

어서와 Java는 처음이지!

제8장 그래픽 사용자 인터페이스

그래픽 사용자 인터페이스는
흔히 GUI라고 불리는 것인가요?

- 그래픽 사용자 인터페이스
- 컨테이너
- GUI 작성절차
- 기초 컴포넌트들

네, GUI입니다. 최근 애플리케이션에서 GUI가 중요하죠. 또 우리가 학습하였던 객체 지향 개념들이 실제로 어떻게 적용되는지를 살펴보는 좋은 사례가 됩니다.





그래픽 사용자 인터페이스

- 그래픽 사용자 인터페이스(Graphical User Interface, 간단히 GUI)는 컴포넌트들로 구성된다.

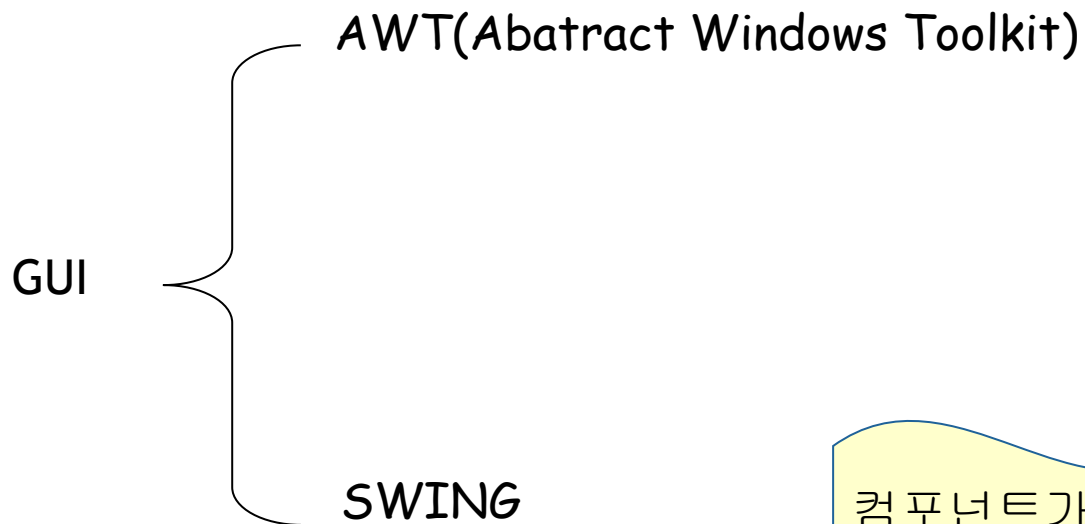
컴포넌트

자바에서는 사용자 인터페이스도 객체로 만들어 집니다. 사실 객체지향 프로그래밍 방법이 많이 사용되는 분야 중의 하나가 사용자 인터페이스 작성입니다.





자바에서 GUI의 종류



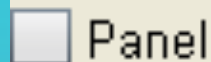
AWT는 운영 체제가 제공하는
자원을 이용하여서
컴포넌트를 생성

컴포넌트가 자바로 작성되어
있기 때문에 어떤 플랫폼에서도
일관된 화면을 보여줄 수 있다.



스윙

Swing Containers



Panel



Tabbed Pane



Split Pane



Scroll Pane



Tool Bar



Desktop Pane



Internal Frame



Layered Pane

Swing Controls



Label



Button



Toggle Button



Check Box



Radio Button



Button Group



Combo Box



List



Text Field



Text Area



Scroll Bar



Slider



Progress Bar



Formatted Field



Password Field



Spinner



Separator



Text Pane



Editor Pane



Tree



Table

Swing Menus



Menu Bar



Menu



Menu Item



Menu Item / CheckBox



Menu Item / RadioButton



Popup Menu



Separator



AWT와 스윙

- 여기서 컴포넌트는 클래스로 간주하면 된다.

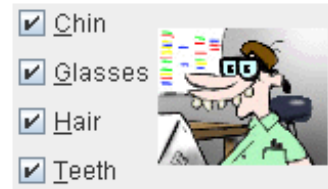
컴포넌트	AWT 버전	스윙 버전
버튼	Button	JButton
레이블	Label	JLabel
리스트	List	JList
...		
패스워드필드	없음	JPasswordField
슬라이더	없음	JSlider



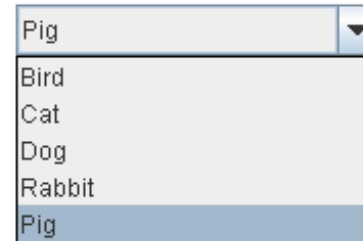
스윙



[JButton](#)



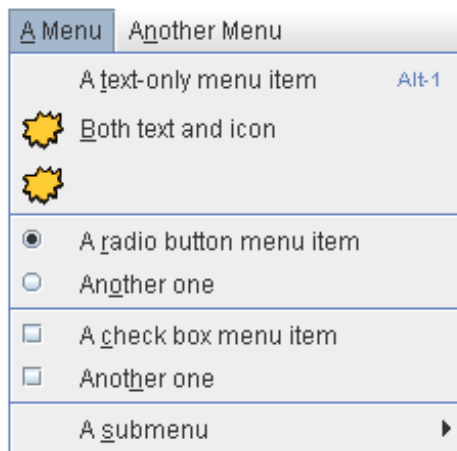
[JCheckBox](#)



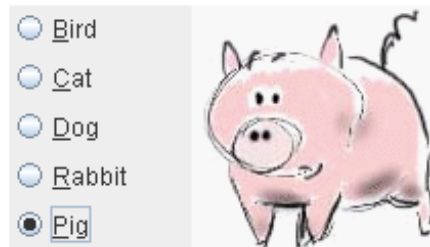
[JComboBox](#)



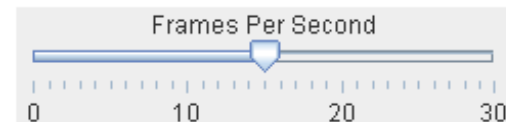
[JList](#)



