

1811943 황성미 8일차 과제

▼ JAVA Swing

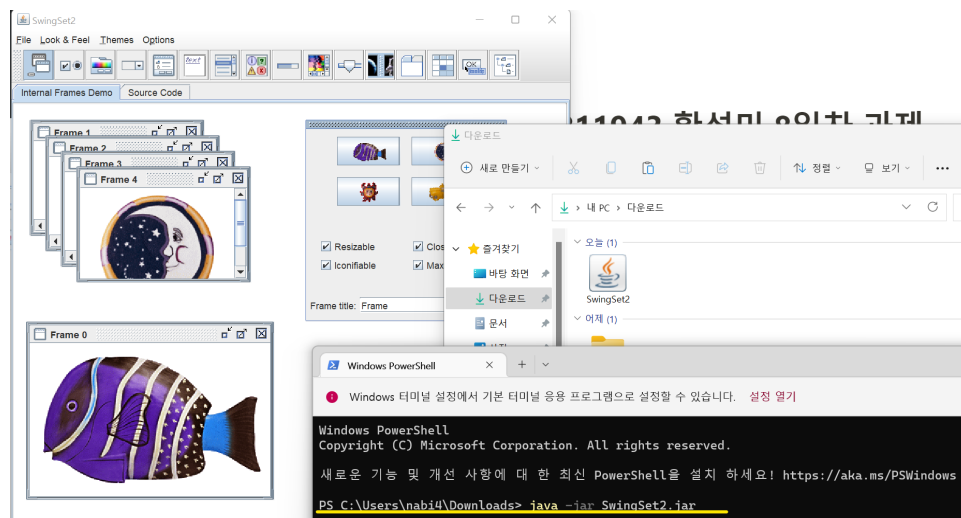
awt 보다 조금 더 기능이 다양함

▼ 5.1 Java Swing

AWT GUI 컴포넌트를 상속받고 있음

javax.swing Package에 들어가있음

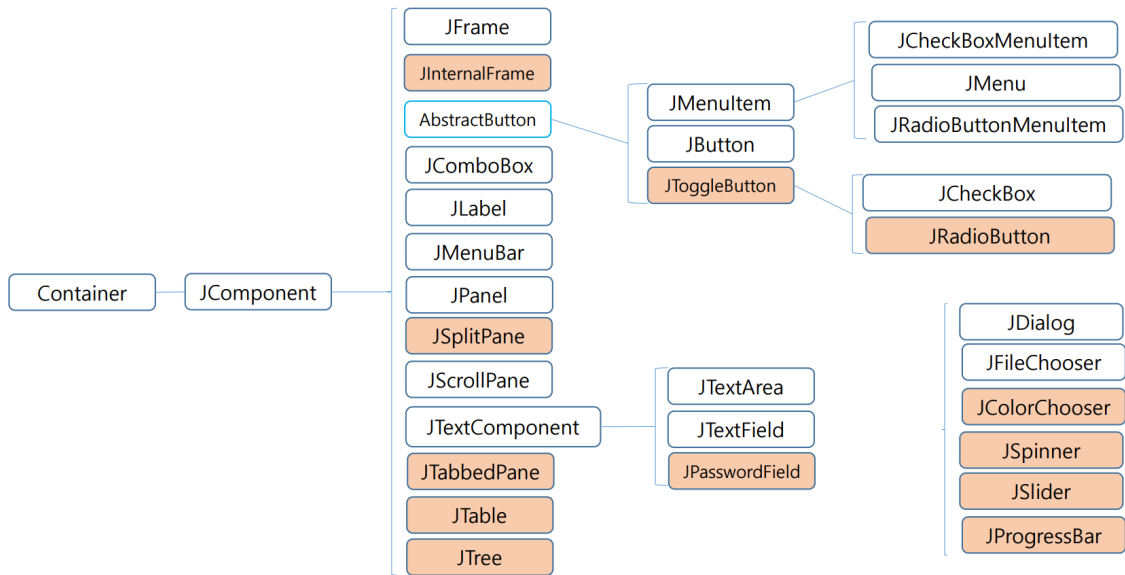
▼ Java Swing Demo



SwingSet2.jar를 다운로드 받은 폴더에서 다음과 같은 코드를 실행시키면 SwingDemo 파일이 실행
됨

▼ 5.2 Swing Components Hierarchy

모든 자바 스윙 컴포넌트는 Jcomponent의 성질을 가지고 있으며 이들은 Container 성질
을 가지고 있다. (다른 컴포넌트들을 포함할 수 있음) 이들은 awt 컴포넌트의 성질을 가지고
있음



