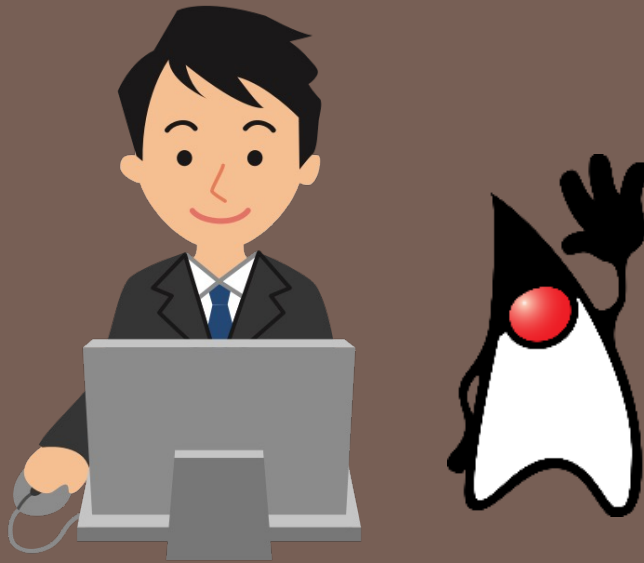


파워자바(개정3판)



11장 스윙 컴포넌트



11장의 목표

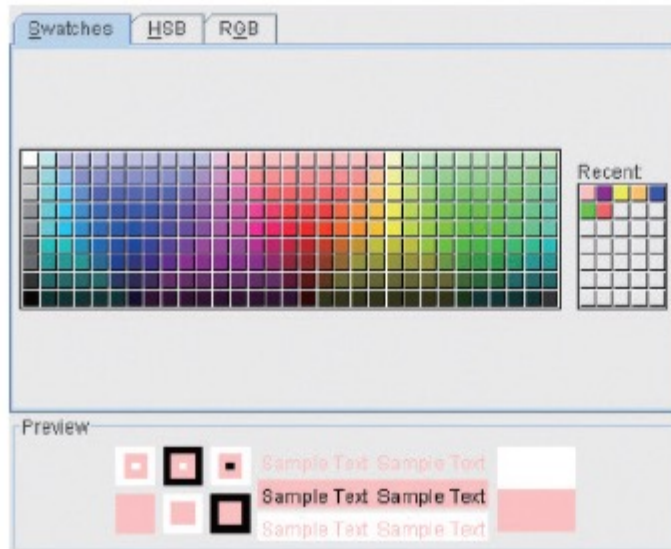
1. 스윙을 사용하여 화면에 이미지를 표시할 수 있나요?
2. 스윙을 사용하여 메모를 입력받을 수 있나요?
3. 스윙을 사용하여 메뉴가 붙어 있는 GUI 애플리케이션을 만들 수 있나요?
4. 라디오 버튼과 체크 박스를 화면에 만들 수 있나요?
5. 슬라이더를 사용하여 수치값을 마우스로 받을 수 있나요?
6. 스윙을 사용하여 피자 주문 화면을 만들 수 있나요?





스윙 컴포넌트 소개

- 스윙(Swing) 컴포넌트는 윈도우 기반 응용 프로그램을 만드는 데 사용되는 JFC(Java Foundation Classes)의 일부이다.
- AWT와 달리, 스윙은 플랫폼 독립적이고 경량 컴포넌트 구조이다. javax.swing 패키지는 JButton, JTextField, JTextArea, JRadioButton, JCheckbox, JMenu, JColorChooser 등과 같은 클래스를 제공한다.



[JColorChooser](#)

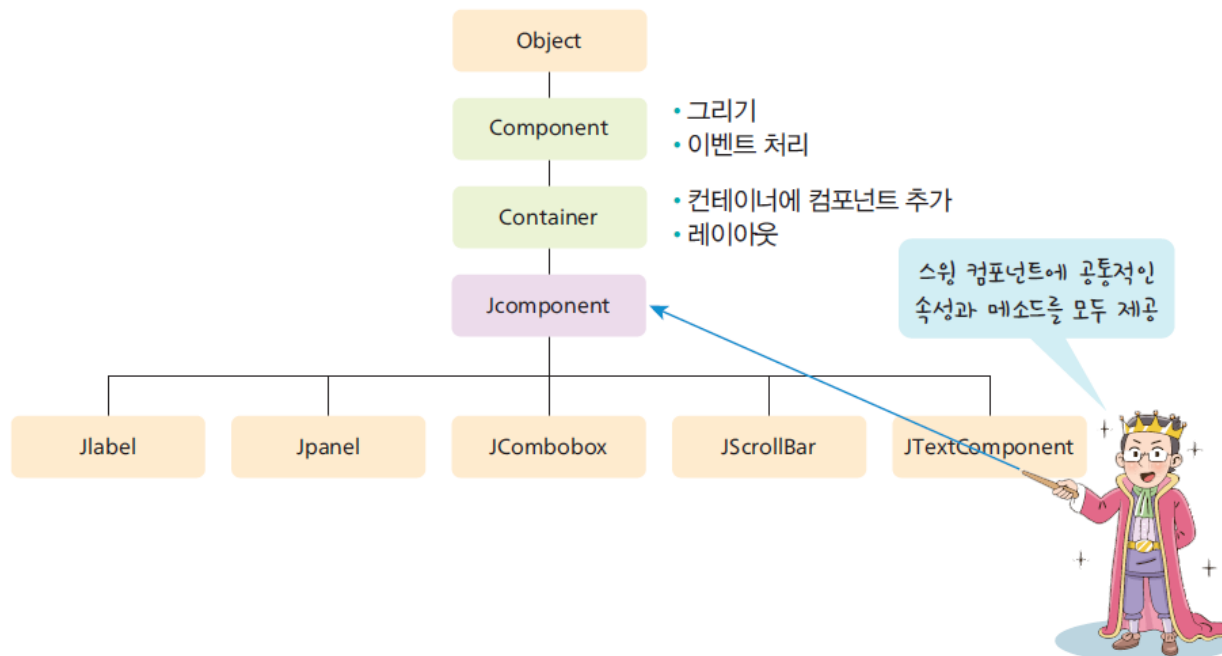


[JEditorPane](#) and [JTextPane](#)



Jcomponent 클래스

- 최상위 컨테이너를 제외하고는 거의 모든 컴포넌트가 JComponent를 상속 받는다. 예를 들어서 JPanel, JLabel, JButton, JComboBox같은 컴포넌트들은 모두 JComponent를 상속받고 있다.
- JComponent 클래스에는 스윙 컴포넌트들이 공통적으로 가져야 하는 속성과 메소드가 포함되어 있다.





Jcomponent 클래스의 메소드

메소드	설명
setBorder(Border), getBorder()	컴포넌트의 경계를 설정한다.
setForeground(Color), setBackground(Color)	컴포넌트의 전경색과 배경색을 설정한다.
getForeground(), getBackground()	컴포넌트의 전경색과 배경색을 얻는다.
setFont(Font), getFont()	컴포넌트의 폰트를 설정한다.
setCursor(Cursor), getCursor()	컴포넌트의 커서를 설정한다.

메소드	설명
setToolTipText(String)	툴팁에 표시되는 텍스트를 설정한다.
setEnabled(boolean), isEnabled()	컴포넌트를 활성화하거나 비활성화한다.
setVisible(boolean), isVisible()	컴포넌트를 화면에 표시한다.



