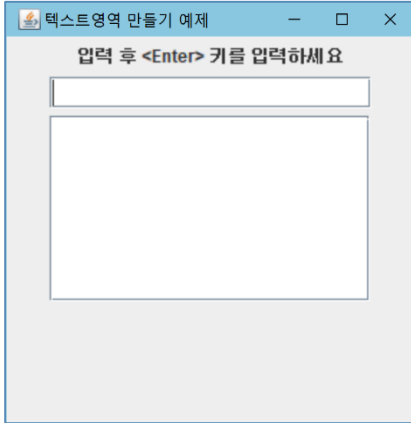
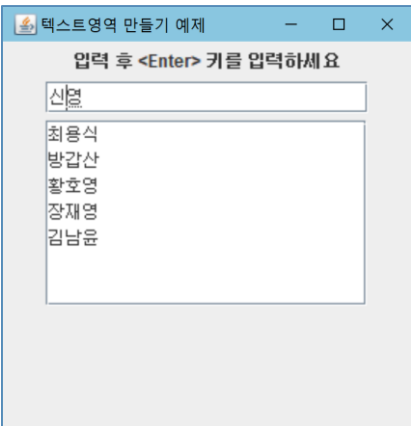


예제 11-9 : JTextArea 컴포넌트 생성



초기화면



텍스트 필드에 입력하고
<Enter> 키를 누르면 텍스트 필드에
입력한 문자열을 텍스트 영역창에 추가

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

```
public class TextAreaEx extends JFrame {
    private JTextField tf = new JTextField(20);
    private JTextArea ta = new JTextArea(7, 20);

    public TextAreaEx() {
        setTitle("텍스트영역 만들기 예제");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        Container c = getContentPane();
        c.setLayout(new FlowLayout());

        c.add(new JLabel("입력 후 <Enter> 키를 입력하세요"));
        c.add(tf);
        c.add(new JScrollPane(ta));

        tf.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                JTextField t = (JTextField)e.getSource();
                ta.append(t.getText() + "\n");
                t.setText("");
            }
        });
        setSize(300,300);
        setVisible(true);
    }

    public static void main(String [] args) {
        new TextAreaEx();
    }
}
```

예제 11-11 : 리스트의 아이템 변경

2

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

public class ListChangeEx extends JFrame {
    private JTextField tf = new JTextField(10);
    private Vector<String> v = new Vector<String>();
    private JList<String> nameList = new JList<String>(v);

    public ListChangeEx() {
        setTitle("리스트 변경 예제");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        Container c = getContentPane();
        c.setLayout(new FlowLayout());

        c.add(new JLabel("이름 입력 후 <Enter> 키"));
        c.add(tf);

        v.add("황기태");
        v.add("이재문");
        nameList.setVisibleRowCount(5);
        nameList.setFixedCellWidth(100);
        c.add(new JScrollPane(nameList));

        setSize(300,300);
        setVisible(true);
    }
}
```

```
// JTextField에 ActionListener 등록. <Enter> 키 처리
tf.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        JTextField t = (JTextField)e.getSource();
        v.add(t.getText());
        t.setText("");
        nameList.setListData(v);
    }
});

public static void main(String [] args) {
    new ListChangeEx();
}
```

- ➔ 예제 11-11~13을 조합하여 다음과 같이 수정
- 1) fruits 배열을 사용하여 콤보박스 생성 (단, apple ~ pear까지 5개만 사용)
 - 2) 콤보박스 옆에 JScrollPane에 리스트 생성하여 배치 (4행, width 100으로 설정)
 - 3) 콤보박스에서 값을 선택하면 리스트에 추가하여 나타나도록 함(ActionListener 사용)
 - 4) 리스트에 추가하면서 해당되는 과일의 그림을 화면에 나타나도록 함 (예제 11-13 활용)

예제 11-12 : 콤보박스 만들기

3

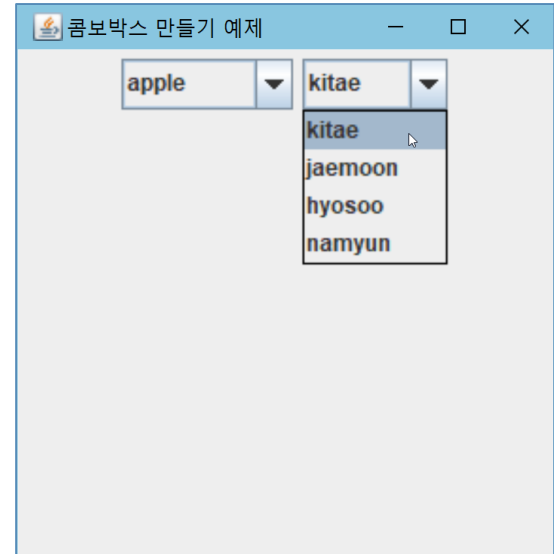
```
import javax.swing.*;
import java.awt.*;
```