흰색 박스 - AWT와 같음, 빨간 박스 - Swing만 가지고 있는 컴포넌트

Swing의 컴포넌트들은 대부분 안드로이드에 포함되어있음


▼ 5.3 Swing Components

JComponent 클래스를 상속함

▼ JComponent

JComponent (Java Platform SE 7)

AbstractButton, BasicInternalFrameTitlePane, Box, Box.Filler, JColorChooser, JComboBox, JFileChooser, JInternalFrame, JInternalFrame.JDesktopIcon, JLabel, JLayer, JLayeredPane, JList, JMenuBar, JOptionPane, JPanel, JPopupMenu, JProgressBar, JRootPane, JScrollbar, JScrollPane,

 <https://docs.oracle.com/javase/7/docs/api/javax/swing/JComponent.html>

JComponent 공식 문서 (메서드 참고)

그 때 그 때 필요한 메서드들은 찾아서 사용하면 됨

paint(Graphics g)

repaint(long tm, int x, int y, int width, int height)가 대표적임

▼ 5.3.1 JFrame

컴포넌트들을 포함시킴

▼ Constructor

JFrame(), JFrame(GraphicsConfiguration gc), JFrame(String title),
JFrame(String title, GraphicsConfiguration gc)

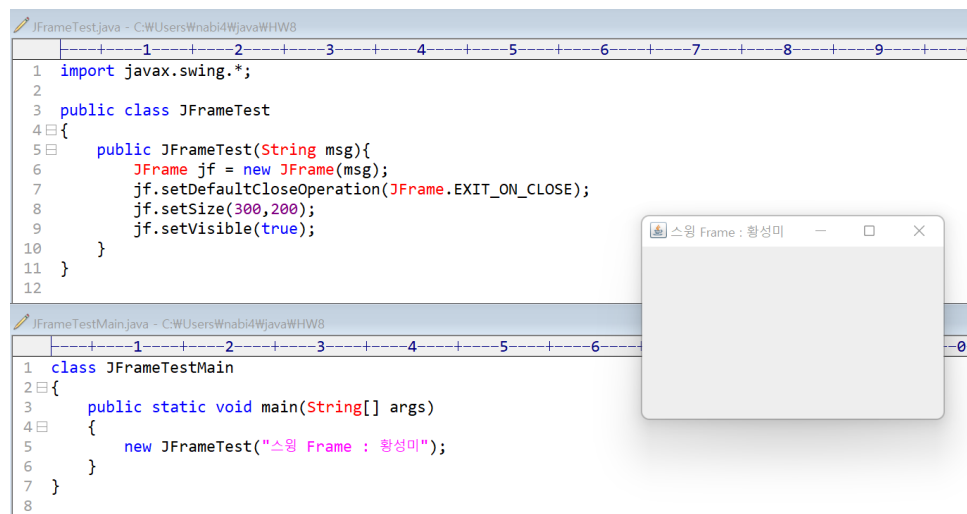
▼ Method

getContentPane() : add 역할의 메서드, setMenuBar(JMenuBar menubar),
setLayout(LayoutManager manager)

setContentPane()의 옵션

- ①DO_NOTHING_ON_CLOSE: 그대로 둔다.
- ②HIDE_ON_CLOSE: 프레임을 안 보이도록 한다.
- ③DISPOSE_ON_CLOSE: 프레임을 안 보이게 하고, dispose() 함수를 이용하여 자원을 반납한다.
- ④EXIT_ON_CLOSE: system.exit() 메소드를 이용해서 프로그램을 종료한다

