

제 11 장 스윙 컴포넌트와 이벤트 핸들링

스윙컴포넌트의 공통 메소드. JComponent의 메소드

3

컴포넌트의 모양과 관련된 메소드

```
void setForeground(Color) 전경색설정  
void setBackground(Color) 배경색설정  
void setOpaque(boolean) 불투명성설정  
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
```

스윙 컴포넌트의 공통 메소드 확인 사례

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

public class SwingAPIEx extends JFrame {
    Container contentPane;
    JLabel la;
    JButton b1, b2, b3, b4;

    SwingAPIEx() {
        setTitle("Swing 공통 메소드 예제");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        contentPane = getContentPane();
        contentPane.setLayout(new FlowLayout());

        b1 = new JButton("위치와 크기 정보");
        b1.addActionListener(new MyButtonListener());
        contentPane.add(b1);

        b2 = new JButton("모양 정보");
        b2.setOpaque(true);
        b2.setForeground(Color.MAGENTA);
        b2.setBackground(Color.YELLOW);
        b2.setFont(new Font("고딕체", Font.ITALIC, 20));
        b2.addActionListener(new MyButtonListener());
        contentPane.add(b2);

        b3 = new JButton("작동하지 않는 버튼");
        b3.setEnabled(false);
        b3.addActionListener(new MyButtonListener());
        contentPane.add(b3);

        b4 = new JButton("숨기기/보이기");
        b4.addActionListener(new MyButtonListener());
        contentPane.add(b4);

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

```
class MyButtonListener implements ActionListener {
    public void actionPerformed(ActionEvent e) {
        Object source = e.getSource();
        if(source == b1) {
            System.out.println("버튼의 위치와 크기");
            System.out.println("위치 = (" + b1.getX() + "," + b1.getY() + ")");
            System.out.println("크기 = (" + b1.getWidth() + "x"
                               + b1.getHeight() + ")");

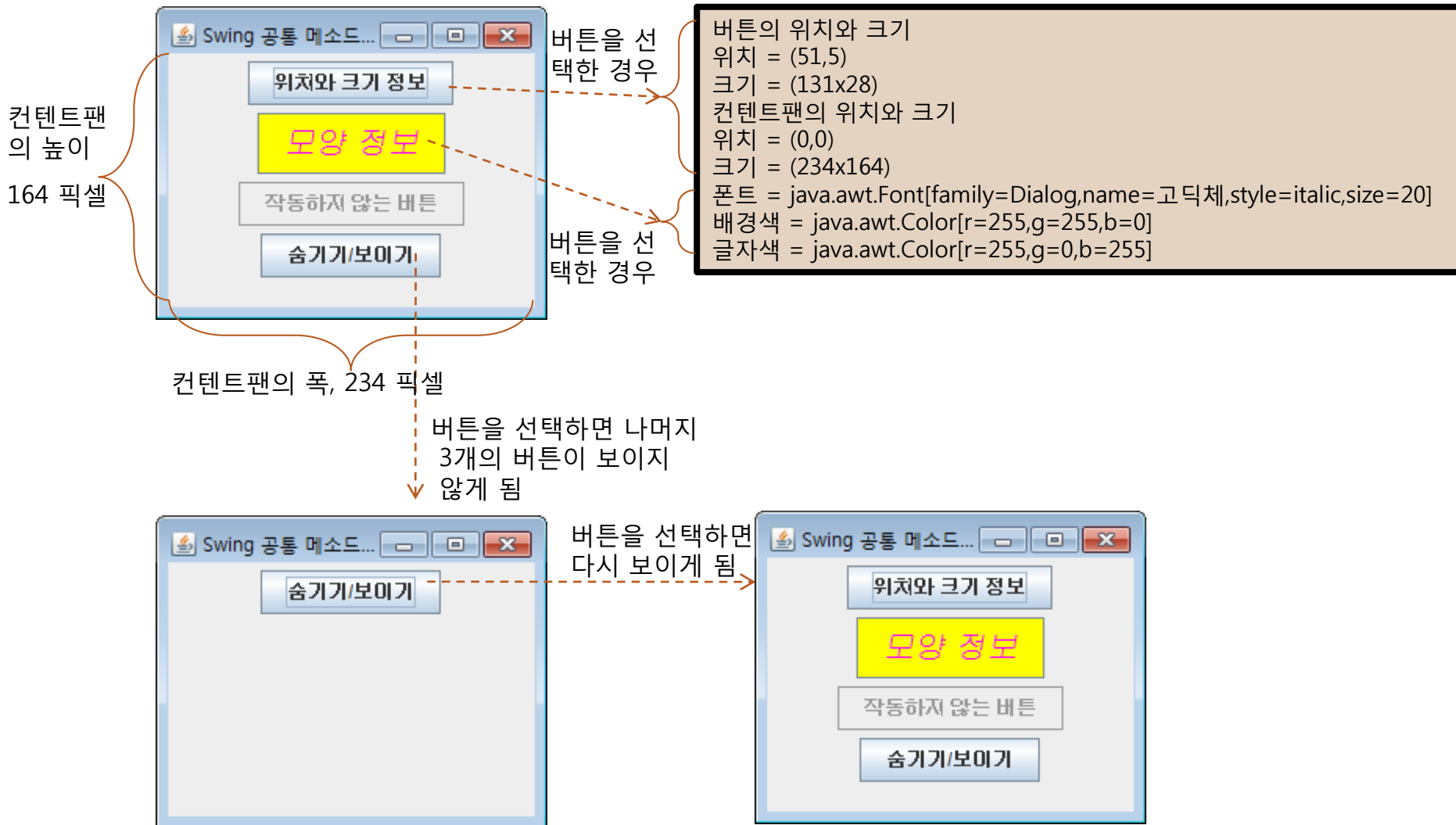
            JPanel c = (JPanel)b2.getParent();
            System.out.println("컨텐츠판의 위치와 크기");
            System.out.println("위치 = (" + c.getX() + "," + c.getY() + ")");
            System.out.println("크기 = (" + c.getWidth() + "x"
                               + c.getHeight() + ")");
        }
        else if(source == b2) {
            System.out.println("폰트 = " + b2.getFont());
            System.out.println("배경색 = " + b2.getBackground());
            System.out.println("글자색 = " + b2.getForeground());
        }
        else {
            if(b1.isVisible()) {
                b1.setVisible(false);
                b2.setVisible(false);
                b3.setVisible(false);
            }
            else {
                b1.setVisible(true);
                b2.setVisible(true);
                b3.setVisible(true);
            }
        }
    }
}

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

실행: 스윙 컴포넌트의 공통 요소

5

콘솔 창에 출력된 내용



JLabel, 레이블 컴포넌트

6

- JLabel의 용도
 - ▣ 텍스트나 이미지를 컴포넌트화 하기 위한 목적
- 레이블 컴포넌트 생성
 - ▣ JLabel()
 - 텍스트나 이미지 정보가 없는 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
 - 수평정렬 값인 hAlignment로 사용가능한 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 종류
 - png, gif, jpg
- ▣ sunset.jpg의 경로명이 "images/sunset.jpg"인 경우

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

□ 수평정렬 값을 가진 레이블 컴포넌트 생성

- ▣ 수평정렬 값으로 사용되는 값을 생성자의 3 번째 인자로 지정
 - 텍스트 이미지 모두 출력하고자 하는 경우 수평정렬 값 지정

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

예제 11-1 : JLabel 컴포넌트 생성 예

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

public class LabelEx extends JFrame {
    Container contentPane;
    LabelEx() {
        setTitle("레이블 예제");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        contentPane = getContentPane();
        contentPane.setLayout(new FlowLayout());
        JLabel textLabel = new JLabel("사랑합니다.");
        ImageIcon beauty = new ImageIcon("images/beauty.jpg");
        JLabel imageLabel = new JLabel(beauty);
        ImageIcon normalIcon = new
            ImageIcon("images/normalIcon.gif");
        JLabel label = new JLabel("보고싶으면 전화하세요",
            normalIcon, SwingConstants.CENTER);

        contentPane.add(textLabel);
        contentPane.add(imageLabel);
        contentPane.add(label);

        setSize(400,600);
        setVisible(true);
    }
    public static void main(String [] args) {
        new LabelEx();
    }
}
```



