operation2.equals("")){
                operation1 = Integer.toString(compute(operation1, operation2,
operator));
                answer.setText(operation1); // display the first sum
                operation2 = "";
                operator = event.getActionCommand();
            }
            // answer +
            else if (!answer.getText().equals("")){
                operation1 = answer.getText();
                operator = event.getActionCommand();
            }
        }
        else if ( event.getSource() == plainJButtonEq ){
            // op1 + op2 =
            if (!operation1.equals("") && !operation2.equals("") &&
!operator.equals("") ){
                answer.setText(Integer.toString(compute(operation1, operation2,
operator)));
                operation1 = "";
                operation2 = "";
                operator = "";
            }
            else{
                answer.setText("");
                operation1 = "";
                operation2 = "";
                operator = "";
            }

        }
        else {
            // enter digits
            if (operator.equals("")) {
                operation1 = event.getActionCommand();
                answer.setText(operation1);
            } else {
                operation2 = event.getActionCommand();
                answer.setText(operation2);
            }
        }
      }
   }
}
```