



# Swing GUI 기본

---

GUI

이벤트, 이벤트 리스너와 이벤트 소스

그림 그리기

내부 클래스

# JFC/Swing 개요

## □ Swing의 등장 배경

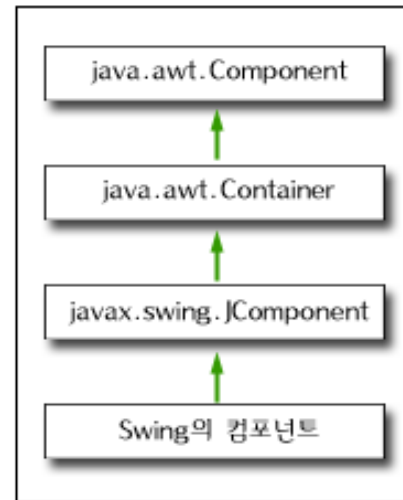
- AWT는 사용하기에는 쉽지만 어느정도 한계가 있음
  - 풍선말 기능 등 실제로 많이 사용하는 기능들이 없음
  - 또한, 운영체제마다 매우 다른 GUI 구현 방식 때문에 AWT의 컴포넌트를 사용함에 있어서 많은 버그가 발견
- Swing은 Top-Level 컨테이너만을 운영체제의 자원을 사용할 뿐 그 하부에 있는 모든 컴포넌트는 직접 자바 코드에 의해 그리는 방식을 사용

	AWT	Swing
사용 구조	Heavy-weight Componet - 컴포넌트를 운영체제의 GUI와 연결시키므로 운영체제에 따라 다른 모양과 배치가 나타난다.	Light-weight Componet - 운영체제가 제공하는 컴포넌트를 사용하지 않고 직접 컴포넌트를 만들어 구현한다.
지원 패키지	java.awt.*;	javax.swing.*;
장점	- 컴포넌트를 추상화함으로써 각 운영체제에서 구현하는것이 편하다.	- 운영체제의 자원을 빌리지 않고 직접 화면에 그리므로 이질적인 운영체제간에 발생한 버그를 자동적으로 해결 - 발행하더라도 자바가상머신에서 해결 가능
단점	- 운영체제에 따른 미묘한 버그 발생 - 불규칙한 컴포넌트의 모양과 레이아웃 설정 문제 발생	- 실행 속도가 느려진다. 실행시 메모리를 많이 차지함 --> JDK1.4 부터 보완됨,

# Swing 개요

## □ Swing의 클래스 구조

- Swing의 컴포넌트는 대부분이 javax.swing.JComponent 클래스로부터 상속 받음



<< 스윙의 클래스 구조 >>

## □ Swing의 특징 및 장점

- 더블 버퍼링 지원
- 각 컴포넌트의 속성값을 설정이 가능하다.
- 컴포넌트에 투명한 효과를 줄수 있다.
- 툴팁(Tool Tip) , 테두리(border) 등을 지원한다.



# Swing 개요 : Hello World

---

```
public static void main(String[] args) {  
    HelloSwing hs = new HelloSwing("수원 시작... 시작이 반이다.");  
    hs.addWindowListener(  
        new WindowAdapter() {  
            public void windowClosing(WindowEvent we) {  
                System.exit(0);  
            }  
        }  
    );  
    hs.pack();  
    hs.setSize(300, 200);  
    hs.setVisible(true);  
}  
}
```



# Swing 개요 : Hello World

---

```
public class HelloWorldSwing {  
    public static void main(String[] args) {  
        JFrame frame = new JFrame("HelloWorldSwing");  
        final JLabel label = new JLabel("Hello World");  
        frame.getContentPane().add(label);  
  
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        frame.pack();  
        frame.setVisible(true);  
    }  
}
```



# Swing 개요 : Hello World

---

```
class HelloSwing extends JFrame {  
    public HelloSwing(String title) {  
        super(title);  
        display();  
    }  
  
    public void display() {  
        Container cpane;  
  
        JMenuBar mb = new JMenuBar();  
        JMenu menu;  
  
        menu = new JMenu("File");  
        menu.add(new JMenuItem("Open"));  
        menu.add(new JMenuItem("Save"));  
        menu.addSeparator();  
        menu.add(new JMenuItem("Exit"));  
  
        mb.add(menu);  
        setJMenuBar(mb);  
  
        cpane = getContentPane();  
        cpane.setLayout(new BorderLayout());  
        cpane.add(new JButton("안녕, 스윙"),BorderLayout.CENTER);  
    }  
}
```

# Swing 데모 보기

## Java Tutorial Swing 데모

- <http://docs.oracle.com/javase/tutorial/uiswing/examples/components/index.html>

Using Swing Components: Examples (The Java™ Tutorial > Creating a GUI with Swing > Using Swing - Windows Internet Explorer)

Complete structure: Note that any examples using images expect their image files to be in a directory named `images` that is in the same directory as the example's `src` files.

Here is a typical setup:

You can find out which files each example needs by consulting the following table or by looking at the comments at the beginning of each source file.

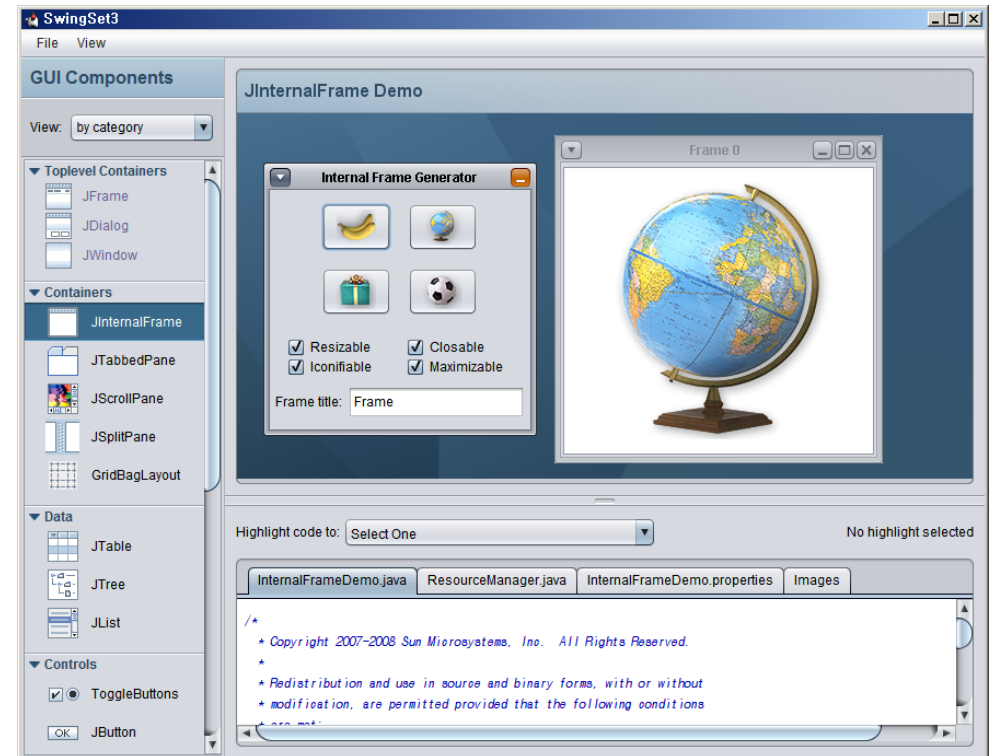
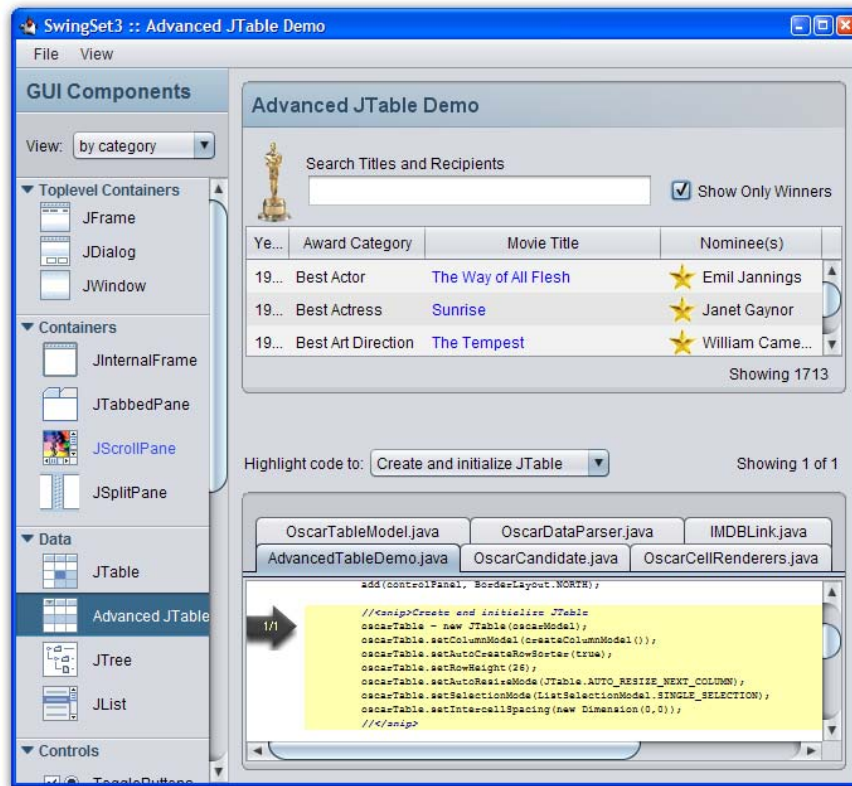
**Table of Examples**

Example	Zip File (contains all files necessary for the example plus NetBeans IDE project metadata)	Source Files (first file has the main method, except for examples that run only as applets)	Image and Other Files	Where Described
BorderDemo <a href="#">[Launch]</a>	Border Demo Project	BorderDemo.java	<code>appv.gif</code>	<a href="#">How to Use Borders</a>
ButtonDemo <a href="#">[Launch]</a>	Button Demo Project	ButtonDemo.java	<code>right.gif</code> , <code>middle.gif</code> , <code>left.gif</code>	<a href="#">How to Use Buttons...</a>
ButtonHtmlDemo <a href="#">[Launch]</a>	Button Html Demo Project	ButtonHtmlDemo.java	<code>right.gif</code> , <code>middle.gif</code> , <code>left.gif</code>	<a href="#">How to Use Buttons...</a>
CheckBoxDemo <a href="#">[Launch]</a>	Check Box Demo Project	CheckBoxDemo.java	All of the images in the <code>images/greek</code> directory.	<a href="#">How to Use Buttons...</a>
ColorChooserDemo <a href="#">[Launch]</a>	Color Chooser Demo Project	ColorChooserDemo.java		<a href="#">How to Use Color Choosers</a>
ColorChooserDemo2 <a href="#">[Launch]</a>	Color Chooser 2 Demo Project	ColorChooserDemo2.java <code>GravonPanel.java</code>	<code>red.gif</code> , <code>yellow.gif</code> , <code>green.gif</code> , <code>blue.gif</code>	<a href="#">How to Use Color Choosers</a>
ComboBoxDemo <a href="#">[Launch]</a>	Combo Box Demo Project	ComboBoxDemo.java	<code>Bird.gif</code> , <code>Cat.gif</code> , <code>Dog.gif</code> , <code>Rabbit.gif</code> , <code>Pig.gif</code>	<a href="#">How to Use Combo Boxes</a>
ComboBoxDemo2 <a href="#">[Launch]</a>	Combo Box 2 Demo Project	ComboBoxDemo2.java		<a href="#">How to Use Combo Boxes</a>
Converter <a href="#">[Launch]</a>	Converter Project	<code>Converter.java</code> <code>ConversionPanel.java</code> <code>ConverterRangeModel.java</code> <code>FollowerRangeModel.java</code> <code>Unit.java</code>		<a href="#">How to Use Panels: Using Models</a>
CustomComboBoxDemo	Custom Combo Box	CustomComboBoxDemo.java	<code>Bird.gif</code> , <code>Cat.gif</code> , <code>Dog.gif</code>	<a href="#">How to Use Combo Boxes</a>