[JMenu](#)



[JRadioButton](#)



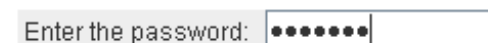
[JSlider](#)



[JSpinner](#)



[JTextField](#)

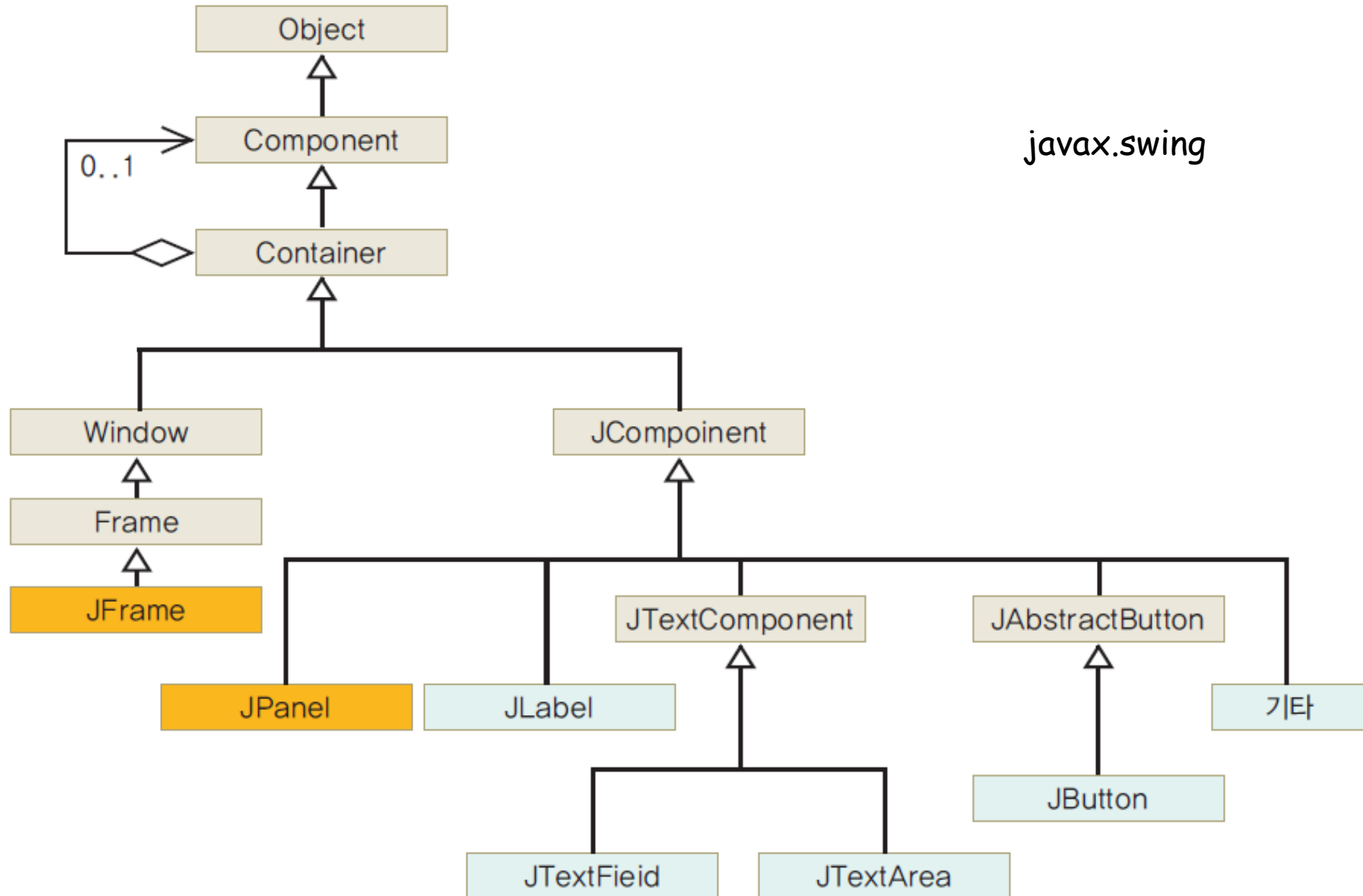


[JPasswordField](#)



스윙 클래스 계층구조

javax.swing





컨테이너와 컴포넌트

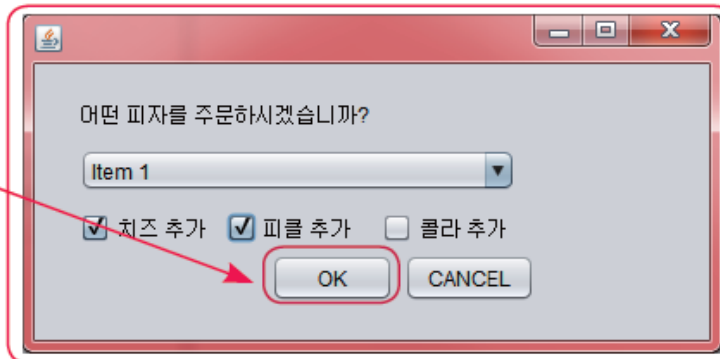
○ 기본 컴포넌트

- JButton, JLabel, JCheckbox, JChoice, JList, JMenu, JTextField, JScrollbar, JTextArea, JCanvas 등이 여기에 속한다.

○ 컨테이너 컴포넌트

- 다른 컴포넌트를 안에 포함할 수 있는 컴포넌트로서 JFrame, JDialog, JApplet, JPanel, JScrollPane 등이 여기에 속한다.

단순
컴포넌트



컨테이너
컴포넌트

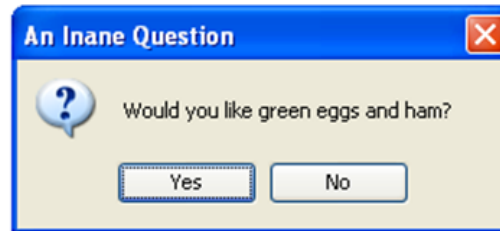
컨테이너는 컴포넌트를
넣을 수 있는 상자같은
것입니다.



