

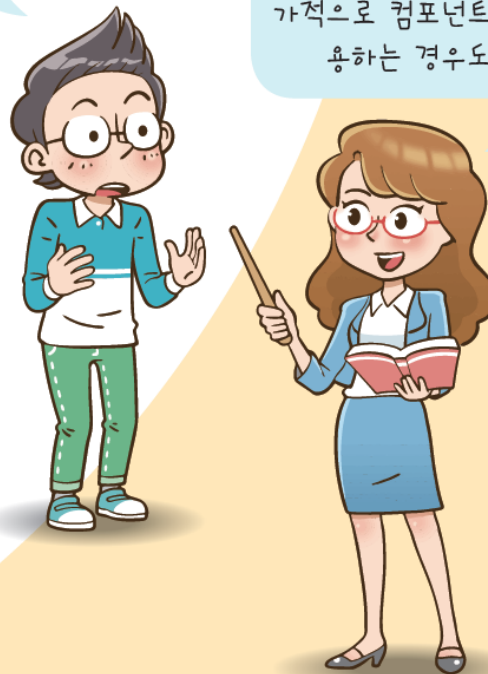
제12장 스윙 컴포넌트

학습목차

- 01 스윙 컴포넌트에 이미지 표시하기
- 02 체크 박스
- 03 라디오 버튼
- 04 텍스트 필드
- 05 텍스트 영역
- 06 콤보 박스
- 07 슬라이더
- 08 파일 선택기
- LAB 피자 주문 화면 만들기
- LAB 화면 캡처 프로그램
- LAB 한글-영문 번역기 작성하기

이러한 컴포넌트들을 사용하면 일반 프로그램을 작성할 수 있나요?

그럼요. 스윙에서 제공하는 것만 사용해도 충분합니다. 물론 추가적으로 컴포넌트를 제작하여 사용하는 경우도 많지만요.



스윙 컴포넌트에 이미지 표시하기

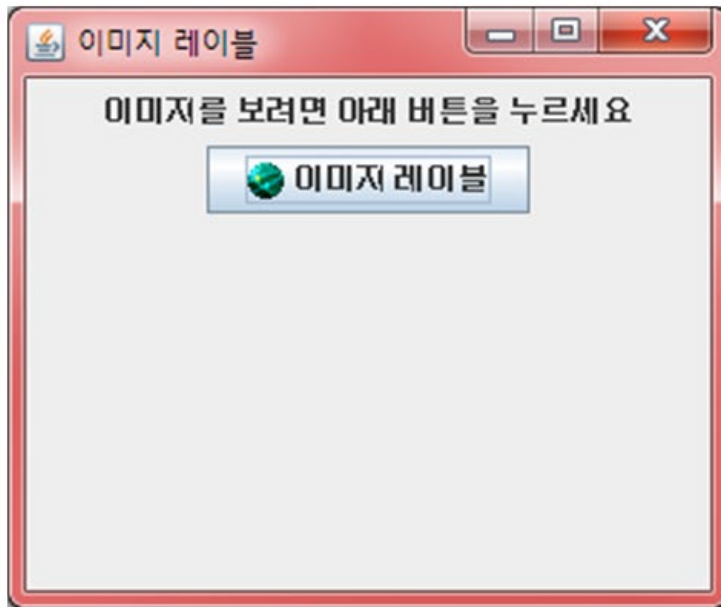
- 모든 스윙 컴포넌트에는 텍스트 옆에 이미지를 추가로 표시할 수 있다. (ImageIcon 클래스, JLabel과 JButton 등 각 컴포넌트의 **setIcon()** 메소드)



```
ImageIcon image = new ImageIcon("image.gif");  
JLabel label = new JLabel("이미지 레이블");  
label.setIcon(image);
```

예제

- 이미지 버튼을 표시하고 사용자가 버튼을 누르면 레이블의 텍스트를 이미지로 바꾸어서 표시하는 프로그램을 작성해보자.



→



예제 소스

// 소스를 입력하고 Ctrl+Shift+O를 눌러서 필요한 파일을 import한다.

```
public class ImageLabelTest extends JFrame implements ActionListener {
    private JPanel panel;
    private JLabel label;
    private JButton button;

    public ImageLabelTest() {
        setTitle("이미지 레이블");
        setSize(300,250);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        panel = new JPanel();
        label = new JLabel("이미지를 보려면 아래 버튼을 누르세요");

        button = new JButton("이미지 레이블");
        ImageIcon icon = new ImageIcon("icon.gif");
        button.setIcon(icon);
        button.addActionListener(this);
    }
}
```

예제 소스 (계속)

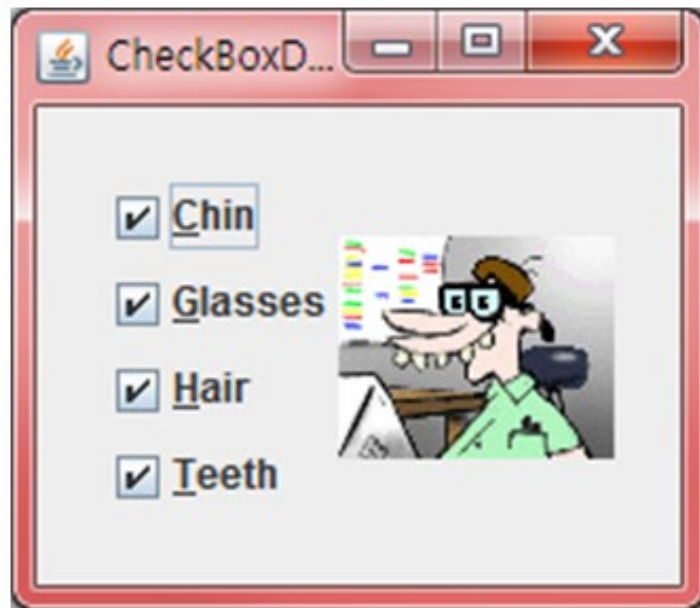
```
    panel.add(label);  
    panel.add(button);  
    add(panel);  
    setVisible(true);  
}
```

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

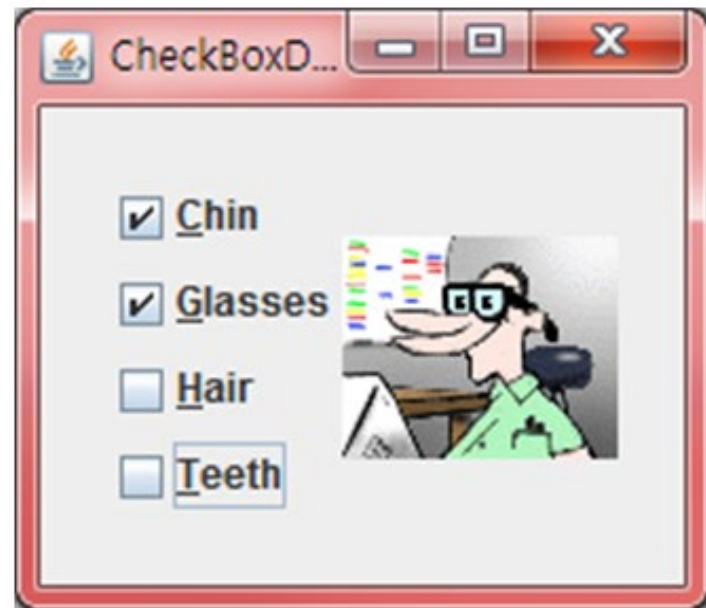
```
public void actionPerformed(ActionEvent e) {  
    ImageIcon dog = new ImageIcon("dog.gif");  
  
    label.setIcon(dog);  
    label.setText(null);  
}  
}
```

체크 박스

- 체크 박스(Check Box)란 사용자가 클릭하여 체크된 상태와 체크되지 않은 상태 중의 하나로 만들 수 있는 컨트롤 (JCheckBox 클래스)



