1. Pascal Triangle (Extend from HW4 Q1)

Write a Java application program that creates a two-dimensional matrix representing the inverse Pascal triangle. Input the size from the user, which is an integer between 1~15. Output the inverse Pascal triangle on the JPanel.

Requirements

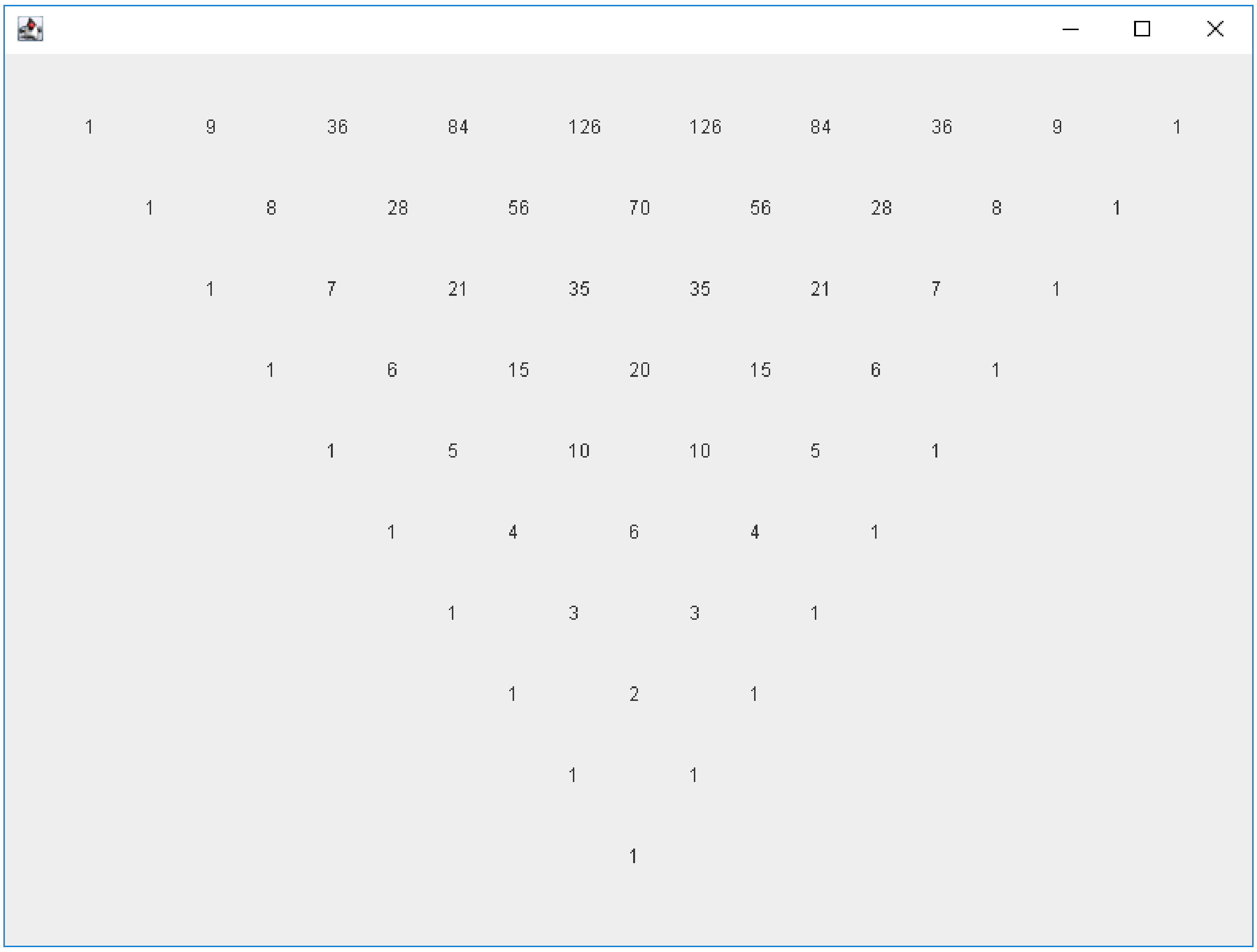
I. Use Enhanced for statement for the array process