▼ Lab



Swing을 이용하여 창을 띄워봄

▼ 5.3.2 JLabel

텍스트와 함께 이미지가 들어갈 수 있음

▼ Constructor

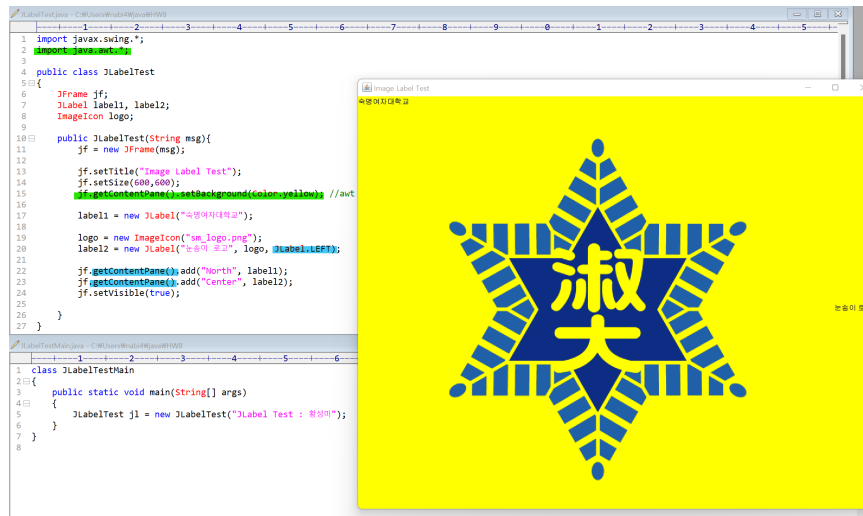
JLabel(), JLabel(String text), JLabel(icon image), JLabel(String text, int alignment), JLabel(String text, icon image)

▼ Method

getText(), setText(String text), getIcon(), setIcon(),
setHorizontalAlignment()_LEFT, RIGHT, CENTER, setVerticalAlignment()_TOP,
BOTTOM, CENTER

▼ Lab(이미지가 있는 라벨과 텍스트만 있는 라벨)

getContentPane() : add명령어



숙명 로고를 java 파일이 있는 폴더에 다운받아놓은 후 getContentPane() 메서드를 사용해 텍스트 옆
에 이미지를 붙여줌

▼ 5.3.3 JPanel

AWT와 동일

▼ Constructor

JPanel(), JPanel(LayoutManager layout)

▼ 5.3.4 AbstractButton

클릭과 선택 등 버튼 클래스에서 필요한 기본적인 메서드들을 정의

JButton, JCheckBox, JRadioButton, JToggleButton, JMenu, JMenuItem

▼ Method

getIcon(), setIcon(Icon icon), getText(), setText(String text),
getHorizontalAlignment(), getVerticalAlignment(), setHorizontalAlignment(h-
value), setVerticalAlignment(v-value)