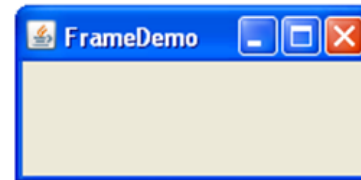


컨테이너의 종류

- 최상위 컨테이너: 절대 다른 컨테이너 안에 포함될 수 없는 컨테이너를 의미한다. 프레임(JFrame), 다이얼로그(JDialog), 애플릿(JApplet) 등이 여기에 해당된다.



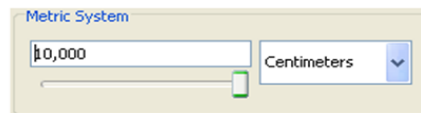
[JApplet](#)



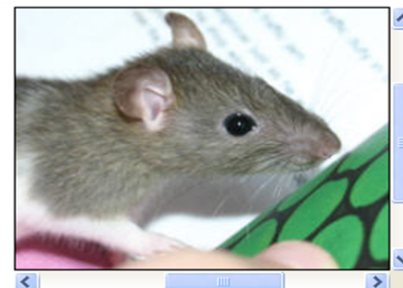
[JDialog](#)

[JFrame](#)

- 일반 컨테이너: 다른 컨테이너 안에 포함될 수 있는 컨테이너로 패널(JPanel), 스크롤 페인(JScrollPane) 등을 의미한다.



[JPanel](#)

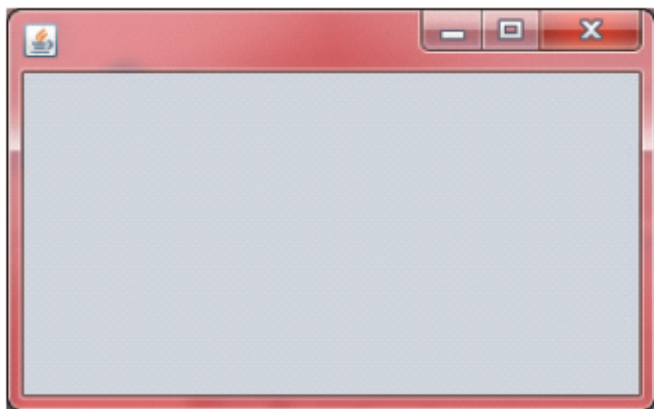


[JScrollPane](#)

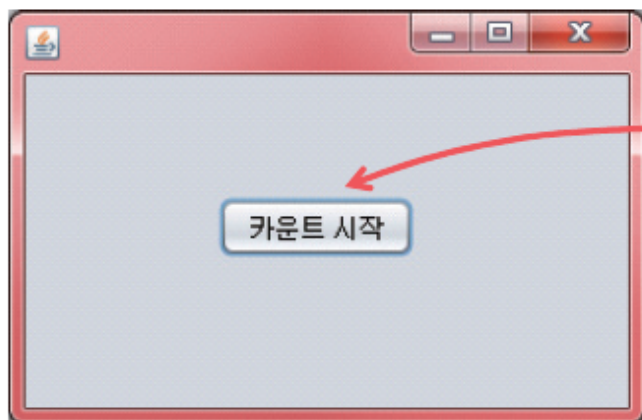


GUI 작성 절차

(1) 컨테이너를 생성한다.



(2) 컴포넌트를 추가한다.



Label	OK Button	Toggle Button
Check Box	Radio Button	Button Group
Combo Box	List	Text Field
Text Area	Scroll Bar	Slider
Progress Bar	Formatted Field	Password Field
Spinner	Separator	Text Pane
Editor Pane	Tree	Table



프레임 생성 #1

```
import javax.swing.*;

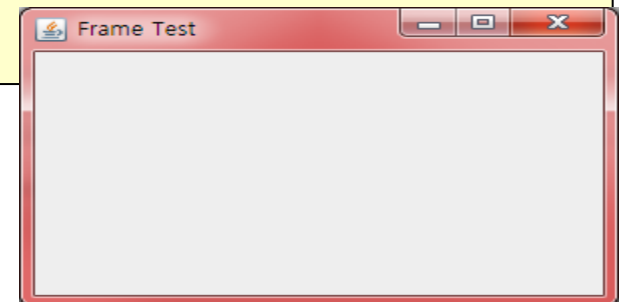
public class FrameTest {

    public static void main(String[] args) {

        JFrame f = new JFrame("Frame Test");

        f.setSize(300, 200);
        f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        f.setVisible(true);
    }
}
```

JFrame의 객체
생성



실행결과





프레임 생성 #2

```
import javax.swing.*;
```

```
public class MyFrame extends JFrame {
```

```
    public MyFrame() {  
        setSize(300, 200);  
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        setTitle("MyFrame");  
        setVisible(true);  
    }  
}
```

Jframe을 상속하
MyFrame을 정의

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

MyFrame의 객체
생성



프레임 생성 #3

JFrame을 상속하여서
MyFrame을 정의

```
import javax.swing.*;
public class MyFrame extends JFrame {
    public MyFrame() {
        setSize(300, 200);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setTitle("MyFrame");
        setVisible(true);
    }
    public static void main(String[] args) {
        MyFrame f = new MyFrame();
    }
}
```

main()이 MyFrame 안으로
이동



컴포넌트 생성과 추가

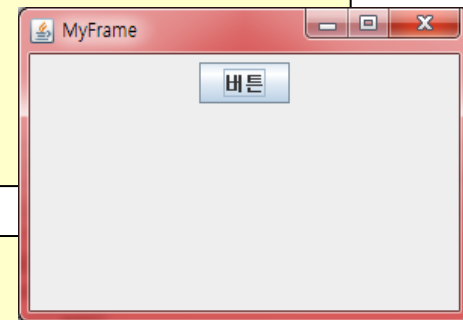
```
import javax.swing.*;  
import java.awt.FlowLayout;
```

```
public class MyFrame extends JFrame {  
    public MyFrame() {  
        setSize(300, 200);  
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        setTitle("MyFrame");  
  
        setLayout(new FlowLayout());  
        JButton button = new JButton("버튼");  
        this.add(button);  
        setVisible(true);  
    }  
}
```

배치 관리자 설정!

