

# Java GUI

## 자바 GUI의 특징

강력한 GUI 컴퓨넌트 제공

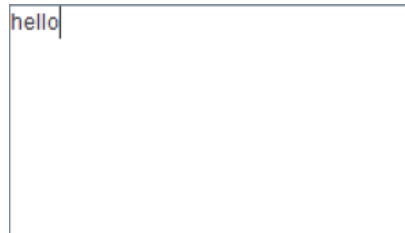
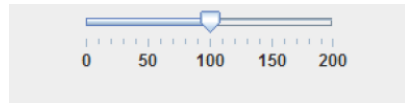
쉬운 GUI 프로그래밍 (관련 클래스들을 패키지로 개발이되어 제공되므로 가져다 사용만 하시면 됩니다.

## 자바의 GUI 프로그래밍 방법

AWT 패키지(java.awt)와 Swing 패키지(javax.swing)가 사용

# JAVA의 GUI 종류

Month:



|                           |  |
|---------------------------|--|
| JApplet                   |  |
| JButton                   |  |
| JCheckBox                 |  |
| JComboBox                 |  |
| JDialog                   |  |
| JEditorPane and JTextPane |  |
| JFrame                    |  |
| JList                     |  |
| JPasswordField            |  |
| JProgressBar              |  |
| JRadioButton              |  |
| JScrollPane               |  |
| JSlider                   |  |
| JSpinner                  |  |
| JSplitPane                |  |
| JTabbedPane               |  |
| JTable                    |  |
| JTextArea                 |  |
| TextField                 |  |
| JToolBar                  |  |
| JToolTip                  |  |
| JTree                     |  |

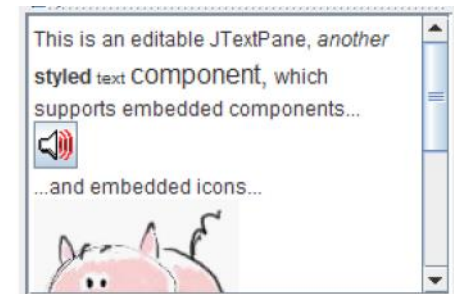
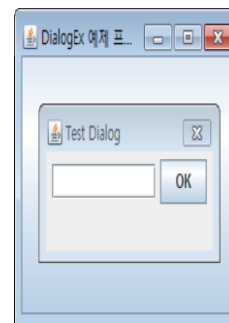
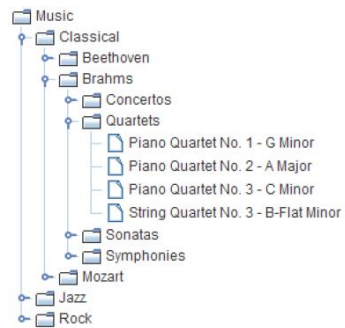
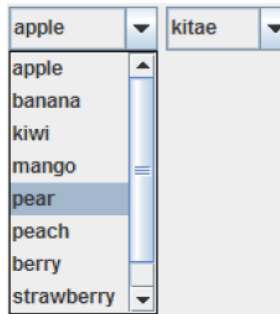