Jcomponent 클래스의 메소드

메소드	설명
<code>addMouseListener(MouseListener)</code> <code>removeMouseListener(MouseListener)</code>	마우스 리스너를 추가하거나 제거한다.
...	...

메소드	설명
<code>repaint()</code>	컴포넌트의 일부나 전체를 다시 그리라고 요청한다.
<code>revalidate()</code>	컨테이너 안의 컴포넌트를 다시 배치하라고 요청한다.
<code>paintComponent(Graphics)</code>	컴포넌트를 그린다. 컴포넌트에서는 이 메소드를 재정의하여서 컴포넌트 위에 그림을 그릴 수 있다.

메소드	설명
<code>add(Component)</code>	컴포넌트를 컨테이너에 추가한다.
<code>remove(Component)</code>	컨테이너에서 컴포넌트를 삭제한다.
<code>getParent()</code>	컴포넌트의 컨테이너를 반환한다.



Jcomponent 클래스의 메소드

메소드	설명
getWidth(), getHeight()	픽셀 단위의 컴포넌트 크기를 반환한다.
getSize()	픽셀 단위의 컴포넌트 크기를 반환한다.
getX(), getY()	부모 컨테이너 안에서의 컴포넌트의 상대적인 위치를 반환한다.
getLocation()	부모 컨테이너 안에서의 상대적인 위치를 반환한다.
setLocation(int, int)	부모 컨테이너 안에서의 컴포넌트의 좌표를 지정한다.
setBounds(int, int, int, int)	부모 컨테이너 안에서의 크기와 위치를 설정한다.

메소드	설명
setPreferredSize(Dimension) setMaximumSize(Dimension) setMinimumSize(Dimension)	컴포넌트의 크기를 설정한다.
setAlignmentX(float), setAlignmentY(float)	컨테이너 안에서 컴포넌트들의 정렬을 지정한다.
setLayout(LayoutManager), getLayout()	배치 관리자를 설정한다.



중간점검

1. 컴포넌트의 폭과 높이를 알고 싶으면 어떤 메소드를 호출하는가?
2. 컴포넌트의 현재 위치를 알고 싶으면 어떤 메소드를 호출하는가?
3. 컨테이너에게 원하는 크기를 전달하고 싶을 때, 사용하는 메소드 중에서 하나만 말해보자.



중간점검



예제: 레이블의 폰트 변경하기

```
public class LabelTest extends JFrame {  
    private JPanel panel;  
    private JLabel label1, label2;  
  
    public LabelTest() {  
        setTitle("레이블 테스트");  
        setSize(400,150);  
  
        panel = new JPanel();  
        label1 = new JLabel("Color Label");  
        label1.setForeground(Color.BLUE);  
        label2 = new JLabel("Font Label");  
        label2.setFont(new Font("Arial", Font.ITALIC, 30));  
        label2.setForeground(Color.ORANGE);  
        panel.add(label1);  
        panel.add(label2);  
        add(panel);  
        setVisible(true);  
        setDefaultCloseOperation(EXIT_ON_CLOSE);  
    }  
  
    public static void main(String[] args) {  
        new LabelTest();  
    }  
}
```



레이블에 이미지 표시하기

- 레이블과 버튼에는 텍스트뿐만 아니라 이미지도 표시할 수 있다.

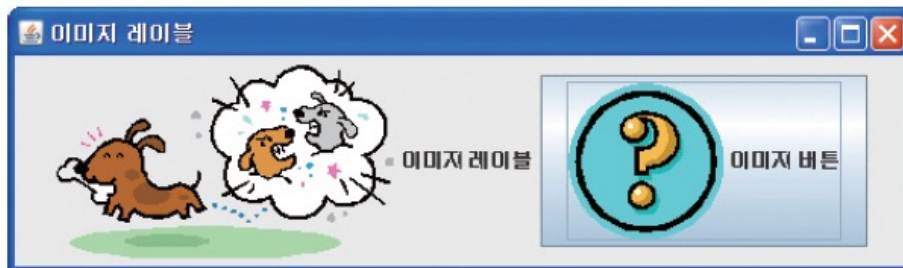


그림 9.10 이미지를 가지고 있는 레이블과 버튼

```
ImageIcon image = new ImageIcon("d://dog.png");  
JLabel label = new JLabel("Dog");  
label.setIcon(image);
```



예제:

- 레이블로 이미지를 표시하고 아래에 버튼을 표시해보자. d: 드라이브에 dog.png 파일이 있어야 한다.





예제:

```
public class ImageLabelTest extends JFrame {  
    private JPanel panel;  
    private JLabel label;  
  
    public ImageLabelTest() {  
        setTitle("레이블 테스트");  
        setSize(400, 250);  
  
        panel = new JPanel();  
        label = new JLabel("");  
        ImageIcon icon = new ImageIcon("d://dog.png");  
        label.setIcon(icon);  
        panel.add(label);  
        add(panel);  
        setVisible(true);  
        setDefaultCloseOperation(EXIT_ON_CLOSE);  
    }  
  
    public static void main(String[] args) {  
        new ImageLabelTest();  
    }  
}
```



텍스트 필드

- 텍스트 필드(text field)는 입력이 가능한 한 줄의 텍스트 필드를 만드는 데 사용

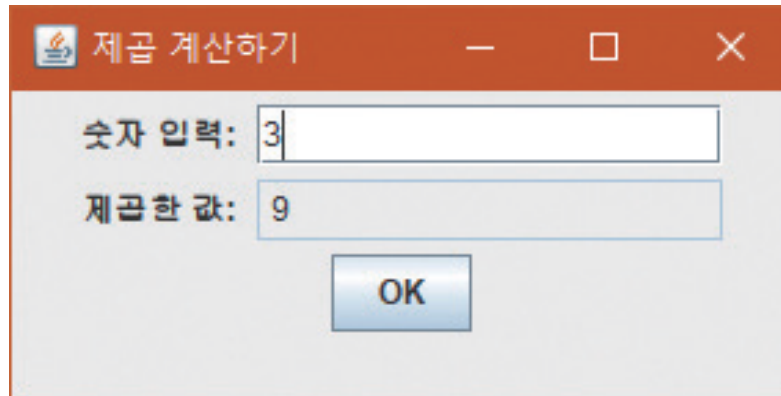
JTextField:	김철수	
JPasswordField:	****	입력한 문자가 보이지 않는다.
JFormattedTextField:	2009. 3. 7	숫자만 입력할 수 있다.

```
JTextField tf = new JTextField(30); // 30자 크기의 텍스트 필드를 만든다.  
tf.setText("아이디를 입력하십시오."); // 텍스트 필드의 텍스트를 설정한다.  
System.out.println(tf.getText()); // 텍스트 필드의 텍스트를 가져온다.
```

```
tf.requestFocus();
```

예제: 사용자로부터 받은 정수의 제공 계산하기

- 텍스트 필드를 이용하여 사용자로부터 정수를 입력받은 후에 정수의 제곱을 구하여 결과를 출력 전용의 텍스트 필드를 이용하여 표시하는 프로그램을 작성하여 보자.