# Swing 개요

## □ SwingSet3

- <http://download.java.net/javadesktop/swingset3/SwingSet3.inlp>

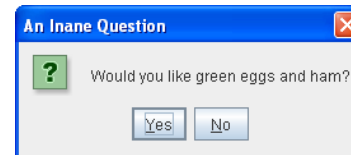


# Swing Visual Overview(Java L&F)

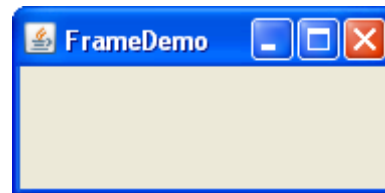
## □ Top-Level 컨테이너



JApplet



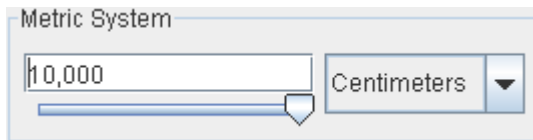
JDialog



JFrame

# Swing Visual Overview(Java L&F)

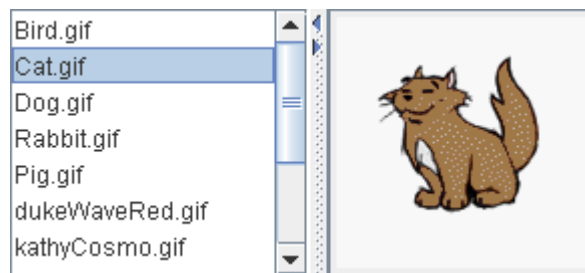
## □ General-Purpose 컨테이너



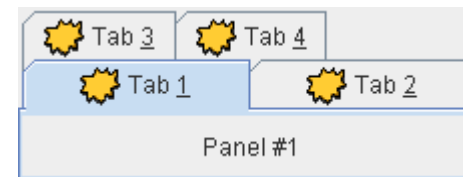
JPanel



JScrollPane



JSplitPane



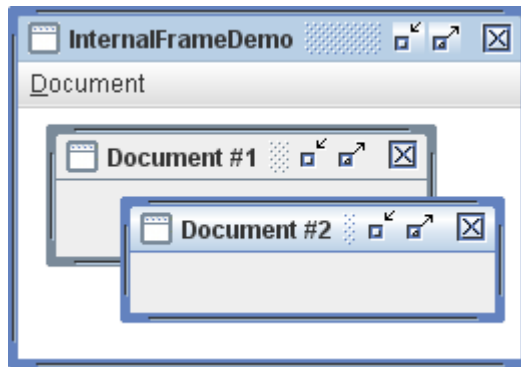
JTabbedPane



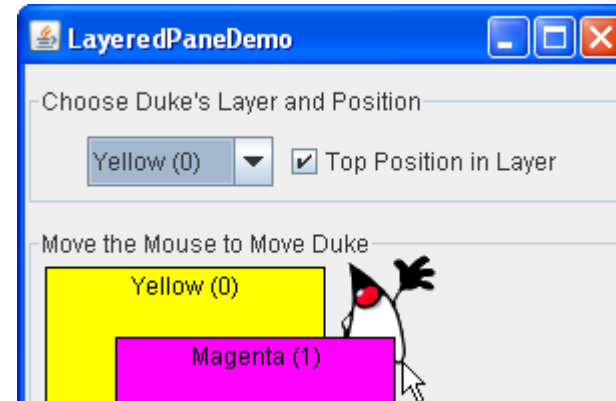
JToolBar

# Swing Visual Overview(Java L&F)

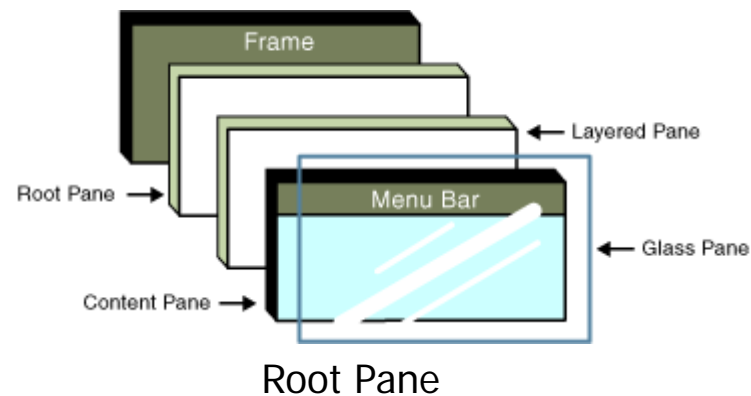
## □ Special-Purpose 컨테이너



JInternalFrame



JLayeredPane

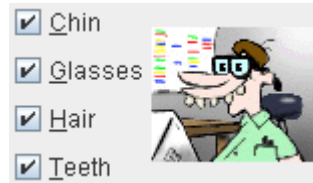


# Swing Visual Overview(Java L&F)

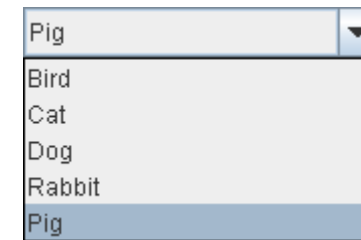
## □ 기본 Controls



JButton



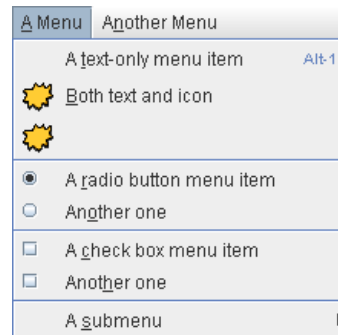
JCheckBox



JComboBox



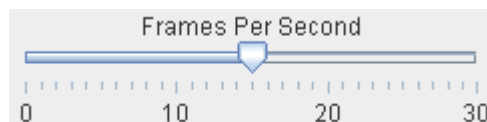
JList



JMenu



JRadioButton



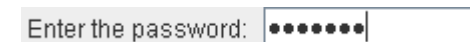
JSlider



JSpinner



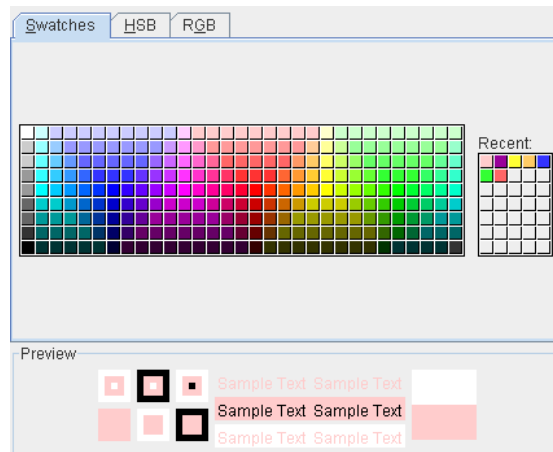
JTextField



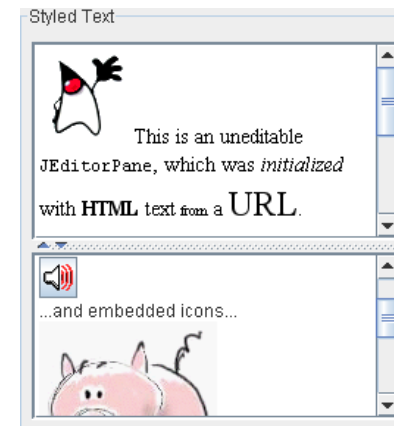
JPasswordField

# Swing Visual Overview(Java L&F)

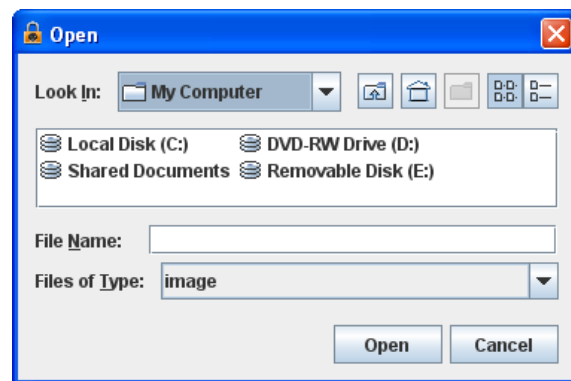
## □ Highly Formatted Information



JColorChooser



JEditorPane/JTextPane



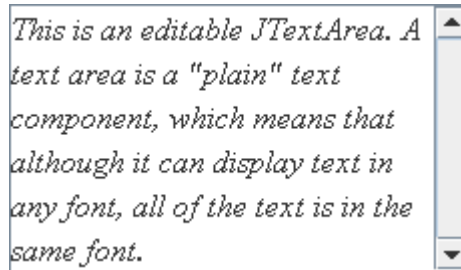
JFileChooser

Host	User	Password	Last Modified
Biocca Games	Freddy	!#asf6Awwzb	Mar 16, 2006
zabble	ichabod	Tazb!34\$fZ	Mar 6, 2006
Sun Developer	fraz@hotmail.co...	AasVW541!fbZ	Feb 22, 2006
Heirloom Seeds	shams@gmail....	bkz[ADF78!	Jul 29, 2005
Pacific Zoo Shop	seal@hotmail.c...	vbAf1 24%z	Feb 22, 2006