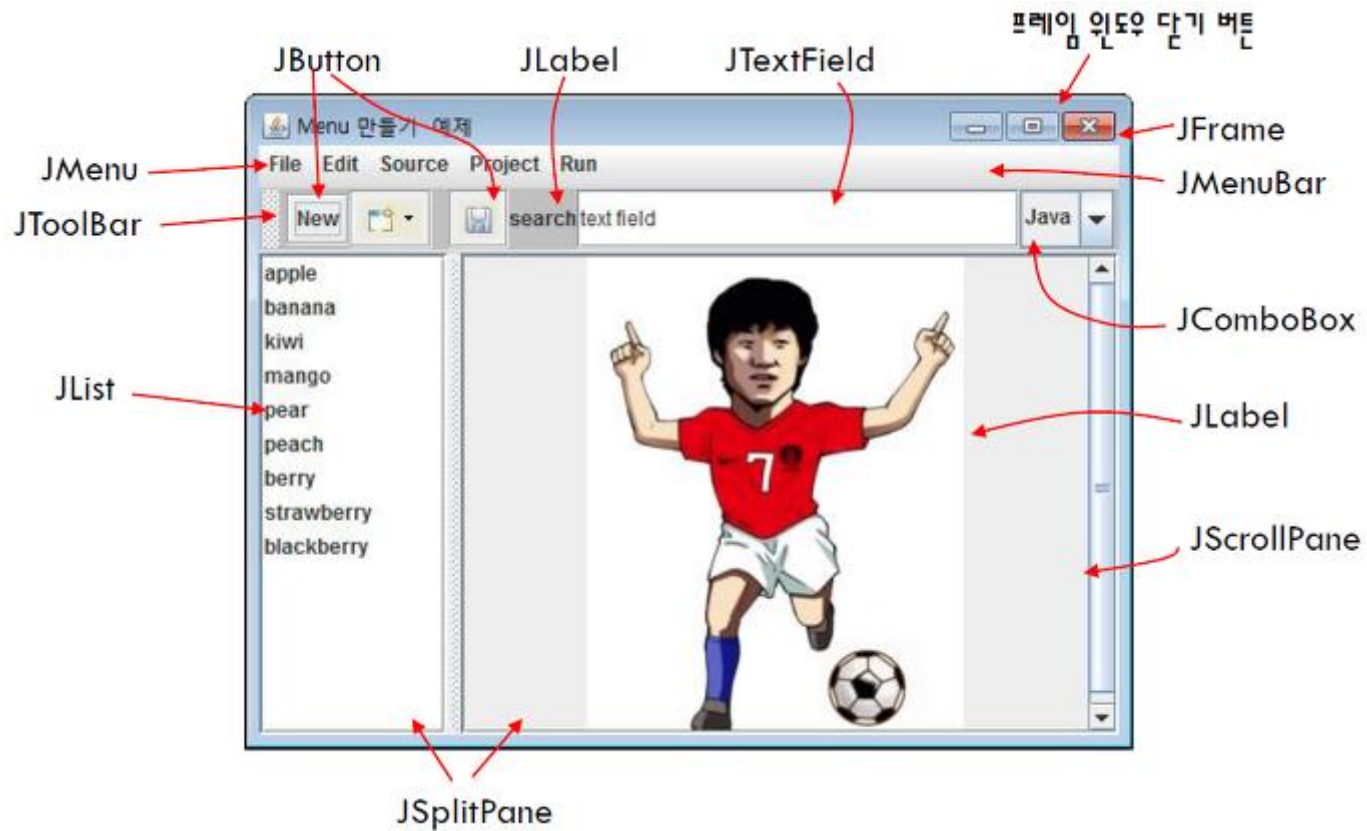
이름 :  학과 : 컴퓨터공학과(수정)