제공 계산하기

숫자 입력: 3

제공한 값: 9

OK



예제: 사용자로부터 받은 정수의 제곱 계산하기

```
public class TextFieldFrame extends JFrame {  
    private JButton button;  
    private JTextField text, result;  
  
    public TextFieldFrame() {  
        setSize(300, 150);  
        setTitle("제곱 계산하기");  
  
        ButtonListener listener = new ButtonListener();  
        JPanel panel = new JPanel();  
        panel.add(new JLabel("숫자 입력: "));  
  
        text = new JTextField(15);  
        text.addActionListener(listener);  
        panel.add(text);  
        panel.add(new JLabel("제곱한 값: "));  
        result = new JTextField(15);  
        result.setEditable(false);  
        panel.add(result);  
        button = new JButton("OK");  
        button.addActionListener(listener);  
        panel.add(button);  
        add(panel);  
        setVisible(true);  
        setDefaultCloseOperation(EXIT_ON_CLOSE);  
    }  
}
```



예제: 사용자로부터 받은 정수의 제곱 계산하기

// 내부 클래스로서 텍스트 필드와 버튼의 액션 이벤트 처리

```
private class ButtonListener implements ActionListener {  
    public void actionPerformed(ActionEvent e) {  
        if (e.getSource() == button || e.getSource() == text) {  
            String name = text.getText();  
            int value = Integer.parseInt(name);  
            result.setText(" " + value * value);  
            text.requestFocus();  
        }  
    }  
}  
  
public static void main(String[] args) {  
    new TextFieldFrame();  
}
```




예제: 패스워드 필드 사용하기

- 패스워드 필드는 용어 그대로 암호를 입력받을 때 사용한다. 패스워드 필드에 사용자가 암호를 입력하면 글자들이 모두 * 문자로 표시된다. 패스워드 필드를 사용하여서 다음과 같은 로그인 윈도우를 작성하여 보자.

The image shows a graphical user interface for a login window. The window has a title bar with the text 'login window' and standard window control buttons (minimize, maximize, close). Inside the window, there are two text input fields. The first field is labeled 'id' and contains the text 'java'. The second field is labeled 'pass' and contains six asterisks '*****'. Below these fields are two buttons: 'login' and 'cancel'.



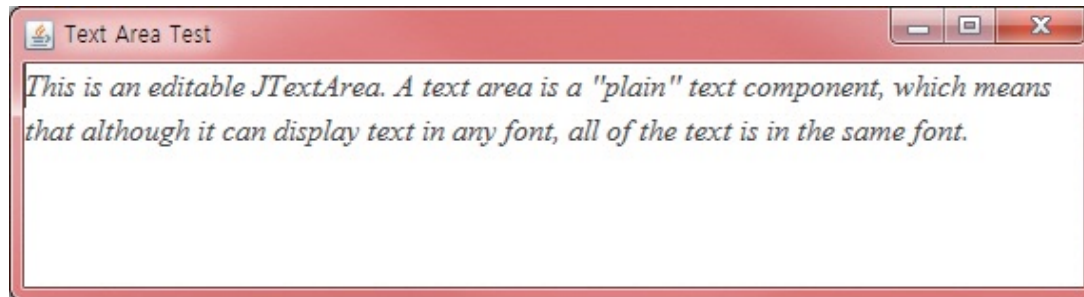
예제: 패스워드 필드 사용하기

```
public class LoginWindow extends JFrame {  
    public LoginWindow()  
    {  
        setTitle("login window");  
        setSize(300, 150);  
  
        JPanel panel = new JPanel();  
        add(panel);  
  
        panel.add(new JLabel("id  "));  
        panel.add(new JTextField(20));  
        panel.add(new JLabel("pass"));  
        panel.add(new JPasswordField(20));  
  
        JButton login = new JButton("login");  
        panel.add(login);  
        JButton cancel = new JButton("cancel");  
        panel.add(cancel);  
  
        setVisible(true);  
        setDefaultCloseOperation(EXIT_ON_CLOSE);  
    }  
    public static void main(String[] args) {  
        new LoginWindow();  
    }  
}
```



텍스트 영역 사용하기

- 텍스트 영역(**TextArea**)은 앞의 텍스트 필드와 비슷하지만 한 줄이 아니라 여러 줄의 텍스트가 들어 갈 수 있다.



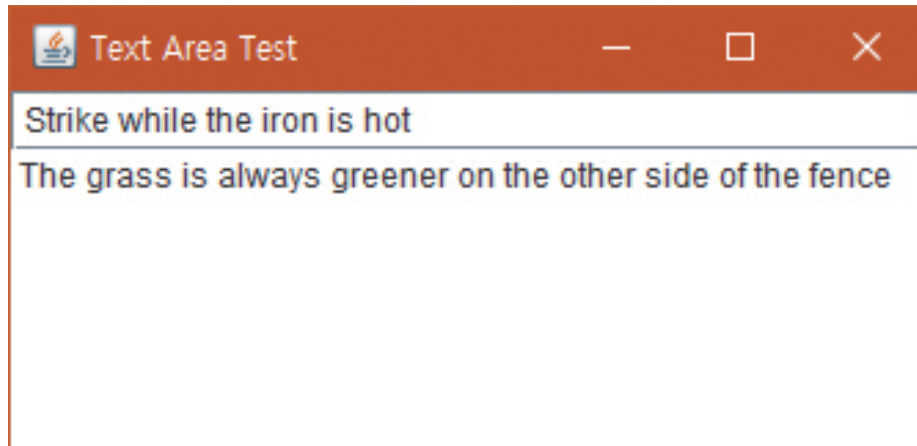
```
textArea = new JTextArea(10, 30);    // 10행 30열의 텍스트 영역을 만든다.
```

```
textArea.append("컵이 많은 개일수록 큰소리로 짖는다");
```



예제: 텍스트 필드 사용하기

- 사용자가 텍스트 필드에 텍스트를 입력하고 엔터키를 누르면 이것을 텍스트 영역에 추가하는 프로그램을 작성하여 보자.





예제: 텍스트 필드 사용하기

```
public class TextAreaFrame extends JFrame implements ActionListener {  
    protected JTextField textField;  
    protected JTextArea textArea;  
  
    public TextAreaFrame() {  
        setTitle("Text Area Test");  
        textField = new JTextField(30);  
        textField.addActionListener(this);  
  
        textArea = new JTextArea(6, 30);  
        textArea.setEditable(false);  
  
        add(textField, BorderLayout.NORTH);  
        add(textArea, BorderLayout.CENTER);  
  
        pack();  
        setVisible(true);  
    }  
}
```



예제: 텍스트 필드 사용하기

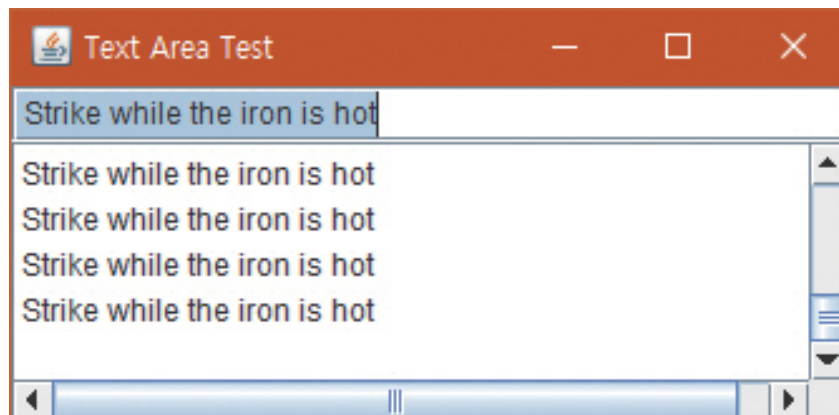
```
public void actionPerformed(ActionEvent evt) {  
    String text = textField.getText();  
    textArea.append(text + "\n");  
    textField.selectAll();  
    textArea.setCaretPosition(textArea.getDocument().getLength());  
}  
  
public static void main(String[] args) {  
    new TextAreaFrame();  
}  
}
```