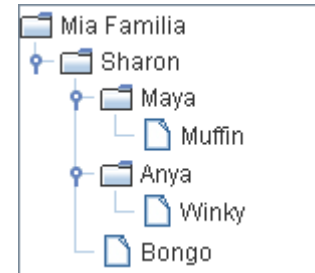
JTable

# Swing Visual Overview(Java L&F)

## □ Highly Formatted Information(계속)



JTextArea



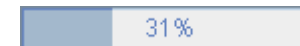
JTree

# Swing Visual Overview(Java L&F)

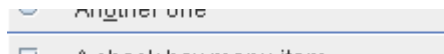
## □ 비편집 정보 디스플레이



JLabel



JProgressBar



JSeparator



JToolTip

# JFrame

## □ 단순한 JFrame 열기

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

```
public class CenteredFrameTest  
{
```

```
    public static void main(String[] args)  
    {
```

```
        CenteredFrame cframe = new CenteredFrame();  
        cframe.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE);  
        cframe.setVisible(true);  
    }
```

```
}
```

```
class CenteredFrame extends JFrame  
{
```

```
    public static final int DEFAULT_WIDTH = 300;  
    public static final int DEFAULT_HEIGHT = 200;
```

```
    public CenteredFrame()  
    {
```

```
        setSize(DEFAULT_WIDTH, DEFAULT_HEIGHT);  
    }
```

```
}
```

Frame을 Visible  
하게 => cframe.show()

종료 버튼 누르면  
자동 프로그램 종료

# JFrame

## □ SwingUtilities.invokeLater 이용한 호출

```
public class CenteredFrameTest
{
    public static void main(String[] args)
    {
        javax.swing.SwingUtilities.invokeLater(
            new Runnable()
            {
                public void run() {
                    CenteredFrame cframe = new CenteredFrame();
                    cframe.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE);
                    cframe.setVisible(true);
                }
            }
        );
    }
}
```

← main thread와 분리된 thread로 호출

invokeLater : asynchronously invocation

# JFrame

## □ Toolkit.getDefaultToolkit()을 이용한 JFrame

### ■ ImageIcon 사용하기

```
class CenteredFrame extends JFrame
{
    public CenteredFrame()
    {
```

```
        Toolkit kit = Toolkit.getDefaultToolkit();
        Dimension screenSize = kit.getScreenSize();
        int screenHeight = screenSize.height;
        int screenWidth = screenSize.width;
```

화면의 중앙에  
Frame 위치시키기

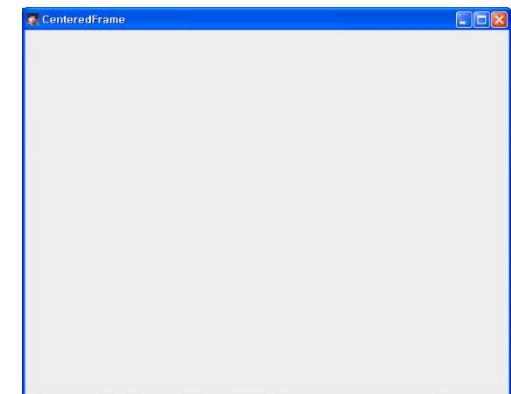
```
        setSize(screenWidth/2, screenHeight/2);
        setLocation( screenWidth/4, screenHeight/4);
```

```
        Image img = kit.getImage("남광우.jpg");
        setIconImage(img);
        setTitle("CenteredFrame");
```

Frame에 Icon 달기

```
    }
}
```

```
public class CenteredFrameTest
{
    public static void main(String[] args)
    {
        CenteredFrame cframe = new CenteredFrame();
        cframe.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE);
        cframe.setVisible(true);
    }
}
```



# JFrame

## □ JFrame에 버튼 넣고, 크기 조절하기

class CenteredFrame2 extends JFrame implements ActionListener

```
{
    boolean sizeToggle = true;
    int screenHeight = 0;
    int screenWidth = 0;

    private JButton b= new JButton("프레임 크기 조절하기");
```

```
public CenteredFrame2()
{
```

```
    this.setLayout(new FlowLayout());
    this.add(b);
    b.addActionListener(this); → 크기조절 Event
```

```
    Toolkit kit = Toolkit.getDefaultToolkit();
    Dimension screenSize = kit.getScreenSize();
    screenHeight = screenSize.height;
    screenWidth = screenSize.width;
```

```
    setSize(screenWidth/2, screenHeight/2);
    setLocation( screenWidth/4, screenHeight/4);
    this.setTitle( "Frame 갖고 놀기 ");
```

```
    Image img = kit.getImage("남광우.jpg");
    setIconImage(img);
}
```

```
public class CenteredFrameTest2
{
    public static void main(String[] args)
    {
        CenteredFrame2 cframe = new CenteredFrame2();
        cframe.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE);
        cframe.setVisible(true);
    }
}
```

→ Frame에 버튼 넣기

```
public void actionPerformed(ActionEvent e)
{
```

```
    if ( sizeToggle == true ) → 버튼 누름을  
    { 파악하여 크기조절
        this.setTitle( "작은 프레임");
        setSize(screenWidth/3, screenHeight/3);
        sizeToggle = false;
```

```
    }
    else
    {
```

```
        this.setTitle( "작은 프레임");
        setSize(screenWidth/2, screenHeight/2);
        sizeToggle = true;
```

```
    }
```

```
}
```

# JFrame

## □ 패널에 사진 보이기

### ■ JFrame + JPanel

```
class CenteredFrame3 extends JFrame implements ActionListener
{
```

```
    boolean sizeToggle = true;
    int screenHeight = 0;
    int screenWidth = 0;
```

```
    // 이미지 패널 표시
```

```
    ipanel = new ImagePanel();
```

```
    add(basePanel, BorderLayout.NORTH);
    add(ipanel, BorderLayout.CENTER);
```

```
    public CenteredFrame3()
    {
```

```
        JButton b;
        ImagePanel ipanel;
```

```
        // 버튼 표시
```

```
        JPanel basePanel = new JPanel();
```

```
        b = new JButton("제목바꾸기");
        b.addActionListener(this);
        basePanel.add(b);
        basePanel.setSize(300, 200);
```

```
        // frame 위치 셋팅
```

```
        Toolkit kit = Toolkit.getDefaultToolkit();
        Dimension screenSize = kit.getScreenSize();
        screenHeight = screenSize.height;
        screenWidth = screenSize.width;
```

```
        setSize(ipanel.getWidth(), basePanel.getHeight()+ipanel.getHeigh
        setLocation( screenWidth/4, screenHeight/4);
        this.setTitle( "Frame 갖고 놀기 ");
```

```
        Image img = kit.getImage("남광우.jpg");
        setIconImage(img);
```

```
    }
```

# JFrame

## □ 패널에 사진 보이기(계속)

```
class ImagePanel extends JPanel
```

```
{
```

```
    Image panellImage;
```

```
    public ImagePanel()
```

```
    {
```

```
        try
```

```
        {
```

```
            panellImage = ImageIO.read( new File("남광우.jpg"));
```

```
        }
```

```
        catch( IOException e)
```

```
        {
```

```
            e.printStackTrace();
```

```
            System.out.println("Exception");
```

```
        }
```

```
        setSize( panellImage.getWidth(this), panellImage.getHeight(this));
```

```
    }
```

```
    public void paintComponent( Graphics g)
```

```
    {
```

```
        super.paintComponent( g );
```

```
        if ( panellImage == null ) return;
```

```
        g.drawImage( panellImage, 0, 0, null );
```

```
        g.drawString("패널에 사진 보이기", 20, 20);
```

```
    }
```

```
}
```

이미지 파일  
읽기

이미지 파일  
그리기



# 컨테이너와 컴포넌트

## □ 컨테이너

- 다른 컴포넌트를 포함할 수 있는 GUI 컴포넌트
  - java.awt.Container를 상속받음
- 다른 컨테이너에 포함될 수 있음
  - AWT 컨테이너
    - Panel, Frame, Applet, Dialog, Window
  - Swing 컨테이너
    - JPanel JFrame, JApplet, JDialog, JWindow

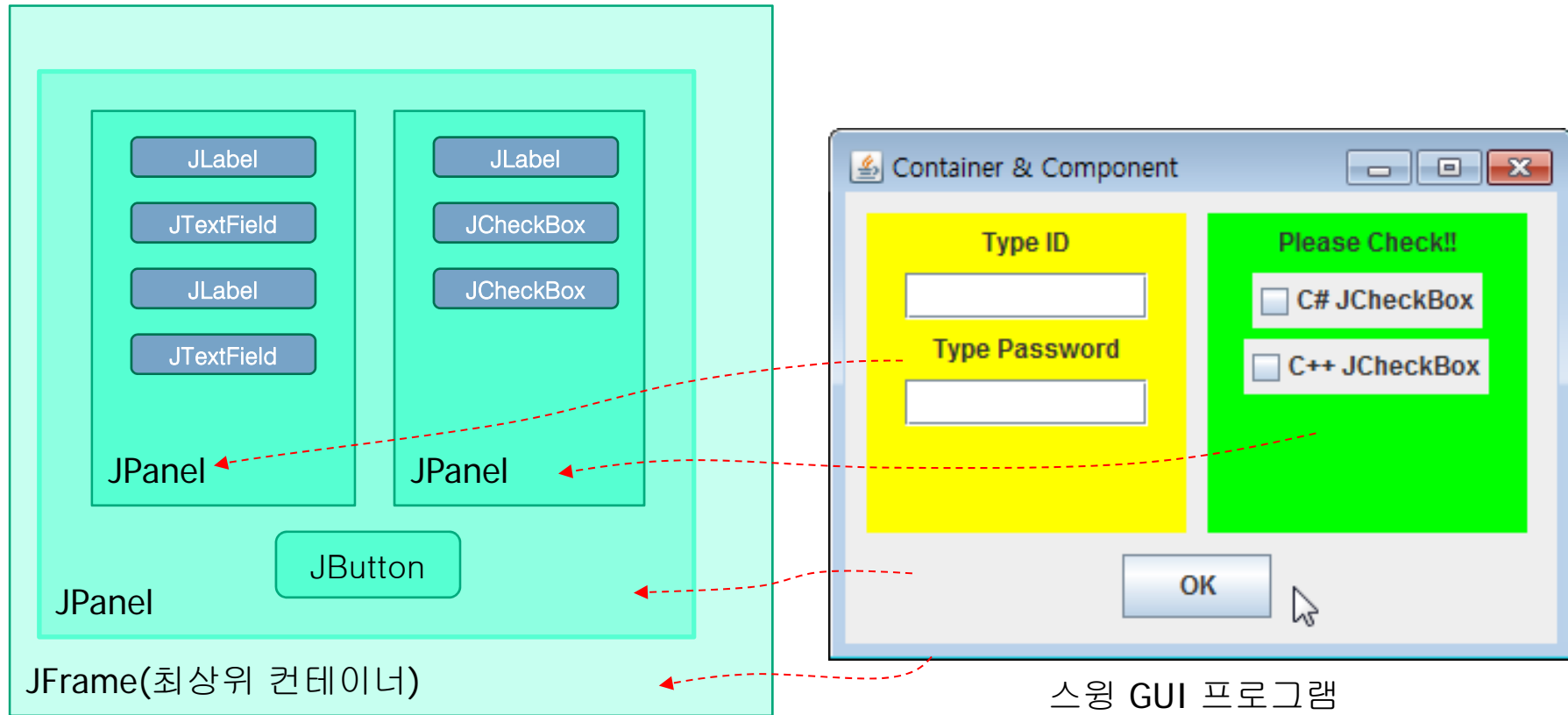
## □ 최상위 컨테이너

- 다른 컨테이너에 속하지 않고 독립적으로 존재 가능한 컨테이너
- 스스로 화면에 자신을 출력하는 컨테이너
  - JFrame, JDialog, JApplet
- 모든 컴포넌트는 컨테이너에 포함되어야 비로소 화면에 출력 가능