JButton, 버튼 컴포넌트

9

- 버튼 컴포넌트
 - ▣ 버튼 모양의 컴포넌트
 - ▣ 버튼은 클릭될 때 Action 이벤트를 발생시킴
- 버튼 컴포넌트 생성
 - ▣ JButton()
 - 텍스트나 이미지 아이콘을 가지지 않은 디폴트 버튼 생성
 - ▣ JButton(Icon icon)
 - 이미지 아이콘만을 가진 버튼 생성
 - ▣ JButton(String text)
 - 텍스트만을 가진 버튼 생성
 - ▣ JButton(String text, Icon icon)
 - 텍스트와 이미지 아이콘을 모두 가진 버튼 생성
- 버튼 컴포넌트 생성 예
 - ▣ "hello" 문자열을 가진 버튼 컴포넌트 생성 예



버튼 이미지

버튼 문자열

```
JButton btn = new JButton("hello");
```

이미지를 가진 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`
 - 생성자 호출 시에 주어진 이미지 아이콘
 - ▣ 버튼 위에 마우스가 올라갈 때 출력되는 이미지 : `rolloverIcon`
 - 이미지 설정 메소드 : `JButton.setRolloverIcon(icon);`
 - ▣ 마우스 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");`

예제 11-2 : 3 개의 이미지 아이콘을 가진 버튼 만들기

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

public class ButtonImageEx extends JFrame {
    Container contentPane;
    ButtonImageEx() {
        setTitle("버튼에 아이콘 달기 예제");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        contentPane = getContentPane();
        contentPane.setLayout(new FlowLayout());

        ImageIcon normalIcon = new
            ImageIcon("images/normalIcon.gif");
        ImageIcon rolloverIcon = new
            ImageIcon("images/rolloverIcon.gif");
        ImageIcon pressedIcon = new
            ImageIcon("images/pressedIcon.gif");

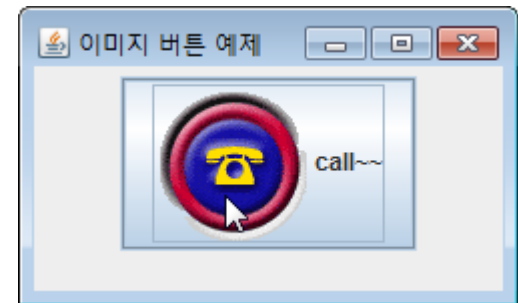
        JButton btn = new JButton("call~~", normalIcon);
        btn.setRolloverIcon(rolloverIcon);
        btn.setPressedIcon(pressedIcon);
        contentPane.add(btn);

