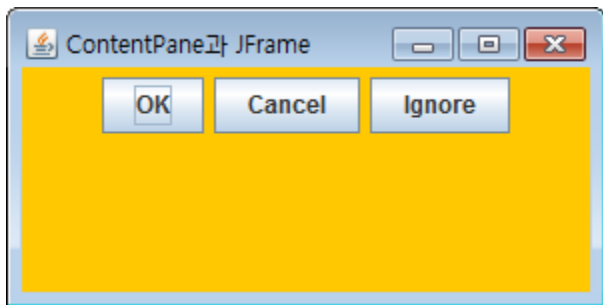


cont.

예제: 3개의 버튼 컴포넌트를 가진 스윙 프레임 만들기

☞ 다음 그림과 같이 콘텐츠팬의 배경색을 오렌지색으로 하고, OK, Cancel, Ignore 버튼을 부착한 스윙 프로그램을 작성하라.



FlowLayout의 배치관리자는
뒤에서 배울 내용으로서,
컴포넌트를 순서대로 부착하는
일을 맡은 객체

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

public class ContentPaneEx extends JFrame {
    ContentPaneEx() {
        setTitle("ContentPane과 JFrame 예제"); // 프레임의 타이틀 달기
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        Container contentPane = getContentPane(); // 콘텐츠팬 알아내기
        contentPane.setBackground(Color.ORANGE); // 오렌지색 배경 설정
        contentPane.setLayout(new FlowLayout()); // 콘텐츠팬에 FlowLayout
        // 배치관리자 달기

        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); // 화면에 프레임 출력
    }

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

(방법 1. main() 안에서 JFrame 객체를 직접 생성) 의 경우 예시

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

class ContentPaneEx {

    public static void main(String[] args) {
        JFrame mf =new JFrame();
        mf.setTitle("ContentPane과 JFrame 예제"); // 프레임의 타이틀 달기
        mf.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        Container contentPane = mf.getContentPane(); // 컨텐트팬 알아내기
        contentPane.setBackground(Color.ORANGE); // 오렌지색 배경 설정
        contentPane.setLayout(new FlowLayout()); // 컨텐트팬에 FlowLayout
                                                // 배치관리자 달기

        contentPane.add(new JButton("OK")); // OK 버튼 달기
        contentPane.add(new JButton("Cancel")); // Cancel 버튼 달기
        contentPane.add(new JButton("Ignore")); // Ignore 버튼 달기

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

(방법 1. main() 안에서 JFrame 객체를 직접 생성) 의 경우 예시

- JDK 1.5 이후는 다음과 같이 가능

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

class ContentPaneEx {

    public static void main(String[] args) {
        JFrame mf = new JFrame();
        mf.setTitle("ContentPane과 JFrame 예제"); // 프레임의 타이틀 달기
        mf.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        mf.getContentPane().setBackground(Color.ORANGE); // 오렌지색 배경 설정, getContentPane() 필요
        mf.setLayout(new FlowLayout()); // 컨텐트팬에 FlowLayout
                                         // 배치관리자 달기

        mf.add(new JButton("OK")); // OK 버튼 달기
        mf.add(new JButton("Cancel")); // Cancel 버튼 달기
        mf.add(new JButton("Ignore")); // Ignore 버튼 달기