## □ 컴포넌트

- 컨테이너에 포함되어야 비로소 화면에 출력될 수 있는 GUI 객체
- 모든 GUI 컴포넌트의 최상위 클래스
  - java.awt.Component
- 스윙 컴포넌트의 최상위 클래스
  - javax.swing.JComponent

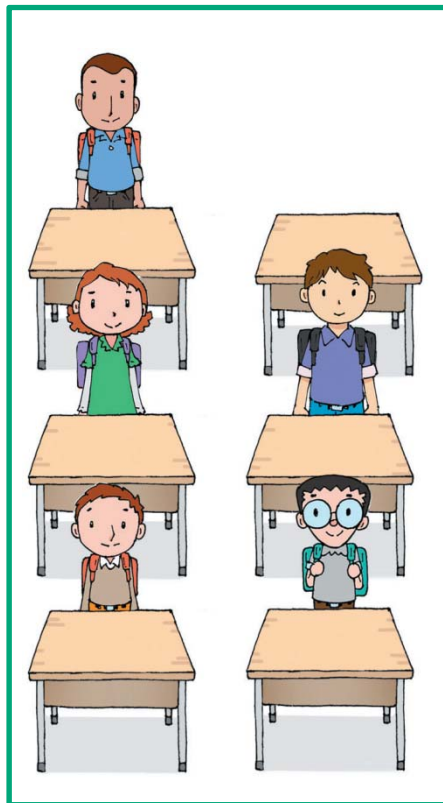
# 컨테이너와 컴포넌트의 포함관계



스윙의 컨테이너와 컴포넌트의 포함 관계

# 컨테이너와 배치 개념

컨테이너(Container)



이쪽으로  
가세요.

컴포넌트  
(Component)

1. 컨테이너마다 하나의 배치관리자가 존재하며, 삽입되는 모든 컴포넌트의 위치와 크기를 결정하고 적절히 배치한다.
2. 컨테이너의 크기가 변하면 내부 컴포넌트들의 위치와 크기를 모두 재조절하고 재배치한다.

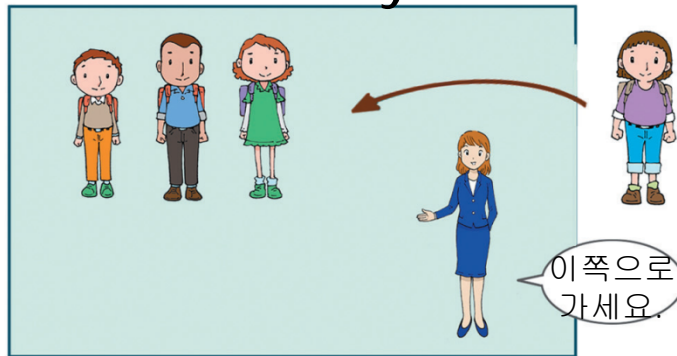


배치관리자  
(Layout Manager)

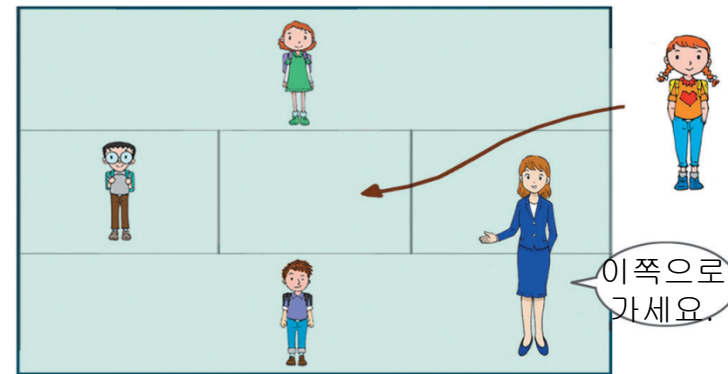
# 배치 관리자 대표 유형 4 가지

- java.awt 패키지에 구현되어 있음

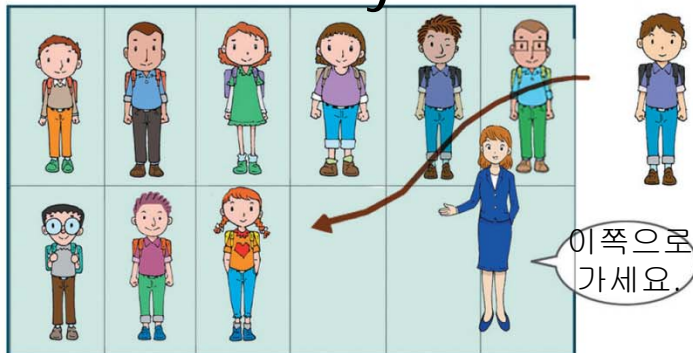
## FlowLayout



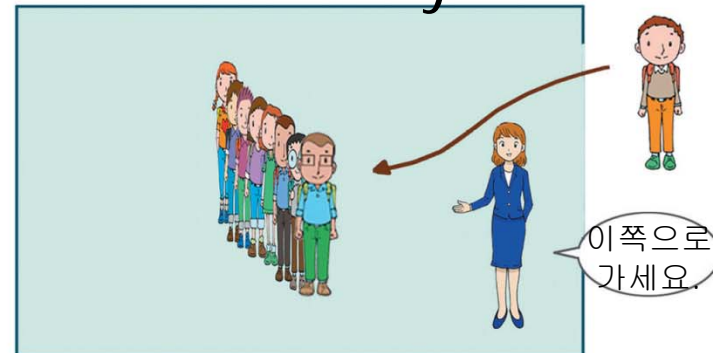
## BorderLayout



## GridLayout



## CardLayout



# 컨테이너와 배치관리자

- 컨테이너의 디폴트 배치관리자
  - 컨테이너는 생성시 디폴트 배치관리자 설정

AWT와 스윙의 컨테이너	디폴트 배치관리자
Window	BorderLayout
Frame, JFrame	BorderLayout
Dialog, JDialog	BorderLayout
Panel, JPanel	FlowLayout
Applet, JApplet	FlowLayout

- 컨테이너에 새로운 배치관리자 설정
  - Container.setLayout(LayoutManager lm)
  - lm을 Container의 새로운 배치관리자로 설정

// JPanel 패널에 BorderLayout 배치관리자를 설정하는 예

```
JPanel p = new JPanel();  
p.setLayout(new BorderLayout());
```

```
JFrame frame = new JFrame();  
Container c = frame.getContentPane(); // 프레임의 콘텐츠팬  
c.setLayout(new FlowLayout()); // 콘텐츠팬에 FlowLayout 설정
```

혹은

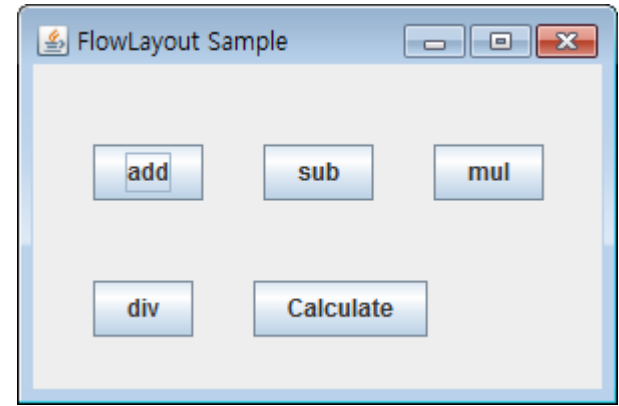
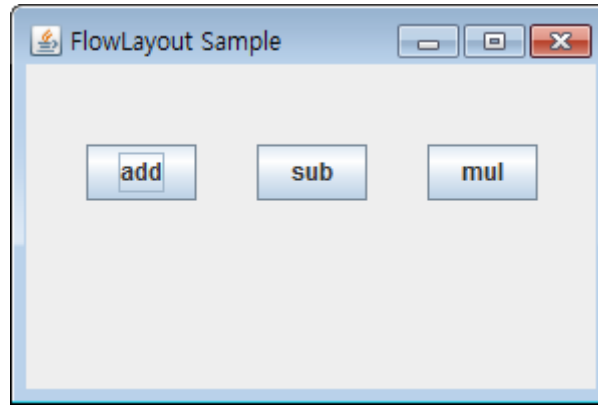
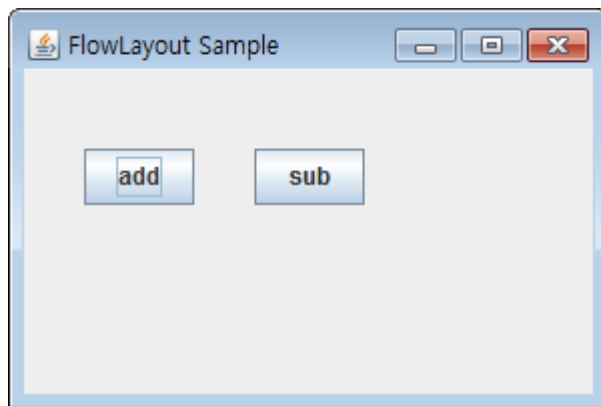
```
frame.setLayout(new FlowLayout()); // JDK 1.5 이후 버전에서
```

# FlowLayout

## □ 배치방법

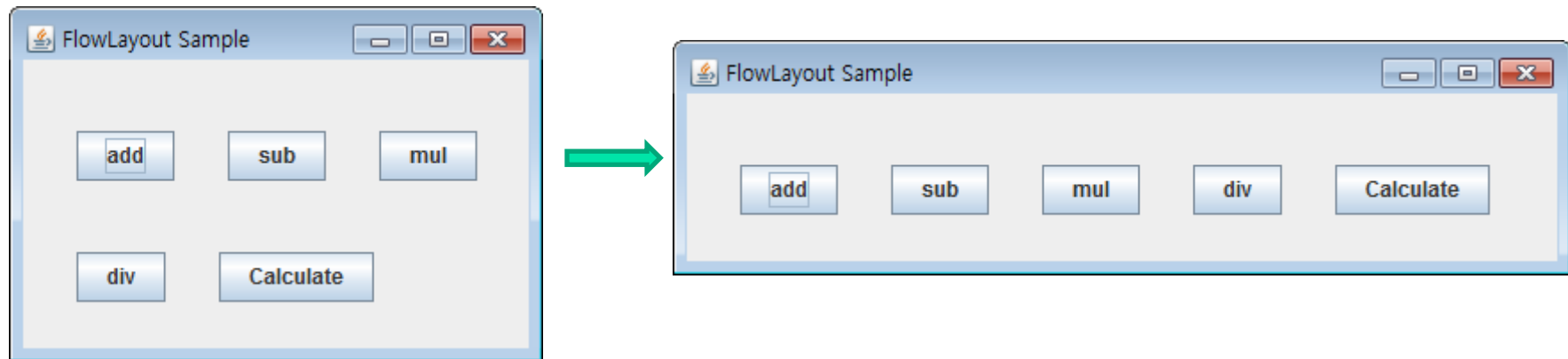
- 컨테이너 공간 내에 왼쪽에서 오른쪽으로 배치
  - 다시 위에서 아래로 순서대로 컴포넌트를 배치한다.

```
container.setLayout(new FlowLayout());  
container.add(new JButton("add"));  
container.add(new JButton("sub"));  
container.add(new JButton("mul"));  
container.add(new JButton("div"));  
container.add(new JButton("Calculate"));
```