->

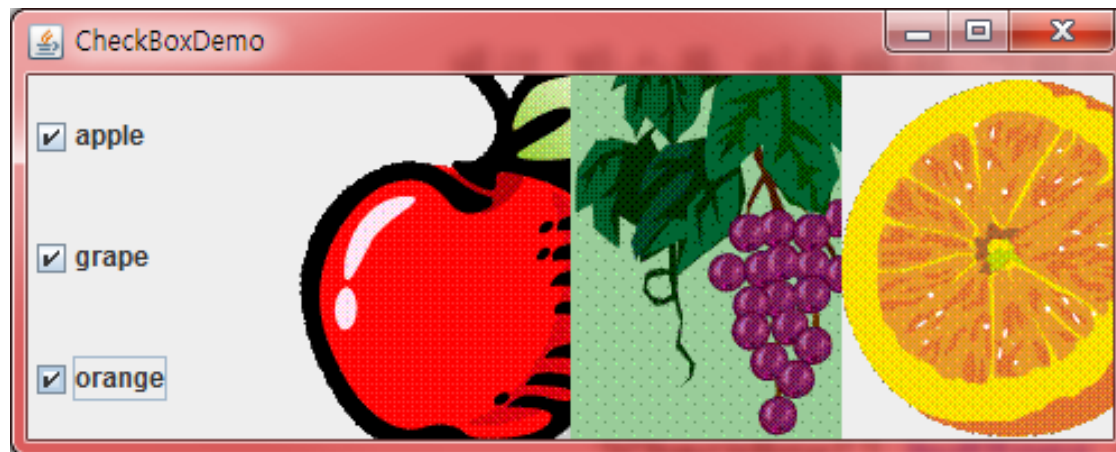


JCheckBox의 메소드

이름		설명
생성자	JCheckBox()	레이블이 없는 check box를 생성한다.
	JCheckBox(String text)	지정된 텍스트의 check box를 생성한다.
	JCheckBox(String text, boolean selected)	지정된 상태와 텍스트를 가지는 check box를 생성한다.
메소드*	String getText()	체크 박스에 표시되는 텍스트를 가져온다.
	boolean isSelected()	만약 체크 박스가 선택되었으면 true를 그렇지 않으면 false를 반환한다.
	void setSelected(boolean value)	매개 변수가 true이면 체크 박스를 체크 상태로, false이면 체크해제 상태로 만든다.
	void setText(String text)	체크 박스 텍스트를 설정한다.

* JButton, JMenuItem, JToggleButton, JCheckBox 등의 부모 클래스인 AbstractButton 클래스에서 정의된 메소드들

예제



예제 소스

// 소스를 입력하고 Ctrl+Shift+O를 눌러서 필요한 파일을 import한다.

```
public class CheckBoxPanel extends JPanel implements ItemListener {  
    JCheckBox[] buttons = new JCheckBox[3];  
    String[] fruits = { "apple", "grape", "orange" };  
    JLabel[] pictureLabel = new JLabel[3];  
    ImageIcon[] icon = new ImageIcon[3];  
}
```

- ❖ ItemListener 인터페이스 : ItemEvent 처리(476쪽), 선택 가능한 컴포넌트에서 사용자가 item을 선택 혹은 선택 해제했을 때 발생
void itemStateChanged(ItemEvent e);

예제 소스

```
public CheckBoxPanel() {  
    super(new GridLayout(0, 4));  
    // 체크 박스 생성  
    for (int i = 0; i < 3; i++) {  
        buttons[i] = new JCheckBox(fruits[i]);  
        buttons[i].addItemListener(this);  
        pictureLabel[i] = new JLabel(fruits[i] + ".gif");  
        icon[i] = new ImageIcon(fruits[i] + ".gif");  
    }  
  
    JPanel checkPanel = new JPanel(new GridLayout(0, 1));  
    for (int i = 0; i < 3; i++)  
        checkPanel.add(buttons[i]);  
  
    add(checkPanel);  
    add(pictureLabel[0]);  
    add(pictureLabel[1]);  
    add(pictureLabel[2]);  
}
```



예제 소스 (계속)

```
/** 체크 박스의 아이템 이벤트를 처리한다. */  
public void itemStateChanged(ItemEvent e) {  
    ImageIcon image = null;  
    Object source = e.getItemSelectable(); // item 이벤트 소스를 반환  
  
    for (int i = 0; i < 3; i++) {  
        if (source == buttons[i]) {  
            if (e.getStateChange() == ItemEvent.DESELECTED)  
                pictureLabel[i].setIcon(null);  
            else  
                pictureLabel[i].setIcon(icon[i]);  
            break;  
        }  
    }  
}
```

❖ ItemEvent 클래스

ItemSelectable getItemSelectable() : 이벤트 소스 반환

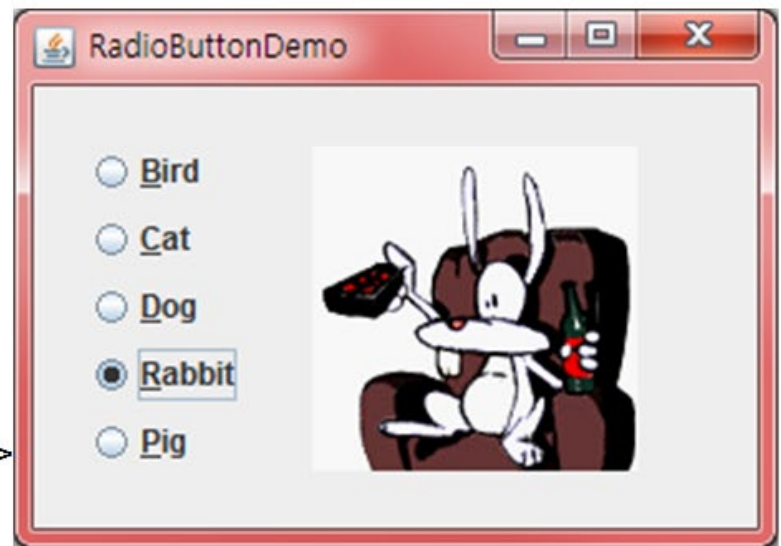
int getStateChange() : 상태 변환 타입 반환 (선택 혹은 선택해제)

예제 소스 (계속)

```
public static void main(String[] args) {  
    JFrame frame = new JFrame("CheckBoxDemo");  
  
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
  
    CheckBoxPanel panel = new CheckBoxPanel();  
    panel.setOpaque(true); // if true, paints every pixel within panel's bound  
    frame.add(panel);  
    frame.setSize(500, 200);  
    frame.setVisible(true);  
    }  
}
```

라디오 버튼

- **라디오 버튼**(Radio Button)은 체크 박스와 비슷하지만 하나의 그룹 안에서는 한 개의 버튼만 선택할 수 있다는 점이 다르다. (JRadioButton, ButtonGroup 클래스)

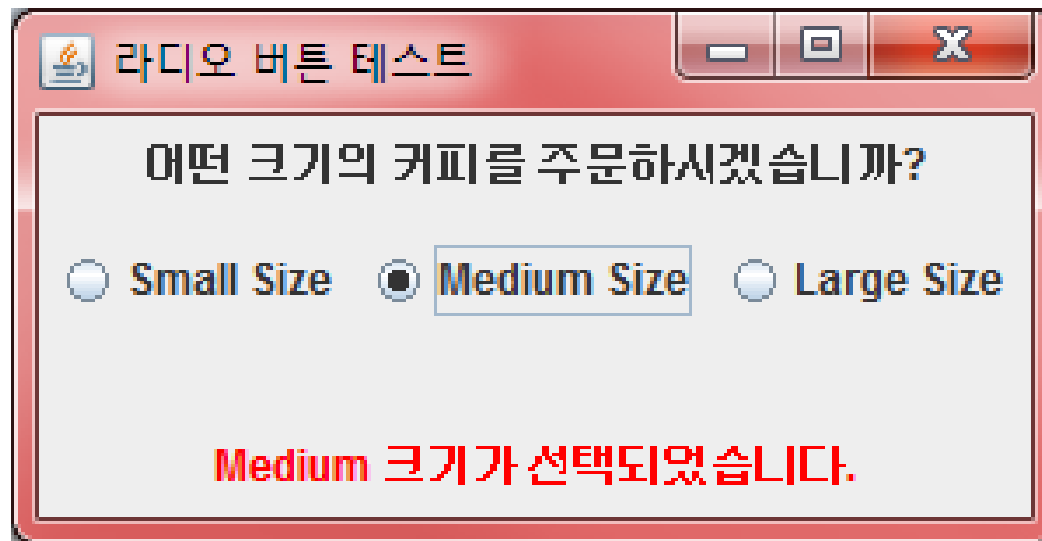


라디오 버튼

- 라디오 버튼을 생성한다.
 - ⊙ `JRadioButton radio1 = new JRadioButton("선택 1");`
 - ⊙ `JRadioButton radio2 = new JRadioButton("선택 2");`
 - ⊙ `JRadioButton radio3 = new JRadioButton("선택 3");`
- `ButtonGroup` 객체를 생성한다.
 - ⊙ `ButtonGroup group = new ButtonGroup();`
- 라디오 버튼들을 `ButtonGroup` 객체에 추가한다.
 - ⊙ `group.add(radio1);`
 - ⊙ `group.add(radio2);`
 - ⊙ `group.add(radio3);`

예제

- 커피의 크기를 선택하는 다음과 같은 화면을 라디오 버튼을 이용하여서 생성하여 보자.