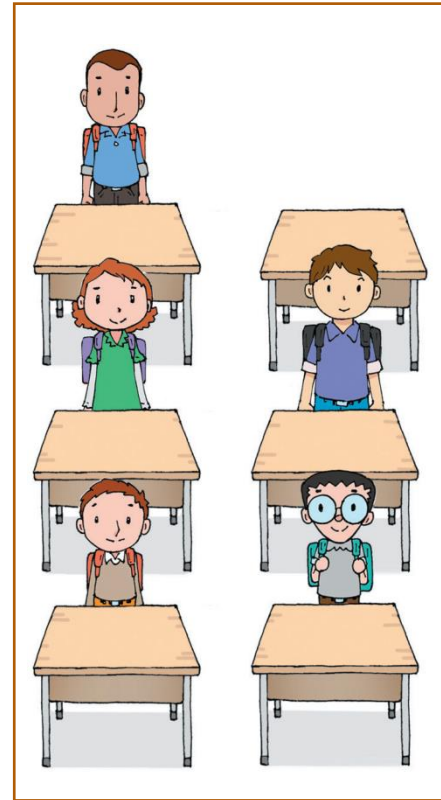
        mf.setSize(300, 150); // 프레임 크기 300x150 설정
        mf.setVisible(true); // 화면에 프레임 출력
    }
}
```

컨테이너와 배치, 배치관리자 개념

- 컨테이너의 배치관리자

- 컨테이너마다 하나의 배치관리자 존재
- 컨테이너에 부착되는 컴포넌트의 위치와 크기 결정
- 컨테이너의 크기가 변경되면, 컴포넌트의 위치와 크기 재결정

컨테이너(Container)



이쪽으로
가세요.



컴포넌트
(Component)

배치관리자
(Layout Manager)

배치 관리자 대표 유형 4 가지

- **FlowLayout 배치관리자**

- 컴포넌트가 삽입되는 순서대로 왼쪽에서 오른쪽으로 배치
- 배치할 공간이 없으면 아래로 내려와서 반복한다.

- **BorderLayout 배치관리자**

- 컨테이너의 공간을 동(EAST), 서(WEST), 남(SOUTH), 북(NORTH), 중앙(CENTER)의 5개 영역으로 나눔
- 5개 영역 중 응용프로그램에서 지정한 영역에 컴포넌트 배치

- **GridLayout 배치관리자**

- 컨테이너를 프로그램에서 설정한 동일한 크기의 2차원 격자로 나눔
- 컴포넌트는 삽입 순서대로 좌에서 우로, 다시 위에서 아래로 배치

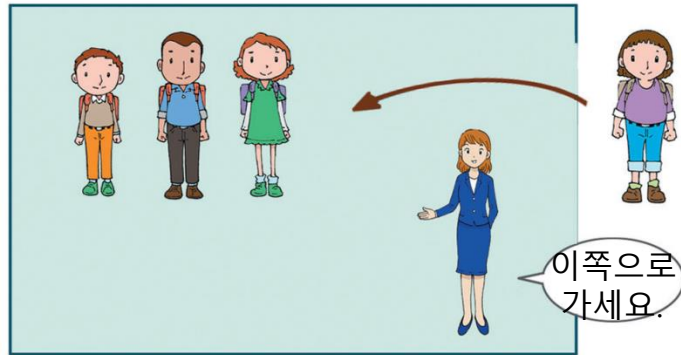
- **CardLayout 배치관리자**

- 컨테이너의 공간에 카드를 쌓아 놓은 듯이 컴포넌트를 포개어 배치

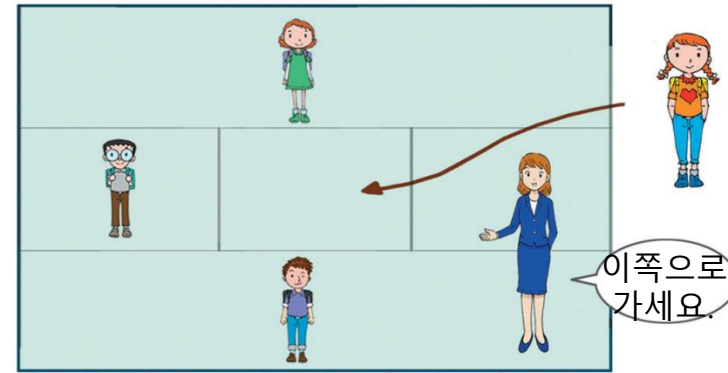
배치 관리자 대표 유형

- java.awt 패키지에 구현되어 있음

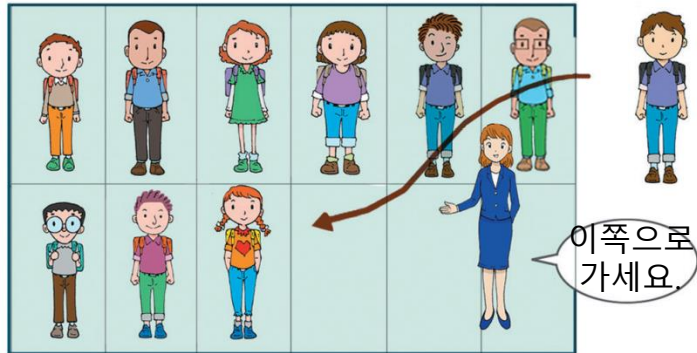
FlowLayout



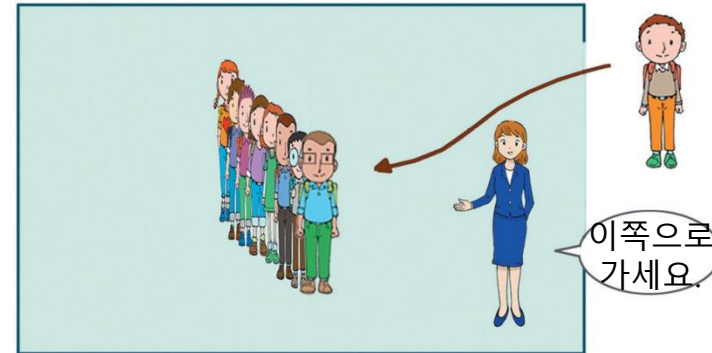
BorderLayout



GridLayout



CardLayout



컨테이너와 디폴트 배치관리자

- 컨테이너의 디폴트 배치관리자

- 컨테이너 생성시 자동으로 생성되는 배치관리자
JFrame

AWT와 스윙 컨테이너	<u>디폴트</u> 배치관리자
Window, JWindow	BorderLayout
Frame, JFrame	BorderLayout
Dialog, JDialog	BorderLayout
Panel, JPanel	FlowLayout
Applet, JApplet	FlowLayout

컨테이너에 새로운 배치관리자 설정

■ 컨테이너에 새로운 배치관리자 설정

- 컨테이너의 setLayout() 메소드 호출

- 사례

- JPanel 컨테이너에 BorderLayout 배치관리자를 설정하는 예

