

# 스윙 컴포넌트의 공통 메소드, JComponent의 메소드

3

## 컴포넌트의 모양과 관련된 메소드

```
void setForeground(Color) 전경색 설정  
void setBackground(Color) 배경색 설정  
void setOpaque(boolean) 불투명성 설정  
void setFont(Font) 폰트 설정  
Font getFont() 폰트 리턴
```

## 컴포넌트의 위치와 크기에 관련된 메소드

```
int getWidth() 폭 리턴  
int getHeight() 높이 리턴  
int getX() x 좌표 리턴  
int getY() y 좌표 리턴  
Point getLocationOnScreen() 스크린 좌표상에서의 컴포넌트 좌표  
void setLocation(int, int) 위치 지정  
void setSize(int, int) 크기 지정
```

## 컴포넌트의 상태와 관련된 메소드

```
void setEnabled(boolean) 컴포넌트 활성화/비활성화  
void setVisible(boolean) 컴포넌트 보이기/숨기기  
boolean isVisible() 컴포넌트의 보이는 상태 리턴
```

## 컨테이너를 위한 메소드

```
Component add(Component) 자식 컴포넌트 추가  
void remove(Component) 자식 컴포넌트 제거  
void removeAll() 모든 자식 컴포넌트 제거  
Component[] getComponents() 자식 컴포넌트 리스트 리턴  
Container getParent() 부모 컨테이너 리턴  
Container getTopLevelAncestor() 최상위 부모 컨테이너 리턴
```

# JLabel, 레이블 컴포넌트

6

- JLabel의 용도
  - 텍스트나 이미지를 컴포넌트화 하여 출력하기 위한 목적
- 레이블 컴포넌트 생성
  - JLabel()
    - 빈 레이블 컴포넌트 생성
  - JLabel(Icon image)
    - 이미지만을 가진 레이블 컴포넌트 생성
  - JLabel(String text)
    - 텍스트만을 가진 레이블 컴포넌트 생성
  - JLabel(String text, Icon image, int hAlignment)
    - 텍스트와 이미지, 수평 정렬 값을 가진 레이블 컴포넌트 생성
    - 수평 정렬 값 hAlignment에 사용 가능한 값들 :
      - *SwingConstants* LEFT, CENTER, RIGHT, LEADING or TRAILING

# 레이블 컴포넌트 생성 예

7

- 단순 텍스트 만을 가진 레이블 컴포넌트 생성

```
JLabel textLabel = new JLabel("사랑합니다");
```

- 이미지를 가진 레이블 컴포넌트 생성

- 이미지 파일로부터 이미지를 읽기 위해 ImageIcon 클래스 사용
- 다룰 수 있는 이미지 : png, gif, jpg
  - sunset.jpg의 경로명이 "images/sunset.jpg"인 경우

```
ImageIcon image = new ImageIcon("images/sunset.jpg");  
JLabel imageLabel = new JLabel(image);
```

- 수평 정렬 값을 가진 레이블 컴포넌트 생성

- 텍스트 이미지 모두 출력하고자 하는 경우 수평 정렬 지정

```
ImageIcon image = new ImageIcon("images/sunset.jpg");  
JLabel label = new JLabel("사랑합니다", image, SwingConstants.CENTER);
```



# 이미지 버튼 컴포넌트 만들기

10

- 하나의 버튼에 3 개의 이미지 연결
  - ▣ 사용자의 마우스 접근에 따라 3 개의 이미지 중 선택 출력
- 3 개의 버튼 이미지
  - ▣ 버튼의 보통 상태 때 출력되는 이미지 : `normalIcon`
    - 생성자에 이미지 아이콘 전달
  - ▣ 버튼에 마우스가 올라갈 때 출력되는 이미지 : `rolloverIcon`
    - 이미지 설정 메소드 : `JButton.setRolloverIcon(Icon);`
  - ▣ 버튼을 누른 상태 때 출력되는 이미지 : `pressedIcon`
    - 이미지 설정 메소드 : `JButton.setPressedIcon(Icon)`
- 이미지 아이콘 생성
  - ▣ `new ImageIcon(이미지 경로명);`
    - `new ImageIcon("images/normalIcon.gif");`



```
getContentPane().setLayout(new FlowLayout(FlowLayout.CENTER, 5, 5));
```

```
JButton btnNewButton = new JButton("연락하세요");
```

```
btnNewButton.setIcon(new ImageIcon("images/btn3.JPG"));
```

```
btnNewButton.setRolloverIcon(new ImageIcon("images/btn2.JPG"));
```

```
btnNewButton.setPressedIcon(new ImageIcon("images/btn1.JPG"));
```

```
getContentPane().add(btnNewButton);
```

```
setSize(300,200);
```

```
setVisible(true);
```