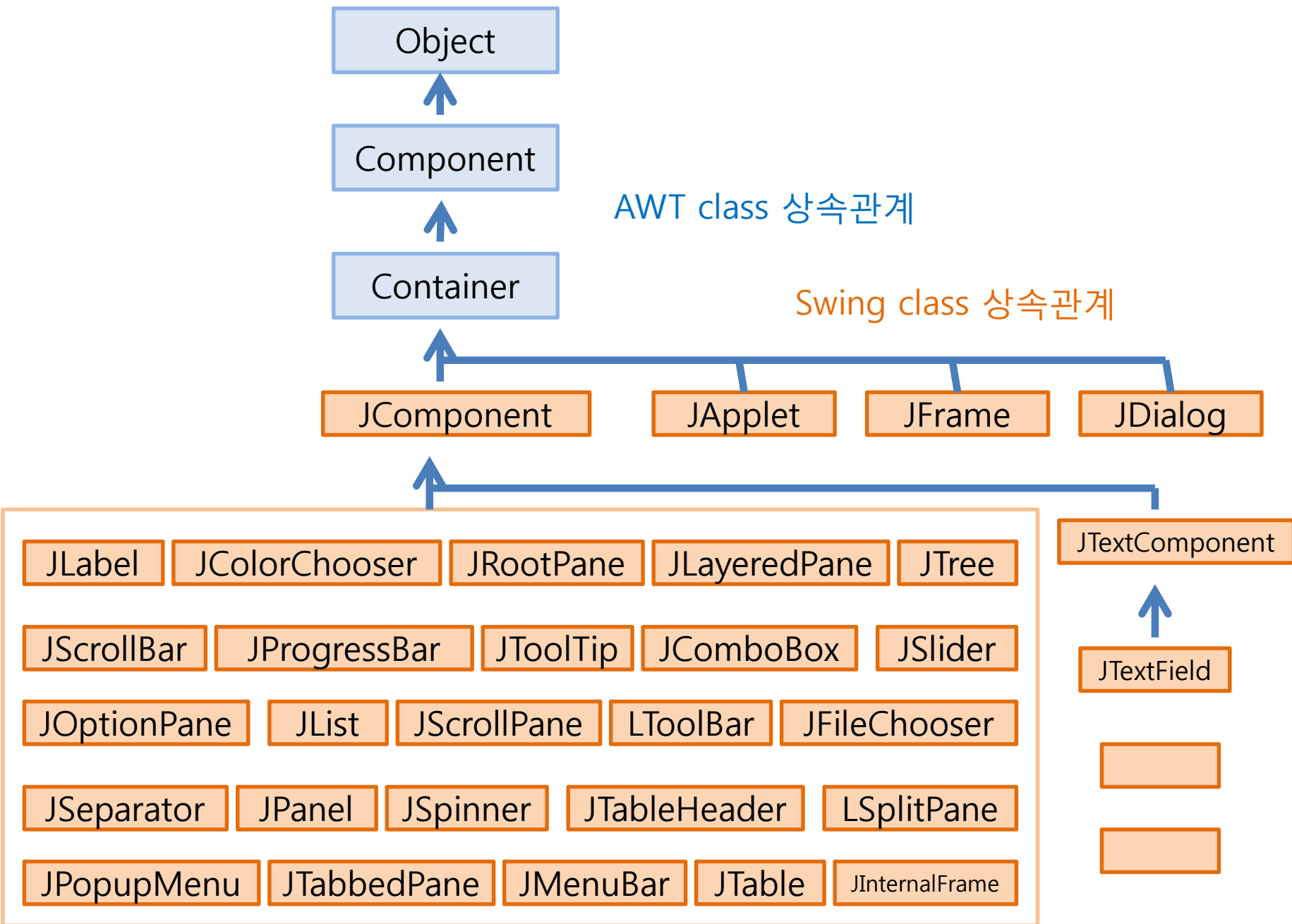
주소 :



Enter the password:





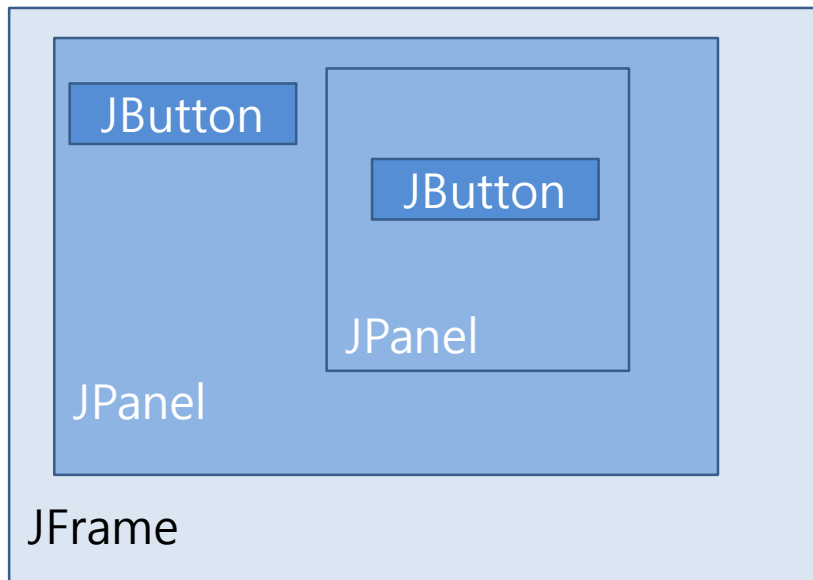


컨테이너 : 다른 component를 포함하는 GUI

스윙 컨테이너 : JFrame, JPanel , JDialog, JWindow, JApplet

최상위 컨테이너 : 다른 컨테이너에 속하지 않고도 독립적으로 존재하여 출력될수 있다. JFrame, JDialog, JApplet

Component : Container와 달리 다른 컴포넌트를 포함할 수 없다.  
스윙컴포넌트는 javax.swing.JComponent 를 상속받는다.



<컨테이너와 컴포넌트의 포함관계 >

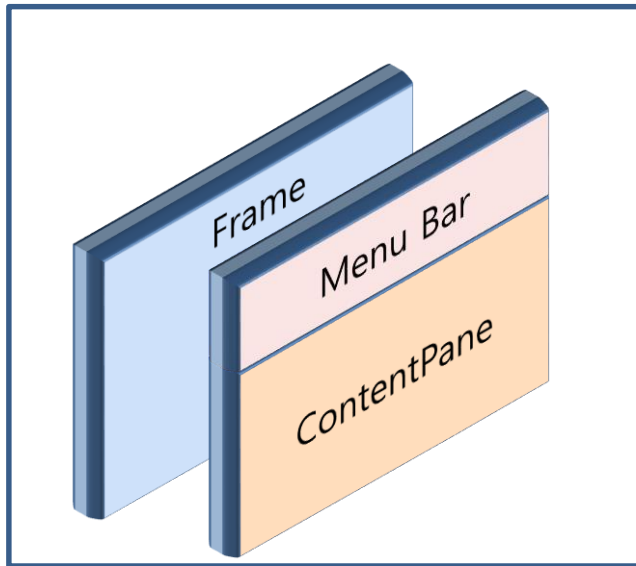
## 스윙 GUI 이용 방법

- 프레임 만들기, 프레임에 스윙 컴포넌트 붙이기, 메인 메소드에서 프레임 사용

스윙 클래스를 import한다.

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

JFrame 구조



## JFrame 구현 방법 1

```
import javax.swing.*;

public class MyFrame {

    public static void main(String[] args) {

        JFrame f = new JFrame();

        f.setTitle("ContentPane과 JFrame"); // 프레임의 타이틀 달기

        f.setSize(300, 150); // 프레임 크기 300x150 설정

        f.setVisible(true); // 프레임을 화면에 출력

    }
}
```



## JFrame 구현 방법 2

```
import javax.swing.*;

public class MyFrame extends JFrame {

    MyFrame() {
        setTitle("ContentPane과 JFrame"); // 프레임의 타이틀 달기

        setSize(300, 150); // 프레임 크기 300x150 설정

        setVisible(true); // 프레임을 화면에 출력
    }

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

```
import javax.swing.*;

public class ContentPaneEx extends JFrame {
    ContentPaneEx() {
        setTitle("ContentPane과 JFrame"); // 프레임의 타이틀 달기
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        // 프레임 윈도우를 닫으면 프로그램 종료
        Container contentPane = getContentPane(); // 콘텐츠 팬을 알아낸다.
        contentPane.setBackground(Color.ORANGE); // 오렌지색 배경 설정
        contentPane.setLayout(new FlowLayout()); // 콘텐츠팬에 FlowLayout 배치관리자
        contentPane.add(new JButton("OK")); // OK 버튼 달기
        contentPane.add(new JButton("Cancel")); // Cancel 버튼 달기
        contentPane.add(new JButton("Ignore")); // Ignore 버튼 달기

        setSize(300, 150); // 프레임 크기 300x150 설정
        setVisible(true); // 프레임을 화면에 출력
    }