예제 소스

```
import javax.swing.*;
import javax.swing.border.Border;

import java.awt.event.*;
import java.awt.*;

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(300, 150);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}
```

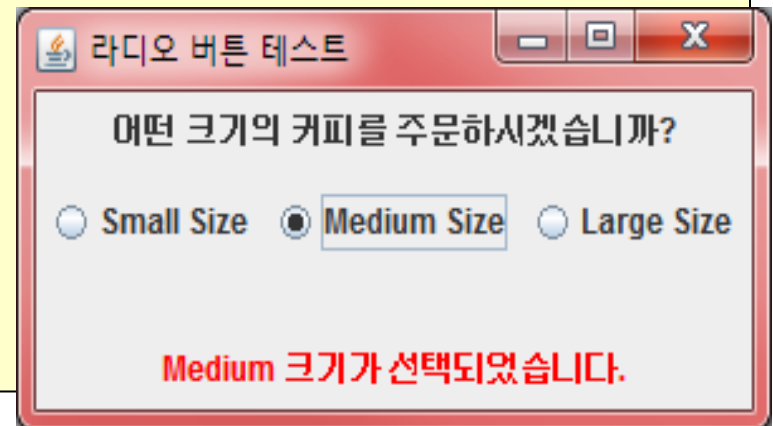

예제 소스 (계속)

```
topPanel = new JPanel();  
JLabel label = new JLabel("어떤 크기의 커피를 주문하시겠습니까?");  
topPanel.add(label);  
add(topPanel, BorderLayout.NORTH);
```

```
sizePanel = new JPanel();  
small = new JRadioButton("Small Size");  
medium = new JRadioButton("Medium Size");  
large = new JRadioButton("Large Size");
```

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

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



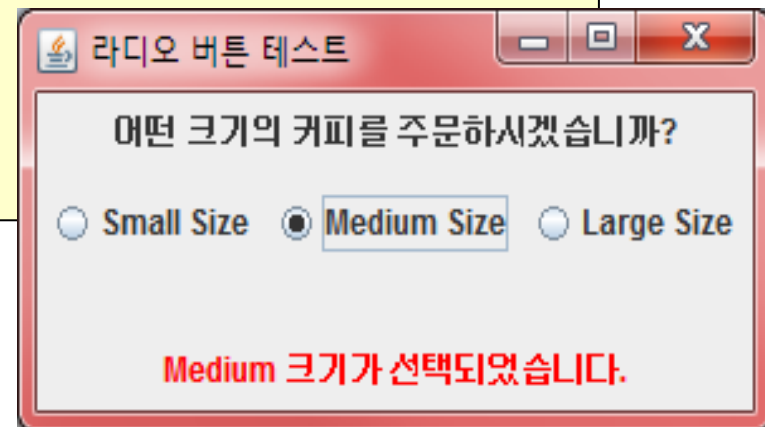
예제 소스 (계속)

```
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);
```

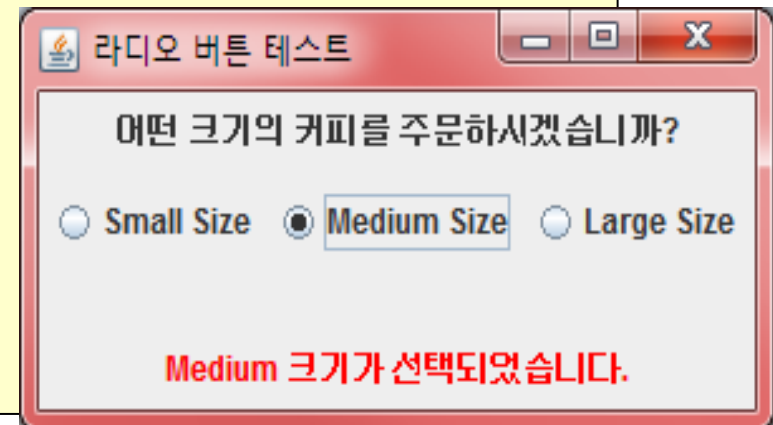
```
add(resultPanel, BorderLayout.SOUTH);  
setVisible(true);
```

```
}
```



예제 소스 (계속)

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



경계 만들기

- **경계(border)**란 시각적으로 컴포넌트들을 그룹핑할 때 사용하는 장식적인 요소 (**Border 인터페이스**)
 - ⊙ `Border border = BorderFactory.createTitledBorder("크기");`
 - ⊙ `sizePanel.setBorder(border);`



텍스트 필드

- 텍스트 필드(text field)는 입력이 가능한 한 줄의 텍스트 필드를 만드는 데 사용된다. (JTextField 클래스)

JTextField:	<input type="text" value="김철수"/>
JPasswordField:	<input type="password" value="****"/>
JFormattedTextField:	<input type="text" value="2009. 3. 7"/>

// 30자 크기의 텍스트 필드를 만든다.

```
JTextField textField1 = new JTextField(30);
```

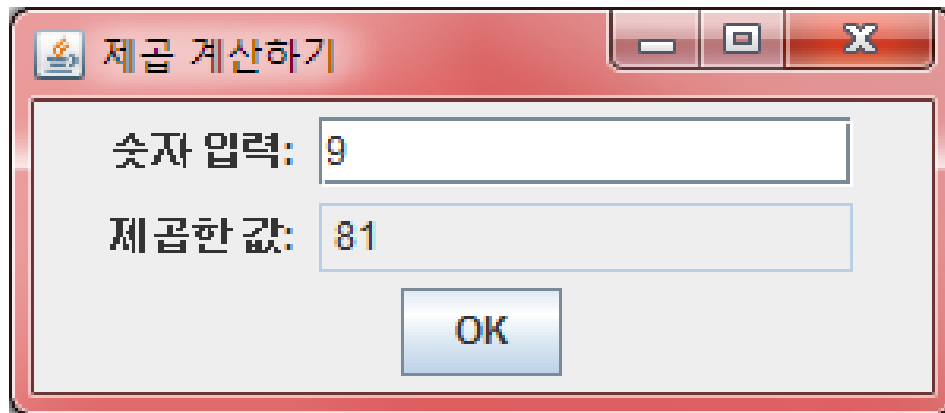
```
JTextField textField2 = new JTextField("Initial String"); // 초기화 문자열
```

```
System.out.println(textField2.getText());
```

```
textField1.setText("Seoul");
```

예제

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



예제 소스

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

```
class TextFieldFrame extends JFrame {  
    private JButton button;  
    private JTextField text, result;
```

```
    public TextFieldFrame() {  
        setSize(300, 130);  
        setTitle("제공 계산하기");  
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

```
        ButtonListener listener = new ButtonListener(); // 리스너 객체 생성
```

예제 소스 (계속)

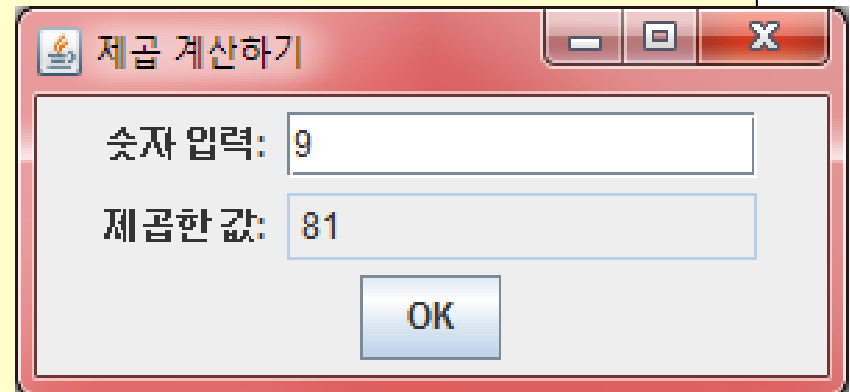
```
JPanel panel = new JPanel();  
panel.add(new JLabel("숫자 입력: "));    // 레이블 생성  
text = new JTextField(15);             // 컬럼수가 15인 텍스트 필드 생성  
text.addActionListener(listener);      // 텍스트 필드에 리스너 연결  
panel.add(text);
```

```
panel.add(new JLabel("제공한 값: "));  
result = new JTextField(15);           // 결과를 나타낼 텍스트 필드  
result.setEditable(false);             // 편집 불가 설정  
panel.add(result);
```

```
button = new JButton("OK");  
button.addActionListener(listener);  
panel.add(button);
```

```
add(panel);  
setVisible(true);
```

```
}
```



예제 소스 (계속)

// 텍스트 필드와 버튼의 액션 이벤트 처리 (내부 클래스로 정의)

```
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(); // text에 input focus를 설정  
        }  
    }  
}  
  
public static void main(String[] args) {  
    new TextFieldFrame();  
}  
}
```

패스워드 필드

- 패스워드 필드(Password Field)에 사용자가 암호를 입력하면 글자들이 모두 * 문자로 표시된다. (JPasswordField)

// 20자 크기의 패스워드 필드를 만든다.