# JCheckBox, 체크박스 컴포넌트

13

- 체크박스
  - 선택(selected)과 비선택(deselected)의 두 상태만 가지는 버튼
- 생성자
  - 디폴트는 선택되지 않은 상태
  - JCheckBox ()
    - 텍스트와 이미지가 없는 토글 버튼 생성
  - JCheckBox(Icon icon)
    - 이미지만 가진 토글 버튼 생성
  - JCheckBox(Icon icon, boolean selected)
    - 이미지와 지정된 선택 상태로 생성
  - JCheckBox(String text)
    - 텍스트만 가진 토글 버튼 생성
  - JCheckBox(String text, boolean selected)
    - 텍스트와 지정된 선택 상태로 생성
  - JCheckBox(String text, Icon icon)
    - 텍스트와 이미지 둘 다 가진 토글 버튼 생성
  - JCheckBox(String text, Icon icon, boolean selected)
    - 텍스트와 이미지를 가지고 지정된 선택상태로 생성



체크박스 문자열

체크박스 이미지

# JCheckBox와 Item 이벤트

16

## □ Item 이벤트

### □ 체크 박스가 선택되거나 해제되는 경우에 발생하는 이벤트

- 사용자가 마우스나 키보드로 체크박스를 선택하거나 해제한 경우
- 프로그램에서 체크박스를 선택하거나 해제한 경우

```
JCheckBox c = new JCheckBox("사과");  
c.setSelected(true); // 선택 상태로 변경
```

### □ 체크박스 컴포넌트의 모양을 변경한 후 이벤트 리스너 호출

### □ ItemEvent 객체 생성

## □ ItemListener 인터페이스의 추상 메소드

### □ public void itemStateChanged(ItemEvent e)

## □ ItemEvent의 주요 메소드

### □ int getStateChange()

- 체크 박스의 상태가 선택 상태인지 해제상태인지 리턴
  - `ItemEvent.SELECTED` 또는 `ItemEvent.DESELECTED`

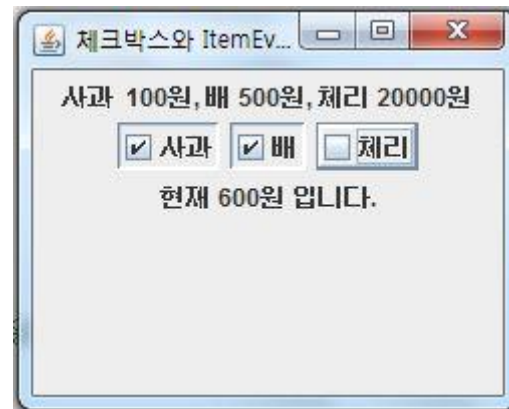
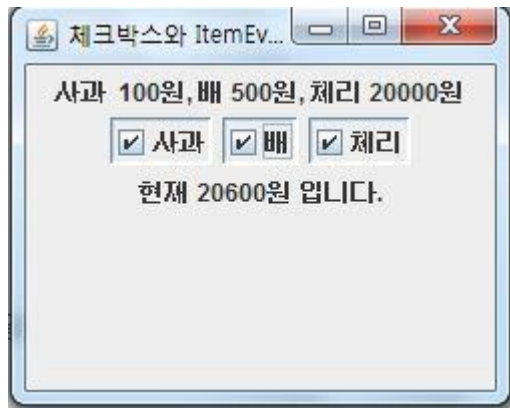
### □ Object getItem()

- 이벤트를 발생시킨 아이템 객체
- 체크박스의 경우 이벤트가 발생한 JCheckBox 객체 리턴



예제 )

아래와 같이 체크박스를 이용하여 가격을 계산하도록 한다.