```
JPanel p = new JPanel();  
p.setLayout(new BorderLayout()); // JPanel에 BorderLayout 설정
```

- 콘텐츠팬의 배치관리자를 FlowLayout 배치관리자로 설정

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

- 오류

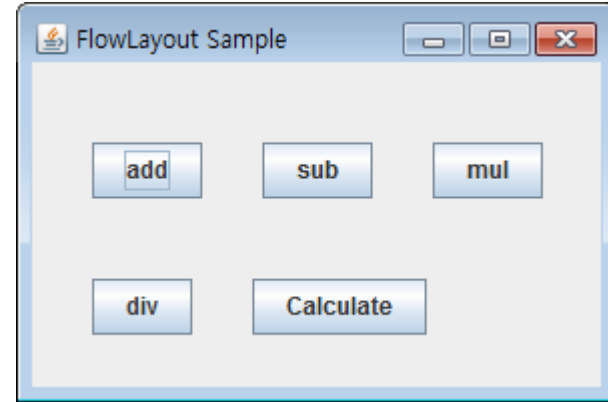
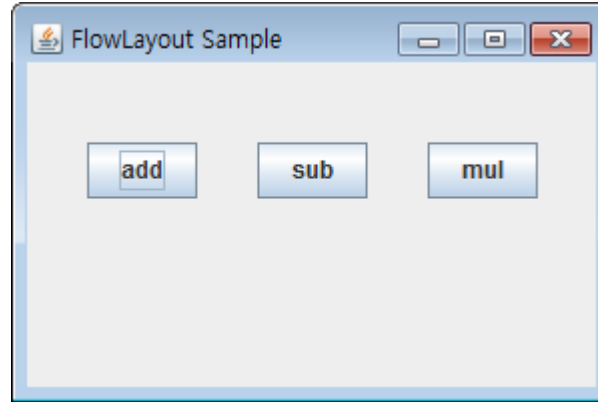
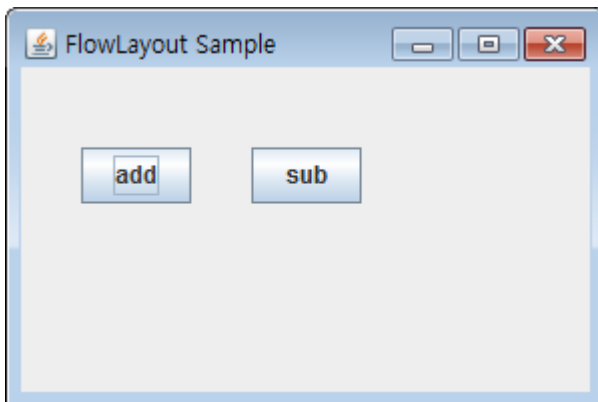
```
 c.setLayout(FlowLayout); // 오류
```

FlowLayout 배치관리자

■ FlowLayout의 배치 방법

- 컴포넌트를 컨테이너 내에 왼쪽에서 오른쪽으로 배치
 - 다시 위에서 아래로 순서대로 배치