▼ 5.3.4.1 JButton

AWT의 Button과 동일하지만 이미지 아이콘을 추가할 수 있음

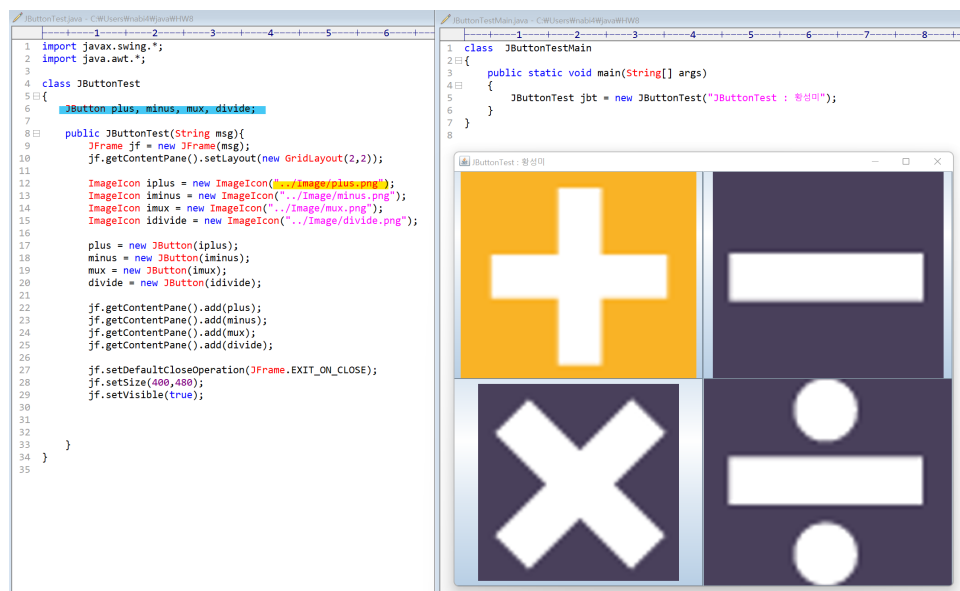
▼ Constructor

JButton(), JButton(String name), JButton(Icon icon), JButton(String name, Icon icon)

▼ Method

addActionListner()

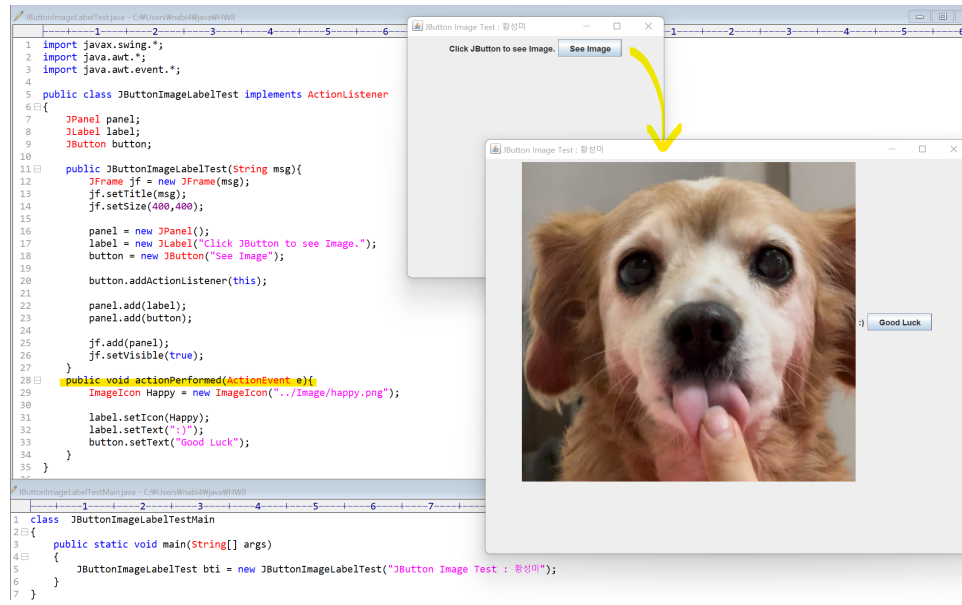
▼ Lab



JButton은 이미지를 삽입할 수 있음.