# FlowLayout

- 컨테이너의 크기가 변하면 배치 관리자에 의해 재배치됨

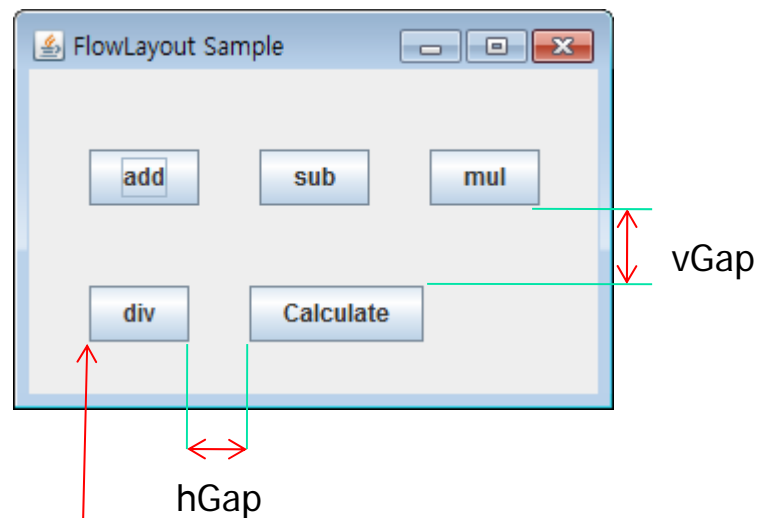


프레임의 크기를 바꾸면 배치도  
변한다.

# FlowLayout – 생성자와 속성

## □ 생성자

- FlowLayout()
- FlowLayout(int align)
- FlowLayout(int align, int hGap, int vGap)
  - align : 컴포넌트의 정렬(5 가지중 많이 사용되는 3 가지)
    - FlowLayout.LEFT, FlowLayout.RIGHT, FlowLayout.CENTER(디폴트)
  - hGap : 좌우 두 컴포넌트 사이의 수평 간격, 픽셀 단위(디폴트 : 5)
  - vGap : 상하 두 컴포넌트 사이의 수직 간격, 픽셀 단위(디폴트 : 5)



FlowLayout.LEFT로 정렬됨

# FlowLayout 사용 예

LEFT로 정렬되는 수평 간격이 30 픽셀, 수직 간격이 40 픽셀인 FlowLayout 사용 예

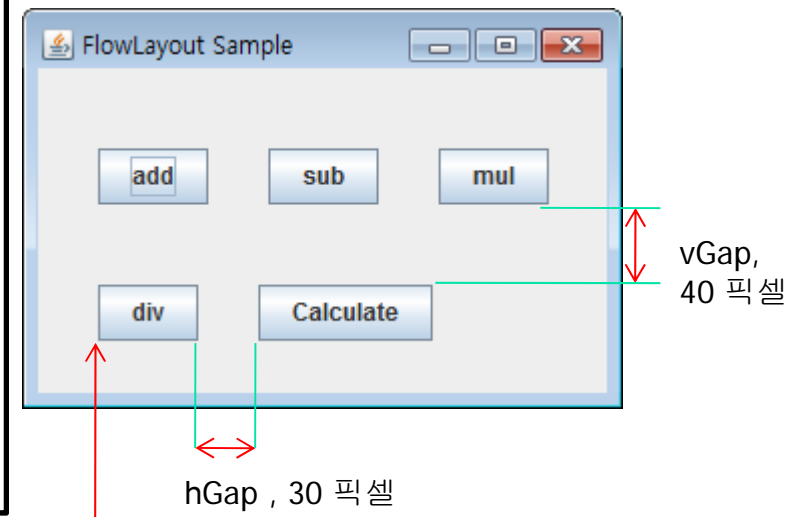
```
import javax.swing.*;
import java.awt.*;

public class FlowLayoutEx extends JFrame {
    FlowLayoutEx() {
        setTitle("FlowLayout Sample");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        setLayout(new FlowLayout(FlowLayout.LEFT, 30, 40));
        add(new JButton("add"));
        add(new JButton("sub"));
        add(new JButton("mul"));
        add(new JButton("div"));
        add(new JButton("Calculate"));

        setSize(300, 250);
        setVisible(true);
    }

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



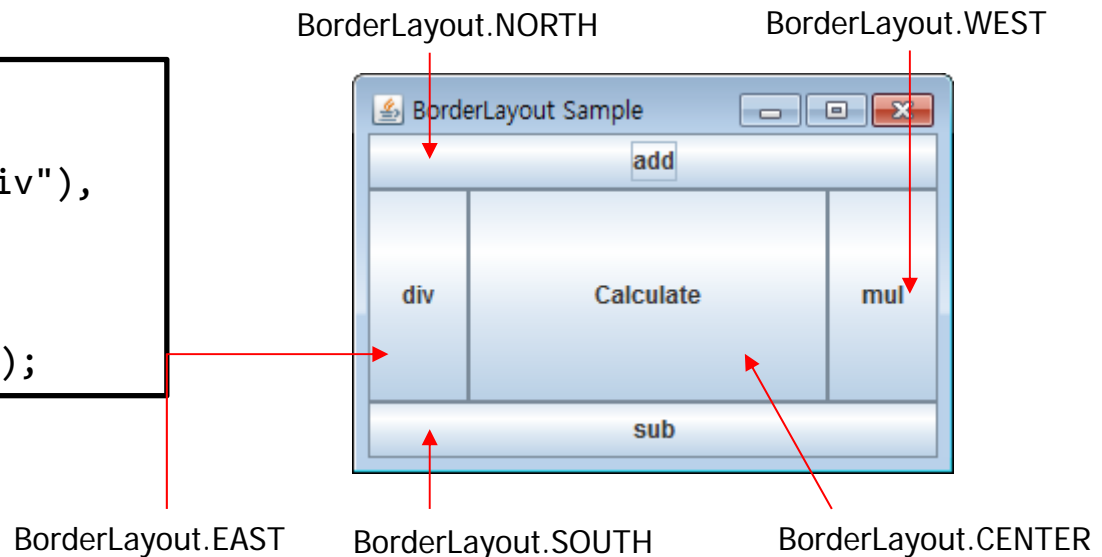
FlowLayout.LEFT로 정렬됨

# BorderLayout

## □ 배치 방법

- 컨테이너 공간을 5 구역으로 분할, 배치
  - East, West, South, North, Center
- 배치 방법
  - `add(Component comp, int index)`
    - comp를 index의 공간에 배치
- 컨테이너의 크기가 변하면 재배포