        setSize(250,200);
        setVisible(true);
    }
    public static void main(String [] args) {
        new ButtonImageEx();
    }
}
```

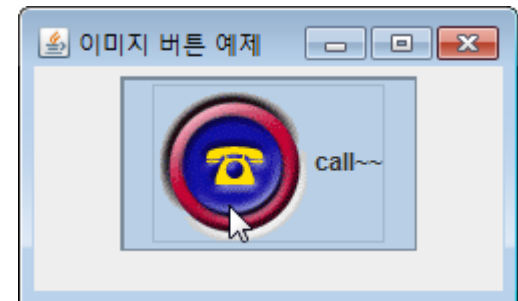
보통 상태에 있는 동안
(normalIcon.gif)



마우스가 버튼 위에
올라간 경우
(rolloverIcon.gif)



마우스가 눌려진 순간
(pressedIcon.gif)



레이블과 버튼의 정렬(Alignment)

12

□ 수평 정렬

- 컴포넌트 영역 내에 이미지와 텍스트의 수평상의 위치 결정
- `void setHorizontalAlignment(int align)`

버튼 영역



왼쪽정렬

`SwingConstants.LEFT`



중앙정렬

`SwingConstants.CENTER`



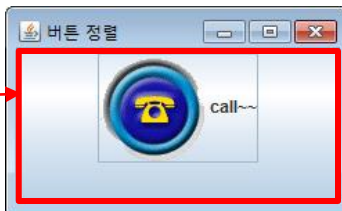
오른쪽정렬

`SwingConstants.RIGHT`

□ 수직 정렬

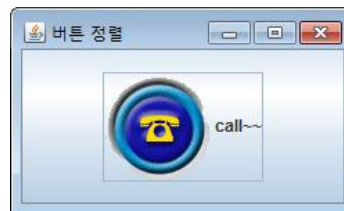
- 컴포넌트 영역 내에 콘텐츠(이미지와 텍스트)의 수직상의 위치
- `void setVerticalAlignment(int align)`

버튼 영역



위쪽정렬

`SwingConstants.TOP`



중앙정렬

`SwingConstants.CENTER`



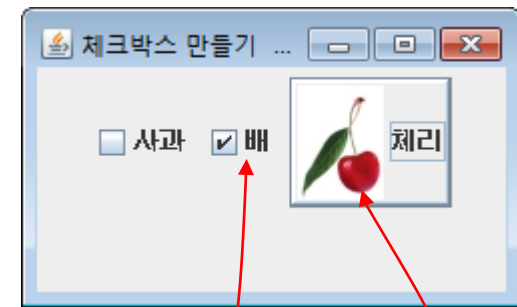
아래쪽정렬

`SwingConstants.BOTTOM`

JCheckBox, 체크박스 컴포넌트

13

- 체크박스
 - ▣ 선택(selected)과 비선택(deselected)의 두 상태만을 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

체크 박스 생성

14

□ 텍스트 정보만을 가진 체크 박스 생성

- "사과" 텍스트를 가진 체크박스 생성

```
JCheckBox c = new JCheckBox("사과");
```

- "배" 텍스트를 가지고 선택상태로 체크박스 생성

```
JCheckBox c = new JCheckBox("배", true);
```

- 체크 박스 모양 ☒ 이 명료하게 출력되고 사용자는 이것을 체크

□ 이미지 아이콘을 가진 체크 박스 생성 예

- 체크 박스 모양 ☒ 이 출력되지 않음

- 따로 선택상태를 표현하는 이미지 아이콘을 설정하여야 함

- cherry.jpg 이미지와 "체리" 텍스트를 가진 체크 박스 생성 예

- 선택 상태의 이미지를 위해 selectedCherry.jpg를 사용하였음

```
ImageIcon cherryIcon = new ImageIcon("images/cherry.jpg");  
ImageIcon selectedCherryIcon = new ImageIcon("images/selectedCherry.jpg");  
JCheckBox cherry = new JCheckBox("체리", cherryIcon);  
cherry.setSelectedIcon(selectedCherryIcon);
```

예제 11-3 : 체크박스 생성 예

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

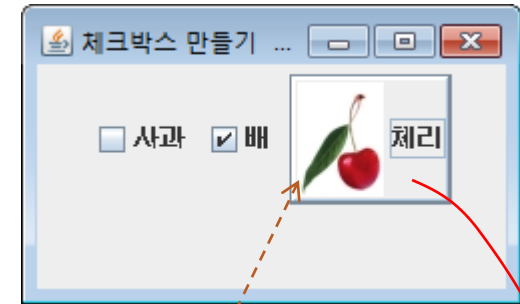
public class CheckBoxEx extends JFrame {
    Container contentPane;
    CheckBoxEx() {
        setTitle("체크박스 만들기 예제");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        contentPane = getContentPane();
        contentPane.setLayout(new FlowLayout());

        ImageIcon cherryIcon = new ImageIcon("images/cherry.jpg");
        ImageIcon selectedCherryIcon =
            new ImageIcon("images/selectedCherry.jpg");

        JCheckBox apple = new JCheckBox("사과");
        JCheckBox pear = new JCheckBox("배", true);
        JCheckBox cherry = new JCheckBox("체리", cherryIcon);
        cherry.setBorderPainted(true);
        cherry.setSelectedIcon(selectedCherryIcon);

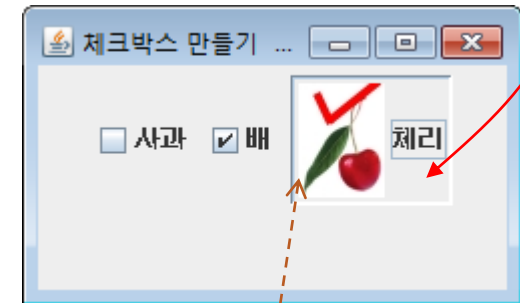
        contentPane.add(apple);
        contentPane.add(pear);
        contentPane.add(cherry);

        setSize(250,150);
        setVisible(true);
    }
    public static void main(String [] args) {
        new CheckBoxEx();
    }
}
```



cherry.jpg(선택되지 않은 상태)

체크 박스를
선택하면



selectedCherry.jpg(선택된 상태)