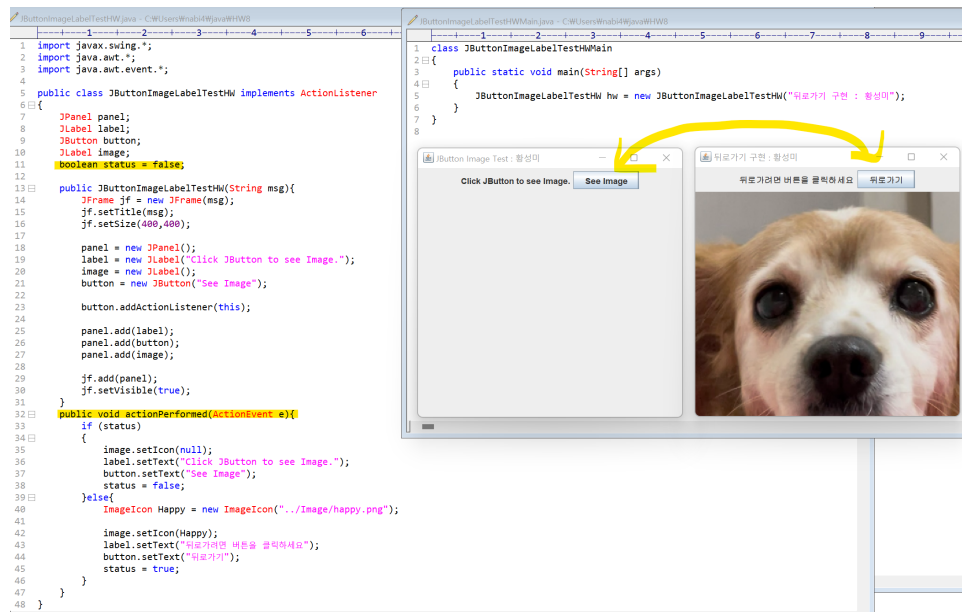
이미지는 폴더를 따로 만들어 폴더 경로를 지정해주면 바로 사용 가능

▼ Lab2



actionPerformed를 구현하여 버튼을 클릭하면 이미지가 뜨도록 함

▼ HW(JButton에 뒤로 가기 기능 구현)



actionPerformed에서 status 변수를 생성하여 클릭을 할 때 마다 이벤트가 다르게 처리되도록 함

▼ 5.3.4.2 JCheckBox, JRadioButton, JToggleButton

AWT의 CheckboxGroup이 JRadioButton 인 것외에 다 동일

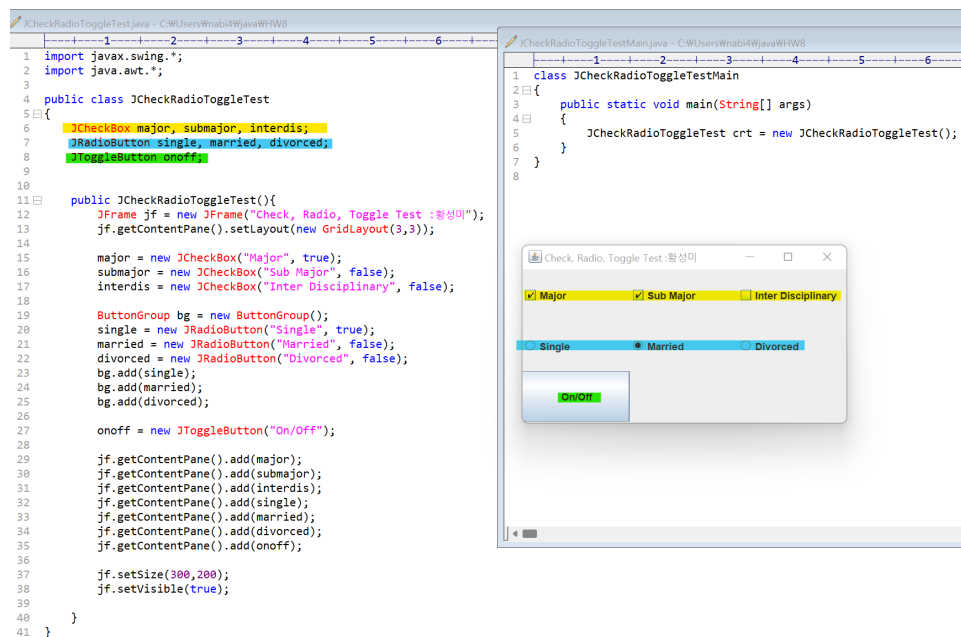
▼ Constructor

JCheckBox(), JCheckBox(String). JCheckBox(String, Boolean),
JRadioButton(String), JRadioButton(Icon), JRadioButton(Icon, String),
JRadioButton(String, CheckboxGroup, Boolean), JRadioButton(Icon, String,
CheckboxGroup, Boolean), JToggleButton()