클래스의 멤버변수(인스턴스변수)로 아래와 같이 둔다.

```
JCheckBox [] fruits = new JCheckBox [3]; // 체크박스 배열  
String [] names = {"사과", "배", "체리"}; // 체크박스 문자열로 사용할 문자열 배열  
JLabel sumLabel; // 계산 합을 출력할 레이블  
int sum = 0; // 가격의 합
```

클래스의 생성자 부분에 아래와 같이 코딩한다.

```
contentPane.setLayout(new FlowLayout());
```

```
// 3개의 체크박스 컴포넌트를 생성하고 컨테넌트팬에 삽입, Item 리스너 등록  
for(int i=0; i<fruits.length; i++) { // fruits.length=3  
    fruits[i] = new JCheckBox(names[i]); // 배열 names[]의 문자열로 체크박스 생성  
    fruits[i].setBorderPainted(true); // 체크박스의 외곽선이 보이도록 설정  
    contentPane.add(fruits[i]); // 체크박스 컴포넌트를 컨테넌트팬에 삽입  
    fruits[i].addItemListener(new MyItemListener()); // 체크박스에 Item 리스너 등록  
}
```

```
class MyItemListener implements ItemListener {  
    // 체크박스의 선택 상태가 변하면 itemStateChanged()가 호출됨  
    public void itemStateChanged(ItemEvent e) {  
        int selected=1; // 1이면 선택 상태, -1이면 해제 상태  
  
        // 선택 상태인지 판별  
        if(e.getStateChange() == ItemEvent.SELECTED)  
            selected = 1; // 선택 상태 의미  
        else  
            selected = -1; // 선택 해제 의미  
  
        // 어떤 체크 박스인지 판별  
        if(e.getItem() == fruits[0]) // 사과 체크박스  
            sum = sum + selected*100;  
        else if(e.getItem() == fruits[1]) // 배 체크박스  
            sum = sum + selected*500;  
        else // 체리 체크박스  
            sum = sum + selected*20000;  
  
        sumLabel.setText("현재 "+sum+"원 입니다."); // 합 출력  
    }  
}
```

# 라디오버튼, JRadioButton

18

- 라디오버튼이란?
  - 여러 버튼으로 그룹을 형성하고, 그룹에 속한 버튼 중 하나만 선택되는 버튼
    - 다른 버튼이 선택되면 이전에 선택된 버튼은 자동으로 해제됨
  - 체크박스와의 차이점
    - 체크박스는 각 체크박스마다 선택/해제가 가능하지만 라디오 버튼은 그룹에 속한 버튼 중 하나만 선택 상태가 됨
  - 이미지를 가진 라디오버튼의 생성 및 다루기는 체크박스와 완전히 동일
- 생성자
  - 디폴트는 선택되지 않은 상태, JCheckBox의 생성자와 동일
  - JRadioButton()
    - 텍스트와 이미지가 없는 토글 버튼 생성
  - JRadioButton(Icon icon)
    - 이미지만 가진 토글 버튼 생성
  - JRadioButton(Icon icon, boolean selected)
    - 이미지와 지정된 선택 상태로 생성
  - JRadioButton(String text)
    - 텍스트만 가진 토글 버튼 생성
  - JRadioButton(String text, boolean selected)
    - 텍스트와 지정된 선택 상태로 생성
  - JRadioButton(String text, Icon icon)
    - 텍스트와 이미지 둘 다 가진 토글 버튼 생성
  - JRadioButton(String text, Icon icon, boolean selected)
    - 텍스트와 이미지를 가지고 지정된 선택상태로 생성



하나의 버튼 그룹에 속한 라디오버튼들

# 라디오 버튼 생성 과정

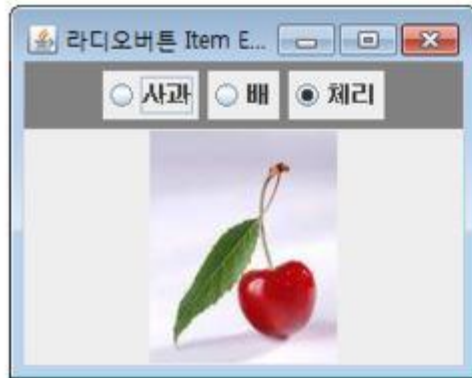
19

1. 버튼 그룹 객체 생성 → `ButtonGroup group = new ButtonGroup();`
2. 라디오버튼 컴포넌트 생성 → `JRadioButton apple= new JRadioButton("사과");`  
`JRadioButton pear= new JRadioButton("배");`  
`JRadioButton cherry= new JRadioButton("체리");`
3. 라디오 버튼을 버튼 그룹에 삽입 → `group.add(apple);`  
`group.add(pear);`  
`group.add(cherry);`
4. 라디오 버튼을 컨테이너에 삽입 → `container.add(apple);`  
`container.add(pear);`  
`container.add(cherry);`