컴포넌트 생성 및 추가

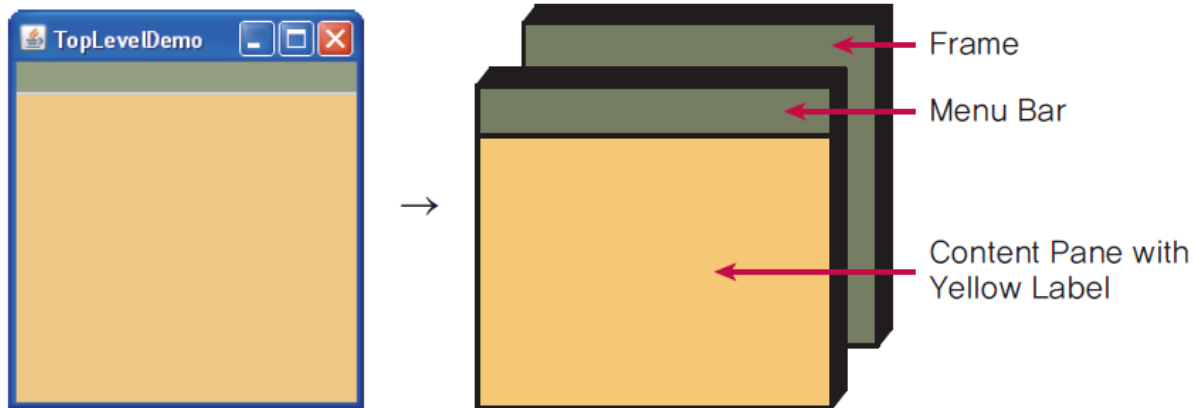
```
public class MyFrameTest {  
    public static void main(String[] args) {  
        MyFrame f = new MyFrame();  
    }  
}
```





JFrame 클래스

- 컨테이너는 컴포넌트들을 트리(tree) 형태로 저장한다. 최상위 컨테이너는 이 트리의 루트 노드가 된다.
- 최상위 컨테이너는 내부에 콘텐츠 페인(content pane)을 가지고 있다. 여기에 화면에 보이는 컴포넌트를 저장한다.
- 최상위 컨테이너에는 메뉴바를 추가할 수 있다.

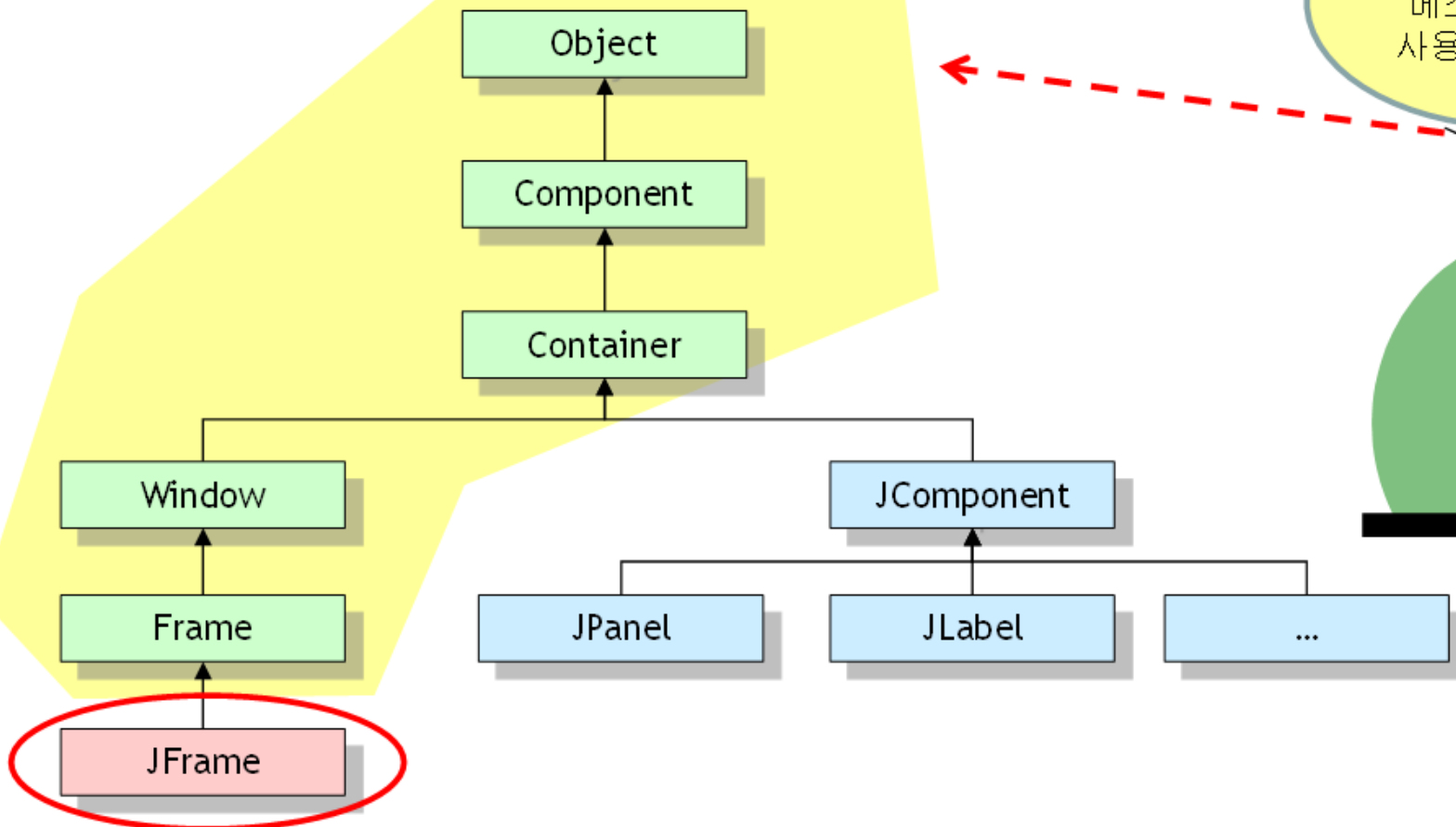




프레임의 속성 변경하기

- 프레임의 속성을 변경시키는 방법을 생각하여 보자.
- 조상 클래스들의 메소드도 사용할 수 있다!