▼ Method

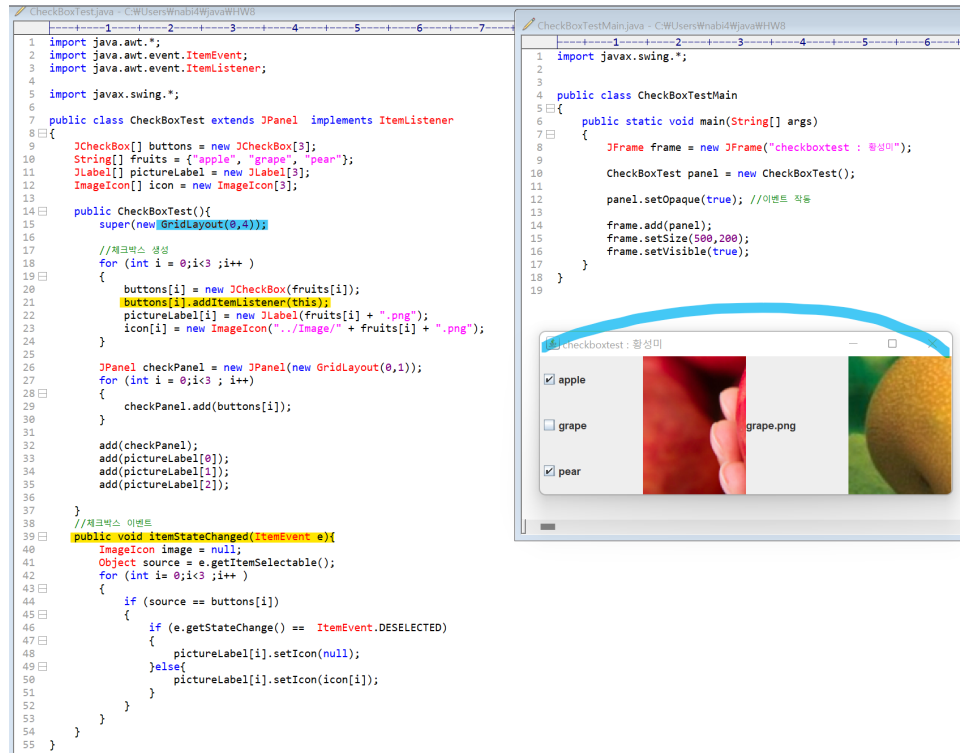
addItemListener()

▼ Lab1



checkbox은 여러개, radiobutton은 한 개만 클릭 가능한 것을 볼 수 있음
true로 설정해두면 미리 체크되어있음

▼ Lab2(Checkbox에 이벤트 핸들링)



for문을 이용해 한 번에 라벨과 이미지를 첨부할 수 있음
 그리드를 0열 4행으로 주어 칸을 분리시킨 후, 한 쪽은 이벤트가 작동하는 체크박스를 나머지 3칸엔
 라벨이 기본 상태였다가 체크박스를 선택하면 사진이 보이도록 코드를 구현함
 이번 코드는 main파일에 frame을 주었음