JCheckBox와 Item 이벤트

16

□ Item 이벤트

▣ 체크 박스가 선택되거나 해제되는 각 경우에 발생하는 이벤트

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

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

▣ 이 이벤트가 발생하면 이미 체크박스 컴포넌트의 상태 변경된 후

▣ ItemEvent 객체 생성

□ ItemListener 인터페이스의 추상 메소드

▣ protected void itemStateChanged(ItemEvent e)

□ ItemEvent의 주요 메소드

▣ int getStateChange()

- 체크박스의 상태가 선택 상태인지 비선택상태인지 리턴
 - *ItemEvent.SELECTED* 또는 *ItemEvent.DESELECTED*

▣ Object getItem()

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

예제 11-4 : ItemEvent를 활용하여 가격 합산하기

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

public class CheckBoxItemEventEx extends JFrame {
    Container contentPane;
    JCheckBox [] fruits = new JCheckBox [3];
    String [] names = {"사과", "배", "체리"};
    JLabel sumLabel;
    int sum = 0;

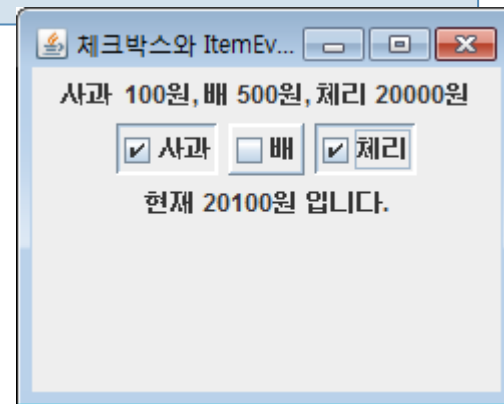
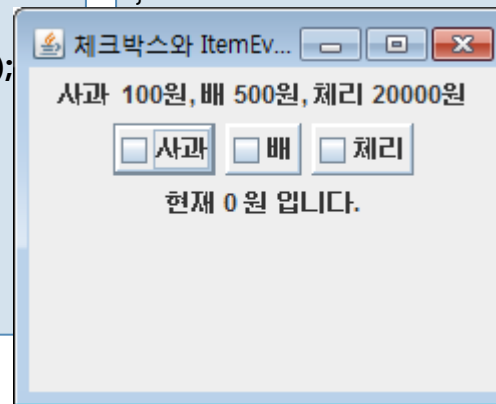
    CheckBoxItemEventEx() {
        setTitle("체크박스와 ItemEvent 예제");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        contentPane = getContentPane();
        contentPane.setLayout(new FlowLayout());

        contentPane.add(
            new JLabel("사과 100원, 배 500원, 체리 20000원"));
        for(int i=0; i<fruits.length; i++) {
            fruits[i] = new JCheckBox(names[i]);
            fruits[i].setBorderPainted(true);
            contentPane.add(fruits[i]);
            fruits[i].addItemListener(new MyItemListener());
        }
        sumLabel = new JLabel("현재 0 원 입니다.");
        contentPane.add(sumLabel);
        setSize(250,200);
        setVisible(true);
    }
}
```

```
class MyItemListener implements ItemListener {
    public void itemStateChanged(ItemEvent e) {
        int selected=1;
        if(e.getStateChange() == ItemEvent.SELECTED)
            selected = 1;
        else
            selected = -1;
        if(e.getItem() == fruits[0])
            sum = sum + selected*100;
        else if(e.getItem() == fruits[1])
            sum = sum + selected*500;
        else
            sum = sum + selected*20000;

        sumLabel.setText("현재 "+sum+"원 입니다.");
    }
}

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



라디오 버튼, JRadioButton

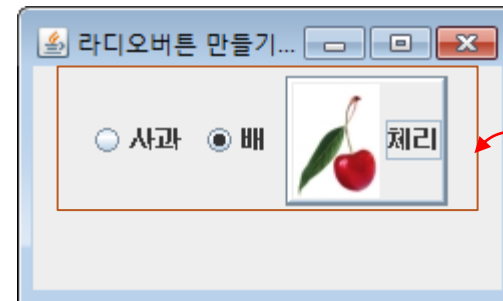
18

□ 라디오버튼이란?

- 여러 버튼으로 그룹을 형성하고, 그룹에 속한 버튼 중 하나만 선택 상태가 되는 버튼
 - 다른 버튼이 선택되면 이전에 선택된 버튼은 자동으로 해제됨
- 체크박스와의 차이점
 - 체크 박스는 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)
 - 텍스트 만을 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



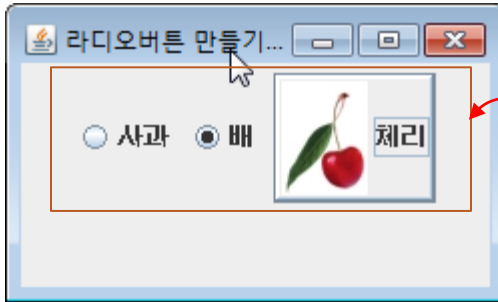
하나의 버튼 그룹에 속한 라디오버튼들

라디오 버튼 생성 과정

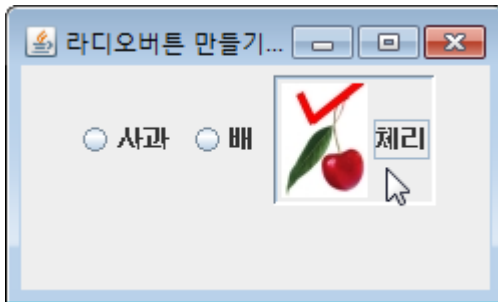
19

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 11-5 : 라디오버튼 생성 예



초기 상태(배가 선택된 상태)