II. Use JFrame-JPanel for displaying area. The Pascal triangle should fit the window size as resizing the window.

Sample input

10

Sample output



```
1    9    36   84   126  126  84   36   9    1
  1    8    28   56   70   56   28   8    1
    1    7    21   35   35   21   7    1
      1    6    15   20   15   6    1
        1    5    10   10   5    1
          1    4    6    4    1
            1    3    3    1
              1    2    1
                1    1
                  1
```

Note

You need to use Java enhanced for loop syntax. If you are not familiar with it, search it from the lecturer's handout or google it.

2. Airline reservation system (Extend from HW4 Q2)

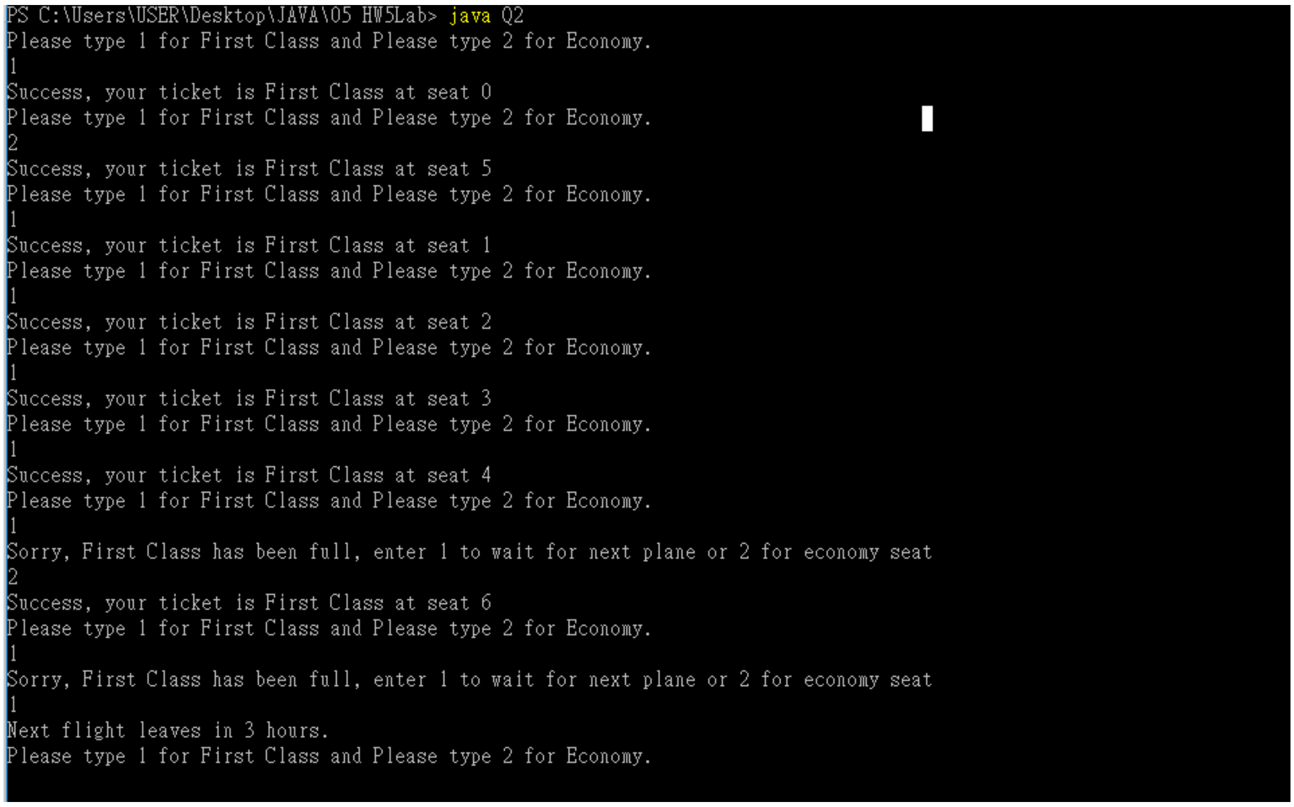
A small airline has just purchased a computer for its new automated reservations system. You've been asked to develop the new system. You're to write an application to assign seats on each flight of the airline's only plane (capacity: 10 seats). Your application should display the following alternatives: Please type 1 for First Class and Please type 2 for Economy. If the user types 1, your application should assign a seat in the firstclass section (seats 1-5). If the user types 2, your application should assign a seat in the economy section (seats 6-10). Your application should then display a boarding pass on a JFrame-JPanel design indicating the person's seat number and whether it's in the first-class or economy section of the plane. Use a one-dimensional array of primitive type boolean to represent the seating chart of the plane. Initialize all the elements of the array to false to indicate that all the seats are empty. As each seat is assigned, set the corresponding element of the array to true to indicate that the seat is no longer available. Your application should never assign a seat that has already been assigned. When the economy section is full, your application should ask the person if it's acceptable to be placed in the first-class section (and vice versa). If yes, make the appropriate seat assignment. If no, display the message "Next flight leaves in 3 hours."