```
public class ComboBoxEx extends JFrame {
    private String [] fruits = {"apple", "banana", "kiwi", "mango", "pear",
                                "peach", "berry", "strawberry", "blackberry"};
    private String [] names = {"kitae", "jaemoon", "hyosoo", "namyun"};
    public ComboBoxEx() {
        setTitle("콤보박스 만들기 예제");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        Container c = getContentPane();
        c.setLayout(new FlowLayout());

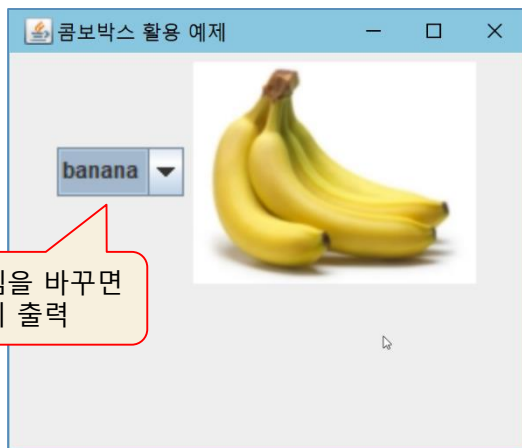
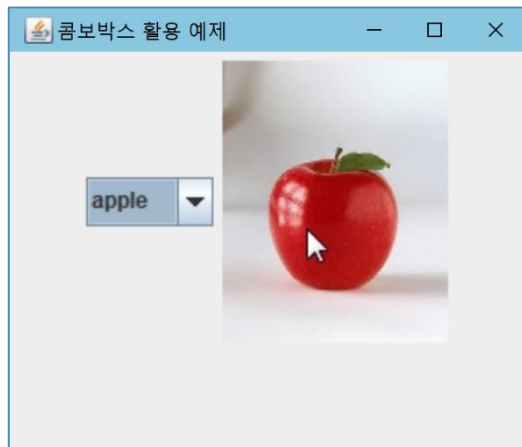
        JComboBox<String> strCombo = new JComboBox<String>(fruits);
        c.add(strCombo);

        JComboBox<String> nameCombo = new JComboBox<String>();
        for(int i=0; i<names.length; i++)
            nameCombo.addItem(names[i]);
        c.add(nameCombo);

        setSize(300,300);
        setVisible(true);
    }
    public static void main(String [] args) {
        new ComboBoxEx();
    }
}
```



예제 11-13 : Action 이벤트를 이용한 콤보박스 활용 예



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

```
public class ComboActionEx extends JFrame {
    private String [] fruits = {"apple", "banana", "kiwi", "mango"};
    private ImageIcon [] images = { new ImageIcon("images/apple.jpg"),
                                     new ImageIcon("images/banana.jpg"),
                                     new ImageIcon("images/kiwi.jpg"),
                                     new ImageIcon("images/mango.jpg")};

    private JLabel imgLabel = new JLabel(images[0]);
    private JComboBox<String> strCombo = new JComboBox<String>(fruits);

    public ComboActionEx() {
        setTitle("콤보박스 활용 예제");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        Container c = getContentPane();
        c.setLayout(new FlowLayout());
        c.add(strCombo);
        c.add(imgLabel);

        strCombo.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                JComboBox<String> cb = (JComboBox<String>)e.getSource();
                int index = cb.getSelectedIndex();
                imgLabel.setIcon(images[index]);
            }
        });
        setSize(300,250);
        setVisible(true);
    }