스크롤 페인

- 스크롤 페인(scroll pane)은 컴포넌트에 스크롤 기능을 제공한다.

```
textArea = new JTextArea(6, 30);    // 텍스트 영역을 생성한다.  
JScrollPane scrollPane = new JScrollPane(textArea); // ❶  
add(scrollPane, BorderLayout.CENTER);
```





중간점검



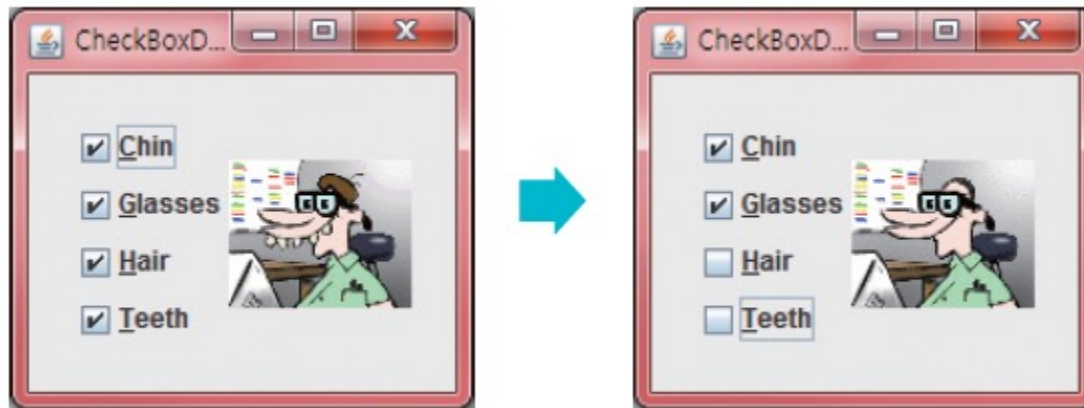
중간점검

1. 사용자로부터 여러 줄의 텍스트를 입력받으려면 어떤 컴포넌트를 사용하는가?
2. 사용자로부터 패스워드를 입력받으려면 어떤 컴포넌트를 사용하는가?
3. 텍스트 필드에서 사용자가 <Enter> 키를 누르면 어떤 이벤트가 발생하는가?



체크박스

- 체크 박스(**check box**)란 사용자가 클릭하여서 체크된 상태와 체크되지 않은 상태 중의 하나로 만들 수 있는 컨트롤이다.



```
onion = new JCheckBox("양파");  
cheese = new JCheckBox("치즈", true);  
onion.setSelected(true);
```



체크박스 이벤트 처리

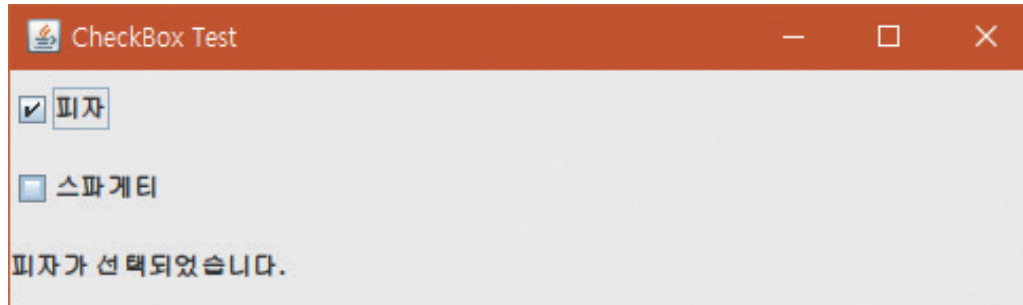
- 사용자가 체크 박스를 클릭하면 **ItemEvent**가 발생한다. **ItemEvent**를 처리하려면 **ItemListener**를 구현하여야 한다. 무명 클래스를 사용하는 것이 편하다

```
JCheckBox check = new JCheckBox("Checkbox", false);
check.addItemListener(new ItemListener() {
    public void itemStateChanged(ItemEvent e) {
        System.out.println("체크상태: " + check.isSelected());
    }
});
```



예제: 체크 박스 사용하기

- 아래와 같은 2개의 체크 박스를 가지는 프로그램을 작성해보자.





예제: 체크 박스 사용하기

```
public class CheckBoxTest extends JFrame {  
    public CheckBoxTest() {  
        setTitle("CheckBox Test");  
        setLayout(new GridLayout(0, 1));  
        JLabel label = new JLabel();  
        JCheckBox checkbox1 = new JCheckBox("피자");  
        JCheckBox checkbox2 = new JCheckBox("스파게티");  
        add(checkbox1);  
        add(checkbox2);  
        add(label);  
        checkbox1.addItemListener(new ItemListener() {  
            public void itemStateChanged(ItemEvent e) {  
                label.setText("피자가 " + (e.getStateChange() == 1 ? "선택되었습니다." :  
"선택해제되었습니다."));  
            }  
        });  
    }  
};
```



예제: 체크 박스 사용하기

```
checkbox2.addItemListener(new ItemListener() {  
    public void itemStateChanged(ItemEvent e) {  
        label.setText("스파게티가 " + (e.getStateChange() == 1 ? "선택되었습니다." :  
"선택해제되었습니다."));  
    }  
});  
setSize(300, 150);  
setVisible(true);  
}  
  
public static void main(String args[]) {  
    new CheckBoxTest();  
}  
}
```



중간점검



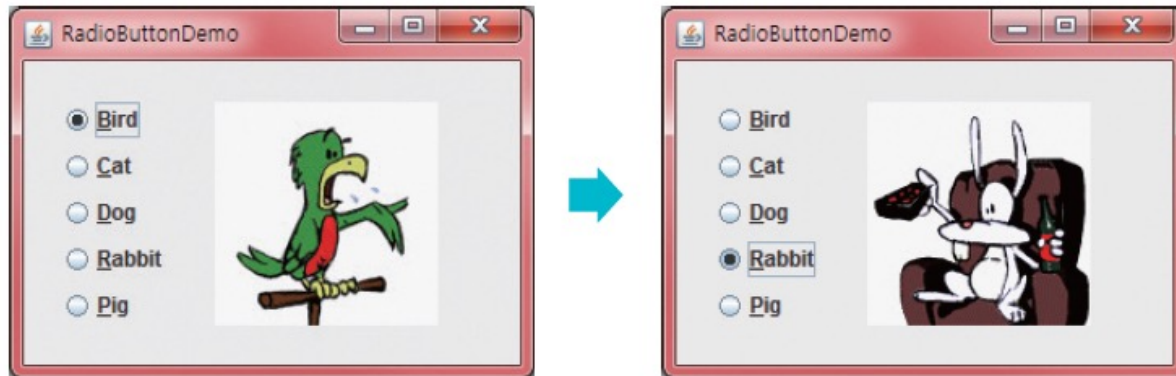
중간점검

1. 체크 박스에서는 어떤 이벤트가 발생하는가?
2. 체크 박스 이벤트를 처리하려면 어떤 방법이 가장 편리한가?