Requirements

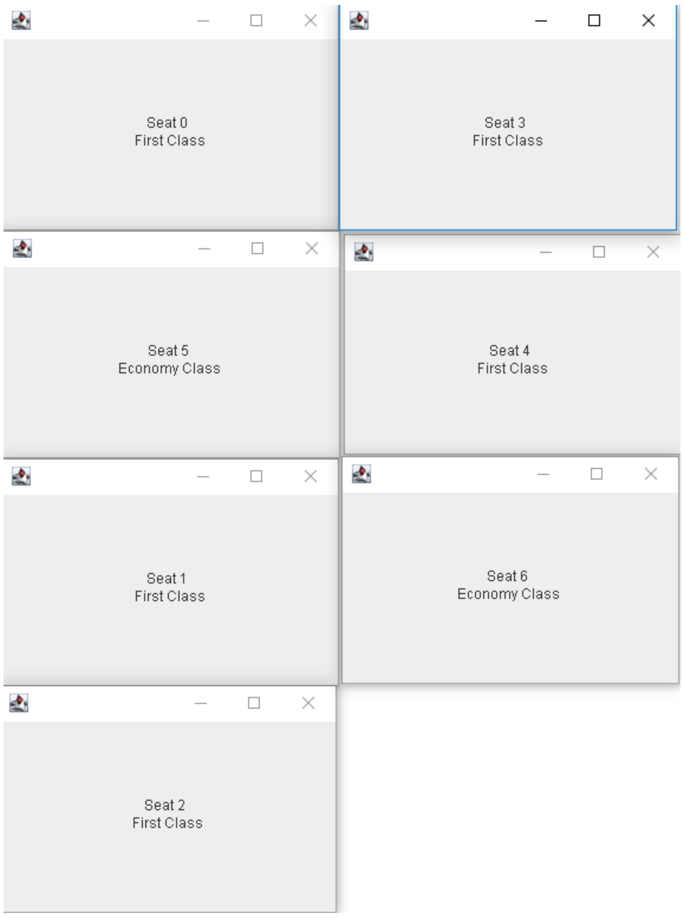
I. Use Enhanced for statement for the array process