```
JPasswordField pwdField1 = new JPasswordField(20);
```

```
JPasswordField pwdField2 = new JPasswordField("Initial"); // 초기화 문자열
```

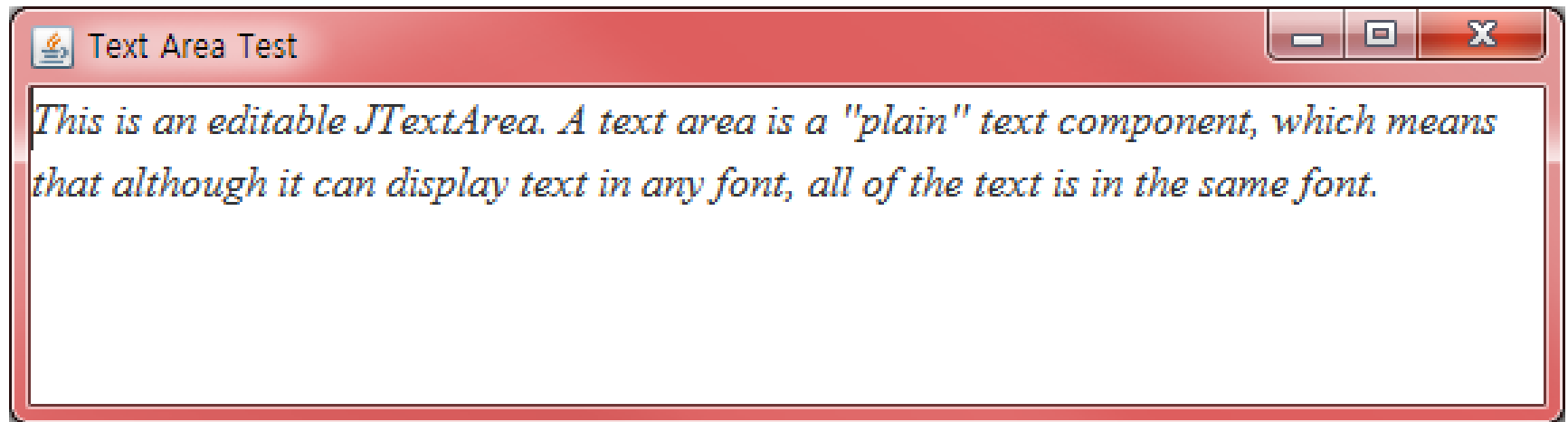
```
char[] password = pwdField2.getPassword();
```

```
pwdField2.setEchoChar('#');
```

메소드	설명
void setEchoChar(char c)	화면에 대신 보여주는 문자를 지정한다.
char getEchoChar()	화면에 대신 보여주는 문자를 반환한다.
char[] getPassword()	사용자가 입력한 암호를 문자 배열 형태로 반환한다.

텍스트 영역

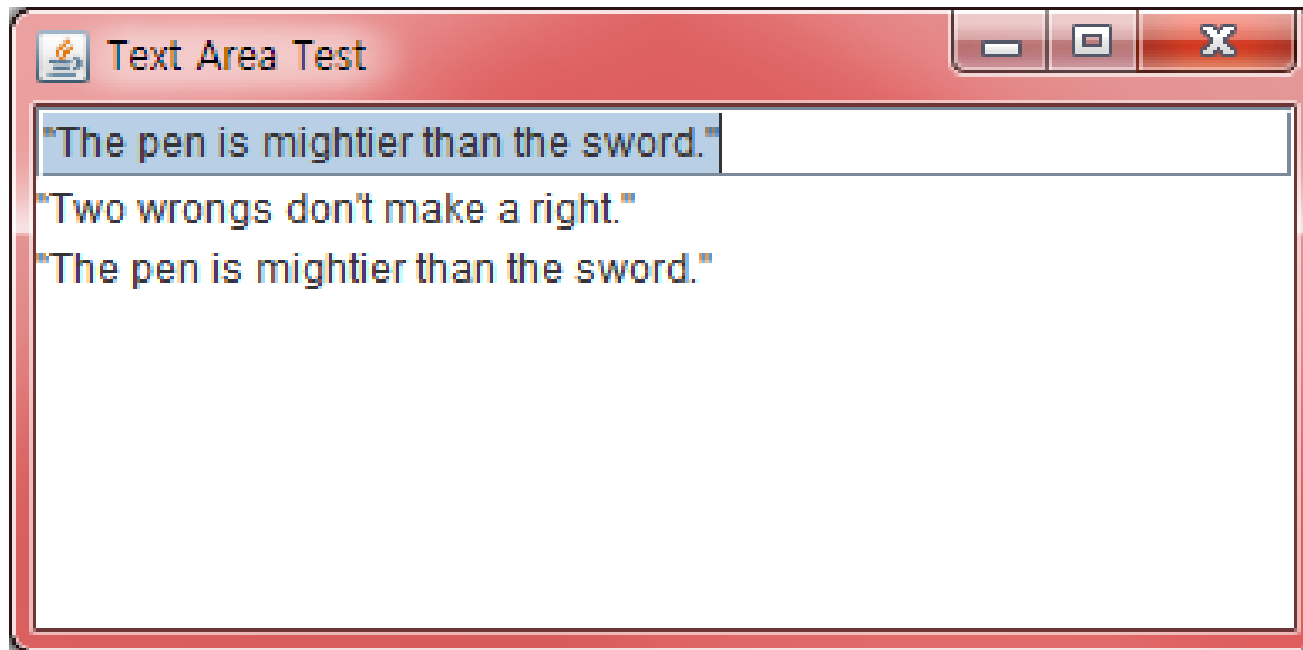
- 텍스트 영역(Text Area): 여러 줄의 텍스트가 들어 갈 수 있는 컴포넌트 (JTextArea 클래스)



생성자	설 명
<code>JTextArea()</code>	비어있는 새로운 <code>TextArea</code> 를 생성한다.
<code>JTextArea(int rows, int columns)</code>	지정된 줄수와 칸수의 비어있는 <code>TextArea</code> 를 생성한다.
<code>JTextArea(String text)</code>	지정된 <code>Text</code> 가 입력된 <code>TextArea</code> 를 생성한다.

예제

- 사용자가 텍스트 필드에 텍스트를 입력하고 [Enter] 키를 누르면 이것을 텍스트 영역에 추가하는 프로그램을 작성하여 보자.



예제 소스

// 소스를 입력하고 Ctrl+Shift+O를 눌러서 필요한 파일을 import한다.

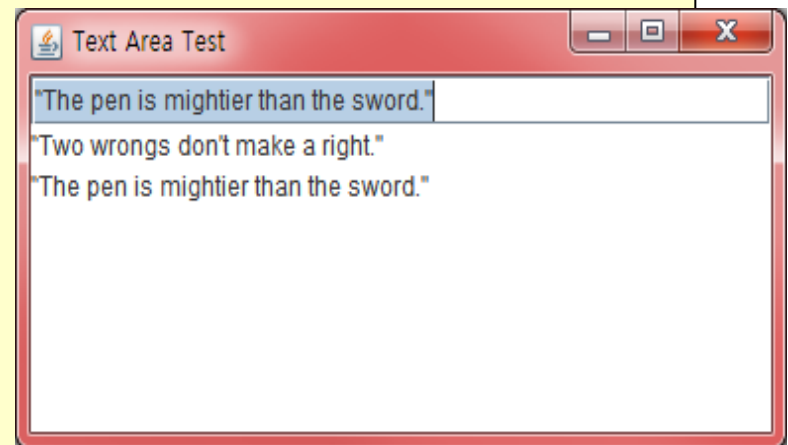
```
public class TextAreaFrame extends JFrame implements ActionListener {  
    protected JTextField textField;  
    protected JTextArea textArea;
```