```
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"));
```

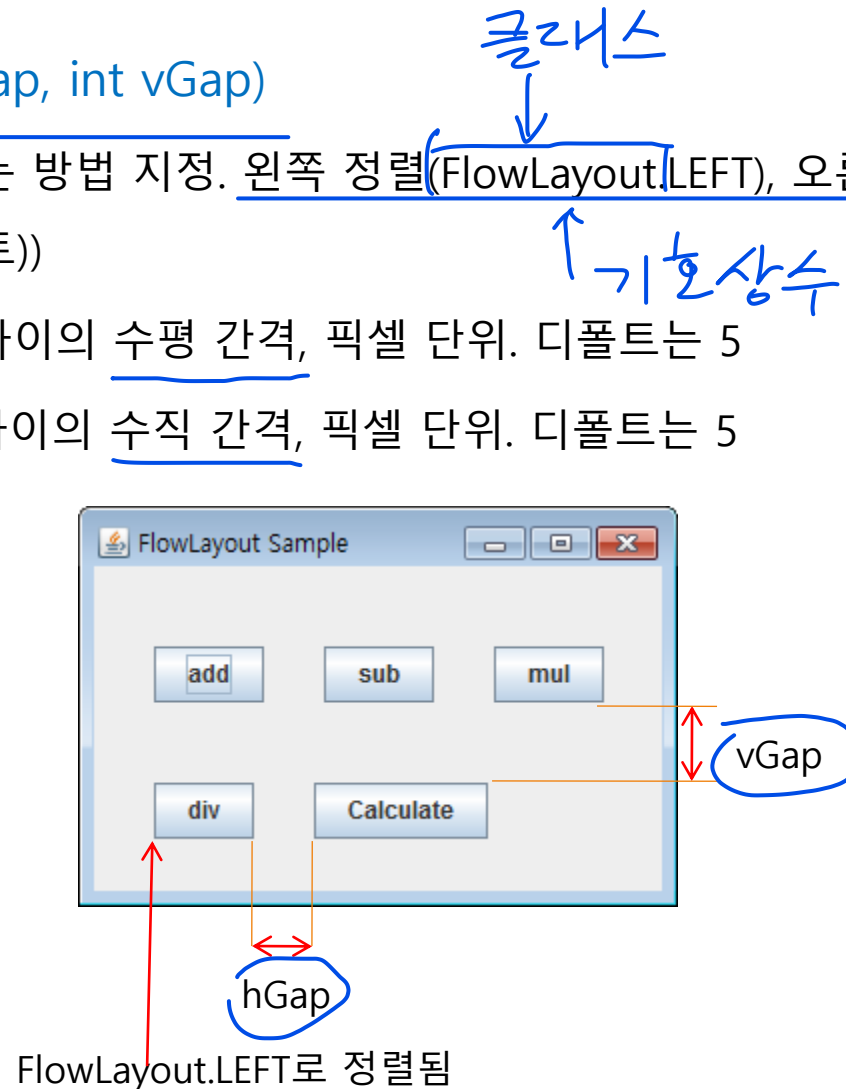


FlowLayout의 생성자

- FlowLayout()

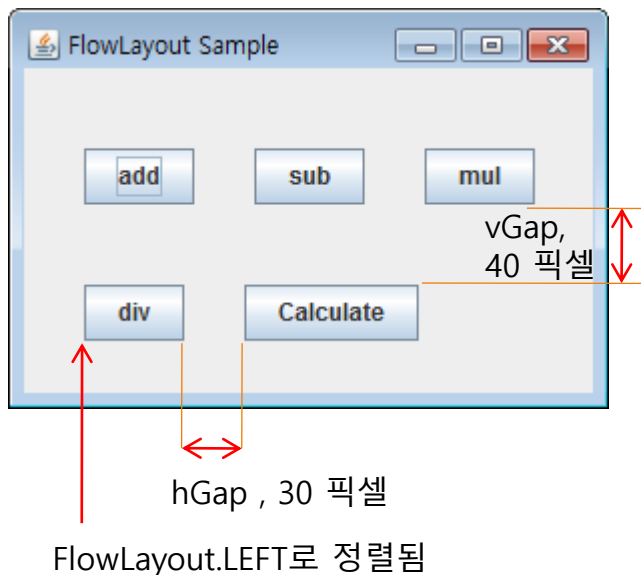
- FlowLayout(int align, int hGap, int vGap)

- align : 컴포넌트를 정렬하는 방법 지정. 왼쪽 정렬(FlowLayout.LEFT), 오른쪽 정렬(FlowLayout.RIGHT), 중앙 정렬(FlowLayout.CENTER(디폴트))
- hGap : 좌우 두 컴포넌트 사이의 수평 간격, 픽셀 단위. 디폴트는 5
- vGap : 상하 두 컴포넌트 사이의 수직 간격, 픽셀 단위. 디폴트는 5



FlowLayout 배치관리자 활용

👉 FlowLayout 배치관리자를 사용하여 다음 그림과 같이 5개의 버튼을 배치하라.



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

public class FlowLayoutEx extends JFrame {
    FlowLayoutEx() {
        setTitle("FlowLayout 예제");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        Container contentPane = getContentPane(); // 콘텐츠팬 알아내기

        // 왼쪽 정렬로, 수평 간격을 30, 수직 간격을 40 픽셀로 배치하는
        // FlowLayout 생성
        contentPane.setLayout(new FlowLayout(FlowLayout.LEFT, 30, 40));
        contentPane.add(new JButton("add"));
        contentPane.add(new JButton("sub"));
        contentPane.add(new JButton("mul"));
        contentPane.add(new JButton("div"));
        contentPane.add(new JButton("Calculate"));

        setSize(300, 200); // 프레임 크기 300x200 설정
        setVisible(true); // 화면에 프레임 출력
    }

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

- JDK 1.5 이상에서는 다음과 같이 작성 가능

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

class FlowLayoutEx{
    public static void main(String[] args) {
        JFrame mf = new JFrame();
        mf.setTitle("FlowLayout 예제");
        mf.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        // 왼쪽 정렬로, 수평 간격을 30, 수직 간격을 40 픽셀로 배치하는
        // FlowLayout 생성
        mf.setLayout(new FlowLayout(FlowLayout.LEFT, 30, 40));
        mf.add(new JButton("add"));
        mf.add(new JButton("sub"));
        mf.add(new JButton("mul"));
        mf.add(new JButton("div"));
        mf.add(new JButton("Calculate"));

        mf.setSize(300, 200); // 프레임 크기 300x200 설정
        mf.setVisible(true); // 화면에 프레임 출력
    }
}
```