체리가 선택된 상태

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

```
public class RadioButtonEx extends JFrame {
    Container contentPane;
    RadioButtonEx() {
        setTitle("라디오버튼 만들기 예제");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        contentPane = getContentPane();
        contentPane.setLayout(new FlowLayout());

        ImageIcon cherryIcon = new ImageIcon("images/cherry.jpg");
        ImageIcon selectedCherryIcon = new
            ImageIcon("images/selectedCherry.jpg");

        ButtonGroup g = new ButtonGroup();
        JRadioButton apple = new JRadioButton("사과");
        JRadioButton pear = new JRadioButton("배", true);
        JRadioButton cherry = new JRadioButton("체리", cherryIcon);
        cherry.setBorderPainted(true);
        cherry.setSelectedIcon(selectedCherryIcon);
        g.add(apple);
        g.add(pear);
        g.add(cherry);

        contentPane.add(apple);
        contentPane.add(pear);
        contentPane.add(cherry);

        setSize(250,150);
        setVisible(true);
    }
    public static void main(String [] args) {
        new RadioButtonEx();
    }
}
```

예제 11-6 : ItemEvent를 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();
}
```

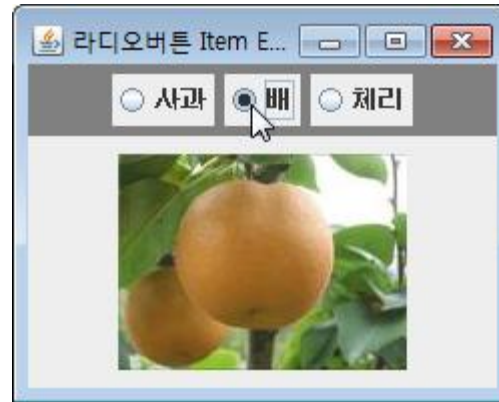
setSelected(true) 메소드 호출로 인해 Item 이벤트가 발생하여 해당하는 이미지 출력됨

예제 실행: ItemEvent 활용, 사진 보여 주기

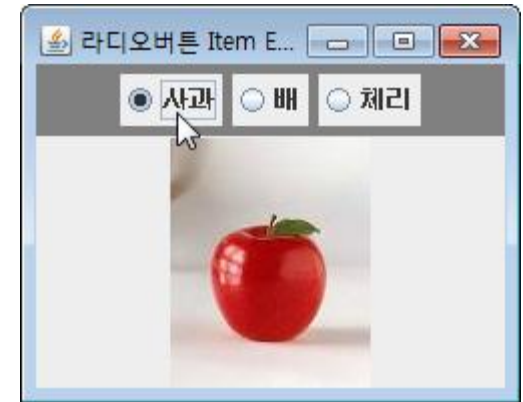
22



초기화면



"배"를 선택한 경우



"사과"를 선택한 경우

JTextField, 텍스트필드 컴포넌트

23

- 텍스트 필드란?
 - ▣ 한 줄 짜리 텍스트(문자열) 입력 창을 구현한 컴포넌트
 - ▣ 텍스트 입력 도중 <Enter>키가 입력되면 Action 이벤트 발생
 - ▣ 입력 가능한 문자 개수와 입력 창의 크기는 서로 다르다.
- 생성자
 - ▣ JTextField()
 - 빈 텍스트 입력 창 생성
 - ▣ JTextField(int columns)
 - 입력 창의 크기가 columns 개, 빈 텍스트 입력 창 생성
 - ▣ JTextField(String text)
 - text 문자열로 초기화된 텍스트 입력 창 생성
 - ▣ JTextField(String text, int columns)
 - 입력 창의 크기가 columns 개이고, text 문자열이 초기 출력된 텍스트 입력 창 생성

예제 11-7 : 간단한 텍스트 필드 만들기

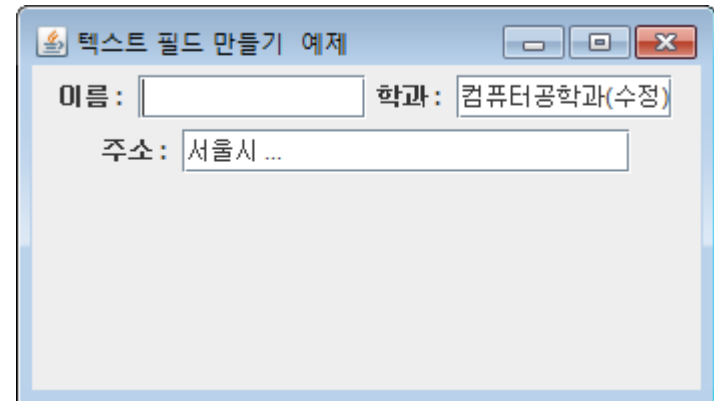
```
import javax.swing.*;
import java.awt.*;

public class TextFieldEx extends JFrame {
    Container contentPane;
    TextFieldEx() {
        setTitle("텍스트 필드 만들기 예제");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        contentPane = getContentPane();
        contentPane.setLayout(new FlowLayout());

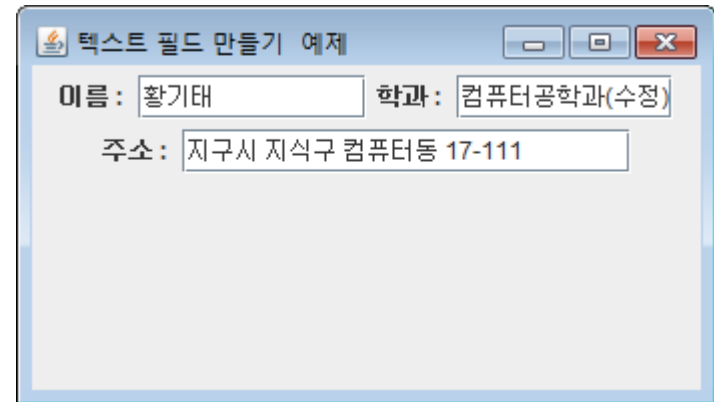
        contentPane.add(new JLabel("이름 : "));
        contentPane.add(new JTextField(10));
        contentPane.add(new JLabel("학과 : "));
        contentPane.add(new JTextField("xxx 공학과"));
        contentPane.add(new JLabel("주소 : "));
        contentPane.add(new JTextField("서울시 ...", 20));