```
    public TextAreaFrame() {  
        setTitle("Text Area Test");  
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

```
        textField = new JTextField(30);  
        textField.addActionListener(this);
```

```
        textArea = new JTextArea(10, 30);  
        textArea.setEditable(false);
```

```
        add(textField, BorderLayout.NORTH);  
        add(textArea, BorderLayout.CENTER);
```



예제 소스 (계속)

```
    pack();
    setVisible(true);
}

public void actionPerformed(ActionEvent e) {
    String text = textField.getText();

    textArea.append(text + "\n");
    textField.selectAll();
    textArea.setCaretPosition(textArea.getDocument().getLength());
}

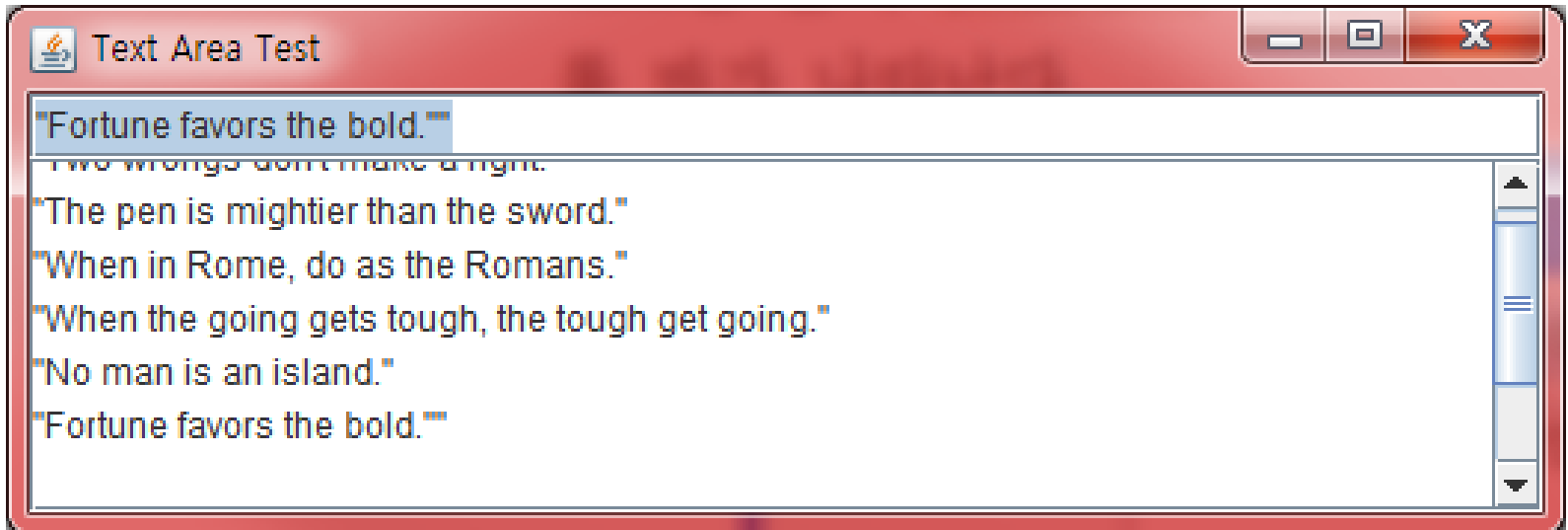
public static void main(String[] args) {
    new TextAreaFrame();
}
}
```

- `public void setCaretPosition(int position)`
Sets the position of the text insertion caret for the `TextComponent`.

스크롤 페인

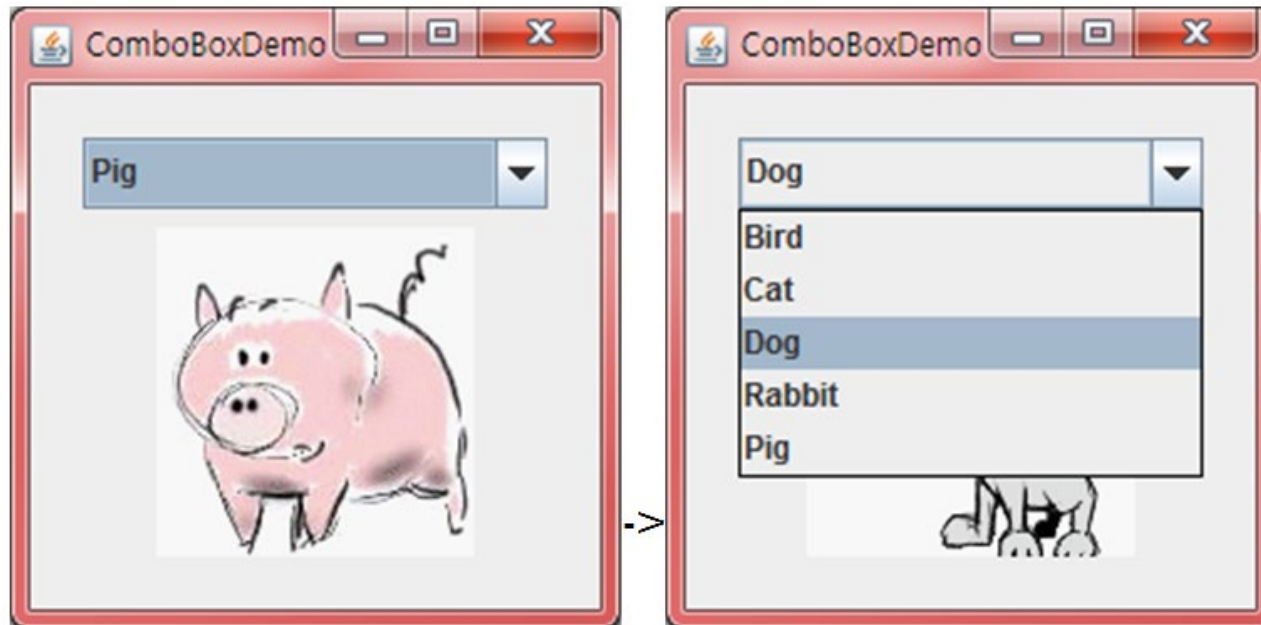
- 텍스트 영역에 스크롤바를 만들려면 **스크롤 페인(ScrollPane)**에 텍스트 영역을 넣어야 한다. (JScrollPane 클래스)

```
textArea = new JTextArea(10, 30); // 텍스트 영역을 생성한다.  
JScrollPane scrollPane = new JScrollPane(textArea);
```



콤보박스

- 콤보 박스(Combo Box)도 여러 항목 중에서 하나를 선택 하는데 사용할 수 있다. (JComboBox 클래스)
 - ◎ 텍스트 박스와 리스트의 결합



콤보박스 메소드

- 콤보 박스를 생성하기 위해서는 먼저 생성자 중에서 하나를 골라서 호출하여야 한다. 기본 생성자는 비어 있는 콤보 박스를 생성한다.

```
JComboBox combo = new JComboBox();
```

- 여기에 항목을 추가하려면 **addItem()** 메소드를 사용한다.