II. Use JPanel for displaying area.

Sample I/O



```
PS C:\Users\USER\Desktop\JAVA\05 HW5Lab> java Q2
Please type 1 for First Class and Please type 2 for Economy.
1
Success, your ticket is First Class at seat 0
Please type 1 for First Class and Please type 2 for Economy.
2
Success, your ticket is First Class at seat 5
Please type 1 for First Class and Please type 2 for Economy.
1
Success, your ticket is First Class at seat 1
Please type 1 for First Class and Please type 2 for Economy.
1
Success, your ticket is First Class at seat 2
Please type 1 for First Class and Please type 2 for Economy.
1
Success, your ticket is First Class at seat 3
Please type 1 for First Class and Please type 2 for Economy.
1
Success, your ticket is First Class at seat 4
Please type 1 for First Class and Please type 2 for Economy.
1
Sorry, First Class has been full, enter 1 to wait for next plane or 2 for economy seat
2
Success, your ticket is First Class at seat 6
Please type 1 for First Class and Please type 2 for Economy.
1
Sorry, First Class has been full, enter 1 to wait for next plane or 2 for economy seat
1
Next flight leaves in 3 hours.
Please type 1 for First Class and Please type 2 for Economy.
```

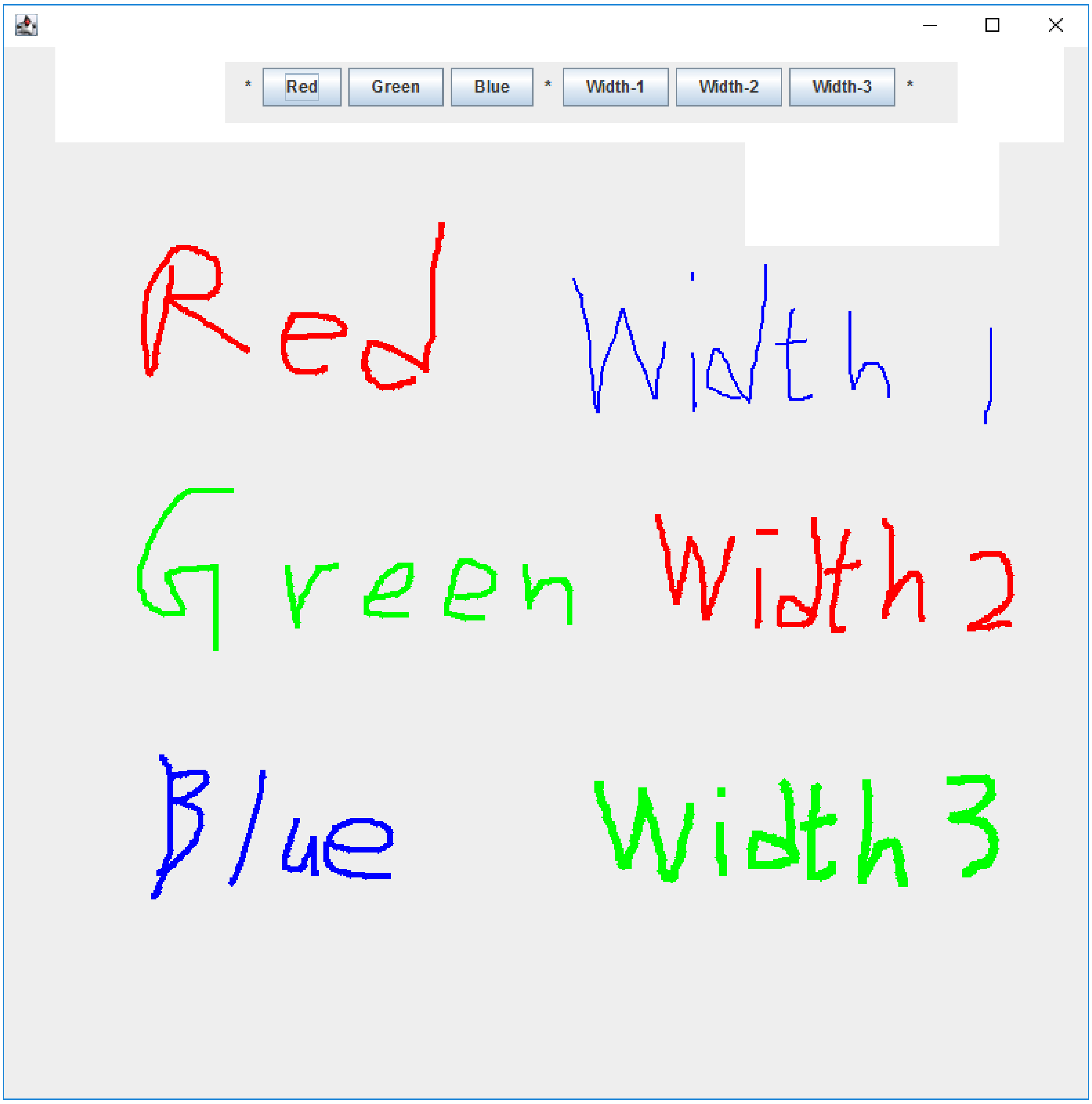


3. Painting

Write a JFrame-JPanel program for painting. Drag the mouse left-button to paint on the JPanel. Create 1 JFrame:

The JFrame (Size: 800*800) should contain 6 Buttons and 1 JPanel. Button Red, Green and Blue to change the color of brush. Button Width-1, Width-2, Width-3 to change the size of brush.

Sample Output



Note:

**** Code given at the end of this file ****

MouseListener & MouseMotionListener functions

public void mousePressed(MouseEvent event) { ... }

public void mouseReleased(MouseEvent event) { ... }

```
public void mouseDragged(MouseEvent event) { ... }
public void mouseClicked(MouseEvent event) { ... }
public void mouseEntered(MouseEvent event) { ... }
public void mouseExited(MouseEvent event) { ... }
public void mouseMoved(MouseEvent event) { ... }
```

Get mouse (x,y)

```
x = event.getX();
y = event.getY();
```

Set color and line-width

```
Graphics g = getGraphics();
Graphics2D g2 = (Graphics2D)g;
g2.setColor(color);
g2.setStroke(new BasicStroke(width));
g2.draw...
```