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

## 컨테이너와 배치(Layout)

- 컨테이너는 반드시 배치관리자를 가진다.
- 배치 관리자의 유형 (FlowLayout, BorderLayout, GridLayout, CardLayout)
- 배치 관리자 java.awt에 존재하는 클래스이다. (import java.awt.\* ;)

## 컨테이너의 디폴트 배치 관리자

-JFrame : *BorderLayout*, JPanel : *FlowLayout*, JDialog : *BorderLayout*  
JApplet : *FlowLayout* , JWindow : *BorderLayout*

## FlowLayout생성자

```
new FlowLayout();//중앙 정렬과 hGap=5, vGap=5인 배치관리자 생성
new FlowLayout(int align) ; //중앙정렬(FlowLayout.CENTER),
//왼쪽정렬(FlowLayout.LEFT) , 오른쪽정렬(FlowLayout.RIGHT)
new FlowLayout(int align, int hGap, int vGap)
//hGap(기본5픽셀:가로컴포넌트간격) hGap(기본5픽셀:세로컴포넌트간격)
```

Left로 정렬되고 수평간격이 30, 수직간격이 40 픽셀인 FlowLayout배치관리자

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

public class FlowLayoutEx extends JFrame {
    FlowLayoutEx() {
        setTitle("FlowLayout Sample"); // 프레임의 타이틀 달기
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        Container c = getContentPane();
        c.setLayout(new FlowLayout(FlowLayout.LEFT, 30, 40));
        c.add(new JButton("add"));
        c.add(new JButton("sub"));
        c.add(new JButton("mul"));
        c.add(new JButton("div"));
        c.add(new JButton("Calculate"));

        setSize(300, 200); // 프레임 크기 300x200 설정
        setVisible(true); // 프레임을 화면에 출력
    }
    public static void main(String[] args) {
        new FlowLayoutEx();
    }
}
```

## BorderLayout 생성자

```
new BorderLayout();  
new BorderLayout(int hGap, int vGap);
```

**BorderLayout배지 관리자를 가진 컨테이너에 컴포넌트 삽입시 위치 명시해야함**

```
add(Component comp, int index);
```

index : 동 - BorderLayout.EAST  
서 - BorderLayout.WEST  
남 - BorderLayout.SOUTH  
북 - BorderLayout.NORTH  
중앙 - BorderLayout.CENTER

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

public class BorderLayoutEx extends JFrame {
    BorderLayoutEx() {
        setTitle("BorderLayout Sample"); // 프레임의 타이틀 달기
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        Container c = getContentPane();
        c.setLayout(new BorderLayout()); // 콘텐츠팬에 BorderLayout 배치관리자 설정
        c.add(new JButton("Calculate"), BorderLayout.CENTER);
        c.add(new JButton("add"), BorderLayout.NORTH);
        c.add(new JButton("sub"), BorderLayout.SOUTH);
        c.add(new JButton("mul"), BorderLayout.EAST);
        c.add(new JButton("div"), BorderLayout.WEST);

        setSize(300, 200); // 프레임 크기 300x200 설정
        setVisible(true); // 프레임을 화면에 출력
    }
    public static void main(String[] args) {
        new BorderLayoutEx();
    }
}
```

## **GridLayout** 생성자와 속성

GridLayout()

hGap=0, vGap=0, rows=1, cols=1

GridLayout(int rows, int cols)

hGap=0, vGap=0

GridLayout(int rows, int cols, int hGap, int vGap)

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

```
public class GridLayoutEx extends JFrame {
    GridLayoutEx() {
        setTitle("GridLayout Sample"); // 프레임의 타이틀 달기
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        Container c = getContentPane();
        GridLayout grid = new GridLayout(4, 2); // 4x2 격자의 GridLayout 배치관리자
        grid.setVgap(5); //격자 사이의 수직 간격을 5 픽셀로 설정
        c.setLayout(grid); // grid를 컨테이너의 배치관리자로 지정
        c.add(new JLabel("이름"));
        c.add(new JTextField(""));
        c.add(new JLabel("학번"));
        c.add(new JTextField(""));
        c.add(new JLabel("학과"));
        c.add(new JTextField(""));
        c.add(new JLabel("과목"));
        c.add(new JTextField(""));
        setSize(300, 200);
        setVisible(true);
    }
    public static void main(String[] args) {
        new GridLayoutEx();
    }
}
```