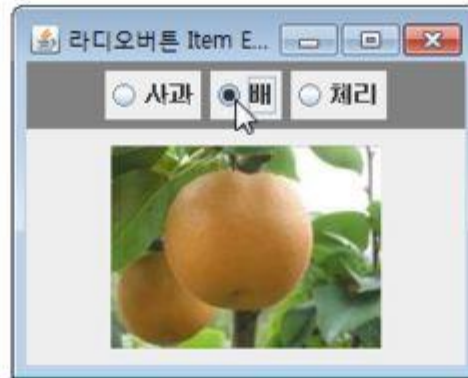


# 예제 실행: ItemEvent 활용, 사진 보여 주기

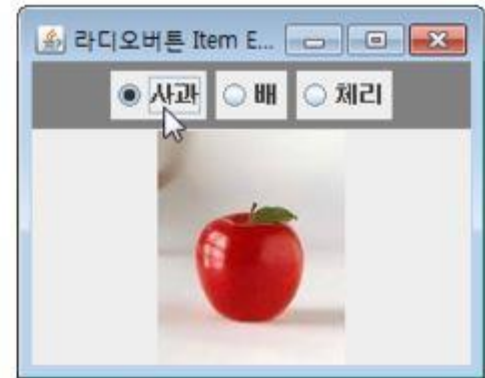
22



초기화면



"배"를 선택한 경우



"사과"를 선택한 경우

## 예제 11-6 : ItemEvent 활용, 사진 보여 주기

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

public class RadioButtonItemEventEx extends JFrame {
    Container contentPane;
    JRadioButton[] radio = new JRadioButton[3];
    String[] text = {"사과", "배", "체리"};
    ImageIcon[] image = {
        new ImageIcon("images/apple.jpg"),
        new ImageIcon("images/pear.jpg"),
        new ImageIcon("images/cherry.jpg")
    };
    JLabel imageLabel = new JLabel();

    RadioButtonItemEventEx() {
        setTitle("라디오버튼 Item Event 예제");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        contentPane = getContentPane();
        contentPane.setLayout(new BorderLayout());

        JPanel panel = new JPanel();
        panel.setBackground(Color.GRAY);

        ButtonGroup g = new ButtonGroup();
        for(int i=0; i<radio.length; i++) {
            radio[i] = new JRadioButton(text[i]);
            g.add(radio[i]);
            panel.add(radio[i]);
            radio[i].addItemListener(new MyItemListener());
        }
        radio[2].setSelected(true);
        contentPane.add(panel, BorderLayout.NORTH);
        contentPane.add(imageLabel, BorderLayout.CENTER);
        imageLabel.setHorizontalAlignment(SwingConstants.CENTER);

        setSize(250,200);
        setVisible(true);
    }
}
```

```
class MyItemListener implements ItemListener {
    public void itemStateChanged(ItemEvent e) {
        if(e.getStateChange() ==
            ItemEvent.DESELECTED)
            return;
        if(radio[0].isSelected())
            imageLabel.setIcon(image[0]);
        else if(radio[1].isSelected())
            imageLabel.setIcon(image[1]);
        else
            imageLabel.setIcon(image[2]);
    }
}

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

setSelected(true) 호출로 인해 Item 이벤트가 발생하며 해당하는 이미지 출력됨

## 텍스트 필드의 주요 메소드

- 텍스트의 편집을 불가능하게 하기
  - ▣ `TextField.setEditable(false);`
- 텍스트 창에 강제로 문자열 출력하기
  - ▣ `TextField.setText("hello");`
- 텍스트 폰트 지정하기
  - ▣ `TextField.setFont(new Font("고딕체", Font.ITALIC, 20));`
- 텍스트 창에 있는 문자열 선택하기
  - ▣ `TextField.select(0, 5);` // 0번 문자에서 5번째까지 문자열 선택