라디오 버튼

- 라디오 버튼(**radio button**)은 체크 박스와 비슷하지만 하나의 그룹 안에서는 한 개의 버튼만 선택할 수 있다는 점이 다르다. 만약 하나의 라디오 버튼을 클릭하면 다른 버튼은 자동적으로 선택이 해제된다.





라디오 버튼

- 라디오 버튼을 만들기 위해서는 두 개의 클래스를 이용한다. 하나는 **JRadioButton**으로 라디오 버튼을 생성하는 데 사용한다.
- 또 하나는 **ButtonGroup**으로 버튼들을 그룹핑하는 데 사용된다.

```
JRadioButton    radio1 = new JRadioButton("Small Size");  
JRadioButton    radio2 = new JRadioButton("Medium Size");  
JRadioButton    radio3 = new JRadioButton("Large Size");  
...
```

```
ButtonGroup    group = new ButtonGroup();
```

ButtonGroup 객체를 생성한다.

```
group.add(radio1);  
group.add(radio2);  
group.add(radio3);
```

라디오 버튼들을 ButtonGroup 객체에 추가한다.



예제: 커피 주문 화면 만들기

- 만약 라디오 버튼이 눌러지는 순간, 어떤 작업을 실행하고 싶다면 액션 이벤트를 처리하면 된다. 코드에서 라디오 버튼이 눌러졌는지를 검사하고 싶다면 `isSelected()`를 사용한다. 만약 강제로 어떤 버튼을 선택되게 하려면 `doClick()`을 사용한다.
- 커피의 크기를 선택하는 다음과 같은 화면을 라디오 버튼을 이용하여서 생성하여 보자.



예제: 커피 주문 화면 만들기

```
public class RadioButtonFrame extends JFrame implements ActionListener {  
  
    private JRadioButton small, medium, large;  
    private JLabel text;  
    private JPanel topPanel, sizePanel, resultPanel;  
  
    public RadioButtonFrame () {  
        setTitle("라디오 버튼 테스트");  
        setSize(500, 150);  
  
        topPanel = new JPanel();  
        topPanel.add(new JLabel("어떤 크기의 커피를 주문하시겠습니까?"));  
  
        add(topPanel, BorderLayout.NORTH);  
  
        sizePanel = new JPanel();  
        small = new JRadioButton("Small Size");  
        medium = new JRadioButton("Medium Size");  
        large = new JRadioButton("Large Size");
```



예제: 커피 주문 화면 만들기

```
ButtonGroup size = new ButtonGroup();
size.add(small);
size.add(medium);
size.add(large);

small.addActionListener(this);
medium.addActionListener(this);

large.addActionListener(this);
sizePanel.add(small);

sizePanel.add(medium);
sizePanel.add(large);

add(sizePanel, BorderLayout.CENTER);

resultPanel = new JPanel();
text = new JLabel("크기가 선택되지 않았습니다.");
text.setForeground(Color.red);
resultPanel.add(text);
add(resultPanel, BorderLayout.SOUTH);
setVisible(true);
```

```
}
```



예제: 커피 주문 화면 만들기

```
public void actionPerformed(ActionEvent e) {  
    if (e.getSource() == small) {  
        text.setText("Small 크기가 선택되었습니다.");  
    }  
    if (e.getSource() == medium) {  
        text.setText("Medium 크기가 선택되었습니다.");  
    }  
    if (e.getSource() == large) {  
        text.setText("Large 크기가 선택되었습니다.");  
    }  
}  
  
public static void main(String[] args) {  
    new RadioButtonFrame();  
}
```



중간점검



중간점검

1. 라디오 버튼 중에서 하나만 눌러지도록 하려면 추가적으로 어떤 객체가 필요한가?
2. 라디오 버튼을 누르면 어떤 이벤트가 발생하는가?



콤보박스

- 콤보 박스(**combo box**)도 여러 항목 중에서 하나를 선택하는 데 사용할 수 있다. 콤보 박스는 텍스트 필드와 리스트의 결합이다. 사용자는 콤보 박스의 텍스트를 직접 입력할 수도 있고 리스트에서 선택할 수도 있다.

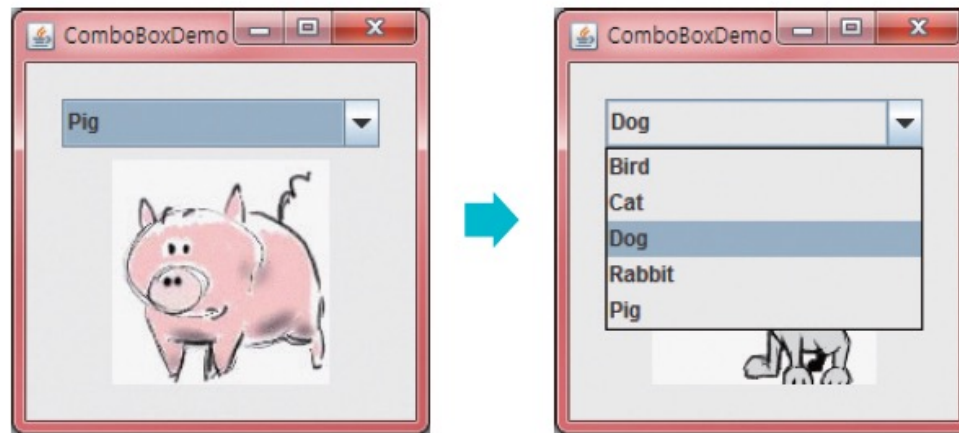


그림 11.5 콤보박스의 모습



콤보 박스

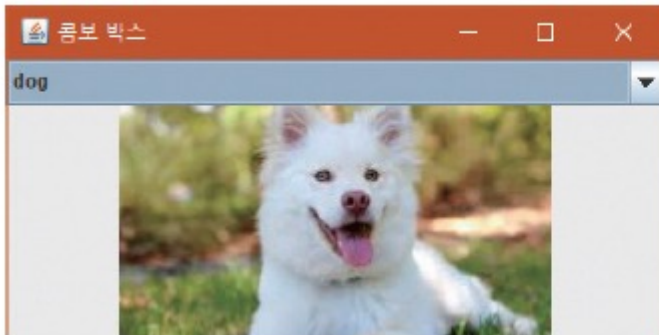
- 콤보 박스를 생성하기 위해서는 먼저 생성자 중에서 하나를 골라서 호출하여야 한다. 생성자는 비어 있는 콤보 박스를 생성한다. 여기에 항목을 추가하려면 addItem() 메소드를 사용한다.

```
JComboBox combo = new JComboBox();  
combo.addItem("dog");  
combo.addItem("lion");  
combo.addItem("tiger");
```



예제: 콤보 박스를 사용하여 이미지 선택하기

- 사용자가 콤보 박스에서 하나의 이미지를 선택하면 이것을 화면에 표시하는 프로그램을 작성해보자.



