



*Power Java*

## 제19장 배치 관리자



© 2009 인피니티북스 All rights reserved



## 이번 장에서 학습할 내용

- 
- 배치 관리자의 개요
  - 배치 관리자의 사용
  - FlowLayout
  - BorderLayout
  - GridLayout
  - BoxLayout
  - CardLayout
  - 절대 위치로 배치

컨테이너  
안에서  
컴포넌트를  
배치하는  
방법에 대하여  
살펴봅시다.



© 2009 인피니티북스 All rights reserved



## 배치 관리자(layout manager)

- 컨테이너 안의 각 컴포넌트의 위치와 크기를 결정하는 작업

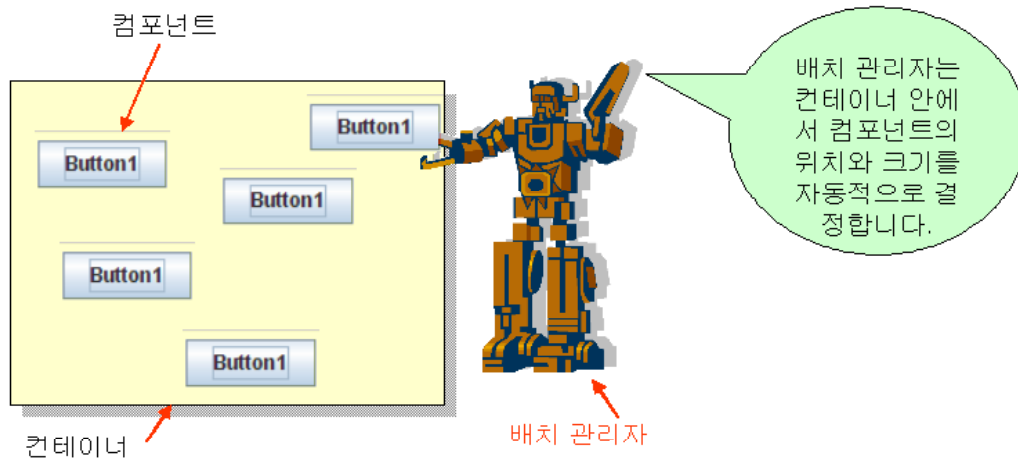


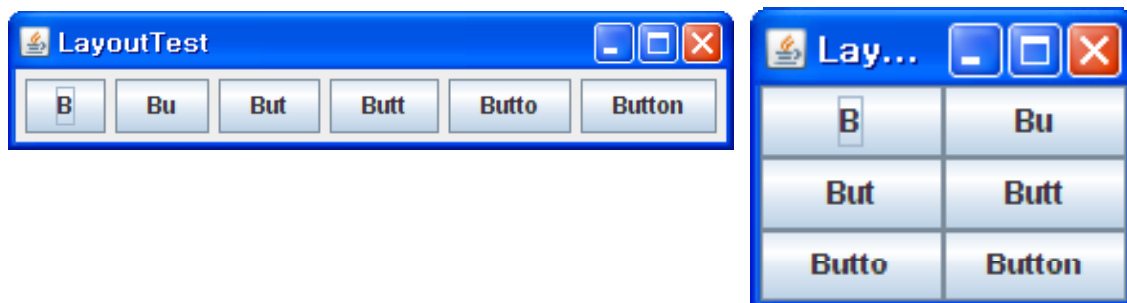
그림 19-1 배치 관리자의 개념

© 2009 인피니티박스 All rights reserved



## 상당히 다르게 보인다.

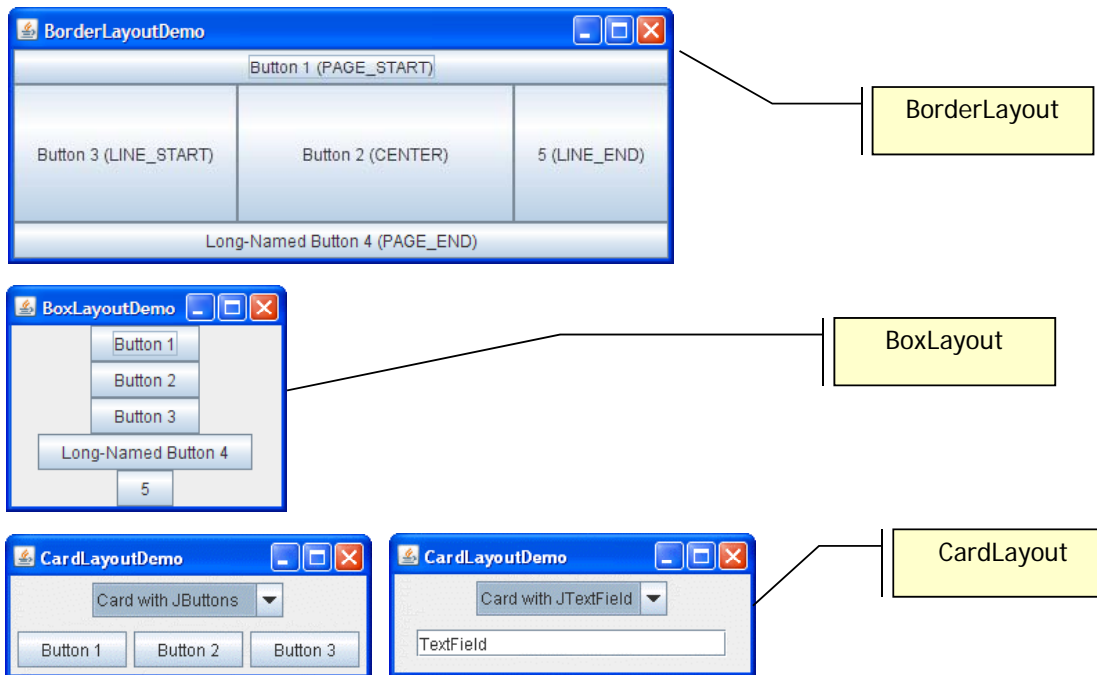
- 배치 관리자에 따라서 상당히 다르게 보인다.