# TextArea, 텍스트영역 컴포넌트

26

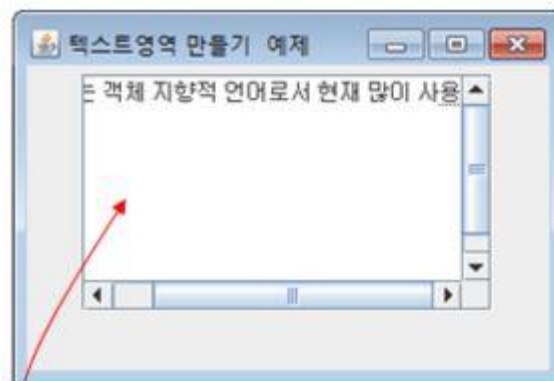
- 텍스트영역이란?
  - ▣ 여러 줄을 입력할 수 있는 텍스트 입력 창
  - ▣ 스크롤바를 지원하지 않는다.
    - JScrollPane 객체에 삽입하는 방식으로 스크롤바 지원
- 생성자
  - ▣ JTextArea()
    - 빈 입력 창 생성
  - ▣ JTextArea(int rows, int cols)
    - 크기가 rows x cols, 빈 입력 창 생성
  - ▣ JTextArea(String text)
    - text 문자열이 출력된 입력 창 생성
  - ▣ JTextArea(String text, int rows, int cols)
    - 크기가 rows x cols, text 문자열이 출력된 입력 창 생성

# 스크롤 가능한 텍스트영역 만들기

27



`new JTextArea("hello", 7, 20);`



`new JScrollPane(new JTextArea("hello", 7, 20));`

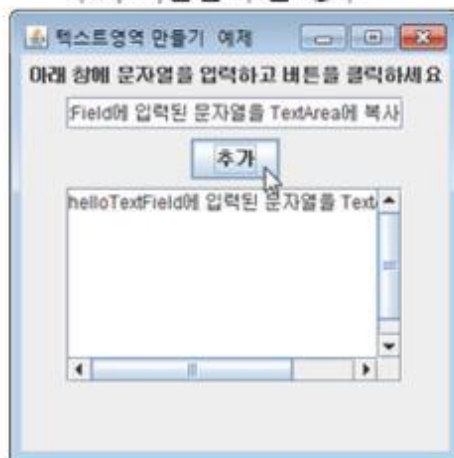


# 예제 11-8 : JTextArea 컴포넌트 생성 예



초기화면

텍스트필드에 입력 후  
추가 버튼을 누른 경우



버튼이 선택되면 ta의  
끝에 tf에 입력된 문자  
열을 추가함

20x7 크기에 "hello"문  
자열을 가진 JTextArea  
컴포넌트 생성

스크롤바를 출력하기  
위해 JTextArea 컴포넌  
트를 JScrollPane에 삽  
입하고 JScrollPane 객  
체를 패널에 삽입