//





예제: 콤보 박스를 사용하여 이미지 선택하기

```
public class ComboBoxFrame extends JFrame implements ActionListener {
    JLabel label;

    public ComboBoxFrame() {
        setTitle("콤보 박스");
        setSize(400, 200);

        String[] animals = { "dog", "cat", "bird" };
        JComboBox animalList = new JComboBox(animals);
        animalList.setSelectedIndex(0);
        animalList.addActionListener(this);

        label = new JLabel();
        label.setHorizontalAlignment(JLabel.CENTER);
        changePicture(animals[animalList.getSelectedIndex()]);
        add(animalList, BorderLayout.NORTH);
        add(label, BorderLayout.CENTER);
        setVisible(true);
    }
}
```



예제: 콤보 박스를 사용하여 이미지 선택하기

```
public void actionPerformed(ActionEvent e) {
    JComboBox cb = (JComboBox) e.getSource();
    String name = (String) cb.getSelectedItem();
    changePicture(name);
}

protected void changePicture(String name) {
    ImageIcon icon = new ImageIcon(name + ".png");
    label.setIcon(icon);
    if (icon != null) {
        label.setText(null);
    } else {
        label.setText("이미지가 발견되지 않았습니다.");
    }
}

public static void main(String[] args) {
    ComboBoxFrame frame=new ComboBoxFrame();
}
}
```



중간점검

1. 콤보 박스에서 어떤 이벤트가 발생하는가?
2. 콤보 박스의 특징은 무엇인가?



중간점검



메뉴 붙이기

- 메뉴바는 윈도우 상단에 위치하면서 여러 개의 메뉴를 가지고 있다. 팝업 메뉴는 사용자가 팝업이 가능한 컴포넌트 위에서 마우스 오른쪽 버튼을 누르면 그 위치에 등장한다.
- 메뉴 항목은 이미지나 텍스트를 가질 수 있다

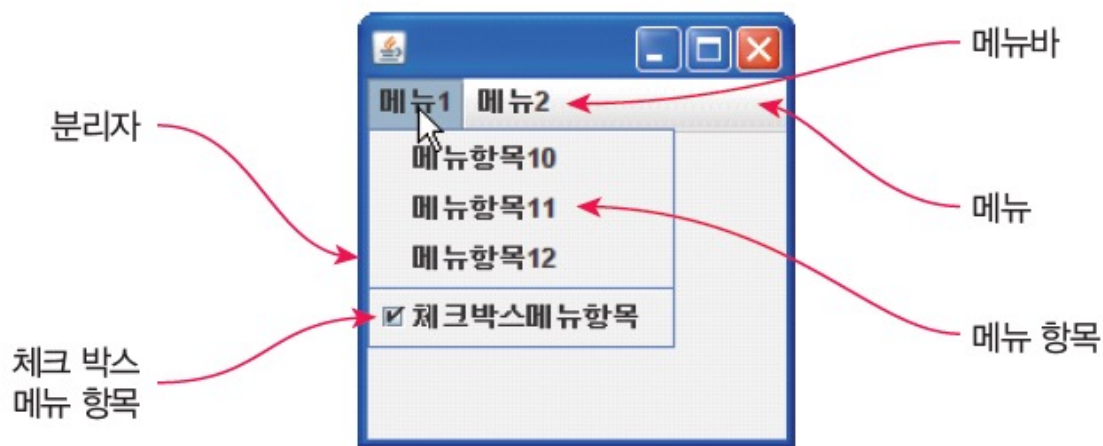


그림 11.6 메뉴에서의 용어



메뉴생성 절차

// ① 메뉴바 관련 변수를 선언한다.

```
JMenuBar menuBar;           // 메뉴바  
JMenu menu;                 // 메뉴  
JMenuItem menuItem;         // 메뉴 항목
```

// ② 메뉴바를 생성한다.

```
menuBar = new JMenuBar();
```

// ③ 메뉴를 생성하여 메뉴바에 추가한다.

```
menu = new JMenu("메뉴1");  
menuBar.add(menu);
```

// ④ 메뉴 항목을 생성하여 메뉴에 추가한다.

```
menuItem = new JMenuItem("메뉴항목1", KeyEvent.VK_T);  
menu.add(menuItem);
```

// ⑤ 프레임에 메뉴바를 설정한다.

```
frame.setJMenuBar(mb);
```



메뉴 이벤트 처리

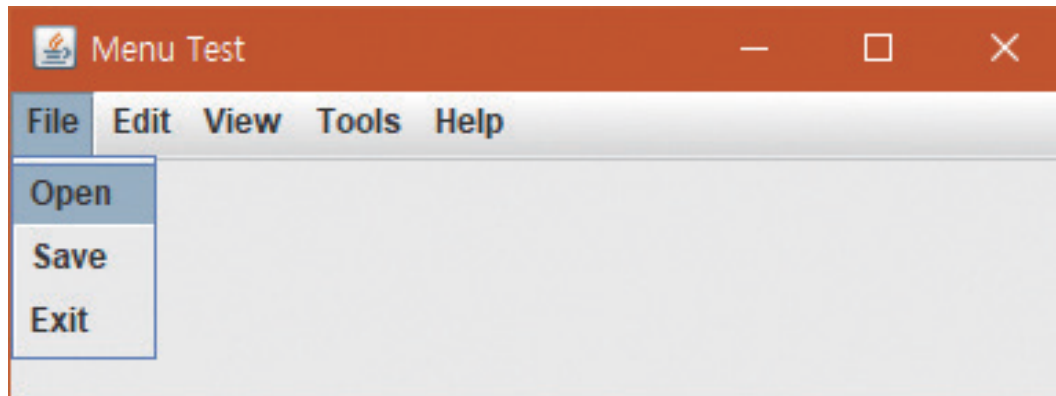
- 각 메뉴 항목에 대하여 액션 이벤트를 처리해주면 된다

```
// 각각의 메뉴 항목에 대하여 이벤트 처리기 등록  
menuItem.addActionListener(this);
```



예제: 메뉴 생성하기

- 프레임을 생성한 후에 메뉴바를 붙인다. 메뉴바에는 **5개의 메뉴**를 생성하여 추가하고 첫 번째 메뉴에 다양한 메뉴
- 항목들을 생성하여 추가하여 보자.