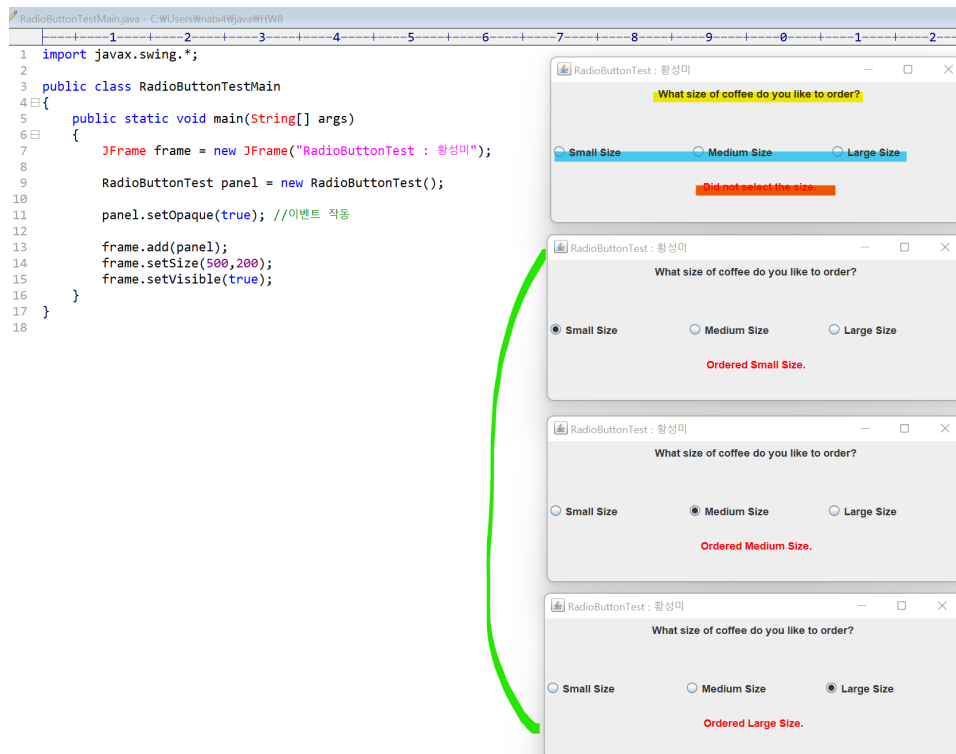
▼ HW(RadioButton test with event)


```

1 import javax.swing.*;
2 import java.awt.*;
3 import java.awt.event.*;
4
5 public class RadioButtonTest extends JPanel implements ActionListener
6 {
7     private JRadioButton small, medium, large;
8     private JLabel text;
9     private JPanel topPanel, radioPanel, textPanel;
10
11     public RadioButtonTest(){
12         setName("RadioButtonTest : 황성미");
13         setSize(300,150);
14         setLayout(new GridLayout(3,1));
15
16         //top
17         topPanel = new JPanel();
18         JLabel topLabel = new JLabel("What size of coffee do you like to order?");
19         topPanel.add(topLabel);
20
21         //radio
22         radioPanel = new JPanel();
23         radioPanel.setLayout(new GridLayout(0,3));
24         ButtonGroup bg = new ButtonGroup();
25         small = new JRadioButton("Small Size", false);
26         medium = new JRadioButton("Medium Size", false);
27         large = new JRadioButton("Large Size", false);
28
29         bg.add(small);
30         bg.add(medium);
31         bg.add(large);
32
33         small.addActionListener(this);
34         medium.addActionListener(this);
35         large.addActionListener(this);
36
37         radioPanel.add(small);
38         radioPanel.add(medium);
39         radioPanel.add(large);
40
41         //text
42         textPanel = new JPanel();
43         text = new JLabel("Did not select the size.");
44         text.setForeground(Color.red);
45
46         //textPanel
47         textPanel = new JPanel();
48         text = new JLabel("Did not select the size.");
49         text.setForeground(Color.red);
50         textPanel.add(text);
51
52         //각 패널의 액션 이벤트 추가
53         add(topPanel);
54         add(radioPanel);
55         add(textPanel);
56         setVisible(true);
57
58         public void actionPerformed(ActionEvent e){
59             if (e.getSource() == small)
60             {
61                 text.setText("Ordered Small Size.");
62             }
63             if (e.getSource() == medium)
64             {
65                 text.setText("Ordered Medium Size.");
66             }
67             if (e.getSource() == large)
68             {
69                 text.setText("Ordered Large Size.");
70             }
71         }
72     }
73 }