        setSize(350,200);
        setVisible(true);
    }

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



초기화면



사용자가 입력한 경우

텍스트 필드의 주요 메소드

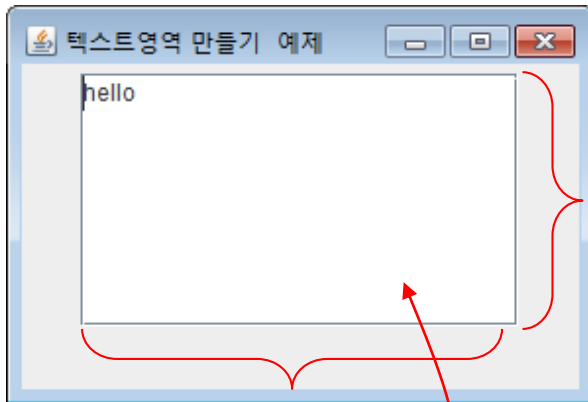
25

- 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

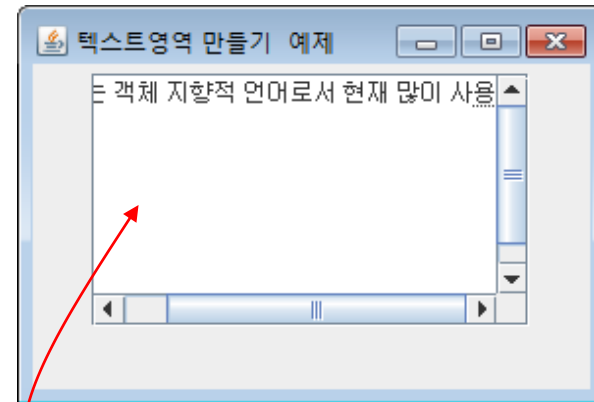
26

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



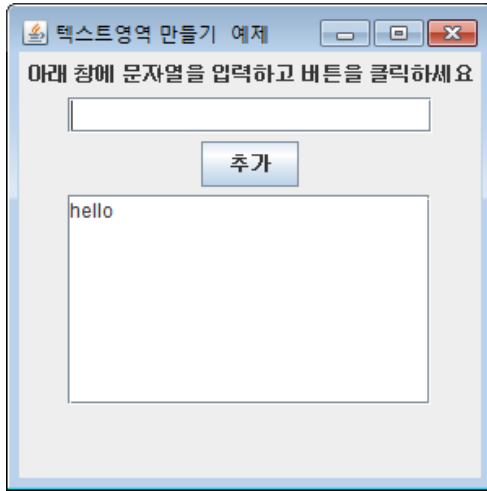
20 문자

```
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
영역 끝에 tf에 입력된
문자열을 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

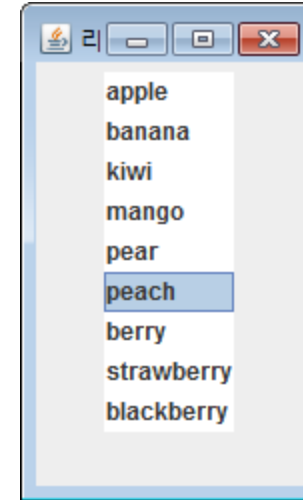
29

- 리스트란?
 - ▣ 사용자에게 하나 이상의 객체 리스트를 보여주고 하나 혹은 다수의 아이템을 선택할 수 있게 하는 컴포넌트
 - ▣ JComboBox와 기본적으로 같은 기능
 - ▣ JList는 스크롤링을 지원하지 않음
 - JScrollPane에 JList를 삽입하여 스크롤링 가능
- 컴포넌트 생성
 - ▣ JList()
 - 비어있는 리스트 생성
 - ▣ JList(Vector listData)
 - 벡트인 listData로부터 리스트 아이템을 공급받는 리스트 컴포넌트 생성
 - read-only : 벡트 listData를 수정하여도 JList를 변경할 수 없음
 - ▣ JList(Object [] listData)
 - 배열 listData로부터 리스트 아이템을 공급받는 리스트 컴포넌트 생성
 - read-only : 배열 listData를 수정하여도 JList를 변경할 수 없음

JList를 생성하는 방법

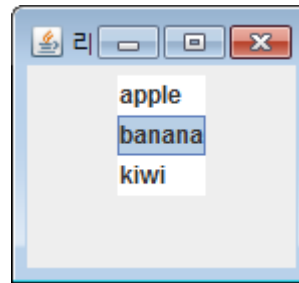
1. 객체 배열로 리스트 데이터를 제공하는 방법

```
String [] fruits= {"apple", "banana", "kiwi", "mango", "pear", "peach",  
                  "berry", "strawberry", "blackberry"};  
JList strList = new JList(fruits);
```



2. Vector로 리스트 데이터를 제공하는 방법

```
Vector v = new Vector();  
v.add("apple");  
v.add("banana");  
v.add("kiwi");  
JList vList = new JList(v);
```



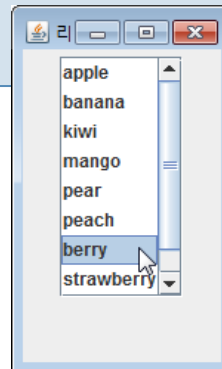
3. 빈 JList 컴포넌트를 생성하고 setListData() 메소드로 리스트 데이터를 제공하는 방법