수퍼 클래스의
메소드는 전부
사용할 수 있죠!





조상 클래스

○ *Component*

- ◎ 컴포넌트 클래스는 화면에 표시되어서 사용자와 상호 작용하는 시각적인 객체를 나타낸다.

○ *Container*

- ◎ 내부에 다른 컴포넌트를 추가할 수 있는 기능을 제공한다. 예를 들어서 이 클래스의 `add()`를 사용하면 컨테이너 안에 컴포넌트를 추가할 수 있다.

○ *Window*

- ◎ 경계선, 타이틀 바, 버튼을 가지고 있는 윈도우를 정의한다.

○ *Frame*

- ◎ 자바 GUI 애플리케이션의 기초가 된다.

○ *JFrame*

- ◎ `Frame` 클래스를 스윙의 출시에 맞추어 변경한 것이다.



중요한 메소드

- *setLocation(x, y)* , *setBounds(x, y, width, height)* , *setSize(width, height)*
 - ⊙ 프레임의 위치와 크기를 설정한다.
- *setIconImage(iconImage)*
 - ⊙ 윈도우 시스템에 타이틀 바, 태스크 스위처에 표시할 아이콘을 알려준다.
- *setTitle()*
 - ⊙ 타이틀 바의 제목을 변경한다.
- *setResizable(boolean)*
 - ⊙ 사용자가 크기를 조절할 수 있는지를 설정한다.



기초 컴포넌트

- 레이블(label)
- 버튼(button)
- 텍스트 필드(text field)

레이블

거리를 입력하시오(단위: km)

10

변환된 거리(단위: 마일)

6.2137

변환

텍스트 필드

버튼



레이블

- 레이블(Label)은 편집이 불가능한 텍스트를 표시.
 - ⦿ (예) JLabel label = **new** JLabel(“안녕하세요?”);





텍스트 필드

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

TextInputDemo

Street address:

City:

State:

Zip code:

No address set.

텍스트 필드



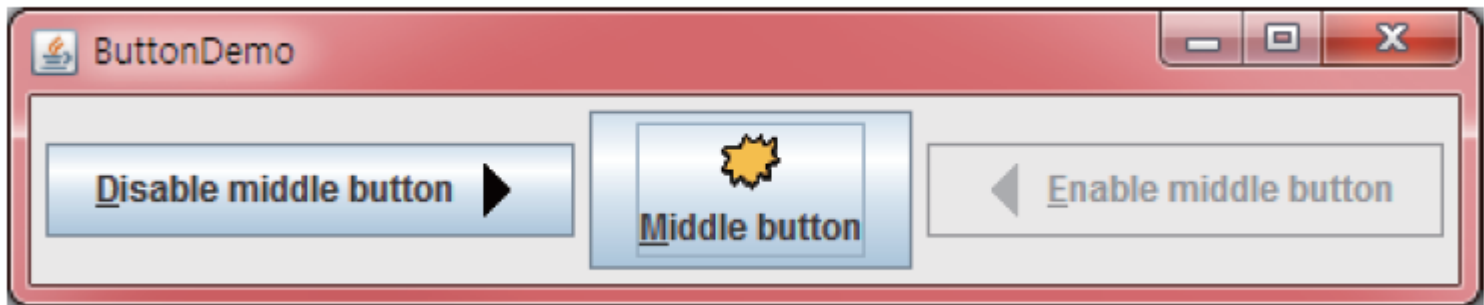
텍스트 필드의 종류

- **TextField** - 기본적인 텍스트 필드
- **FormattedTextField** - 사용자가 입력할 수 있는 문자를 제한한다.
- **PasswordField** - 사용자가 입력하는 내용이 보이지 않는다.
- **ComboBox** - 사용자가 직접 입력할 수도 있지만 항목 중에서 선택할 수 있다.
- **Spinner** - 텍스트 필드와 버튼이 조합된 것으로 사용자는 이전 버튼과 다음 버튼을 이용하여 선택할 수 있다.



버튼

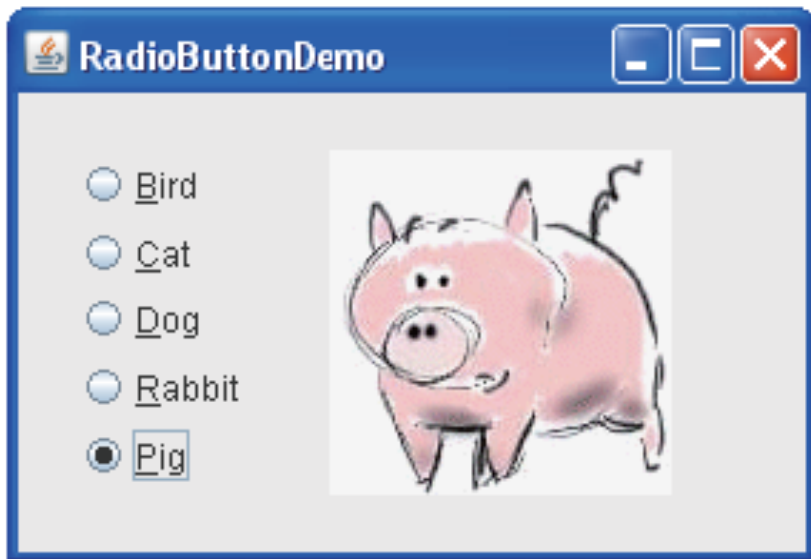
- 버튼은 사용자가 클릭했을 경우, 이벤트를 발생하여 원하는 동작을 하게 하는데 이용된다





버튼의 종류

- JButton - 가장 일반적인 버튼이다.
- JCheckBox - 체크박스 버튼
- JRadioButton - 라디오 버튼으로 그룹 중의 하나의 버튼만 체크할 수 있다.