```

질문 / 라디오 버튼 / 답 을 각각의 패널을 주어 틀이 깨지지 않게 했고 각각의 버튼을 누를 때 마다 이벤트가 동작하여 글씨가 바뀌도록 함



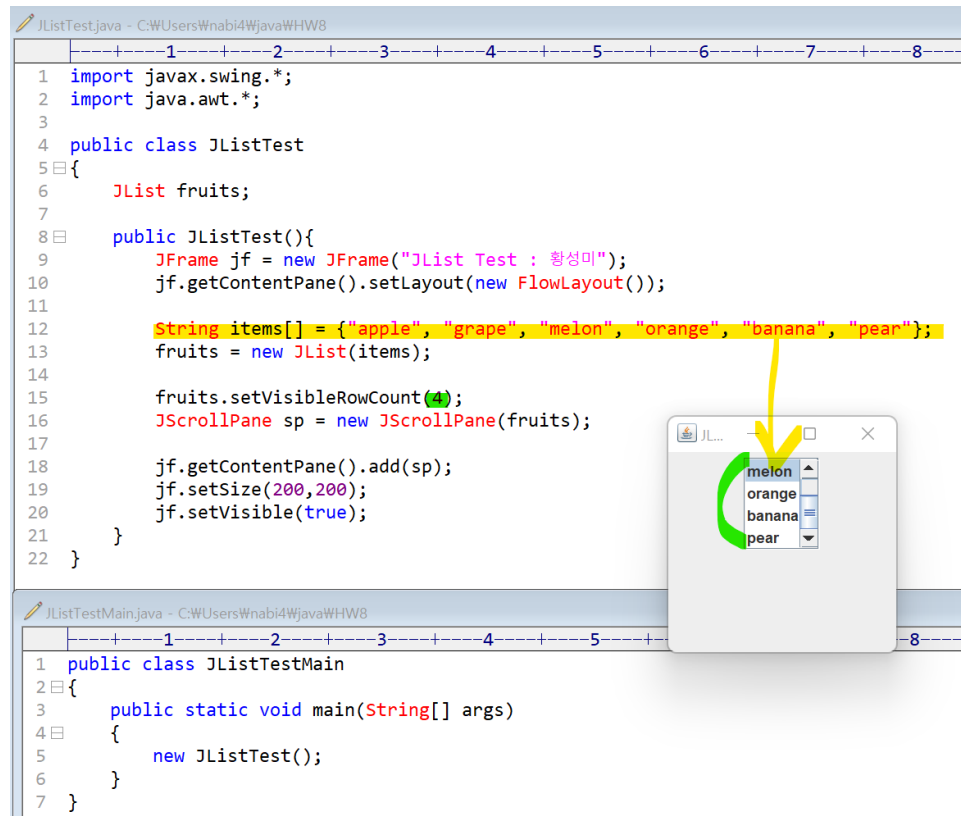
▼ 5.3.5 JList

AWT와 동일

▼ Constructor

JList(), JList(Object[]), JList(Vector), JList(ListModel)

▼ Lab



설정한 개수만큼 리스트들이 보여지는 것을 확인할 수 있음

▼ 5.3.6 JComboBox

AWT의 Choice와 동일

▼ Constructor

JComboBox(), JComboBox(Object[] rows), JComboBox(Vector rows)

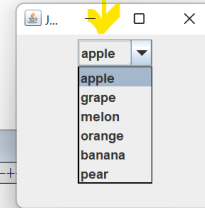
▼ Method

addItem(), insertItemAt(object, index), removeItemAt(index), removeAllItems(),
getItemAt(index), getItemCount(), getSelectedIndex(), getSelectedItem()

▼ Lab

```
JComboBoxTest.java - C:\Users\Wnabi4\java\HW8
1 import javax.swing.*;
2 import java.awt.*;
3
4 public class JComboBoxTest
5 {
6     JComboBox fruits;
7
8     public JComboBoxTest(){
9         JFrame jf = new JFrame("JComboBox Test : 황성미");
10        jf.getContentPane().setLayout(new FlowLayout());
11
12        String items[] = {"apple", "grape", "melon", "orange", "banana", "pear"};
13
14        fruits = new JComboBox(items);
15
16        jf.getContentPane().add(fruits);
17        jf.setSize(200,200);
18        jf.setVisible(true);
19    }
20 }
21

JComboBoxTestMain.java - C:\Users\Wnabi4\java\HW8
1 class JComboBoxTestMain
2 {
3     public static void main(String[] args)
4     {
5         new JComboBoxTest();
6     }
7 }
```



콤보박스 형태로 보여짐