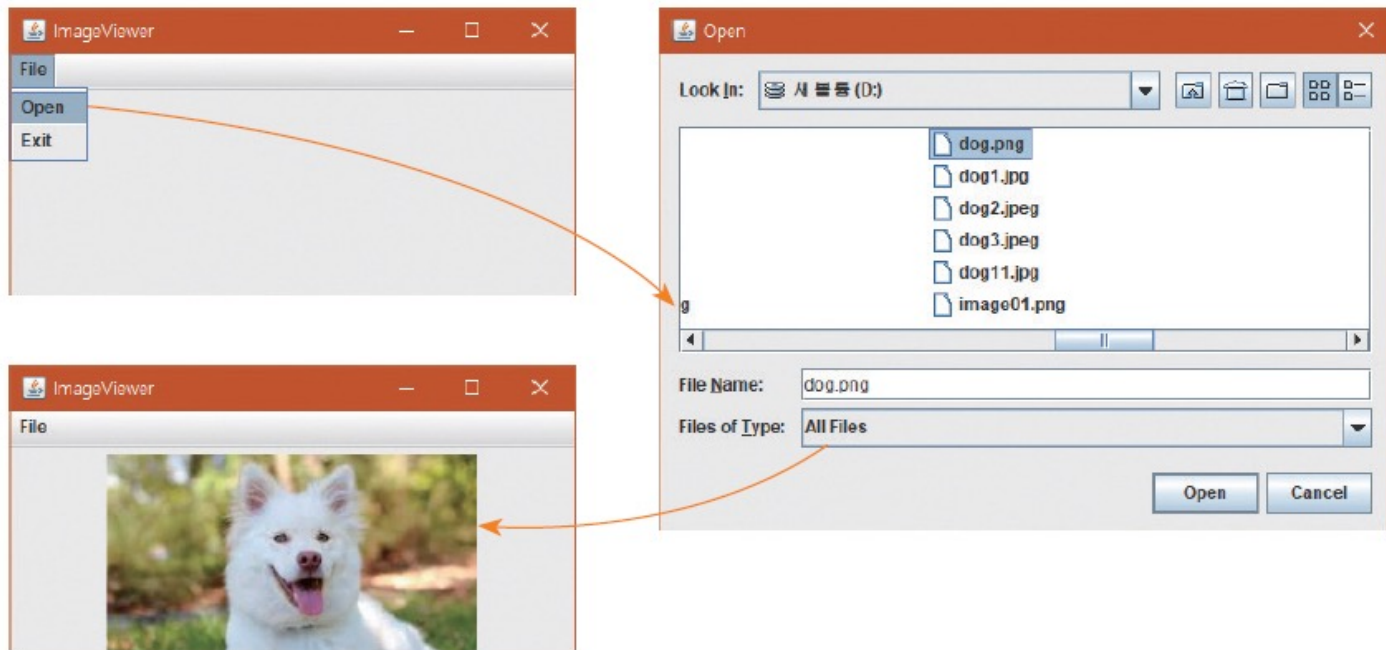
예제: 메뉴 생성하기

```
public class MenuTest extends JFrame {  
  
    public MenuTest() {  
        setTitle("Menu Test");  
        setSize(400, 150);  
        JMenuBar menuBar = new JMenuBar();  
        JMenu menu = new JMenu("File");  
        menu.add(new JMenuItem("Open"));  
        menu.add(new JMenuItem("Save"));  
        menu.add(new JMenuItem("Exit"));  
        menuBar.add(menu);  
  
        menuBar.add(new JMenu("Edit"));  
        menuBar.add(new JMenu("View"));  
        menuBar.add(new JMenu("Tools"));  
        menuBar.add(new JMenu("Help"));  
  
        setJMenuBar(menuBar);  
        setVisible(true);  
    }  
  
    public static void main(String args[]) {  
        MenuTest f = new MenuTest();  
    }  
}
```




예제: 이미지 뷰어 만들기

- 메뉴와 레이블을 이용하여서 이미지를 화면에 보여주는 프로그램을 작성해 보자. “Open”과 “Exit”로 이루어진 메뉴를 만들고 “Open” 메뉴를 선택하면 다음과 같이 파일을 선택하는 대화 상자가 나타난다. 이미지 파일이 선택되면 이것을 레이블을 이용하여 화면에 표시한다.





예제: 이미지 뷰어 만들기

```
public class ImageViewer extends JFrame implements ActionListener {  
    private JLabel label;  
    private JMenuItem open, exit;  
  
    public ImageViewer() {  
        setTitle("ImageViewer");  
        setSize(350, 200);  
  
        JMenuBar mbar = new JMenuBar();  
        JMenu m = new JMenu("File");  
        open = new JMenuItem("Open");  
        open.addActionListener(this);  
        m.add(open);  
        exit = new JMenuItem("Exit");  
        exit.addActionListener(this);  
        m.add(exit);  
        mbar.add(m);  
        setJMenuBar(mbar);  
  
        label = new JLabel();  
        JPanel panel = new JPanel();  
        panel.add(label, "Center");  
        add(panel);  
        setVisible(true);  
    }  
}
```



예제: 이미지 뷰어 만들기

```
public void actionPerformed(ActionEvent e) {
    Object source = e.getSource();
    if (source == open) {
        JFileChooser chooser = new JFileChooser();
        int r = chooser.showOpenDialog(this);
        if (r == JFileChooser.APPROVE_OPTION) {
            String name =
chooser.getSelectedFile().getAbsolutePath();
            label.setIcon(new ImageIcon(name));
        }
    } else if (source == exit)
        System.exit(0);
}

public static void main(String[] args) {
    ImageViewer f = new ImageViewer();
}
}
```



중간점검



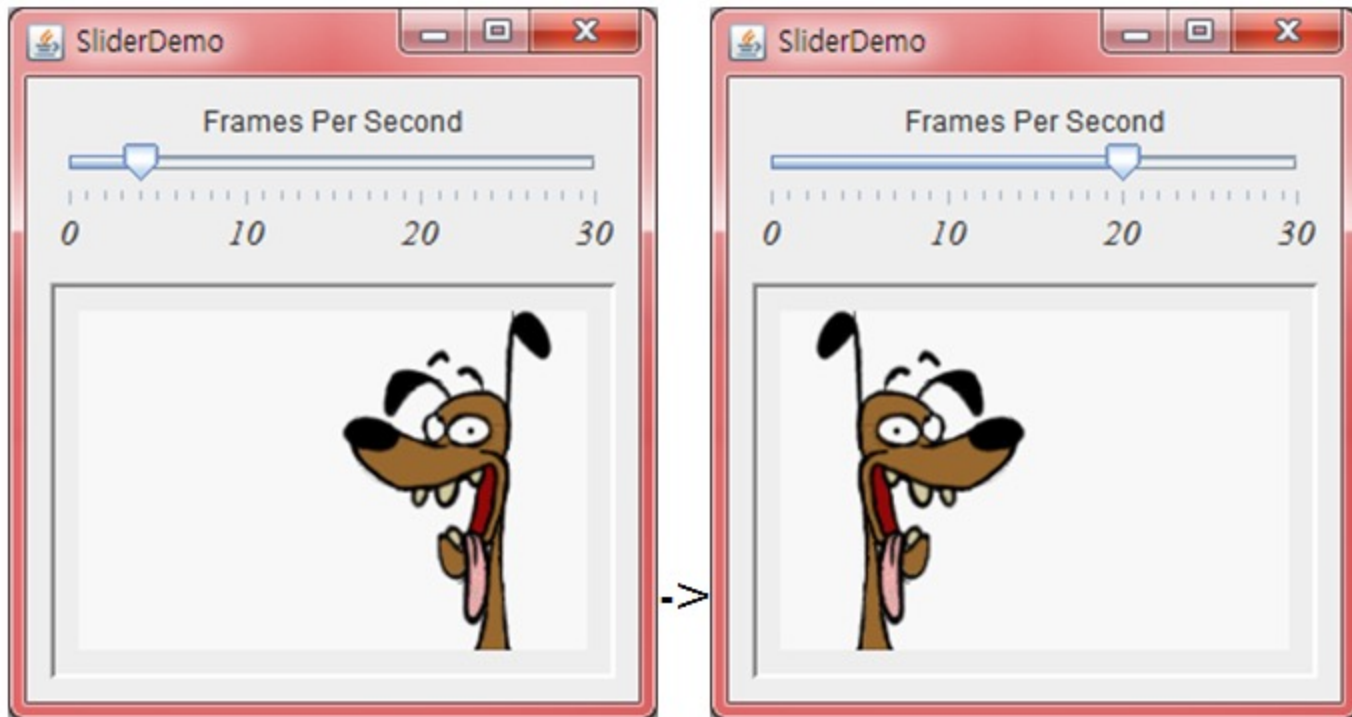
중간점검

1. 메뉴가 만들어지는 과정을 설명해보자.
2. 메뉴 항목을 선택하면 어떤 이벤트가 발생하는가?