LAB#1: 온도 변환기

- 이제까지 학습한 내용을 바탕으로 화씨 온도를 섭씨 온도로 변환해주는 애플리케이션을 작성하여 보자.

A screenshot of a Windows-style application window titled "온도변환기" (Temperature Converter). The window has a red title bar with standard minimize, maximize, and close buttons. Inside the window, there are two input fields. The first is labeled "화씨 온도" (Fahrenheit Temperature) and contains the value "100". The second is labeled "섭씨 온도" (Celsius Temperature) and is currently empty. Below these fields is a blue button with the text "변환" (Convert).

Label	Value
화씨 온도	100
섭씨 온도	

변환



SOLUTION

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

```
public class Mylab {  
    public static void main(String[] args) {  
        JFrame f = new JFrame();  
        JPanel panel = new JPanel();  
        f.add(panel);  
  
        JLabel label1 = new JLabel("화씨 온도");  
        JLabel label2 = new JLabel("섭씨 온도");  
        JTextField field1 = new JTextField(15);  
        JTextField field2 = new JTextField(15);  
        JButton button = new JButton("변환");
```



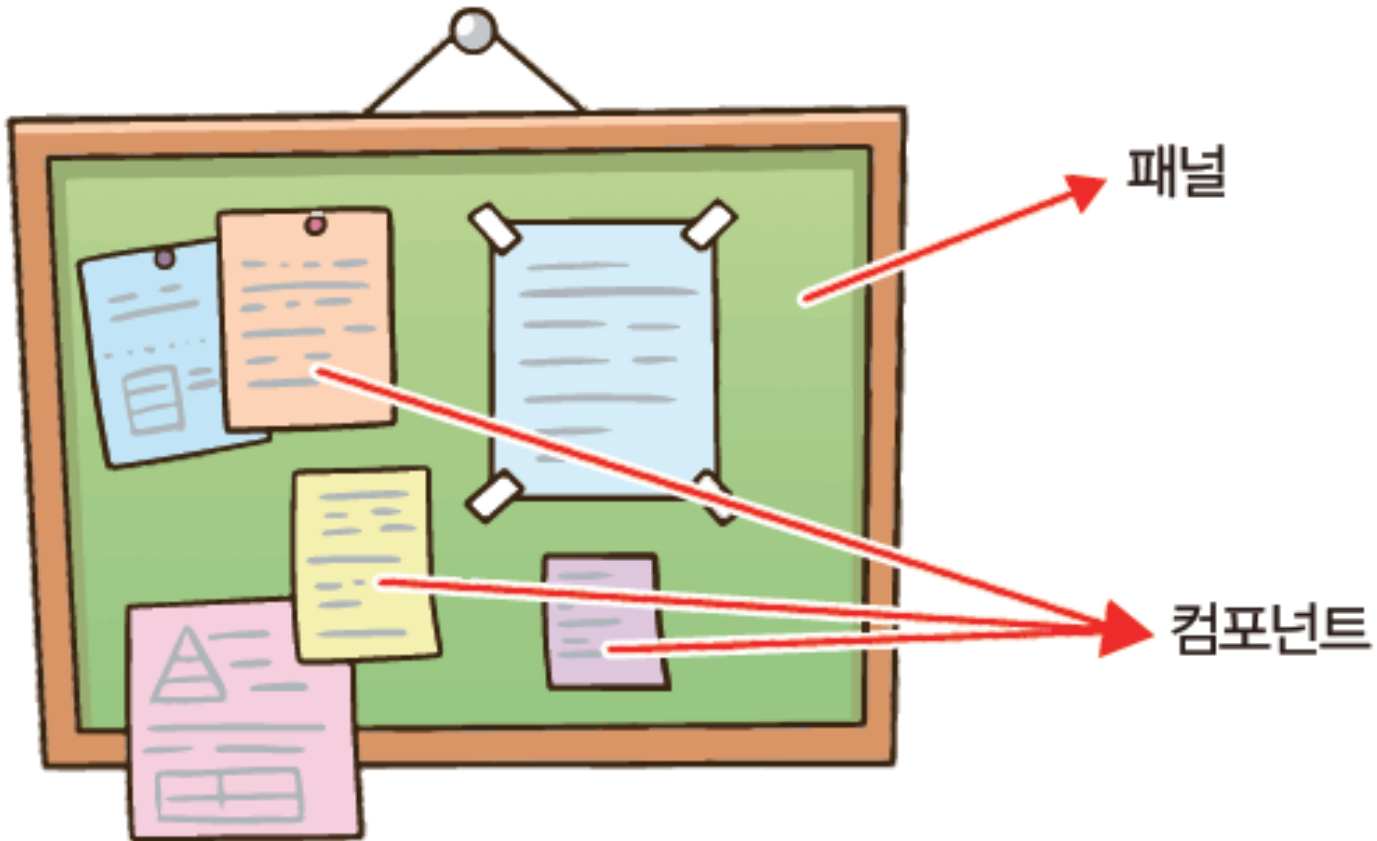
SOLUTION

```
panel.add(label1);  
panel.add(field1);  
panel.add(label2);  
panel.add(field2);  
panel.add(button);  
f.setSize(300, 150);  
f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
f.setTitle("온도변환기");  
f.setVisible(true);  
}  
}
```