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

public class TextAreaEx extends JFrame {
    Container contentPane;
    TextAreaEx() {
        setTitle("텍스트 영역 만들기 예제");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        contentPane = getContentPane();
        contentPane.add(new MyCenterPanel(),
            BorderLayout.CENTER);
        setSize(300,300);
        setVisible(true);
    }
    class MyCenterPanel extends JPanel {
        JTextField tf;
        JButton btn;
        JTextArea ta;
        MyCenterPanel() {
            tf = new JTextField(20);
            btn = new JButton("추가");
            btn.addActionListener(new ActionListener() {
                public void actionPerformed(ActionEvent e) {
                    ta.append(tf.getText()+"\n");
                }
            });
            ta = new JTextArea("hello", 7, 20);
            add(new JLabel("아래 창에 문자열을 입력하고 버튼을 클릭하세요"));
            add(tf);
            add(btn);
            add(new JScrollPane(ta));
        }
    }
    public static void main(String[] args) {
        new TextAreaEx();
    }
}
```

# JList, 리스트 컴포넌트

29

## □ 리스트 컴포넌트란?

- 객체 리스트를 보여주고 하나 혹은 다수의 아이템을 선택할 수 있게 하는 컴포넌트
- JComboBox와 기본적으로 같은 기능
- JList는 스크롤 지원하지 않음
  - JScrollPane에 JList를 삽입하여 스크롤 가능

## □ 컴포넌트 생성

- JList()
  - 빈 리스트 생성
- JList(Vector listData)
  - 벡트 listData로부터 아이템을 공급받는 구성되는 리스트 생성
  - read-only : 벡트 listData를 추후 수정해도 리스트 변경 안 됨
- JList(Object [] listData)
  - 배열 listData로부터 리스트 아이템을 공급받는 리스트 컴포넌트 생성
  - read-only : 배열 listData를 추후 수정해도 리스트 변경 안 됨

# JComboBox, 콤보박스 컴포넌트

32

- 콤보박스란?
  - 텍스트 필드와 버튼, 그리고 드롭다운 리스트로 구성



- 컴포넌트 생성
  - JComboBox()
    - 비어 있는 콤보박스 생성
  - JComboBox(ComboBoxModel model)
    - model에 의해 아이템을 공급 받는 콤보박스 생성
  - JComboBox(Object [] items)
    - items 배열로부터 아이템을 공급받는 콤보박스 생성
  - JComboBox(Vector items)
    - items 벡터로부터 아이템을 공급받는 콤보박스 생성

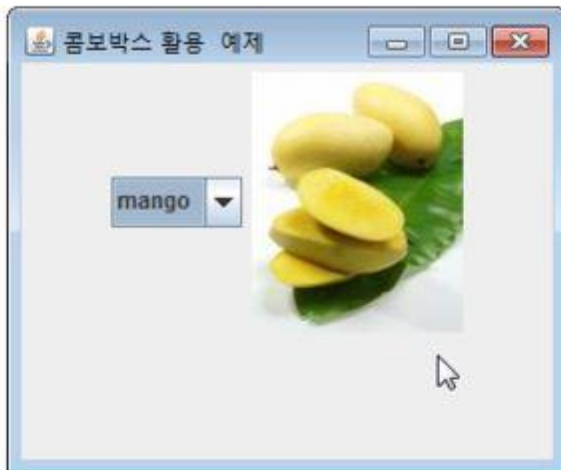
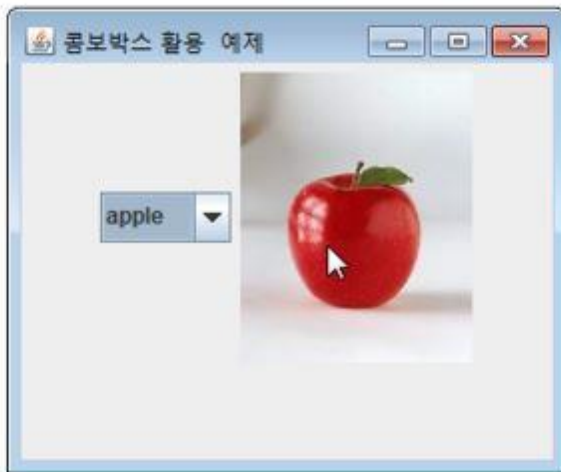
# JComboBox와 Action 이벤트

34

- 콤보박스의 아이템 선택시 Action 이벤트 발생
  - ▣ ActionListener 이용
- 콤보박스의 아이템의 선택시 Item 이벤트 발생
  - ▣ ItemListener 이용
  - ▣ 새로운 아이템이 선택되면 두 번의 Item 이벤트 발생
    - 새로 아이템이 선택되었음을 알리는 Item 이벤트 발생
    - 이전에 선택된 아이템이 해제됨을 알리는 Item 이벤트 발생
  - ▣ 선택 상태인 아이템을 선택하는 경우 Item 이벤트 발생 **않음**
- 현재 선택된 아이템 알아내기
  - ▣ `int JComboBox.getSelectedIndex()`
    - 선택 상태인 아이템의 인덱스 번호 리턴
  - ▣ `Object JComboBox.getSelectedItem()`
    - 선택 상태인 아이템 객체의 레퍼런스 리턴



## 예제 11-11 : Action 이벤트를 이용한 콤보 박스 활용 예



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

public class ComboActionEx extends JFrame {
    Container contentPane;
    String [] fruits = {"apple", "banana", "kiwi", "mango"};
    ImageIcon [] images = {
        new ImageIcon("images/apple.jpg"),
        new ImageIcon("images/banana.jpg"),
        new ImageIcon("images/kiwi.jpg"),
        new ImageIcon("images/mango.jpg") };
    JLabel imgLabel = new JLabel(images[0]);

    ComboActionEx() {