슬라이더

- 슬라이더(**slider**)는 사용자가 특정한 범위 안에서 하나의 값을 선택할 수 있는 컴포넌트이다.





슬라이더와 이벤트 처리

```
slider = new JSlider(0, 100, 50);
```

```
slider.setMajorTickSpacing(10);    // 큰 눈금 간격
slider.setMinorTickSpacing(1);     // 작은 눈금 간격
slider.setPaintTicks(true);        // 눈금을 표시한다.
slider.setPaintLabels(true);       // 값을 레이블로 표시한다.
```

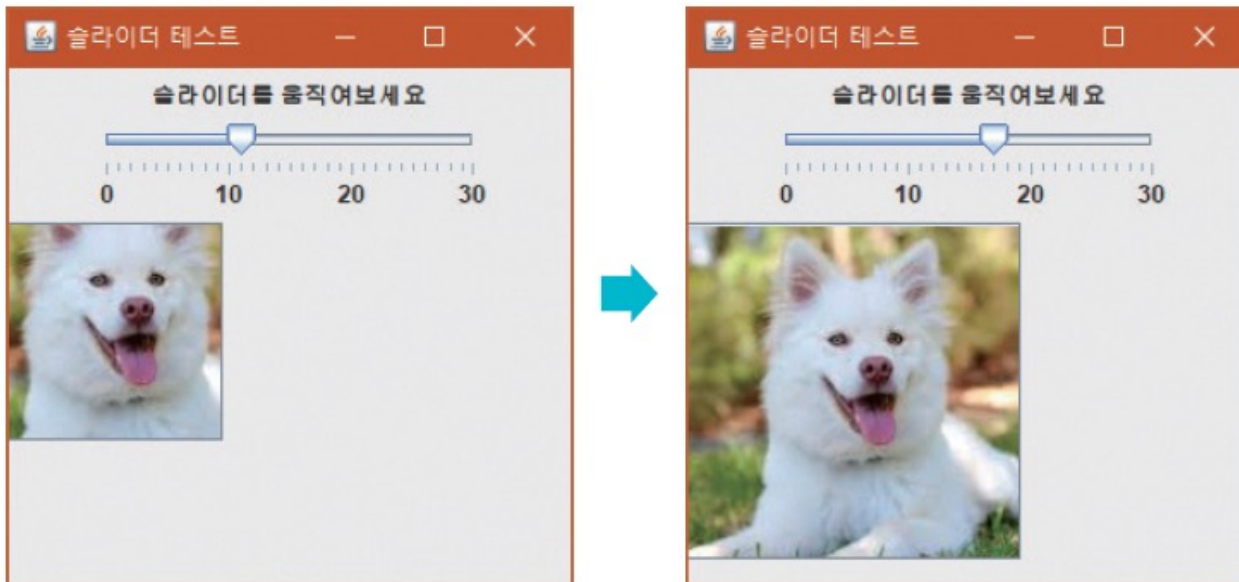
```
slider.addChangeListener(this);
```

```
public void stateChanged(ChangeEvent e) {
    JSlider source = (JSlider) e.getSource();
    if (!source.getValueIsAdjusting()) {
        int value = (int) source.getValue();
        button.setSize(value * 10, value * 10);
    }
}
```



예제

- 다음 예제에서는 슬라이더를 움직이면 표시되는 이미지의 크기가 변경된다.





예제

```
public class SliderFrame extends JFrame implements ChangeListener {
    static final int INIT_VALUE = 15;
    private JButton buttonOK;
    private JSlider slider;
    private JButton button;

    public SliderFrame() {
        JPanel panel;

        setTitle("슬라이더 테스트");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        panel = new JPanel();
        JLabel label = new JLabel("슬라이더를 움직여보세요", JLabel.CENTER);
        label.setAlignmentX(Component.CENTER_ALIGNMENT);
        panel.add(label);

        slider = new JSlider(0, 30, INIT_VALUE);
        slider.setMajorTickSpacing(10);           // 큰 눈금 간격
        slider.setMinorTickSpacing(1);            // 작은 눈금 간격
        slider.setPaintTicks(true);               // 눈금을 표시한다.
        slider.setPaintLabels(true);              // 값을 레이블로 표시한다.
        slider.addChangeListener(this);           // 이벤트 리스너를 붙인다.
        panel.add(slider);
    }
}
```




예제

```
button = new JButton("");
    ImageIcon icon = new ImageIcon("dog.gif");
    button.setIcon(icon);
    button.setSize(INIT_VALUE * 10, INIT_VALUE * 10);
    panel.add(button);
    add(panel);

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

public void stateChanged(ChangeEvent e) {
    JSlider source = (JSlider) e.getSource();
    if (!source.getValueIsAdjusting()) {
        int value = (int) source.getValue();
        button.setSize(value * 10, value * 10);
    }
}

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



중간점검

1. 슬라이더에서 발생하는 이벤트는?
2. 0에서 200까지의 범위를 가지고 초기값이 10인 슬라이더를 생성하는 문장을 작성해보자.



중간점검



Mini Project: 피자주문

- 피자를 주문할 수 있는 애플리케이션을 작성하여 보자. 다음의 스케치를 참조하라.

원도우 제목

피자 주문

자바 피자에 오신 것을 환영합니다.

종류	추가 토핑	크기
<input checked="" type="radio"/> 콤보	<input type="checkbox"/> 피망	<input type="radio"/> Small
<input type="radio"/> 포테이토	<input type="checkbox"/> 치즈	<input checked="" type="radio"/> Medium
<input type="radio"/> 불고기	<input type="checkbox"/> 페페로니	<input type="radio"/> Large
	<input type="checkbox"/> 베이컨	

주문 취소



Mini Project: 입회원서

- 다음과 같은 동아리 입회원서 형식을 스윙 컴포넌트를 사용하여 제작해보자. GUI는 “Window-Builder”를 사용하여 비주얼하게 작성하여도 된다.



Summary

- 스윙 컴포넌트들은 계층적인 구조로 되어 있다. 거의 모든 스윙 컴포넌트의 부모 클래스는 **JComponent**로서 컴포넌트의 모양, 크기, 위치를 변경할 수 있는 많은 메소드들을 제공한다.
- **JLabel**을 이용하여 문자열이나 이미지를 표시할 수 있다. 폰트는 변경 가능하다.
- **JCheckBox**는 선택과 선택해제의 두 가지의 상태를 가지는 특수한 버튼이다.
- **JRadioButton**은 여러 개의 버튼 중에서 오직 하나의 버튼만 선택되는 특수한 버튼이다.
- **JTextField**는 한 줄의 문자열을 입력받을 때 사용한다. 사용자가 <Enter> 키를 누르면 액션 이벤트가 발생한다.
- **JTextArea**는 여러 줄의 문자열을 입력할 수 있는 영역을 제공한다.
- **JSlider**는 사용자가 일정 범위의 수치값을 마우스로 입력하게 해주는 컴포넌트이다.
- **Change** 이벤트는 컴포넌트에 어떤 변화가 일어날 때 발생하는 이벤트이다.
- **JList**는 여러 개의 아이템을 리스트 형식으로 보여주고 사용자가 그 중에서 하나를 선택하도록 하는 컴포넌트이다.





Q & A

