

Chapter 11 Swing Component

11.1 Swing Component 소개

- Swing Component: Window 기반 응용 프로그램을 만드는 데 사용되는 Java Foundation Class(JFC)
- JFC: Desktop Application 개발을 쉽게 하는 GUI component 집합
- Swing component는 Abstract Windowing Toolkit(AWT) API를 기반으로 하여 완전히 자바로 작성되었다.
- javax.swing package는 JButton, JTextField, JTextArea, JRadioButton, JCheckbox, JMenu, JColorChooser 등과 같은 클래스를 제공한다.

JComponent class

- JComponent: Swing Component들이 공통적으로 가져야 하는 속성과 메소드를 가지고 있다.
- 최상위 Component를 제외하고 거의 모든 Component가 JComponent를 상속받는다.
- JComponent Class는 Component와 Container Class를 상속하므로 스윙 컴포넌트들은 이 클래스들이 제공하는 기능을 사용할 수 있다.
- Component Class: 화면 그리기와 이벤트 처리 기능을 제공한다.
- Container Class: 주로 컨테이너 클래스에 컴포넌트를 추가하거나 제거하는 기능과 레이아웃 기능을 제공한다.

컴포넌트의 외관 변경

Method	Description
setBorder(Border), getBorder()	컴포넌트들의 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.

컴포넌트들의 상태 변경

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

공통 이벤트 처리

Method	Description
addMouseListener(MouseListener), removeMouseListener(MouseListener)	마우스 리스너를 추가하거나 제거한다.

11.2 Label과 Button의 고급 기능

Method	Description
repaint()	컴포넌트의 일부나 전체를 다시 그리라고 요청한다.
revalidate()	컨테이너 안의 컴포넌트를 다시 배치하라고 요청한다.
paintComponent(Graphics)	컴포넌트를 그린다. 컴포넌트에서는 이 메소드를 재정의하여서 컴포넌트 위에 그림을 그릴 수 있다.

컨테이너에 추가/제거

Method	Description
add(Component)	컴포넌트를 컨테이너에 추가한다.
remove(Component)	컨테이너에서 컴포넌트를 삭제한다.
getParent()	컴포넌트의 컨테이너를 반환한다.

레이아웃

Method	Description
getWidth(), getHeight()	픽셀 단위의 컴포넌트 크기를 반환한다.
getSize()	픽셀 단위의 컴포넌트 크기를 반환한다.
getX(), getY()	부모 컨테이너 안 컴포넌트의 상대적인 위치를 반환한다.
getLocation()	부모 컨테이너 안에서의 상대적인 위치를 반환한다.
getLocation(int, int)	부모 컨테이너 안에서의 컴포넌트의 좌표를 지정한다.
setBounds(int, int, int, int)	부모 컨테이너 안에서의 크기와 위치를 지정한다.

크기와 위치 정보 얻기

레이블의 색상과 폰트 변경하기

Method	Description
setPreferredSize(Dimension), setMaximumSize(Dimension), setMinimumSize(Dimension)	컴포넌트의 크기를 설정한다.
setAlignmentX(float), setAlignment(float)	컨테이너 안에서 컴포넌트들의 정렬을 지정한다.
setLayout(LayoutManager), getLayout()	배치 관리자를 설정한다.

예제 11-1 레이블의 색상과 폰트 변경하기

```

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.JLabel;
import java.awt.Color;
import java.awt.Font;

public class LabelTest1 extends JFrame {
    private JPanel panel;
    private JLabel label1, label2;

    public LabelTest1() {
        setLocation(600, 400);
        setSize(400, 150);
        setTitle("Font Test");

        panel = new JPanel();
        label1 = new JLabel("Color Label");
        label1.setForeground(Color.BLUE);

        label2 = new JLabel();
        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) {
        LabelTest l = new LabelTest();
    }
}

```

레이블에 이미지 표시하기

- ImageIcon: JPEG, GIF, PNG 이미지 파일을 읽을 수 있다.
- ImageIcon instance를 생성하고 setIcon Method를 사용하여 Label에 이미지를 지정할 수 있다.

```

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