```
container.setLayout(new  
BorderLayout());  
container.add(new JButton("div"),  
                BorderLayout.WEST);  
container.add(new  
JButton("Calculate"),  
                BorderLayout.CENTER);
```



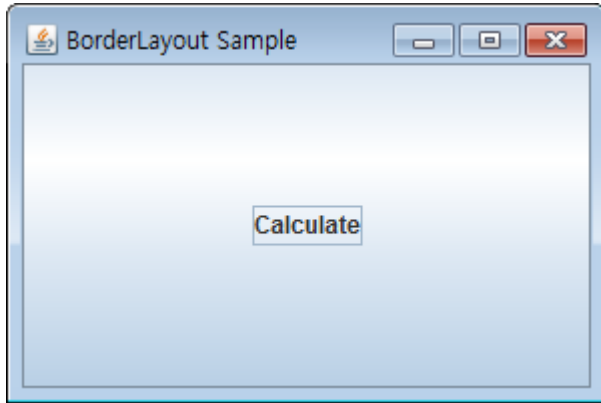


# BorderLayout 생성자와 속성

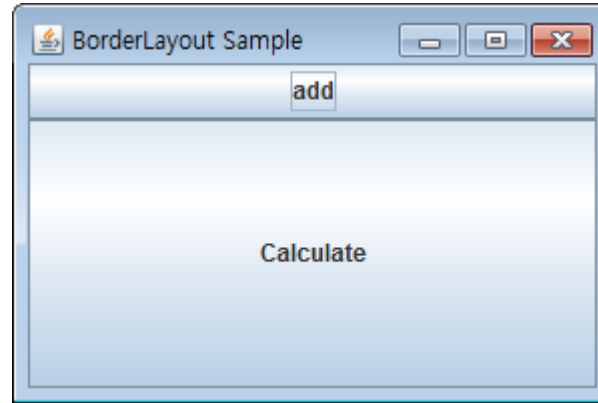
## □ 생성자

- BorderLayout()
- BorderLayout(int hGap, int vGap)
  - hGap : 좌우 두 컴포넌트 사이의 수평 간격, 픽셀 단위(디폴트 : 0)
  - vGap : 상하 두 컴포넌트 사이의 수직 간격, 픽셀 단위(디폴트 : 0)

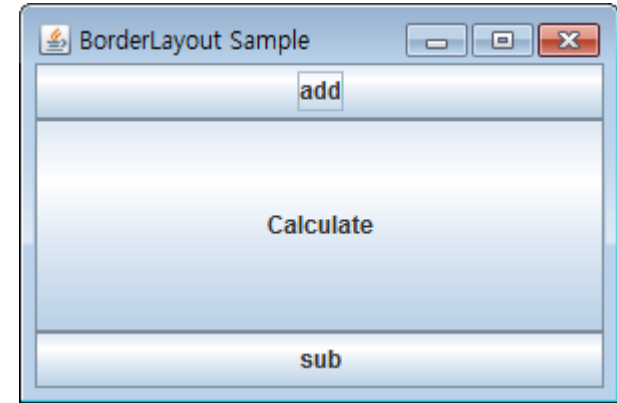
# BorderLayout의 사용예



CENTER에 컴포넌트가 삽입될 때

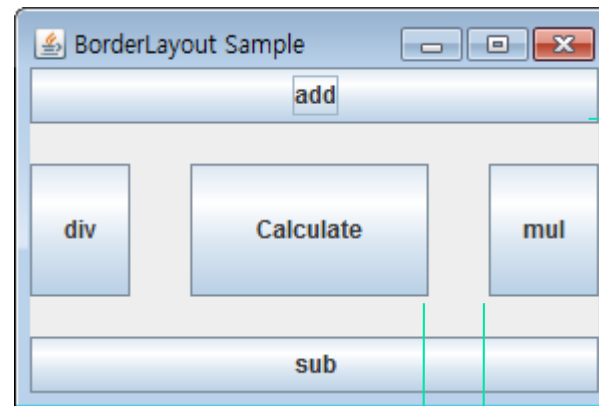


CENTER와 NORTH에 컴포넌트가 삽입될 때



CENTER, NORTH, SOUTH에  
컴포넌트가 삽입될 때

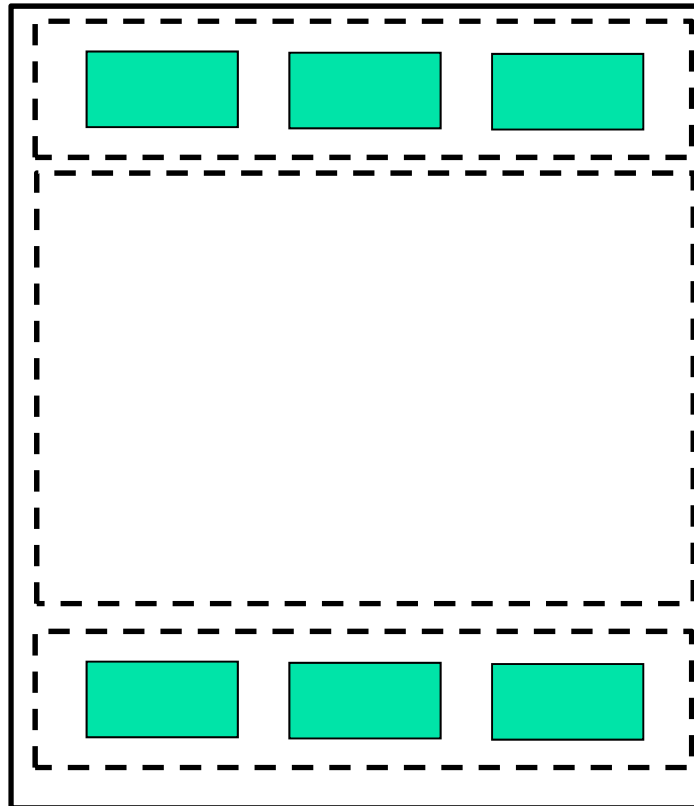
`new BorderLayout(30,20);`  
으로 배치관리자를  
생성하였을 때



vGap,  
20 픽셀

hGap , 30 픽셀

# BorderLayout의 사용예



# BorderLayout 사용 예

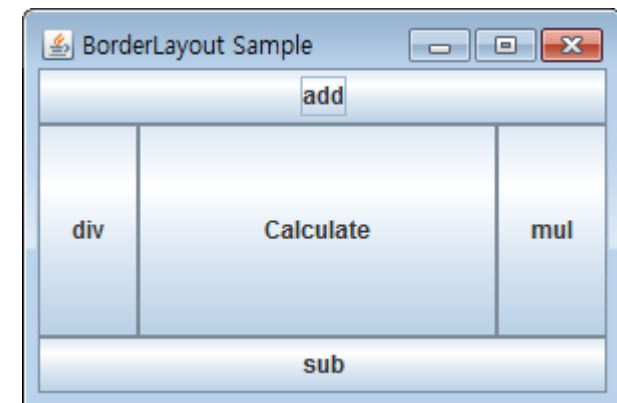
```
import javax.swing.*.*;
import java.awt.*.*;

public class BorderLayoutEx extends JFrame {
    BorderLayoutEx() {
        setTitle("BorderLayout Sample");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        setLayout(new BorderLayout());
        add(new JButton("add"), BorderLayout.NORTH);
        add(new JButton("sub"), BorderLayout.SOUTH);
        add(new JButton("mul"), BorderLayout.EAST);
        add(new JButton("div"), BorderLayout.WEST);
        add(new JButton("Calculate"), BorderLayout.CENTER);

        setSize(300, 200);
        setVisible(true);
    }

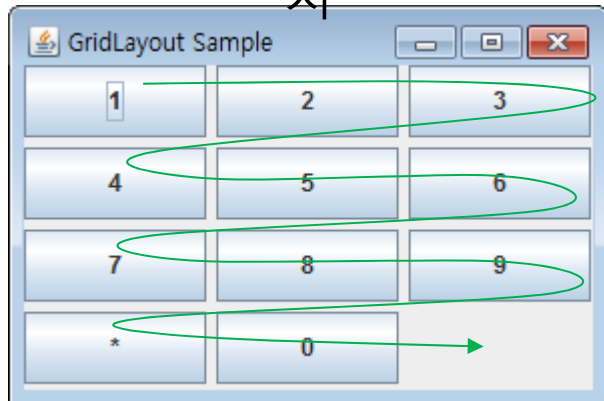
    public static void main(String[] args) {
        new BorderLayoutEx();
    }
}
```



# GridLayout

## 배치방법

- 컨테이너 공간을 동일한 사각형 격자(그리드)로 분할하고 각 셀에 하나의 컴포넌트 배치
  - 격자 구성은 생성자에서 행수와 열수로 지정
  - 셀에 왼쪽에서 오른쪽으로, 다시 위에서 아래로 순서대로 배치



```
container.setLayout(new GridLayout(4,3,5,5)); // 4×3 분할로 컴포넌트 배치  
container.add(new JButton("1")); // 상단 왼쪽 첫 번째 셀에 버튼 배치  
container.add(new JButton("2")); // 그 옆 셀에 버튼 배치
```

- 4x3 그리드 레이아웃 설정
- 총 11 개의 버튼이 순서대로 add 됨
- 수직 간격 vGap : 5 픽셀
- 수평 간격 hGap : 5 픽셀

- 컨테이너의 크기가 변하면 재배포
  - 크기 재조정

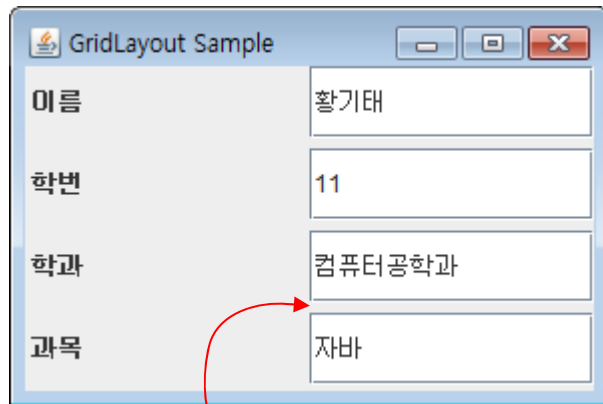


# GridLayout 생성자와 속성

## □ 생성자

- GridLayout()
- GridLayout(int rows, int cols)
- GridLayout(int rows, int cols, int hGap, int vGap)
  - rows : 격자의 행수 (디폴트 : 1)
  - cols : 격자의 열수 (디폴트 : 1)
  - hGap : 좌우 두 컴포넌트 사이의 수평 간격, 픽셀 단위(디폴트 : 0)
  - vGap : 상하 두 컴포넌트 사이의 수직 간격, 픽셀 단위(디폴트 : 0)
  - rows x cols 만큼의 셀을 가진 격자로 컨테이너 공간을 분할, 배치

# GridLayout으로 입력 폼 만들기



두 행 사이의 수직 간격  
vGap이 5 픽셀로 설정됨

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

public class GridLayoutEx extends JFrame {
    GridLayoutEx() {
        setTitle("GridLayout Sample");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        GridLayout grid = new GridLayout(4, 2);
        grid.setVgap(5);
        setLayout(grid);
        add(new JLabel(" 이름"));
        add(new JTextField(""));
        add(new JLabel(" 학번"));
        add(new JTextField(""));
        add(new JLabel(" 학과"));
        add(new JTextField(""));
        add(new JLabel(" 과목"));
        add(new JTextField(""));

        setSize(300, 200);
        setVisible(true);
    }

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



# 배치관리자 없는 컨테이너(위치지정)

- 배치관리자가 없는 컨테이너 개념
  - 응용프로그램에서 컴포넌트의 절대 크기와 절대 위치를 스스로 결정
- 용도
  - 컴포넌트의 크기나 위치를 개발자 임의로 결정하고자 하는 경우
  - 게임 프로그램과 같이 시간이나 마우스/키보드의 입력에 따라 컴포넌트들의 위치와 크기가 수시로 변하는 경우
  - 여러 컴포넌트들이 서로 겹치는 효과를 연출하고자 하는 경우
- 컨테이너의 배치 관리자 제거 방법
  - `Container.setLayout(null);`