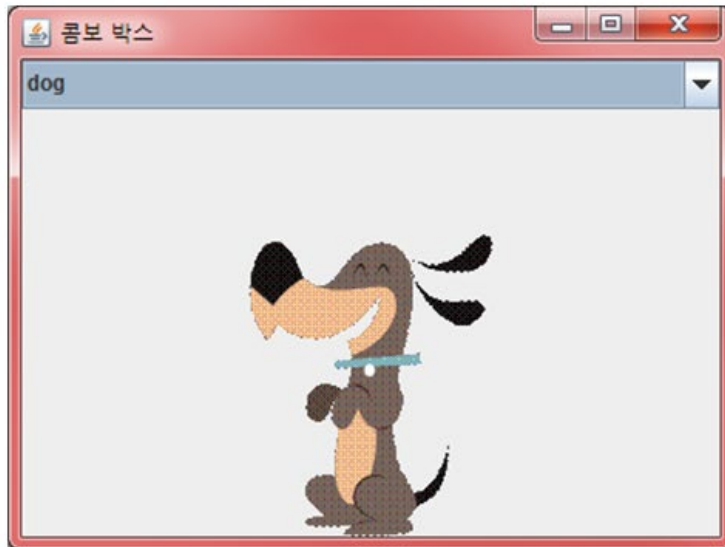
```
combo.addItem("dog");
```

```
combo.addItem("lion");
```

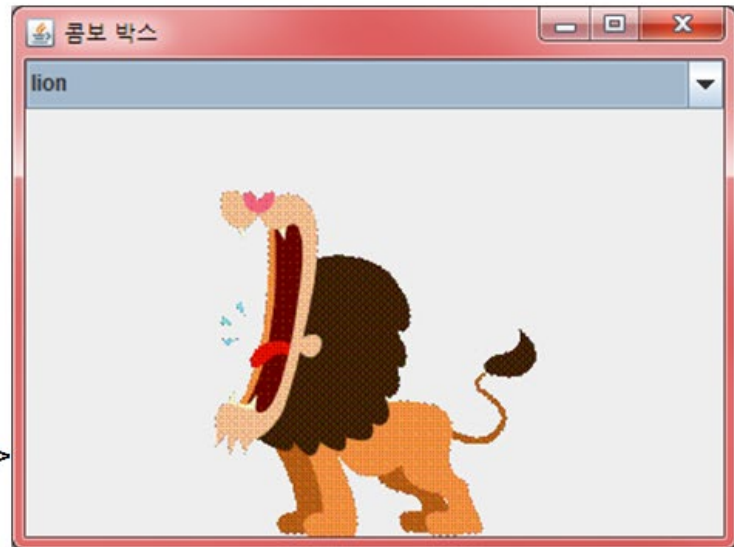
```
combo.addItem("tiger");
```

예제

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



→



예제 소스

// 소스를 입력하고 Ctrl+Shift+O를 눌러서 필요한 파일을 import한다.

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

```
    public ComboBoxFrame() {  
        setTitle("콤보 박스");  
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        setSize(300, 200);  
  
        String[] animals = { "dog", "lion", "tiger" };  
        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.PAGE_START);  
        add(label, BorderLayout.PAGE_END);  
        setVisible(true);  
    }
```

예제 소스 (계속)

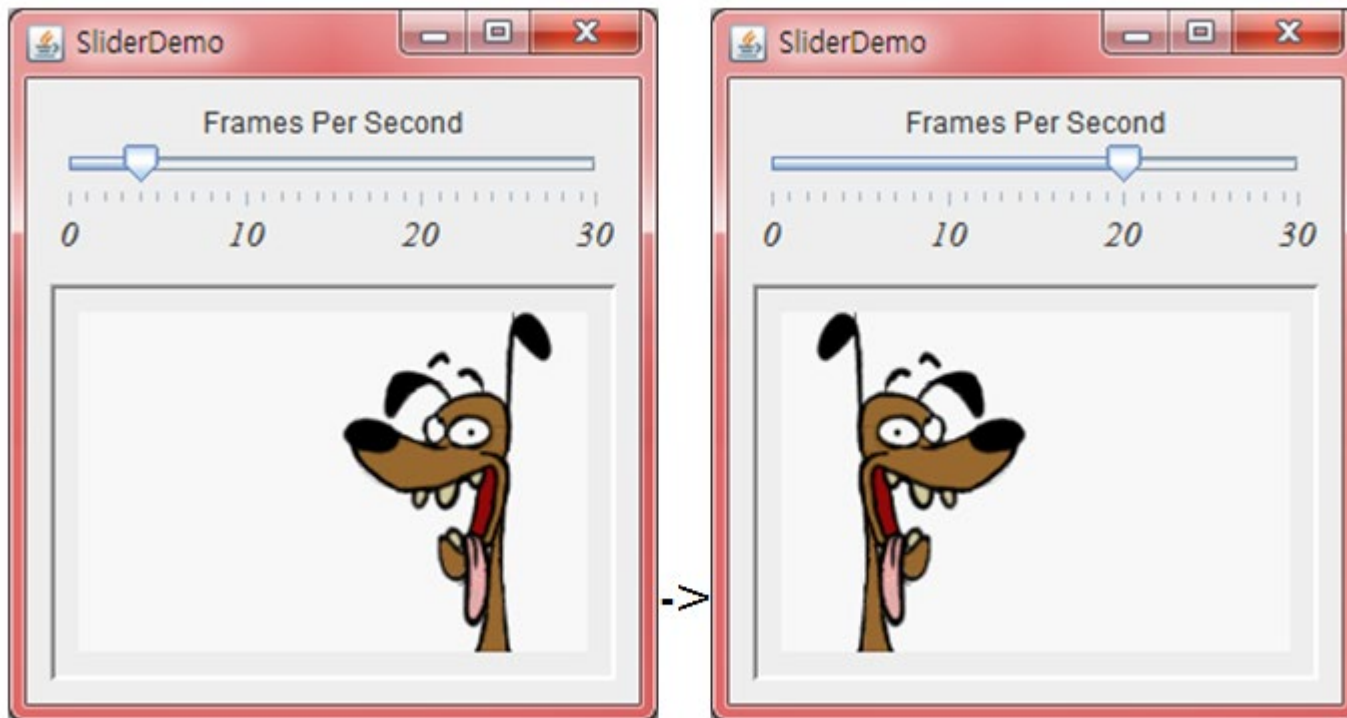
```
public void actionPerformed(ActionEvent e) {
    JComboBox cb = (JComboBox) e.getSource();
    String name = (String) cb.getSelectedItem();
    changePicture(name);
}

protected void changePicture(String name) {
    ImageIcon icon = new ImageIcon(name + ".gif");
    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();
}
}
```

슬라이더

- 슬라이더(Slider)는 사용자가 특정한 범위 안에서 하나의 값을 선택할 수 있는 컴포넌트이다. (JSlider 클래스)



예제

- 다음 예제에서는 슬라이더를 움직이면 표시되는 이미지의 크기가 변경된다.



예제 소스

// 소스를 입력하고 Ctrl+Shift+O를 눌러서 필요한 파일을 import한다.

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

    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);
    }
}
```

❖ ChangeEvent 클래스

❖ ChangeListener 인터페이스 : void stateChanged(ChangeEvent e);

예제 소스 (계속)

```
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);

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);
}
```

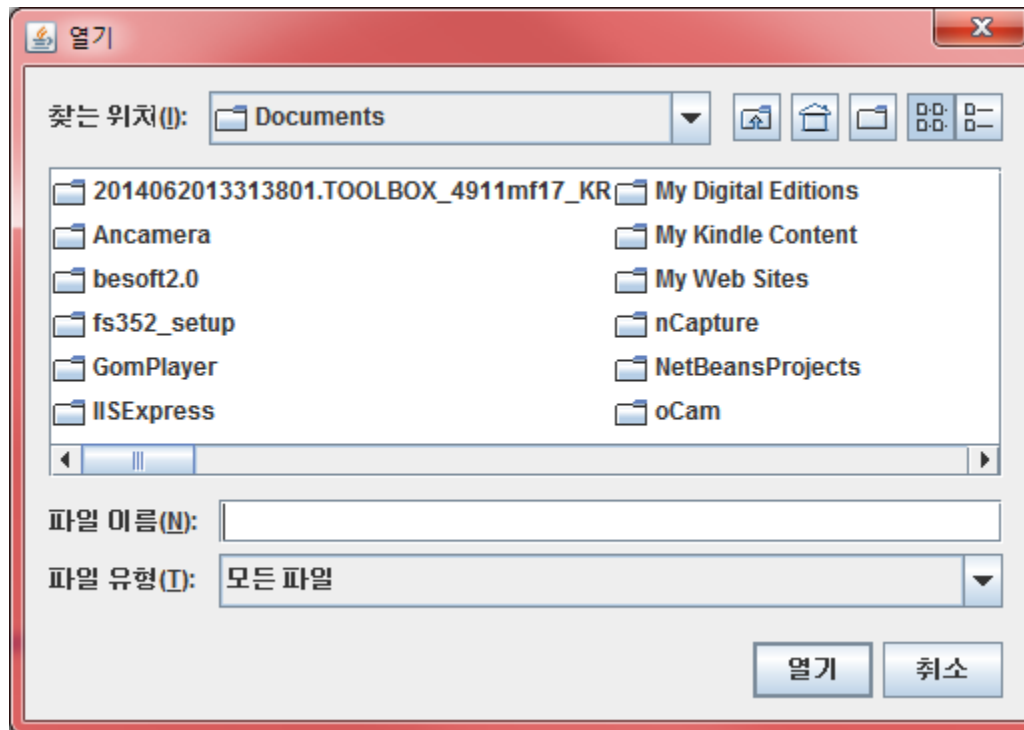

예제 소스 (계속)