**컨테이너의 배치관리자가 없는 경우**  
컴포넌트의 절대적 위치와 크기 설정

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

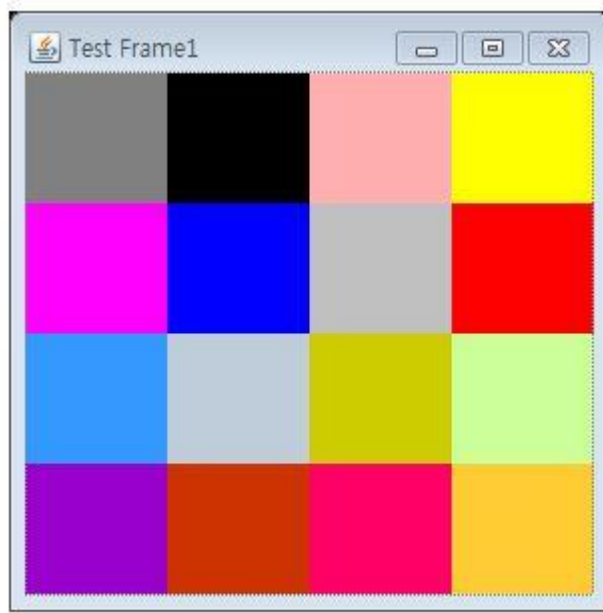
```
public class NullContainerEx extends JFrame {  
    NullContainerEx() {  
        setTitle("Null Container Sample");  
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        Container c = getContentPane();  
        c.setLayout(null); // 컨텐트팬의 배치관리자 제거  
  
        // JLabel 컴포넌트 생성하고 직접 위치와 크기를 지정한다.  
        JLabel la = new JLabel("Hello, Press Buttons!");  
        la.setLocation(130, 50); // la를 (130,50) 위치로 지정  
        la.setSize(200, 20); // la를 200x20 크기로 지정  
        c.add(la); // la를 컨텐트팬에 부착  
  
        // 9개의 버튼 컴포넌트를 생성하고 모두 동일한 크기로 설정한다.  
        // 위치는 서로 다르게 설정  
        for(int i=1; i<=9; i++) {  
            JButton b = new JButton(Integer.toString(i)); // 버튼 생성  
            b.setLocation(i*15, i*15); // 버튼의 위치 설정  
            b.setSize(50, 20); // 9 개의 버튼의 크기는 동일하게 50x20 크기  
            c.add(b); // 버튼을 컨텐트팬에 부착  
        }  
    }  
}
```

```
        setSize(300, 200);  
        setVisible(true);  
    }  
    public static void main(String[] args) {  
        new NullContainerEx();  
    }  
}
```

## 실습)

다음과 같은 화면을 구성해 보자 (GridLayout 4\*4로 구성한다)  
윈도우빌더와 같이 사용해 보자, 레이블의 색깔은 마음대로 정한다.

참고 : Label 컴포넌트의 background의 색깔을 보이게 하는 메소드는  
JComponent의 setOpaque(true) 메소드를 사용하면 된다.  
`label.setOpaque(true);`



## 실습)

20개의 10\*10크기의 JLabel 컴포넌트가 프레임내의 (50,50)위치에서 (250,250) 내의 영역에서 랜덤한 위치에 출력되도록 스윙 프로그램을 작성하여라

*프레임의 크기는 300\*300으로 하고 , JLabel의 배경색은 모두 파란색으로 하라.*

*힌트) JLabel의 좌표값 `int x= .. Random()` , `int y= ...Random()`*

*`label.setLocation(x,y);`*

*`label.setSize(10,10);`*

*`label.setOpaque(true);`*

*`label.setBackgroundColor.blue);`*