```
ImageIcon [] images = {new ImageIcon("images/icon1.png"),  
                       new ImageIcon("images/icon2.png"), new ImageIcon("images/icon3.png"),  
                       new ImageIcon("images/icon4.png")  
};  
JList imageList = new JList();  
imageList.setListData(images);
```



4. 스크롤링을 지원하는 방법

```
JList scrollList = new JList(fruits);  
new JScrollPane(scrollList);
```



예제 11-9 : 다양한 리스트 컴포넌트 생성 예



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

public class ListEx extends JFrame {
    Container contentPane;
    String [] fruits= {"apple", "banana", "kiwi", "mango", "pear",
        "peach", "berry", "strawberry", "blackberry"};
    ImageIcon [] images = {new ImageIcon("images/icon1.png"),
        new ImageIcon("images/icon2.png"),
        new ImageIcon("images/icon3.png"),
        new ImageIcon("images/icon4.png")}
};

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

    JList strList = new JList(fruits);
    contentPane.add(strList);

    JList imageList = new JList();
    imageList.setListData(images);
    contentPane.add(imageList);

    JList scrollList = new JList(fruits);
    contentPane.add(new JScrollPane(scrollList));

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

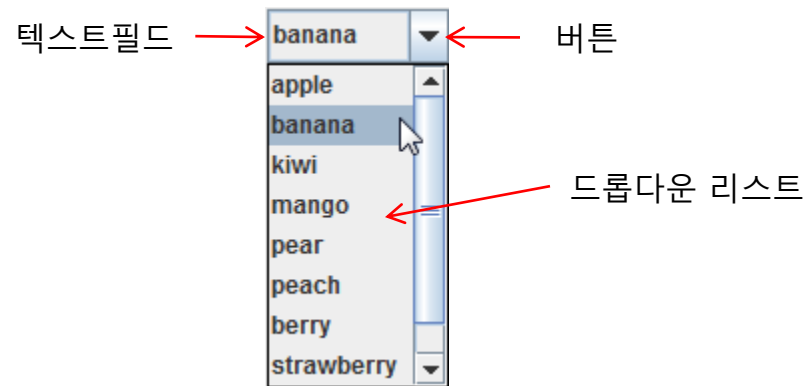
public static void main(String [] args) {
    new ListEx();
}
}
```

JComboBox, 콤보박스 컴포넌트

32

□ 콤보박스란?

- ▣ 텍스트 필드와 버튼, 그리고 드롭다운 리스트로 구성



□ 컴포넌트 생성

▣ JComboBox()

- 아이템이 비어 있는 콤보 박스 컴포넌트 생성

▣ JComboBox(ComboBoxModel model)

- model에 의해 아이템을 공급 받는 콤보박스 컴포넌트 생성

▣ JComboBox(Object [] items)

- items 배열로부터 아이템을 공급받는 콤보박스 컴포넌트 생성

▣ JComboBox(Vector items)

- items 벡트로부터 아이템을 공급받는 콤보박스 컴포넌트 생성

예제 11-10 : 콤보 박스 컴포넌트 만들기 예

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

public class ComboBoxEx extends JFrame {
    Container contentPane;
    String [] fruits = {"apple", "banana", "kiwi", "mango", "pear",
        "peach", "berry", "strawberry", "blackberry"};
    String [] names = {"kitae", "jaemoon", "hyosoo", "namyun"};

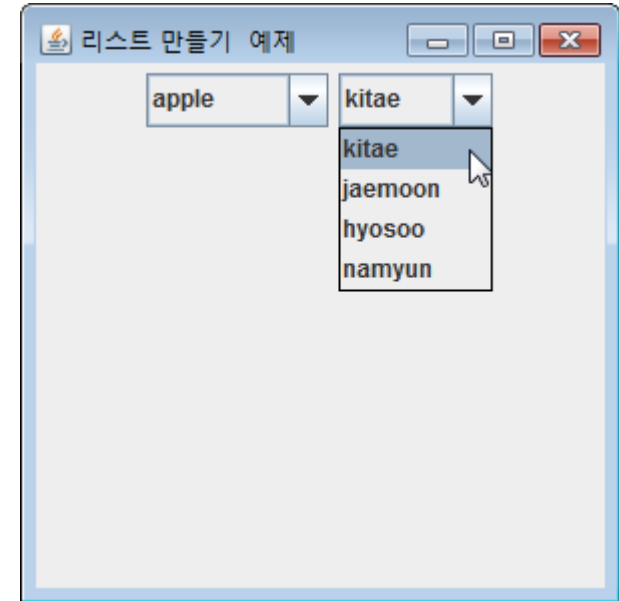
    ComboBoxEx() {
        setTitle("리스트 만들기 예제");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        contentPane = getContentPane();
        contentPane.setLayout(new FlowLayout());

        JComboBox strCombo = new JComboBox(fruits);
        contentPane.add(strCombo);

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

        contentPane.add(nameCombo);
        setSize(300,300);
        setVisible(true);
    }