```
public void stateChanged(ChangeEvent e) {  
    JSlider source = (JSlider) e.getSource();  
  
    if (!source.getValueIsAdjusting()) { // knob의 드래그가 끝나면(false)  
        int value = (int) source.getValue();  
        button.setSize(value * 10, value * 10);  
    }  
}  
  
public static void main(String[] args) {  
    new SliderFrame();  
}
```



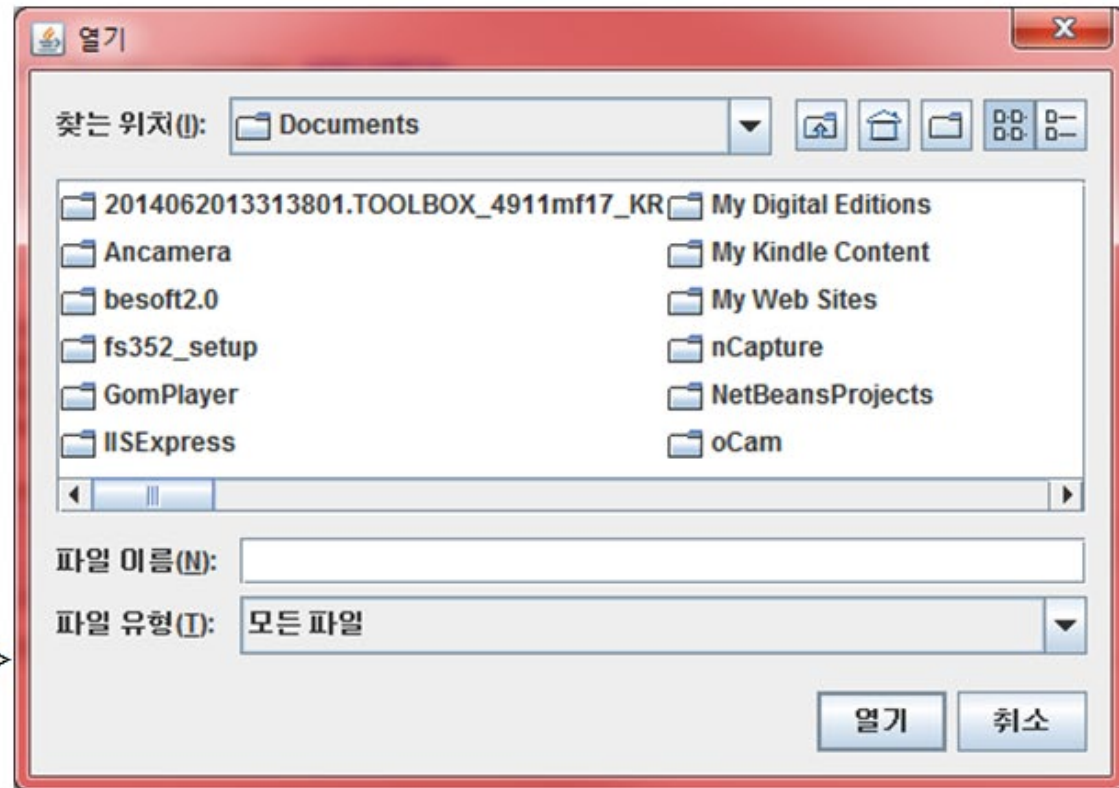
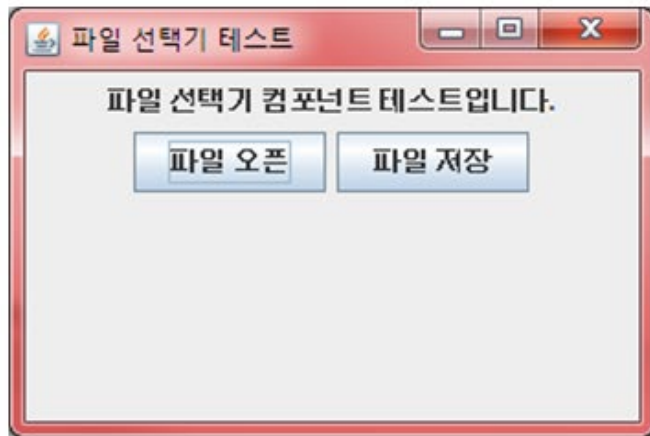
파일 선택기

- **파일 선택기**(File Chooser)는 파일 시스템을 탐색하여 파일이나 디렉토리(폴더)를 선택하는 GUI를 제공한다. 파일 선택기를 표시하려면 JFileChooser 클래스를 사용한다.



예제

- 사용자가 "파일 오픈" 혹은 "파일 저장" 버튼을 누르면 파일을 선택하는 대화 상자가 나오도록 프로그램을 작성해보자.



예제 소스

// 소스를 입력하고 Ctrl+Shift+O를 눌러서 필요한 파일을 import한다.

```
public class FileChooserTest extends JFrame implements ActionListener {
    JButton openButton, saveButton;
    JFileChooser fc;

    public FileChooserTest() {
        setTitle("파일 선택기 테스트");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setSize(300, 200);

        fc = new JFileChooser();

        JLabel label = new JLabel("파일 선택기 컴포넌트 테스트입니다.");
        openButton = new JButton("파일 오픈");
        openButton.addActionListener(this);

        saveButton = new JButton("파일 저장");
        saveButton.addActionListener(this);
    }
}
```

예제 소스 (계속)

```
JPanel panel = new JPanel();
panel.add(label);
panel.add(openButton);
panel.add(saveButton);

add(panel);
setVisible(true);
}

public void actionPerformed(ActionEvent e) {
    // “파일 오픈”버튼에 대한 액션 이벤트 처리
    if (e.getSource() == openButton) {
        int returnVal = fc.showOpenDialog(this);

        if (returnVal == JFileChooser.APPROVE_OPTION) {
            File file = fc.getSelectedFile();
            // 실제 파일을 오픈한다.
        } else {
            // 사용자 취소
        }
    }
}
```

예제 소스 (계속)

```
// “파일 저장”버튼에 대한 액션 이벤트 처리
} else if (e.getSource() == saveButton) {
    int returnVal = fc.showSaveDialog(this);

    if (returnVal == JFileChooser.APPROVE_OPTION) {
        File file = fc.getSelectedFile();
        // 실제 파일에 저장한다.
    } else {
        // 사용자 취소
    }
}

}

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

LAB: 피자 주문 화면

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

윈도우 제목

피자 주문

자바 피자에 오신 것을 환영합니다.

종류

- 콤보
- 포테이토
- 불고기

추가 토핑

- ☐ 피망
- ☐ 치즈
- ☐ 페페로니
- ☐ 베이컨

크기

- Small
- Medium
- Large

주문 취소

(1) WelcomePanel

```
class WelcomePanel extends JPanel
{
    private JLabel message;
    public WelcomePanel()
    {
        message = _____; // 레이블 생성
    }
    add(message);
}
```


(2) JPanel

```
class JPanel extends JPanel {  
    private JRadioButton combo, potato, bulgogi;  
  
    public JPanel() {  
        setLayout(new GridLayout(3, 1);  
  
        combo = new JRadioButton("콤보", true);  
        potato = new _____("포테이토");  
        bulgogi = new _____("불고기");  
        ButtonGroup bg = new ButtonGroup();  
        bg.add(combo);  
        bg.add(potato);  
        bg.add(bulgogi);  
  
        setBorder(BorderFactory.createTitledBorder("종류"));  
        add(combo);  
        add(potato);  
        add(bulgogi);  
    }  
}
```

(3) ToppingPanel – 체크박스 버튼 구현

```
class ToppingPanel extends JPanel { // 토핑 패널 정의 JPanel 상속

    private JCheckBox pepper, cheese, peperoni, bacon; // 체크박스 버튼

    public ToppingPanel() { // 생성자
        setLayout(new GridLayout(4, 1)); // 배치관리자 일렬

        pepper = new JCheckBox("피망", true); // 체크박스 버튼 생성
        cheese = new JCheckBox("치즈");
        peperoni = new JCheckBox("페페로니");
        bacon = new JCheckBox("베이컨");

        ...
    }
}
```

(4) SizePanel

```
class SizePanel extends JPanel { // 사이즈 패널 정의 JPanel 상속
    private JRadioButton samll, medium, large; // 라디오 버튼 참조변수
    private ButtonGroup bg; // 버튼 그룹 참조변수

    public SizePanel() { // 생성자
        setLayout(new GridLayout(3, 1)); // 배치관리자

        samll = new JRadioButton("Small", true); // 라디오 버튼 생성
        medium = new JRadioButton("Medium");
        large = new JRadioButton("Large");
        bg = new ButtonGroup(); // 버튼 그룹 생성
        bg.add(samll);
        bg.add(medium);
        bg.add(large);

        ...
    }
}
```

(5) 결합하기 (소스: 586-589쪽)

WelcomePanel을 BorderLayout의 North에 배치

자바 피자에 오신 것을 환영합니다.