        setTitle("리스트 만들기 예제");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        contentPane = getContentPane();
        contentPane.setLayout(new FlowLayout());

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

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

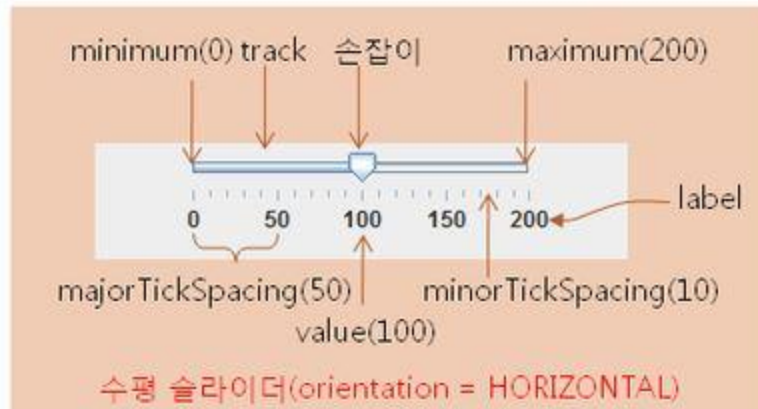


# 슬라이더, JSlider

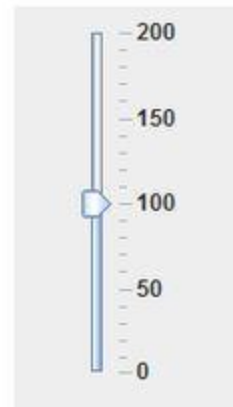
36

## □ 슬라이더란?

- 일정 범위 내에서 마우스로 움직이면서 값을 선택하는 컴포넌트
- 슬라이더 구성 요소



수직 슬라이더  
(orientation = VERTICAL)



## □ 슬라이더 생성

- 슬라이더의 디폴트 값
  - minimum=0, maximum=100, value=50인 수평 슬라이더
- JSlider()
- 디폴트 슬라이더 생성
- JSlider(int orientation)
- orientation의 방향으로 구성된 슬라이더 생성
- JSlider(int min, int max, int val)
- minimum, maximum, value 값이 각각 min, max, val로 초기화된 수평 슬라이더 생성
- JSlider(int orientation, int min, int max, int val)
- minimum, maximum, value 값이 각각 min, max, val로 초기화된 슬라이더 생성. 방향은 orientation

# 슬라이더의 모양 제어

37

- 슬라이더 방향 설정
  - ▣ void setOrientation(int orientation)
    - orientation : JSlider.HORIZONTAL, JSlider.VERTICAL
- 최대 최소 값 설정
  - ▣ void setMaximum(int max)
  - ▣ void setMinimum(int min)
- label 보이기/감추기
  - ▣ void setPaintLabels(boolean b)
    - b가 true이면 label 출력
- tick 보이기/감추기
  - ▣ void setPaintTicks(boolean b)
    - b가 true이면 눈금 출력
- track 보이기/감추기
  - ▣ void setPaintTrack(boolean b)
    - b가 true이면 track 출력
- 큰 눈금 간격 지정
  - ▣ void setMajorTickSpacing(int space)
- 작은 눈금 간격 지정
  - ▣ void setMinorTickSpacing(int space)
- 슬라이더 값 제어
  - ▣ void setValue(int n)
    - n이 슬라이더의 값이 되며 이에 따라 슬라이더의 손잡이 위치가 변경된다.

# JSlider와 Change 이벤트

39

- Change 이벤트
  - ▣ 이벤트 소스 컴포넌트의 값이 변경되었을 때
  - ▣ 리스너
    - ChangeListener 인터페이스
  - ▣ ChangeEvent와 ChangeListener 클래스가 속한 패키지
    - javax.swing.event 패키지에 정의
- ChangeListener의 메소드
  - ▣ public void stateChanged(ChangeEvent e)
- JSlider의 경우
  - ▣ value가 변경될 때 Change 이벤트 발생
    - 사용자가 슬라이더의 value 값을 변경하는 동안 계속 발생
    - 응용프로그램에서 JSlider.setValue(int n)을 호출하여 value 값을 변경할 때

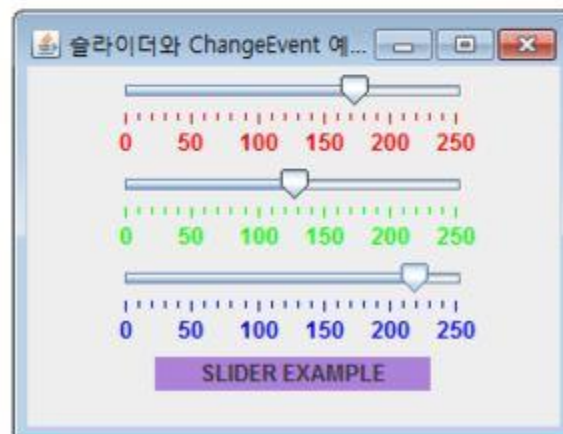
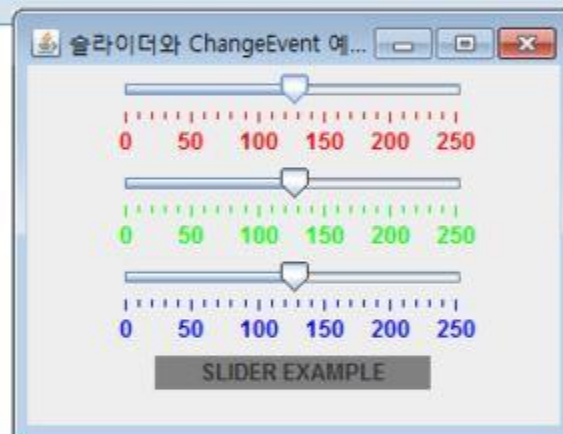
## 예제 11-13 : JSlider와 Change이벤트를 활용한 색깔 다루기

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

public class SliderChangeEx extends JFrame {
    Container contentPane;
    JLabel colorLabel;
    JSlider [] sl = new JSlider [3];
    SliderChangeEx() {
        setTitle("슬라이더와 ChangeEvent 예제");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        contentPane = getContentPane();
        contentPane.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());
            contentPane.add(sl[i]);
        }
        sl[0].setForeground(Color.RED);
        sl[1].setForeground(Color.GREEN);
        sl[2].setForeground(Color.BLUE);
        colorLabel.setOpaque(true);
        colorLabel.setBackground(
            new Color(sl[0].getValue(), sl[1].getValue(), sl[2].getValue()));
        contentPane.add(colorLabel);
        setSize(300,300);
        setVisible(true);
    }
}
```