패널

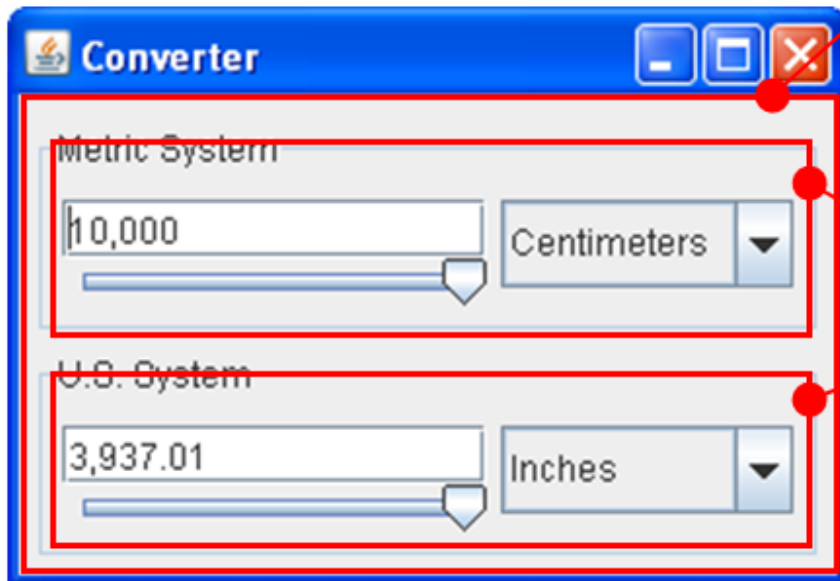
- 패널 (panel)은 컴포넌트들을 포함하고 있도록 설계된 컨테이너 중의 하나이다





패널의 용도

- 예를 들어서 다음과 같이 화면을 2개로 나누어서 표시하고자 할 때 패널을 사용할 수 있다.



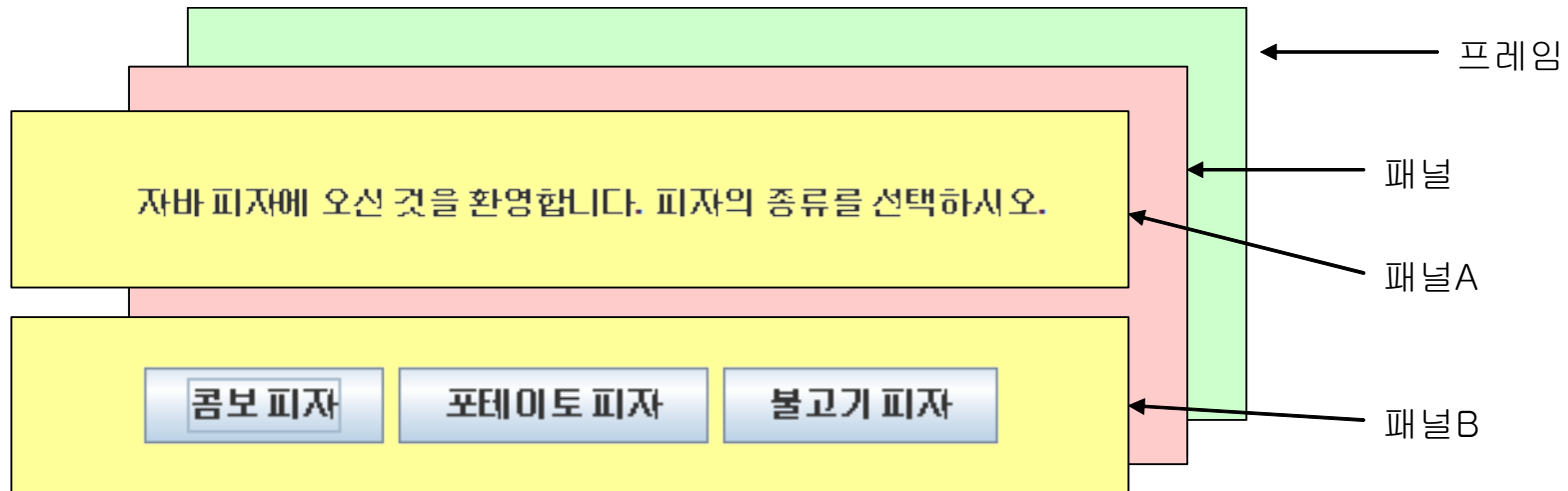
빨간색으로 표시된 패널은 전체 콘텐츠를 보여주는 배경 컨테이너로 동작한다. 배치 관리자는 수직 BoxLayout

제목을 가지고 있는 패널로 배치 관리자로 수평 BoxLayout을 사용한다.



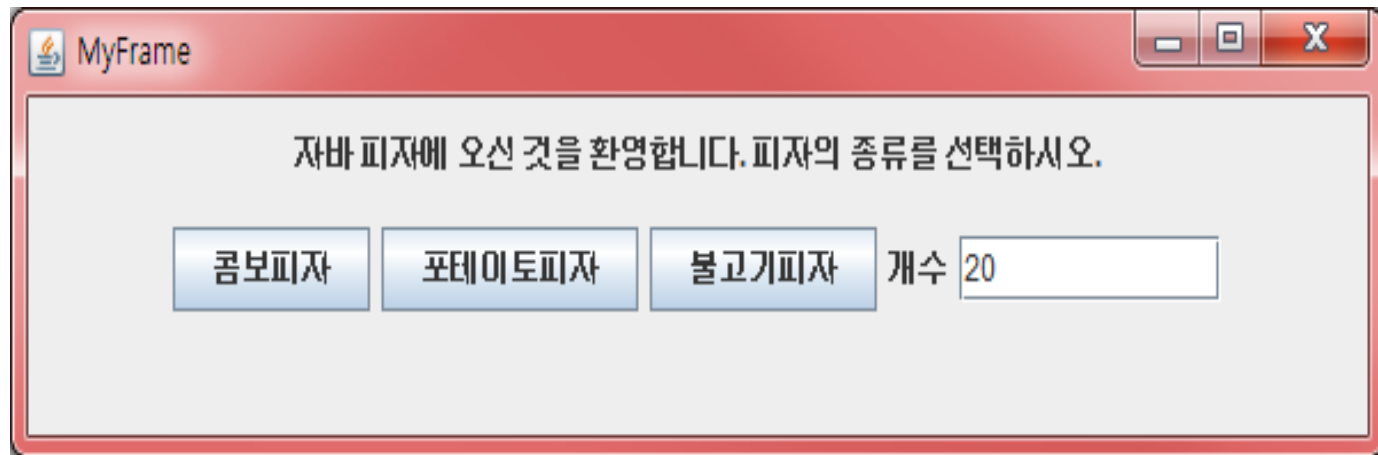
LAB#2: 피자 주문 화면 제작

- 패널 안에 다른 패널이 포함될 수 있다. 이것을 이용하여서 다음 그림처럼 프로그램의 화면을 디자인하라.