```

예제 11-2 레이블로 이미지 표시하기

```

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.JLabel;
import javax.swing.ImageIcon;

public class ImageTest extends JFrame {
    private JPanel panel;
    private JLabel label;

    public ImageTest() {
        setLocation(200, 200);
        setSize(1000, 1000);
        setTitle("Image Test");

        panel = new JPanel();
        label = new JLabel();

        ImageIcon icon = new ImageIcon("./Retriever.jpeg");
        label.setIcon(icon);

        panel.add(label);
        add(panel);

        setVisible(true);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
    }

    public static void main(String [] args) {
        ImageTest i = new ImageTest();
    }
}

```

11.3 Text field and text Area

- Text field: 입력이 가능한 한 줄의 텍스트 필드
- JTextField 두 개의 자식 클래스를 갖는다.
- JPasswordField: 패스워드 입력처럼 사용자가 입력하는 문자를 보여주지 않는 컴포넌트
- JFormattedTextField: 사용자가 입력할 수 있는 문자 집합을 제한한다.

```

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(); // 텍스트 필드가 사용자로부터 입력을 받으려면 키보드 포커스가 있어야 한다.

```

예제 11-3 사용자로부터 받은 정수의 제곱 계산하기

```
import java.lang.Integer;

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.JButton;
import javax.swing.JLabel;
import javax.swing.JTextField;

import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;

public class TextFieldFrame extends JFrame {
    private JButton button;
    private JTextField text, result;
    private ButtonListener listener = new ButtonListener();

    public TextFieldFrame() {
        setLocation(600, 400);
        setSize(300, 150);
        setTitle("제공 계산하기");

        JPanel panel = new JPanel();

        panel.add(new JLabel("Number Input: "));
        text = new JTextField(15);
        text.addActionListener(listener);
        panel.add(text);

        panel.add(new JLabel("제공한 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);
    }

    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) {
        TextFieldFrame f = new TextFieldFrame();
    }
}

```

예제 11-4

```

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.JButton;
import javax.swing.JLabel;
import javax.swing.JTextField;
import javax.swing.JPasswordField;

public class WindowLogin extends JFrame {
    public WindowLogin() {
        setLocation(600, 400);
        setSize(230, 150);
        setTitle("로그인");

        JPanel panel = new JPanel();
        panel.add(new JLabel("ID"));
        panel.add(new JTextField(15));
        panel.add(new JLabel("PW"));
        panel.add(new JPasswordField(15));

        JButton button1 = new JButton("로그인");
        panel.add(button1);

        JButton button2 = new JButton("취소");
        panel.add(button2);

        add(panel);

        setVisible(true);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
    }

    public static void main(String [] args) {
        WindowLogin w = new WindowLogin();
    }
}

```

11.4 Check Box

- Check Box : 사용자가 클릭하여서 체크된 상태와 체크되지 않은 상태 중에 하나로 만들 수 있는 컨트롤

```
onion = new JCheckBox("양파");
cheese = new JCheckBox("치즈", true); // 생성자에서 초기 상태를 주지 않으면 체크되지 않는 체크 박스가 생성된다.
onion.setSelected(true); // 강제로 true로 설정한다.
```

```
// 익명 클래스로 ItemListener를 구현하여 ItemEvent를 처리한다.
JCheckBox check = new JCheckBox("CheckBox", false);
check.addItemListener(new ItemListener() {
    public void itemStateChanged(ItemEvent e) {
        System.out.println("체크 상태" + check.isSelected());
    }
});
```