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

주문 취소

TypePanel을 BorderLayout의 West에 배치

ToppingPanel을 BorderLayout의 Center에 배치

SizePanel을 BorderLayout의 East에 배치

LAB: 한글-영문 번역기 작성

- 왼쪽 텍스트 영역에 사용자가 한글을 입력하고 "변환" 버튼을 누르면 미리 입력된 몇 개의 단어만을 영어로 변환한다.

윈도우 제목

텍스트 변환

텍스트 영어

Text English

변환 취소

Solution

// 소스를 입력하고 Ctrl+Shift+O를 눌러서 필요한 파일을 import한다.

```
public class TextConverter extends JFrame {  
    JButton converter;  
    JButton canceler;  
    JTextArea textIn;  
    JTextArea textOut;  
  
    public TextConverter() {  
        super("텍스트 변환");  
  
        // 텍스트 영역  
        textIn = new JTextArea(10, 14);  
        textOut = new JTextArea(10, 14);  
        textIn.setLineWrap(true); // 자동 줄바꿈  
        textOut.setLineWrap(true);  
        textOut.setEnabled(false); // 비활성화
```

Solution (계속)

// 텍스트 영역을 관리할 패널

```
JPanel textAreaPanel = new JPanel(new GridLayout(1, 2, 20, 20));  
textAreaPanel.add(textIn);  
textAreaPanel.add(textOut);
```

// 버튼

```
converter = new JButton("변환");  
canceler = new JButton("취소");  
converter.addActionListener(new ActionListener());  
canceler.addActionListener(new ActionListener());
```

// 버튼 패널

```
JPanel buttonPanel = new JPanel(new FlowLayout());  
buttonPanel.add(converter);  
buttonPanel.add(canceler);
```

// 메인 패널

```
JPanel mainPanel = new JPanel(new BorderLayout(10, 10));
```

Solution (계속)

```
mainPanel.add(BorderLayout.CENTER, textAreaPanel);  
mainPanel.add(BorderLayout.SOUTH, buttonPanel);
```

```
// 프레임 설정
```

```
setLayout(new FlowLayout(FlowLayout.CENTER, 20, 20));  
add(mainPanel);  
pack();  
setDefaultCloseOperation(EXIT_ON_CLOSE);  
setVisible(true);  
}
```

```
// 버튼의 액션 이벤트를 처리 할 버튼 액션 리스너 클래스
```

```
private class ButtonActionListener implements ActionListener {  
    public void actionPerformed(ActionEvent e) {  
        // TODO Auto-generated method stub  
        if (e.getSource() == converter) {  
            textOut.setText("");  
            String result = toEnglish(textIn.getText());  
            textOut.append(result);  
        }  
    }  
}
```


Solution (계속)

```
        if (e.getSource() == canceler) {  
            textOut.setText("");  
        }  
    }
```

// 한국어를 영어로 변환하는 메소드

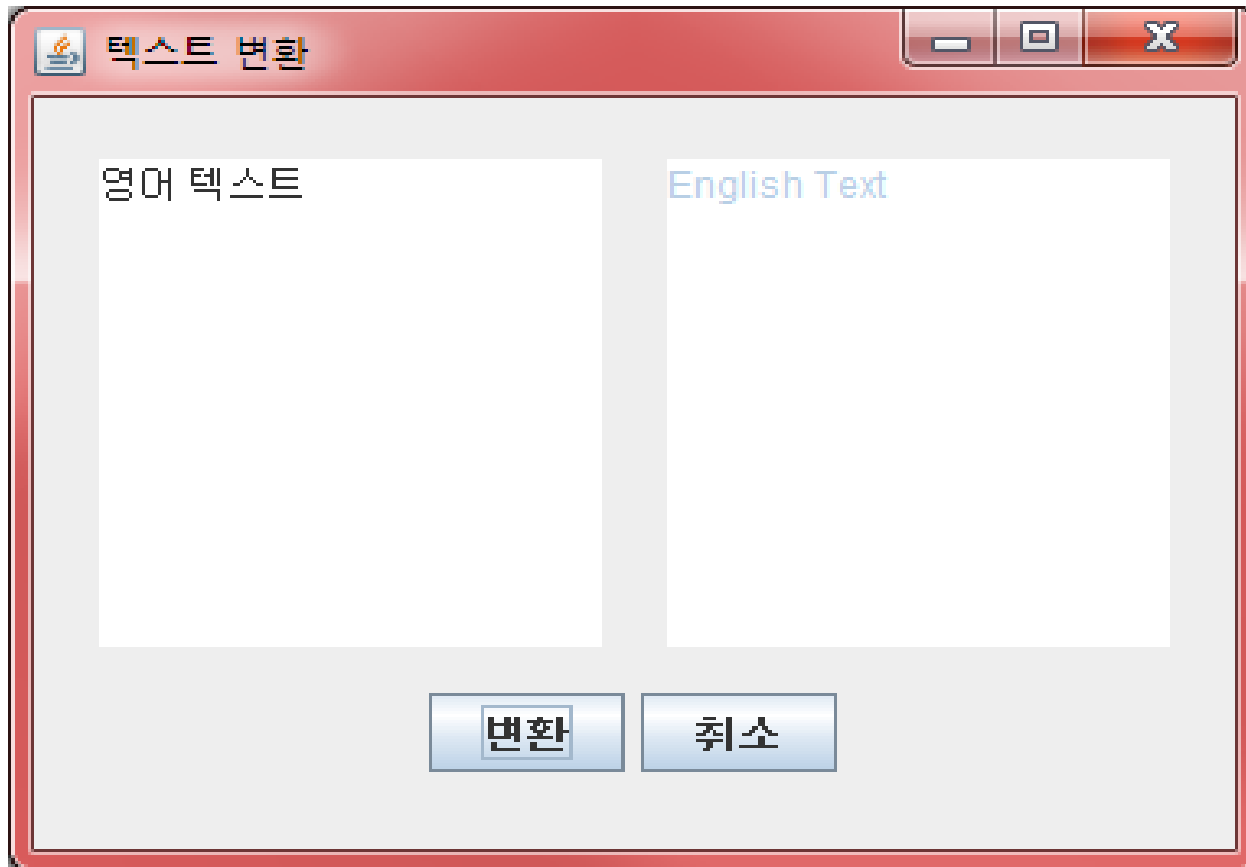
```
private String toEnglish(String korean) {  
    String result = korean;  
    result = result.replace("텍스트", "Text");  
    result = result.replace("영어", "English");  
    return result;  
}
```

```
}
```

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

```
}
```

실행 결과



제12장 스윙 컴포넌트 - 끝

학습목차

- 01 스윙 컴포넌트에 이미지 표시하기
- 02 체크 박스
- 03 라디오 버튼
- 04 텍스트 필드
- 05 텍스트 영역
- 06 콤보 박스
- 07 슬라이더
- 08 파일 선택기
- LAB 피자 주문 화면 만들기
- LAB 화면 캡처 프로그램
- LAB 한글-영문 번역기 작성하기

이러한 컴포넌트들을 사용하면 일반 프로그램을 작성할 수 있나요?

그럼요. 스윙에서 제공하는 것만 사용해도 충분합니다. 물론 추가적으로 컴포넌트를 제작하여 사용하는 경우도 많지만요.