■ 콘텐츠팬 활용

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

class FlowLayoutEx{
    public static void main(String[] args) {
        JFrame mf = new JFrame();
        mf.setTitle("FlowLayout 예제");
        mf.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        Container contentPane = mf.getContentPane(); // 콘텐츠팬 알아내기

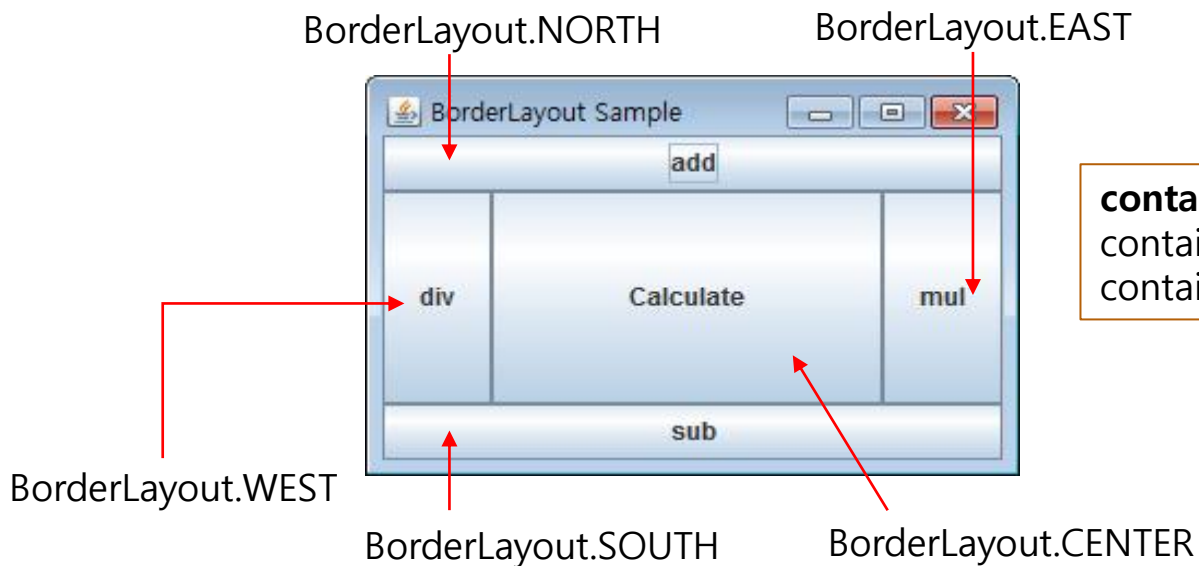
        // 왼쪽 정렬로, 수평 간격을 30, 수직 간격을 40 픽셀로 배치하는
        // FlowLayout 생성
        contentPane.setLayout(new FlowLayout(FlowLayout.LEFT, 30, 40));
        contentPane.add(new JButton("add"));
        contentPane.add(new JButton("sub"));
        contentPane.add(new JButton("mul"));
        contentPane.add(new JButton("div"));
        contentPane.add(new JButton("Calculate"));

        mf.setSize(300, 200); // 프레임 크기 300x200 설정
        mf.setVisible(true); // 화면에 프레임 출력
    }
}
```

BorderLayout 배치관리자

■ BorderLayout의 배치 방법

- 컨테이너 공간을 5 구역으로 분할, 배치
 - 동, 서, 남, 북, 중앙
- 배치 방법
 - add(Component comp, int index)
 - comp를 index의 공간에 배치



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

BorderLayout 생성자와 add() 메소드

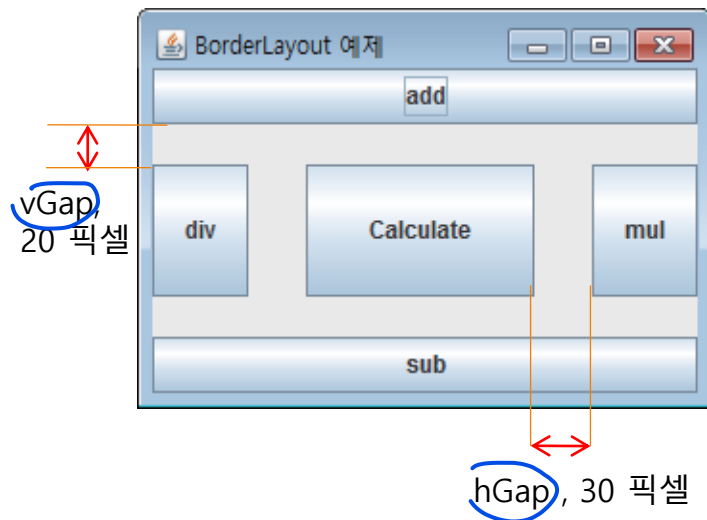
- BorderLayout()
- BorderLayout(int hGap, int vGap)
 - hGap : 좌우 두 컴포넌트 사이의 수평 간격, 픽셀 단위(디폴트 : 0)
 - vGap : 상하 두 컴포넌트 사이의 수직 간격, 픽셀 단위(디폴트 : 0)

▪ add() 메소드

- void add(Component comp, int index)
 - comp 컴포넌트를 index 위치에 삽입한다.
 - index : 컴포넌트의 위치
 - 동 : BorderLayout.EAST 서 : BorderLayout.WEST
 - 남 : BorderLayout.SOUTH 북 : BorderLayout.NORTH
 - 중앙 : BorderLayout.CENTER

BorderLayout 배치관리자 활용

👉 BorderLayout 배치관리자를 사용하여 다음 그림과 같이 5개의 버튼을 배치하라.



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