예제 11-6 체크 박스 사용하기

```
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JCheckBox;
import java.awt.GridLayout;
import java.awt.event.ItemListener;
import java.awt.GridBagLayout;
import java.awt.event.ItemEvent;

public class CheckBoxTest1 extends JFrame {
    JLabel label;
    JCheckBox checkbox1, checkbox2;
    public CheckBoxTest1() {
        setLocation(500, 400);
        setSize(400, 200);
        setTitle("Example 11-6 Check Box Test");
        setLayout(new GridLayout(0, 1));

        label = new JLabel();
        checkbox1 = new JCheckBox("Pizza");
        checkbox2 = new JCheckBox("Spagetti");

        add(label);
        add(checkbox1);
        add(checkbox2);

        checkbox1.addItemListener(new ItemListener() {
            public void itemStateChanged(ItemEvent e) {
                label.setText("Pizza가 " + (e.getStateChange() == 1 ? "선택되었습니다." : "선택되지 않았습니다.));
            }
        });

        checkbox2.addItemListener(new ItemListener() {
```

```

        public void itemStateChanged(ItemEvent e) {
            label.setText("Spagetti가 " + (e.getStateChange() == 1 ? "선택되었습니다." : "선택되지 않았습니다."));
        }

        setVisible(true);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
    }

    public static void main(String [] args) {
        CheckBoxTest c = new CheckBoxTest();
    }
}

```

11.5 Radio Button

- Radio Button : 하나의 그룹 안에서 한 개의 버튼만 선택할 수 있고 하나의 라디오 버튼을 클릭하면 다른 버튼은 자동적으로 선택이 해제된다.

```

JRadioButton radio1 = new JRadioButton("Small Size"); // 1. 라디오 버튼을 생성한다.
JRadioButton radio2 = new JRadioButton("Medium Size");
JRadioButton radio3 = new JRadioButton("Large Size");

ButtonGroup group = new ButtonGroup(); // 2-1 버튼 그룹을 생성한다.

group.add(radio1); // 2. 버튼들을 그룹핑한다.
group.add(radio2); // 2-2 버튼 그룹에 추가한다.
group.add(radio3);

```

라디오 버튼 이벤트 처리

- 라디오 버튼이 눌러지는 순간 어떤 작업을 실행하고 싶다 -> `ActionEvent`
- 코드에서 라디오 버튼이 눌러졌는지 검사하고 싶다 -> `isSelected()`
- 강제로 어떤 버튼을 선택되게 하고 싶다 -> `doClick()`

예제 11-7

```

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.JLabel;
import javax.swing.JRadioButton;
import javax.swing.border.Border;
import javax.swing.ButtonGroup;
import java.awt.event.ActionListener;
import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.event.ActionEvent;

```



```
public class RadioButtonTest extends JFrame implements ActionListener {
    JPanel topPanel, sizePanel, resultPanel;
    JRadioButton small, medium, large;
    JLabel text;

    public void actionPerformed(ActionEvent e) {
        if (e.getSource() == small) {
            text.setText("small 사이즈가 선택되었습니다.");
        }
        else if (e.getSource() == medium) {
            text.setText("medium 사이즈가 선택되었습니다.");
        }
        else
            text.setText("Large 사이즈가 선택되었습니다.");
    }

    public RadioButtonTest() {
        setLocation(500, 400);
        setSize(400, 150);
        setTitle("Example 11-7 Coffee Ordering Page");

        topPanel = new JPanel();
        topPanel.add(new JLabel("원하시는 커피의 크기를 선택하세요."));
        add(topPanel, BorderLayout.NORTH);

        sizePanel = new JPanel();
        small = new JRadioButton("small");
        medium = new JRadioButton("medium");
        large = new JRadioButton("large");

        ButtonGroup group = new ButtonGroup();
        group.add(small);
        group.add(medium);
        group.add(large);

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

        small.addActionListener(this);
        medium.addActionListener(this);
        large.addActionListener(this);
        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);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
    }
}
```

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

11.6 Combo box

- Combo box : 여러 항목 중에서 하나를 선택하는 데 사용한다.
- 콤보 박스는 텍스트 필드와 리스트의 결합이다.

```
JComboBox combo = new JComboBox(); // 생성자는 비어 있는 콤보 박스를 생성한다.  
combo.addItem("dog"); // 항목을 추가한다.₩₩₩  
combo.addItem("lion");  
combo.addItem("tiger");
```

11.7 Menu 붙이기

11.8 Slider