    public static void main(String [] args) {
        new ComboActionEx();
    }
}
```

예제 11-15 : JSlider와 Change이벤트를 활용한 색깔 다루기

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

public class SliderChangeEx extends JFrame {
    private JLabel colorLabel;
    private JSlider [] sl = new JSlider [3];
    public SliderChangeEx() {
        setTitle("슬라이더와 ChangeEvent 예제");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        Container c = getContentPane();
        c.setLayout(new FlowLayout());
        colorLabel = new JLabel(" SLIDER EXAMPLE ");
        for(int i=0; i<sl.length; i++) {
            sl[i] = new JSlider(JSlider.HORIZONTAL, 0, 255, 128);
            sl[i].setPaintLabels(true);
            sl[i].setPaintTicks(true);
            sl[i].setPaintTrack(true);
            sl[i].setMajorTickSpacing(50);
            sl[i].setMinorTickSpacing(10);
            sl[i].addChangeListener(new MyChangeListener());
            c.add(sl[i]);
        }
        sl[0].setForeground(Color.RED);
        sl[1].setForeground(Color.GREEN);
        sl[2].setForeground(Color.BLUE);

        int r = sl[0].getValue();
        int g = sl[1].getValue();
        int b = sl[2].getValue();

        colorLabel.setOpaque(true);
        colorLabel.setBackground(new Color(r,g,b));
        c.add(colorLabel);
        setSize(300,230);
        setVisible(true);
    }
}
```

```
class MyChangeListener implements ChangeListener {
    public void stateChanged(ChangeEvent e) {
        int r = sl[0].getValue();
        int g = sl[1].getValue();
        int b = sl[2].getValue();
        colorLabel.setBackground(new Color(r,g,b));
    }
}

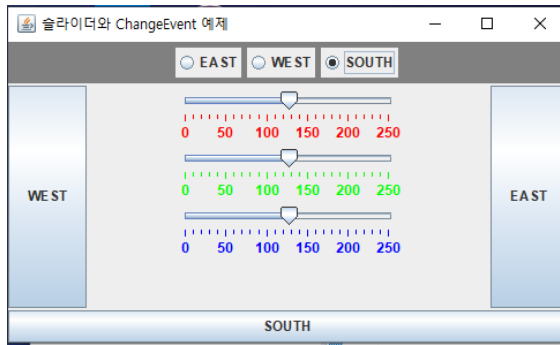
public static void main(String [] args) {
    new SliderChangeEx();
}
```

- ➔ 눈금 숫자는 100 단위로 나오도록 수정하고, 작은 눈금은 20 간격으로 나오도록 수정
- ➔ JLabel을 " SLIDER EXAMPLE " 대신 본인의 " 학번 이름 " 으로 변경하고, 슬라이더에 의해 바탕색이 아닌 글씨의 색을 변경하는 것으로 수정. (바탕색은 사용하지 않고 투명하게 함)

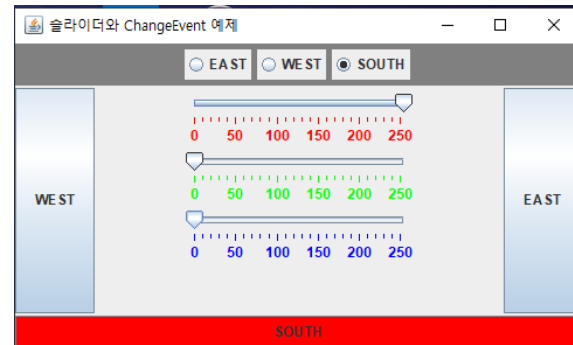
과제 : 다음과 같은 프로그램을 작성하시오.

6

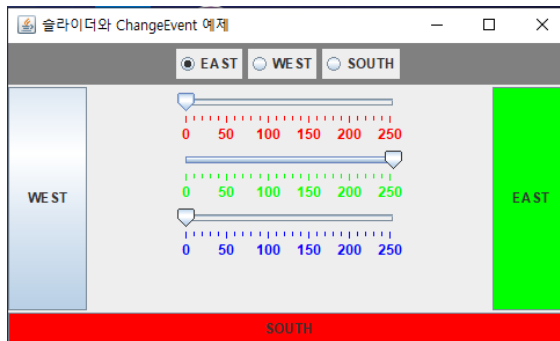
1. 아래와 같은 창을 띄운다.
 - 1) 초기 화면에는 상단에 3개의 라디오 버튼, 동/서/남에 버튼, 중앙에 슬라이더 3개 위치
 - 2) 라디오 버튼 영역은 회색으로 설정(JPanel), SOUTH 라디오 버튼 선택 상태
2. R/G/B 슬라이더를 움직이면 선택 된 버튼의 색을 rgb 값에 의해 변경
 - 다른 버튼 색은 그대로 유지
 - 라디오 버튼에 ItemListener, 슬라이더에 ChangeListener 사용



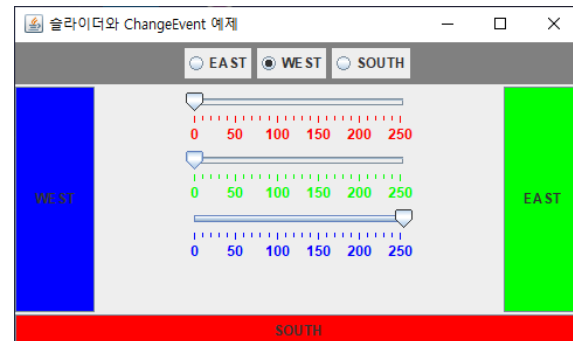
[초기화면]



[SOUTH 선택 상태]



[EAST 선택 상태]



[WEST 선택 상태]