```
public class BorderLayoutEx extends JFrame {
    BorderLayoutEx() {
        setTitle("BorderLayout 예제");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        Container contentPane = getContentPane(); // 콘텐츠팬 알아내기

        // 콘텐츠팬에 BorderLayout 배치관리자 설정
        contentPane.setLayout(new BorderLayout(30, 20));
        contentPane.add(new JButton("Calculate"), BorderLayout.CENTER);
        contentPane.add(new JButton("add"), BorderLayout.NORTH);
        contentPane.add(new JButton("sub"), BorderLayout.SOUTH);
        contentPane.add(new JButton("mul"), BorderLayout.EAST);
        contentPane.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();
    }
}
```

- JDK 1.5 이상에서는 다음과 같이 작성 가능

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

class BorderLayoutEx {
    public static void main(String[] args) {
        JFrame mf = new JFrame();
        mf.setTitle("BorderLayout 예제");
        mf.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        // 콘텐츠팬에 BorderLayout 배치관리자 설정
        mf.setLayout(new BorderLayout(30, 20));
        mf.add(new JButton("Calculate"), BorderLayout.CENTER);
        mf.add(new JButton("add"), BorderLayout.NORTH);
        mf.add(new JButton("sub"), BorderLayout.SOUTH);
        mf.add(new JButton("mul"), BorderLayout.EAST);
        mf.add(new JButton("div"), BorderLayout.WEST);

        mf.setSize(300, 200); // 프레임 크기 300x200 설정
        mf.setVisible(true); // 프레임을 화면에 출력
    }
}
```

■ 콘텐츠팬 활용

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

class BorderLayoutEx {
    public static void main(String[] args) {
        JFrame mf = new JFrame();
        mf.setTitle("BorderLayout 예제");
        mf.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        Container contentPane = mf.getContentPane(); // 콘텐츠팬 알아내기

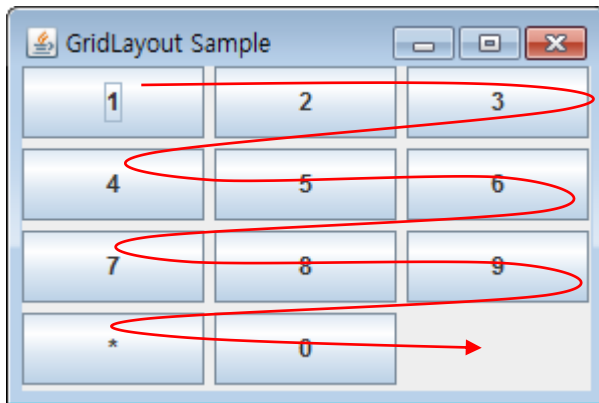
        // 콘텐츠팬에 BorderLayout 배치관리자 설정
        contentPane.setLayout(new BorderLayout(30, 20));
        contentPane.add(new JButton("Calculate"), BorderLayout.CENTER);
        contentPane.add(new JButton("add"), BorderLayout.NORTH);
        contentPane.add(new JButton("sub"), BorderLayout.SOUTH);
        contentPane.add(new JButton("mul"), BorderLayout.EAST);
        contentPane.add(new JButton("div"), BorderLayout.WEST);

        mf.setSize(300, 200); // 프레임 크기 300x200 설정
        mf.setVisible(true); // 프레임을 화면에 출력
    }
}
```

GridLayout 배치관리자

■ GridLayout의 배치 방법

- 컨테이너 공간을 동일한 사각형 격자(그리드)로 분할하고 각 셀에 컴포넌트 하나씩 배치
 - 생성자에 행수와 열수 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
```