© 2009 인피니티박스 All rights reserved



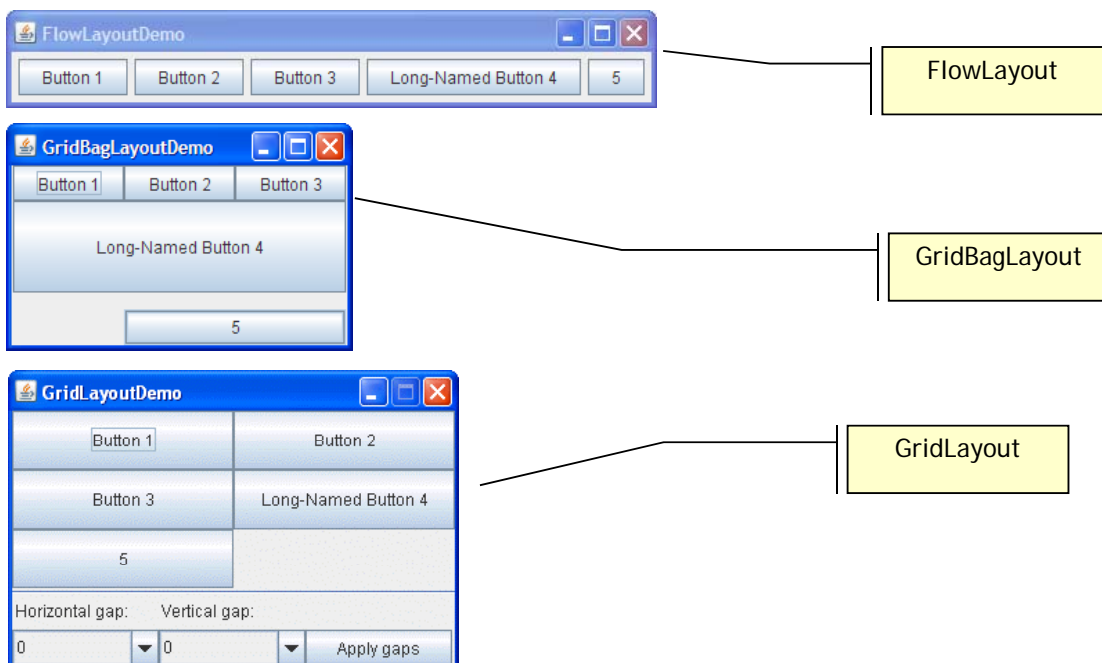
## 배치 관리자의 종류



© 2009 인피니티박스 All rights reserved



## 배치 관리자의 종류



© 2009 인피니티박스 All rights reserved



## 배치 관리자의 설정

1. 생성자를 이용하는 방법

```
JPanel panel = new JPanel(new BorderLayout());
```

2. `setLayout()` 메소드 이용

```
panel.setLayout(new FlowLayout());
```



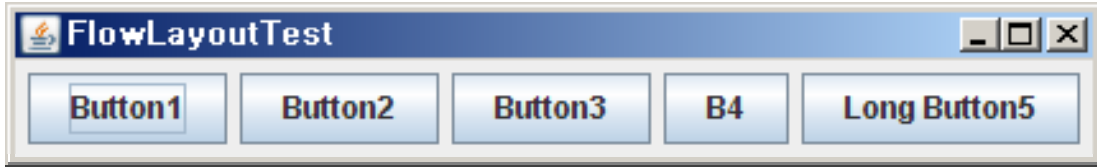
## 크기와 정렬 힌트

- 프로그래머가 컴포넌트의 크기와 힌트를 배치 관리자에게 주고 싶은 경우에는 `setMinimumSize()`, `setPreferredSize()`, `setMaximumSize()` 메소드를 사용
  - (예) `component.setMaximumSize(new Dimension(Integer.MAX_VALUE, Integer.MAX_VALUE));`



# FlowLayout

- 컴포넌트들을 왼쪽에서 오른쪽으로 버튼을 배치한다.
- 패널과 애플릿의 디폴트 배치 관리자이다.



© 2009 인피니티박스 All rights reserved

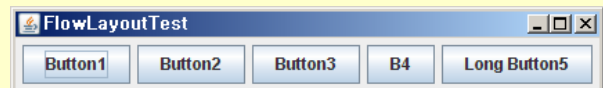


## FlowLayout 예제



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

```
class MyFrame extends JFrame {  
    public MyFrame() {  
  
        setTitle("FlowLayoutTest");  
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
  
        JPanel panel;  
        // 패널을 생성하고 배치 관리자를 FlowLayout으로 설정  
        panel = new JPanel();  
        panel.setLayout(new FlowLayout(FlowLayout.CENTER));  
        // 패널에 버튼을 생성하여 추가  
        panel.add(new JButton("Button1"));  
        panel.add(new JButton("Button2"));  
        panel.add(new JButton("Button3"));  
        panel.add(new JButton("B4"));  
        panel.add(new JButton("Long Button5"));  
        add(panel);  
  