```
// JPanel 에 배치관리자를 삭제하는 예  
  
JPanel p = new JPanel();  
p.setLayout(null);
```

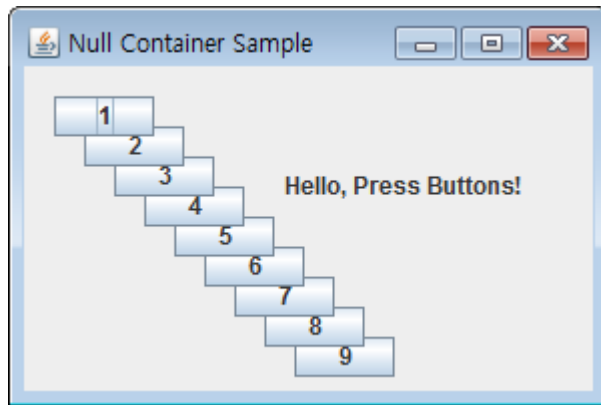


# 배치관리자 없는 컨테이너

## □ 컴포넌트의 크기와 위치 설정

- 프로그램 내에서 이루어져야 함
- 컴포넌트들이 서로 겹치는 효과 연출 가능
- 다음 메소드 이용
  - 컴포넌트 크기 설정
    - `Component.setSize(int width, int height);`
  - 컴포넌트 위치 설정
    - `Component.setLocation(int x, int y);`
  - 컴포넌트 위치와 크기 동시 설정
    - `Component.setBounds(int x, int y, int width, int height);`

# 배치관리자 없는 컨테이너에 컴포넌트 위치와 크기를 절대적으로 지정



원하는 위치에 원하는 크기로  
컴포넌트를 마음대로  
배치할 수 있다.

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

public class NullContainerEx extends JFrame {
    NullContainerEx() {
        setTitle("Null Container Sample");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLayout(null);

        JLabel la = new JLabel("Hello, Press Buttons!");
        la.setLocation(130, 50);
        la.setSize(200, 20);
        add(la);
        for(int i=1; i<=9; i++) {
            JButton b = new JButton(Integer.toString(i));
            b.setLocation(i*15, i*15);
            b.setSize(50, 20);
            add(b);
        }
        setSize(300, 200);
        setVisible(true);
    }

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

# Layout의 예

원도우 제목

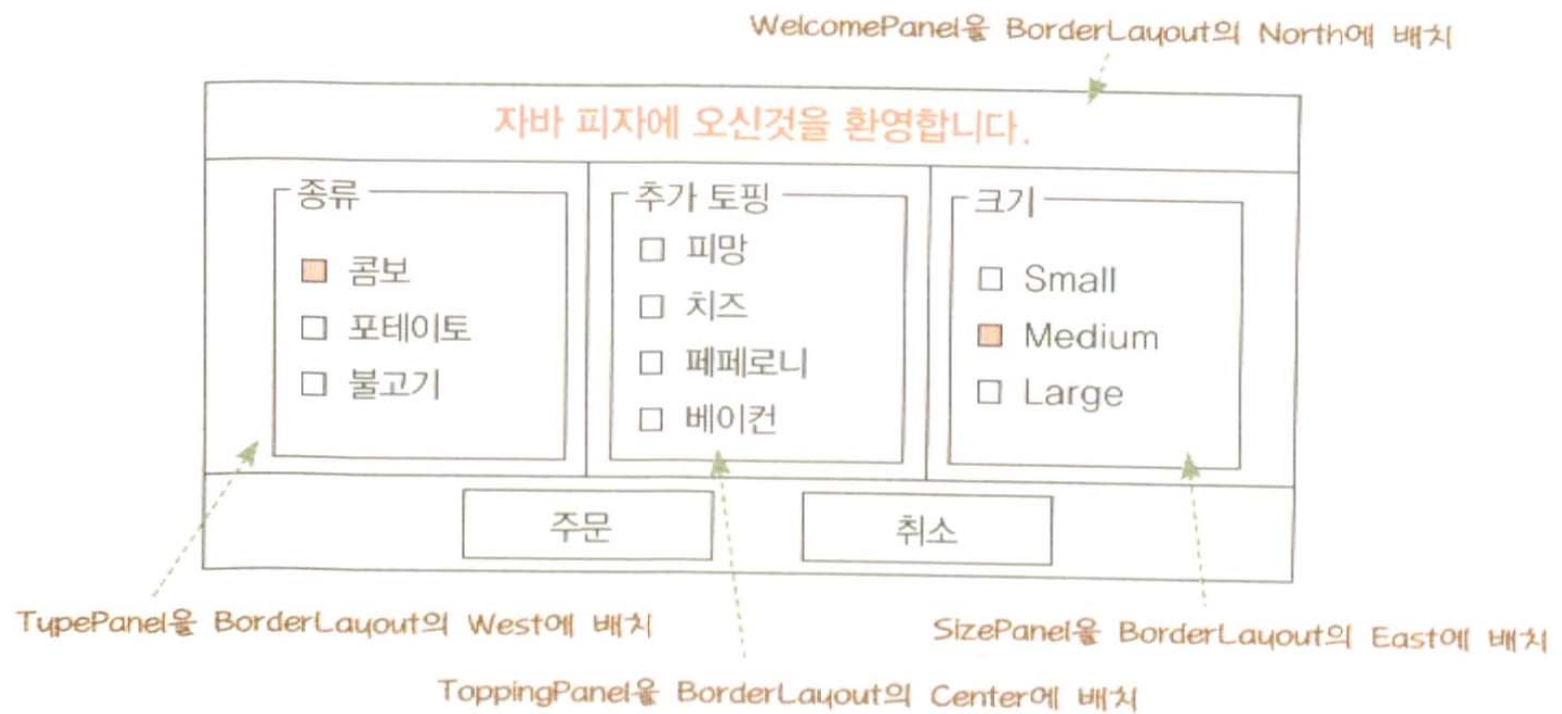
피자 주문

자바 피자에 오신것을 환영합니다.

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

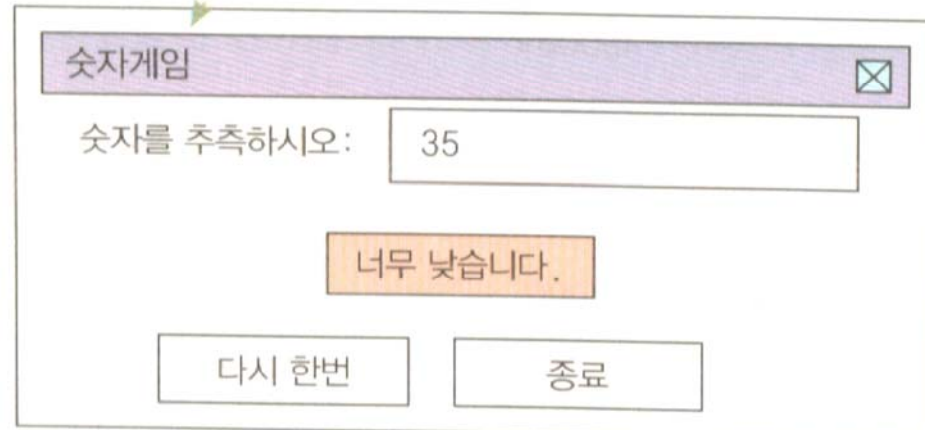
주문 취소

# Layout의 예



# Layout의 예

원도우 제목



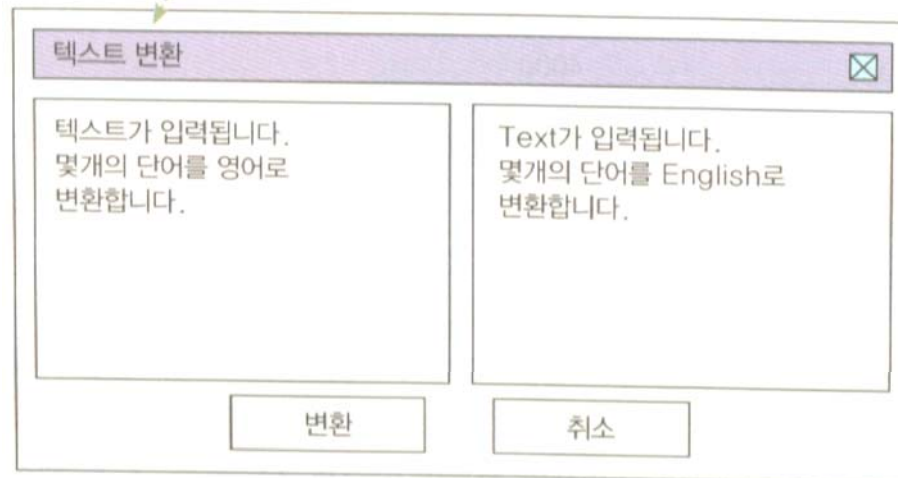
숫자게임

숫자를 추측하시오: 35

너무 낮습니다.

다시 한번 종료

원도우 제목



텍스트 변환


텍스트가 입력됩니다.  
몇개의 단어를 영어로  
변환합니다.

Text가 입력됩니다.  
몇개의 단어를 English로  
변환합니다.

변환 취소



# Layout의 예제



이메일

비밀번호

로그인

☒ 로그인 상태 유지    [도움이 필요하세요?](#)

## □ JLabel에 HTML 쓰기

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

public class JLabels extends JFrame {
    public static void main(String[] args) {

        JLabels jlabels = new JLabels();

        jlabels.setVisible(true);
    }

    public JLabels() {
        super("Using HTML in JLabels");

        Container content = getContentPane();

        // 컬러 이름 쓰기
        String labelText =
            "<html><FONT COLOR=RED>남</FONT> " +
            "<FONT COLOR=BLUE>광우</FONT> (kwnam@kunsan.ac.kr)</html>";

        JLabel coloredLabel =
            new JLabel(labelText, JLabel.CENTER);

        coloredLabel.setBorder (BorderFactory.createTitledBorder("이름"));
        content.add(coloredLabel, BorderLayout.NORTH);
    }
}
```

## □ JLabel에 HTML 쓰기(계속)

```
// 굵은 글씨 소속 쓰기
labelText =
    "<html><B>군산대학교</B> <I>컴퓨터정보과학과</I> 교수</html>";
JLabel boldLabel =
    new JLabel(labelText, JLabel.CENTER);
boldLabel.setBorder
    (BorderFactory.createTitledBorder("소속"));
content.add(boldLabel, BorderLayout.CENTER);

// 설명 쓰기..
labelText =
    "<html>The MCA Laboratory is a division " +
    "of the Kunsan National University." +
    "<P>" +
    "MCA Lab. include:" +    "<UL>" +
    "  <LI>Mobile Embbeded Applications" +    "  <LI>Convergence Application" +
    "  <LI>Wireless Network Applications" +    "  <LI>Telematics Software" +
    "  <LI>Location-Based Applications" +    "</UL>";
JLabel fancyLabel =    new JLabel(labelText, new ImageIcon("F:WWW사진WWW남광우.jpg"), JLabel.CENTER);
fancyLabel.setBorder (BorderFactory.createTitledBorder("Multi-line HTML"));

content.add(fancyLabel, BorderLayout.SOUTH);

pack();

}
```

# Image JButton

## □ Image JButton

### ■ Image를 갖는 JButton 만들기

```
Container content = getContentPane();  
content.setBackground(Color.white);  
content.setLayout(new FlowLayout());
```

```
JButton button1 = new JButton("Java");  
content.add(button1);
```

```
ImageIcon cup = new ImageIcon("images/cup.gif");  
JButton button2 = new JButton(cup);  
content.add(button2);
```

```
JButton button3 = new JButton("Java", cup);  
content.add(button3);
```

```
JButton button4 = new JButton("Java", cup);  
button4.setHorizontalTextPosition(SwingConstants.LEFT);  
content.add(button4);
```



SwingConstants.RIGHT  
SwingConstants.LEFT  
SwingConstants.CENTER  
SwingConstants.LEADING  
SwingConstants.TRAILING (the default)

# JTextField

## JTextField



```
JTextField field = new JTextField(20);  
JTextField field = new JTextField("Your name");  
System.out.println(field.getText());  
field.setText("whatever");  
field.setText("");  
field.addActionListener(myActionListener);  
field.selectAll();  
field.requestFocus();
```