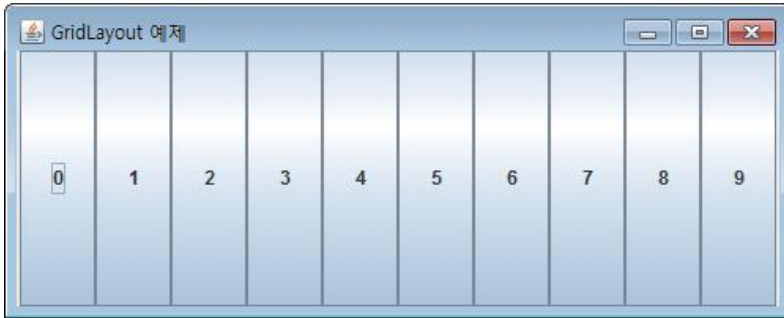
- 4x3 그리드 레이아웃 설정
- 총 11 개의 버튼이 순서대로 add 됨
- 수직 간격 vGap : 5 픽셀
- 수평 간격 hGap : 5 픽셀

GridLayout 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

GridLayout 배치관리자를 사용하는 예

👉 GridLayout을 활용하여 다음 그림과 같이 한 줄에 10개의 버튼을 동일한 크기로 배치하는 스윙 프로그램을 작성하라.



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

public class GridLayoutEx extends JFrame {
    public GridLayoutEx() {
        super("GridLayout 예제");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        Container contentPane = getContentPane();

        // 1x10의 GridLayout 배치관리자
        contentPane.setLayout(new GridLayout(1, 10));

        for(int i=0; i<10; i++) { // 10개의 버튼 부착
            String text = Integer.toString(i); // i를 문자열로 변환
            JButton button = new JButton(text);
            contentPane.add(button); // 컨테이너에 버튼 부착
        }
        setSize(500, 200);
        setVisible(true);
    }
    public static void main(String[] args) {
        new GridLayoutEx();
    }
}
```

- JDK 1.5 이상에서는 다음과 같이 작성 가능

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

class GridLayoutEx {
    public static void main(String[] args) {
        JFrame mf = new JFrame();
        mf.setTitle("GridLayout 예제");
        mf.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        // 1x10의 GridLayout 배치관리자
        mf.setLayout(new GridLayout(1, 10)); 10행 10열

        for(int i=0; i<10; i++) { // 10개의 버튼 부착
            String text = Integer.toString(i); // i를 문자열로 변환
            JButton button = new JButton(text);
            mf.add(button); // 컨테트팬에 버튼 부착
        }
        mf.setSize(500, 200);
        mf.setVisible(true);
    }
}
```

- 컨텐츠팬 활용

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

class GridLayoutEx {
    public static void main(String[] args) {
        JFrame mf = new JFrame();
        mf.setTitle("GridLayout 예제");
        mf.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        Container contentPane = mf.getContentPane();

        // 1x10의 GridLayout 배치관리자
        contentPane.setLayout(new GridLayout(1, 10));

        for(int i=0; i<10; i++) { // 10개의 버튼 부착
            String text = Integer.toString(i); // i를 문자열로 변환
            JButton button = new JButton(text);
            contentPane.add(button); // 컨텐츠팬에 버튼 부착
        }
        mf.setSize(500, 200);
        mf.setVisible(true);
    }
}
```


배치관리자 없는 컨테이너

■ 배치관리자가 없는 컨테이너가 필요한 경우

- 응용프로그램에서 직접 컴포넌트의 크기와 위치를 결정하고자 하는 경우
 1. 컴포넌트의 크기나 위치를 개발자 임의로 결정하고자 하는 경우
 2. 게임 프로그램과 같이 시간이나 마우스/키보드의 입력에 따라 컴포넌트들의 위치와 크기가 수시로 변하는 경우
 3. 여러 컴포넌트들이 서로 겹쳐 출력하고자 하는 경우

■ 컨테이너의 배치 관리자 제거 방법

- `container.setLayout(null);`

```
JPanel p = new JPanel();  
p.setLayout(null); // JPanel의 배치관리자 삭제
```

- 컨테이너의 배치관리자가 없어지면, 컴포넌트에 대한 어떤 배치도 없음
 - 추가된 컴포넌트의 크기가 0으로 설정, 위치는 예측할 수 없게 됨

// 패널 p에는 배치관리자가 없으면 아래 두 버튼은 배치되지 않는다.