실행 결과





SOURCE

// 소스만 입력하고 Ctrl+Shift+O를 누른다.

```
public class MyFrame extends JFrame {  
  
    public MyFrame() {  
        setSize(600, 150);  
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        setTitle("MyFrame");  
  
        JPanel panel = new JPanel();  
        JPanel panelA = new JPanel();  
        JPanel panelB = new JPanel();  
  
        JLabel label1 = new JLabel("자바 피자에 오신 것을 환영합니  
다. 피자의 종류를 선택하십시오.");  
        panelA.add(label1);
```



SOURCE

```
 JButton button1 = new JButton("콤보피자");
 JButton button2 = new JButton("포테이토피자");
 JButton button3 = new JButton("불고기피자");
 panelB.add(button1);
 panelB.add(button2);
 panelB.add(button3);

 JLabel label2 = new JLabel("개수");
 JTextField field1 = new JTextField(10);
 panelB.add(label2);
 panelB.add(field1);

 panel.add(panelA);
 panel.add(panelB);
 add(panel);
 setVisible(true);

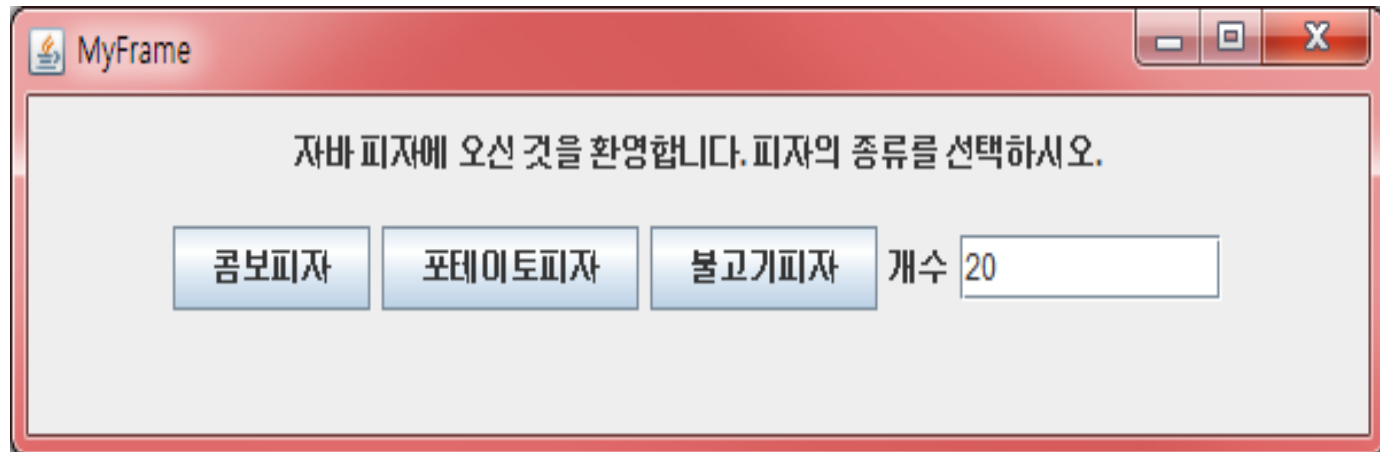
    }

}
```



SOURCE

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



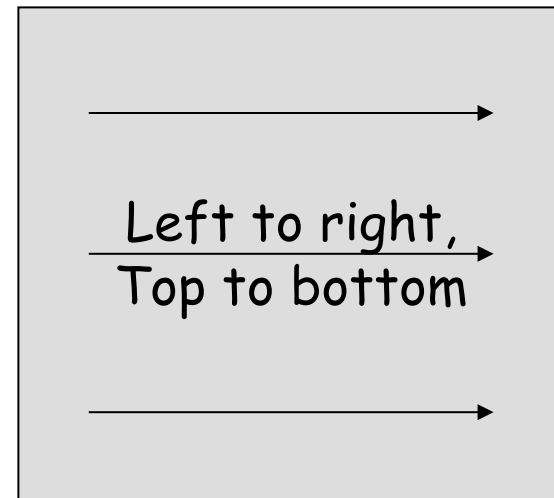
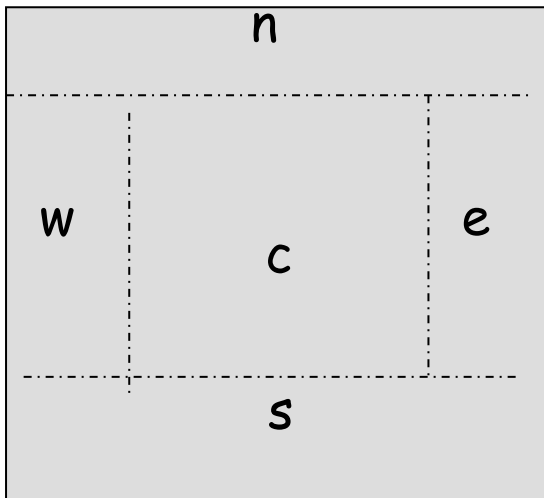


점검 문제

1. 패널에 버튼 3개를 추가하는 경우와 프레임에 버튼 3개를 추가하는 경우를 비교하여 보자. 어떻게 외관이 달라지는가?

프레임기본배치: **BORDERLAYOUT**

패널기본배치: **FLOWLAYOUT**





Q & A