# Layout+ 예제

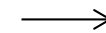
## □ 학생정보 입력화면

학생정보

이름

학번

전화

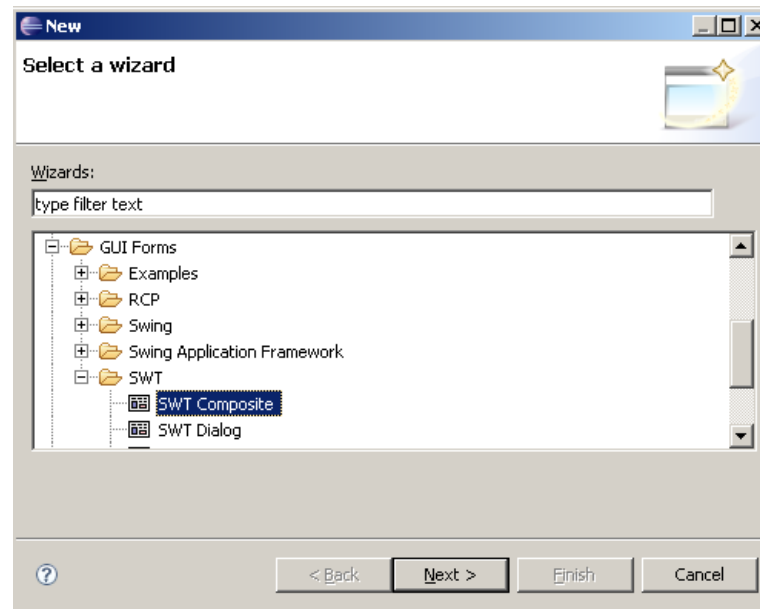


저장을 누르면  
system.out.println으로  
정보 출력

# \*Jigloo Tool의 설치

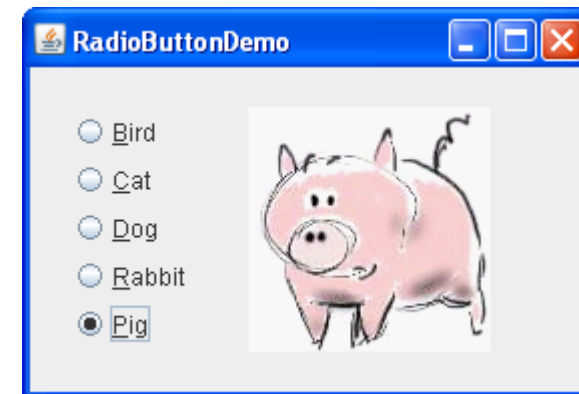
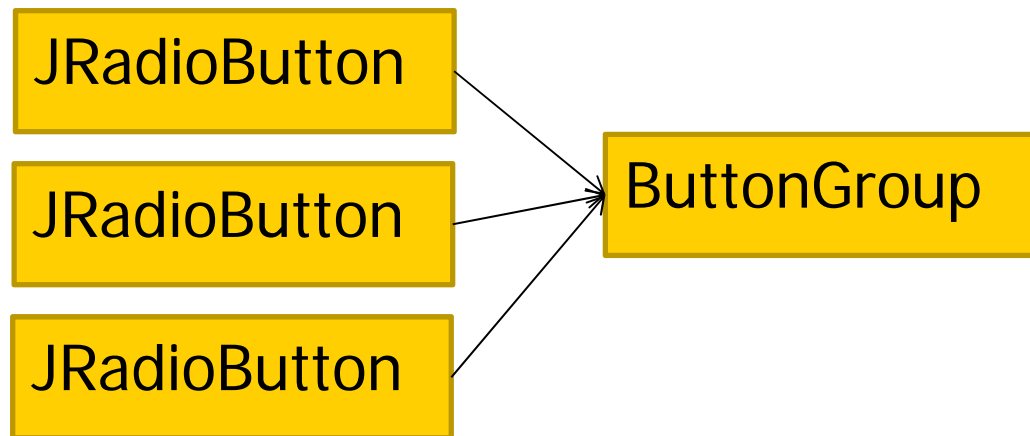
## □ Jigloo

- 가장 많이 사용되는 Swing/SWT GUI 툴중 하나
- Install New Software에서
  - <http://cloudgarden1.com/update-site/>
  - 등록후 설치



# RadioButton

## □ RadioButton/ButtonGroup



<http://docs.oracle.com/javase/tutorial/uiswing/examples/components/RadioButtonDemoProject/src/components/RadioButtonDemo.java>

# RadioButton

## □ RadioButtonDemo

단축키 ALT-B

```
//In initialization code:
//Create the radio buttons.
JRadioButton birdButton = new JRadioButton(birdString);
birdButton.setMnemonic(KeyEvent.VK_B);
birdButton.setActionCommand(birdString);
birdButton.setSelected(true);

JRadioButton catButton = new JRadioButton(catString);
catButton.setMnemonic(KeyEvent.VK_C);
catButton.setActionCommand(catString);

JRadioButton dogButton = new JRadioButton(dogString);
dogButton.setMnemonic(KeyEvent.VK_D);
dogButton.setActionCommand(dogString);

JRadioButton rabbitButton
    = new JRadioButton(rabbitString);
rabbitButton.setMnemonic(KeyEvent.VK_R);
rabbitButton.setActionCommand(rabbitString);

JRadioButton pigButton = new JRadioButton(pigString);
pigButton.setMnemonic(KeyEvent.VK_P);
pigButton.setActionCommand(pigString);
```

```
//Group the radio buttons.
ButtonGroup group = new ButtonGroup();
group.add(birdButton);
group.add(catButton);
group.add(dogButton);
group.add(rabbitButton);
group.add(pigButton);

//Register a listener for the radio buttons:
birdButton.addActionListener(this);
catButton.addActionListener(this);
dogButton.addActionListener(this);
rabbitButton.addActionListener(this);
pigButton.addActionListener(this);

...
public void actionPerformed(ActionEvent e) {
    picture.setIcon(new ImageIcon("images/"
        + e.getActionCommand()
        + ".gif"));
}
```

full source :

<http://docs.oracle.com/javase/tutorial/uiswing/examples/components/RadioButtonDemoProject/src/components/RadioButtonDemo.java>

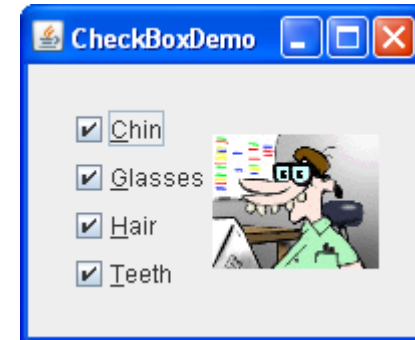
# JCheckBox

## JCheckBox

```
JCheckBox chinButton = new JCheckBox("Chin");  
chinButton.setMnemonic(KeyEvent.VK_C);  
chinButton.setSelected(true);  
chinButton.addItemListener(this);
```

```
public void itemStateChanged(ItemEvent e)  
{  
    String onOrOff = "off";  
    if (chinButton.isSelected()) onOrOff = "on";  
    System.out.println("Check box is " + onOrOff);  
}
```

```
check.setSelected(true);  
check.setSelected(false);
```





# JCheckBox

---

```
public CheckBoxDemo() {  
    super(new BorderLayout());  
    //Create the check boxes.  
    chinButton = new JCheckBox("Chin");  
    chinButton.setMnemonic(KeyEvent.VK_C);  
    chinButton.setSelected(true);  
  
    chinButton.addItemListener(this);  
  
    //Put the check boxes in a column in a panel  
    JPanel checkPanel = new JPanel(new GridLayout(0, 1));  
    checkPanel.add(chinButton);  
    add(checkPanel, BorderLayout.LINE_START);  
  
}
```

full source :

<http://docs.oracle.com/javase/tutorial/uiswing/examples/components/CheckBoxDemoProject/src/components/CheckBoxDemo.java>

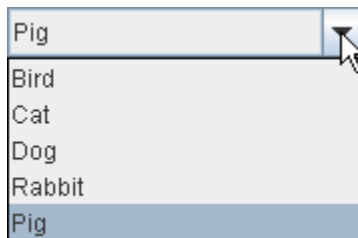


# JCheckBox

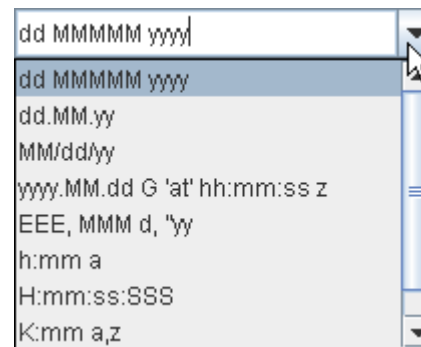
```
public void itemStateChanged(ItemEvent e) {
    int index = 0;
    char c = '-';
    Object source = e.getItemSelectable();
    if (source == chinButton) {
        index = 0;
        c = 'c';
    } else if (source == glassesButton) {
        index = 1;
        c = 'g';
    } else if (source == hairButton) {
        index = 2;
        c = 'h';
    } else if (source == teethButton) {
        index = 3;
        c = 't';
    }
    //Now that we know which button was pushed, find out
    //whether it was selected or deselected.
    if (e.getStateChange() == ItemEvent.DESELECTED) {
        c = '-';
    }
    //Apply the change to the string.
    choices.setCharAt(index, c);
    updatePicture();
}
```

# JComboBox

## □ JComboBox의 형태



Uneditable JComboBox



Editable JComboBox

# JComboBox

## □ Uneditable JComboBox

```
String[] petStrings = { "Bird", "Cat", "Dog", "Rabbit",  
"Pig" };
```

```
//Create the combo box, select item at index 4.  
//Indices start at 0, so 4 specifies the pig.
```

```
JComboBox petList = new JComboBox(petStrings);  
petList.setSelectedIndex(4);  
petList.addActionListener(this);
```



Editable로 전환

```
patternList.setEditable(true);
```



# JComboBox

## □ Uneditable JComboBox : Event 처리

```
public class ComboBoxDemo ... implements ActionListener {  
    ...  
    petList.addActionListener(this) {  
    ...  
    public void actionPerformed(ActionEvent e) {  
        JComboBox cb = (JComboBox)e.getSource();  
        String petName = (String)cb.getSelectedItem();  
        updateLabel(petName);  
    }  
    ...  
}
```

# JComboBox

```
public JComboBoxDemo() {
    super(new BorderLayout());
    String[] petStrings = { "Bird", "Cat", "Dog", "Rabbit", "Pig"
    };
    //Create the combo box, select the item at index 4.
    //Indices start at 0, so 4 specifies the pig.
    JComboBox petList = new JComboBox(petStrings);
    petList.setSelectedIndex(4);
    petList.addActionListener(this);
    //Set up the picture.
    picture = new JLabel();
    picture.setFont(picture.getFont().deriveFont(Font.ITALIC));
    picture.setHorizontalAlignment(JLabel.CENTER);
    updateLabel(petStrings[petList.getSelectedIndex()]);
    picture.setBorder(BorderFactory.createEmptyBorder(10,0,0,0));
    //The preferred size is hard-coded to be the width of the
    //widest image and the height of the tallest image + the
    border.
    //A real program would compute this.
    picture.setPreferredSize(new Dimension(177, 122+10));
    //Lay out the demo.
    add(petList, BorderLayout.PAGE_START);
    add(picture, BorderLayout.PAGE_END);
    setBorder(BorderFactory.createEmptyBorder(20,20,20,20));
}

/** Listens to the combo box. */
public void actionPerformed(ActionEvent e) {
    JComboBox cb = (JComboBox)e.getSource();
    String petName = (String)cb.getSelectedItem();
    updateLabel(petName);
}
```