```
p.add(new JButton("click")); // 폭과 높이가 0인 상태로 화면에 보이지 않는다.  
p.add(new JButton("me!")); // 폭과 높이가 0인 상태로 화면에 보이지 않는다.
```

컴포넌트의 절대 위치와 크기 설정

■ 배치관리자가 없는 컨테이너에 컴포넌트를 삽입할 때

- 프로그램에서 컴포넌트의 절대 크기와 위치 설정
- 컴포넌트들이 서로 겹치게 할 수 있음

■ 컴포넌트의 크기와 위치 설정 메소드

- void setSize(int width, int height) // 컴포넌트 크기 설정
- void setLocation(int x, int y) // 컴포넌트 위치 설정
- void setBounds(int x, int y, int width, int height) // 위치와 크기 동시 설정

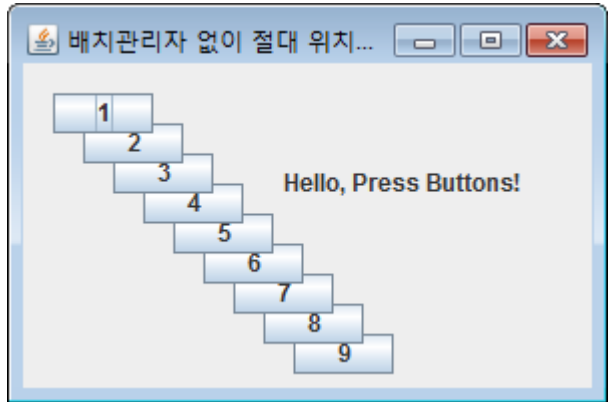
- 예) 버튼을 100×40 크기로 하고, JPanel의 (50, 50) 위치에 배치

```
JPanel p = new JPanel();
p.setLayout(null); // 패널 p의 배치관리자 제거

JButton clickButton = new JButton("Click");
clickButton.setSize(100, 40); // 버튼 크기를 100×40으로 지정
clickButton.setLocation(50, 50); // 버튼 위치를 (50, 50)으로 지정
p.add(clickButton); // 패널 내 (50, 50)에 100×40 크기의 버튼 출력
```

배치관리자 없는 컨테이너에 컴포넌트를 절대 위치와 절대 크기로 지정

☞ 다음 그림과 같이 콘텐츠팬에
배치관리자를 삭제하고
9개의 버튼과 하나의 문자열을
출력하는 프로그램을 작성하라.



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

public class NullContainerEx extends JFrame {
    NullContainerEx() {
        setTitle("배치관리자 없이 절대 위치에 배치하는 예제");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        Container contentPane = getContentPane();
        contentPane.setLayout(null); // 콘텐츠팬의 배치관리자 제거

        JLabel la = new JLabel("Hello, Press Buttons!");
        la.setLocation(130, 50); // la를 (130,50) 위치로 지정
        la.setSize(200, 20); // la를 200x20 크기로 지정
        contentPane.add(la); // la를 콘텐츠팬에 부착

        // 9개의 버튼 컴포넌트를 생성하고 동일한 크기로 설정한다.
        // 위치는 서로 겹치게 설정한다.
        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); // 버튼의 크기는 동일하게 50x20
            contentPane.add(b); // 버튼을 콘텐츠팬에 부착
        }
        setSize(300, 200);
        setVisible(true);
    }

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

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

▪ JDK 1.5 이상에서는 다음과 같이 작성 가능

```
class NullContainerEx {
    public static void main(String[] args) {
        JFrame mf = new JFrame();
        mf.setTitle("배치관리자 없이 절대 위치에 배치하는 예제");
        mf.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        mf.setLayout(null); // 콘텐츠팬의 배치관리자 제거

        JLabel la = new JLabel("Hello, Press Buttons!");
        la.setLocation(130, 50); // la를 (130,50) 위치로 지정
        la.setSize(200, 20); // la를 200x20 크기로 지정
        mf.add(la); // la를 콘텐츠팬에 부착

        // 9개의 버튼 컴포넌트를 생성하고 동일한 크기로 설정한다.
        // 위치는 서로 겹치게 설정한다.
        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); // 버튼의 크기는 동일하게 50x20
            mf.add(b); // 버튼을 콘텐츠팬에 부착
        }
        mf.setSize(300, 200);
        mf.setVisible(true);
    }
}
```

■ 콘텐츠팬 활용

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

class NullContainerEx {
    public static void main(String[] args) {
        JFrame mf = new JFrame();
        mf.setTitle("배치관리자 없이 절대 위치에 배치하는 예제");
        mf.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        Container contentPane = mf.getContentPane();
        contentPane.setLayout(null); // 콘텐츠팬의 배치관리자 제거

        JLabel la = new JLabel("Hello, Press Buttons!");
        la.setLocation(130, 50); // la를 (130,50) 위치로 지정
        la.setSize(200, 20); // la를 200x20 크기로 지정
        contentPane.add(la); // la를 콘텐츠팬에 부착

        // 9개의 버튼 컴포넌트를 생성하고 동일한 크기로 설정한다.
        // 위치는 서로 겹치게 설정한다.
        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); // 버튼의 크기는 동일하게 50x20
            contentPane.add(b); // 버튼을 콘텐츠팬에 부착
        }
        mf.setSize(300, 200);
        mf.setVisible(true);
    }
}
```