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



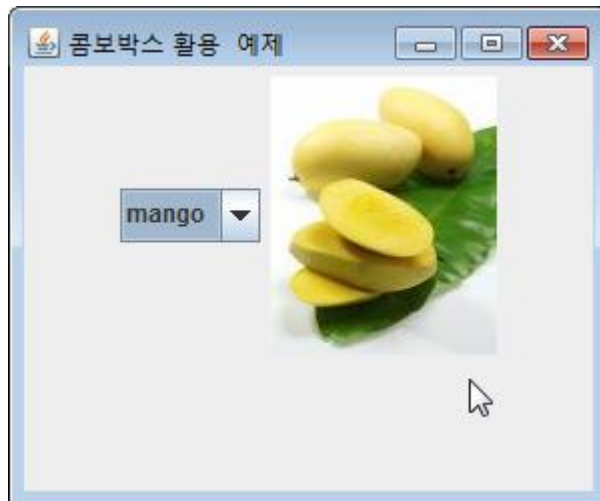
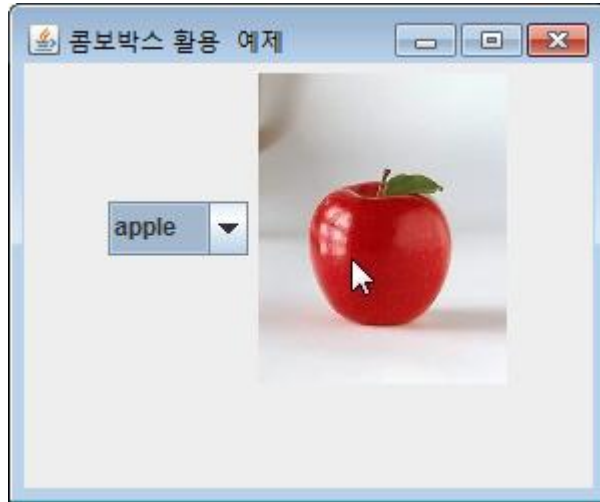
addItem() 메소드를 호출하여
아이템을 동적으로 삽입할 수 있다.

JComboBox와 Action 이벤트

34

- 콤보박스에서 아이템 선택시 Action 이벤트 발생
 - ▣ ActionListener 이용
 - ▣ 한 번의 아이템 선택시 한 번의(ActionEvent) 발생
- 콤보박스에서 아이템의 선택시 Item 이벤트 발생
 - ▣ ItemListener 이용
 - ▣ 새로운 아이템이 선택되면 2 번의 Item 이벤트 발생
 - 새로 아이템이 선택되었음을 알리는 Item 이벤트 발생
 - 이전에 선택된 아이템이 해제됨을 알리는 Item 이벤트 발생
 - ▣ 사용자가 아이템을 선택하지만 선택된 아이템이 변경되지 않을 경우에는 Item 이벤트가 발생하지 않음
- 현재 선택된 아이템 알아내기
 - ▣ int JComboBox.getSelectedIndex()
 - 선택 상태인 아이템의 인덱스 번호 리턴
 - ▣ Object JComboBox.getSelectedItem()
 - 선택 상태인 아이템 객체 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);
    }

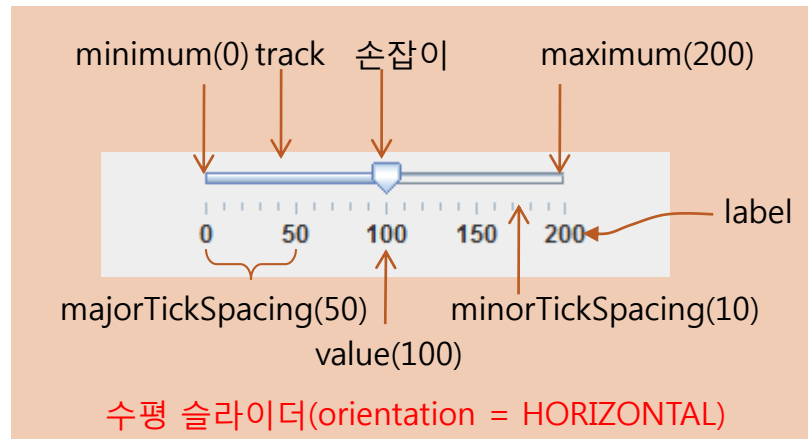
    public static void main(String [] args) {
        new ComboActionEx();
    }
}
```

슬라이더, JSlider

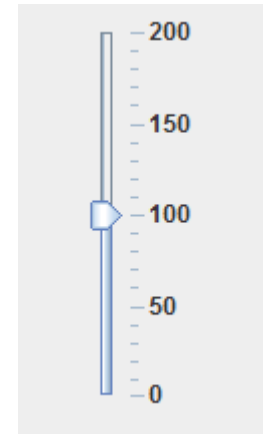
36

□ 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 컴포넌트 생성
- JSlider(int orientation)
 - orientation의 방향으로 구성된 슬라이더 컴포넌트 생성
- JSlider(int min, int max, int val)
 - minimum, maximum, value 값이 각각 min, max, val로 초기화된 수평 슬라이더 컴포넌트 생성
- JSlider(int orientation, int min, int max, int val)
 - minimum, maximum, value 값이 각각 min, max, val로 초기화된 슬라이더 컴포넌트 생성. 방향은 orientation

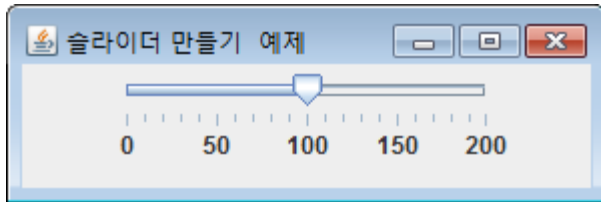
슬라이더의 모양 제어

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.

예제 11-12 : JSlider로 슬라이더 생성 및 모양 제어 예

38



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

```
public class SliderEx extends JFrame {
    Container contentPane;
    SliderEx() {
        setTitle("슬라이더 만들기 예제");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        contentPane = getContentPane();
        contentPane.setLayout(new FlowLayout());

        JSlider slider = new JSlider(JSlider.HORIZONTAL, 0, 200, 100);
        slider.setPaintLabels(true);
        slider.setPaintTicks(true);
        slider.setPaintTrack(true);
        slider.setMajorTickSpacing(50);
        slider.setMinorTickSpacing(10);

        contentPane.add(slider);
        setSize(300,100);
        setVisible(true);
    }

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

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();
}
```