4. Painting (Continue)

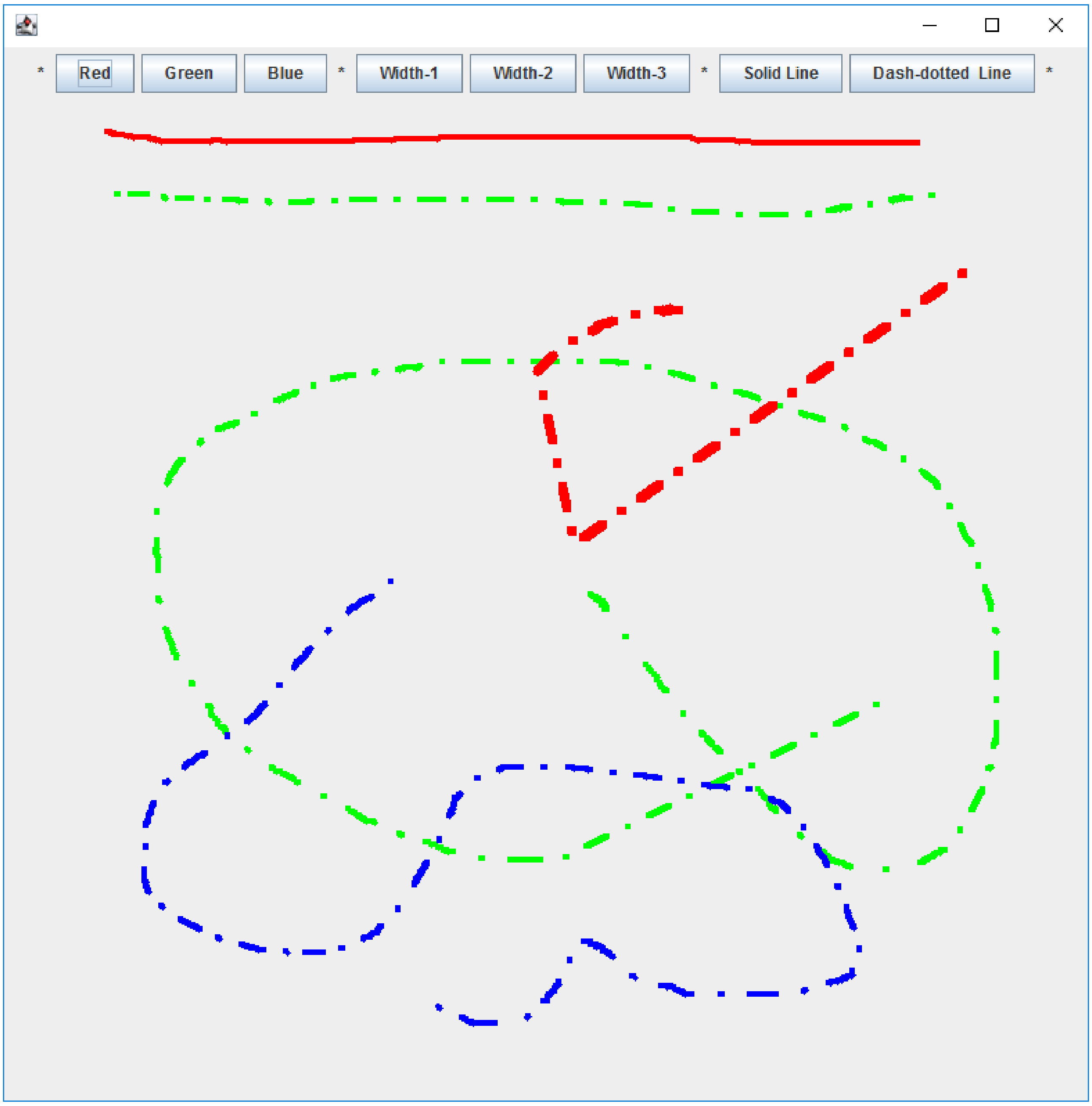
Please modify your program from Q3. Implementing two new buttons, Solid Line and Dash-dotted Line.

Requirement

A cycle of dash-dotted line should length 50 pixels.

If you have no idea about how to make it, the TA will give a hint at 20:30.

Sample Output



Code for Q3 and Q4   // you may or may not use all of them

Main

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

public class Q3 {

	public static void main(String[] args) {

		JFrame  frame1 = new JFrame();
		Q3_0 panel1 = new Q3_0();

		JButton btn_red   = new JButton("Red");
		JButton btn_green = new JButton("Green");
		// ...

		btn_red.   addActionListener((e)->   // ...
		//...

		panel1.setLayout(new FlowLayout // ...
		panel1.add(new JLabel(" * "));
		panel1.add(btn_red)
		// ...

		frame1.pack();
		frame1.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
		frame1.setSize(800, 800);
		frame1.setVisible(true);
	}

}
```

JPanel

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

public class Q3_0 extends JPanel implements MouseListener, MouseMotionListener {

	// ... some variable

	public Q3_0() {
		addMouseListener(this);
		addMouseMotionListener(this);
		// ... some initialize
	}
	public void mousePressed(MouseEvent event) {
		// ...
	}

	public void mouseReleased(MouseEvent event) {
		// ...
	}

	public void mouseDragged(MouseEvent event) {
		// ...
	}

	public void mouseClicked(MouseEvent event) {
		// ...
	}

	public void mouseEntered(MouseEvent event) {
		// ...
	}
	public void mouseExited(MouseEvent event) {
		// ...
	}
	public void mouseMoved(MouseEvent event) {
		// ...
	}

	public void setColor(Color c) {
		// ...
	}
	public void setWidth(float w) {
		// ...
	}
	public void setLineShape(int s) {
		// ...
	}

}
```