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

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



예제

다음과 같은 프로그램을 만들자

힌트)

The screenshot shows a Windows application window titled "Yoon Mi\_young". The window contains a form for breaking down a total amount into Korean currency units. At the top, there is a text box labeled "금액" (Amount) containing the value "543200". To the right of this text box is a button labeled "계산" (Calculate). Below the "금액" text box, there are nine rows, each representing a currency unit: "오만원" (50,000 won), "만원" (10,000 won), "오천원" (5,000 won), "천원" (1,000 won), "오백원" (500 won), "백원" (100 won), "오십원" (50 won), "십원" (10 won), and "일원" (1 won). Each row has a text box for the count and a checkbox to the right. The counts are: 10 for 오만원, 4 for 만원, 3 for 천원, 2 for 백원, and 0 for 일원. The checkboxes for 오만원, 만원, 오천원, 천원, 오백원, 백원, 오십원, and 십원 are checked, while the checkbox for 일원 is not checked.

금액	543200	계산
오만원	10	<input checked="" type="checkbox"/>
만원	4	<input checked="" type="checkbox"/>
오천원		<input checked="" type="checkbox"/>
천원	3	<input checked="" type="checkbox"/>
오백원		<input checked="" type="checkbox"/>
백원	2	<input checked="" type="checkbox"/>
오십원		<input checked="" type="checkbox"/>
십원		<input checked="" type="checkbox"/>
일원	0	<input type="checkbox"/>



10개의 레이블을 순서대로 클릭하는 간단한 게임을 만들어 보자  
0~9까지의 숫자를 가진 레이블 컴포넌트를 10개 만들고 이들을  
프레임 내의 랜덤한 위치에 배치한다. 사용자가 9~0까지 순서대로  
클릭하면 다시 10개의 레이블을 랜덤한 위치에 배치한다.

클릭된 레이블 컴포넌트는 화면에 보이지 않게 하며 번호 순서대로  
클릭되게 하라.