        pack();  
        setVisible(true);  
    }  
}
```





# BorderLayout

- BorderLayout은 5개의 영역으로 구분하고 각각의 영역에 컴포넌트를 배치



- PAGE\_START (또는 NORTH)
- PAGE\_END (또는 SOUTH)
- LINE\_START (또는 WEST)
- LINE\_END (또는 EAST)
- CENTER

© 2009 인피니티박스 All rights reserved



## BorderLayout 예제



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

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

```
        setTitle("BorderLayoutTest");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

```
        // 프레임은 디폴트로 BorderLayout 이므로 사실은 불필요
        setLayout(new BorderLayout());
```

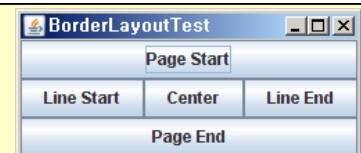
```
        // 버튼을 추가한다.
```

```
        add(new JButton("Center"), BorderLayout.CENTER);
        add(new JButton("Line Start"), BorderLayout.LINE_START);
        add(new JButton("Line End"), BorderLayout.LINE_END);
        add(new JButton("Page Start"), BorderLayout.PAGE_START);
        add(new JButton("Page End"), BorderLayout.PAGE_END);
```

```
        pack();
        setVisible(true);
```

```
    }
```

```
}
```





# GridLayout

- GridLayout은 컴포넌트들을 격자 모양으로 배치한다.



© 2009 인피니티박스 All rights reserved



## GridLayout 예제

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

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

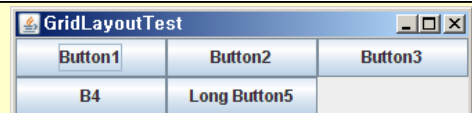
```
        setTitle("GridLayoutTest");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

```
        setLayout(new GridLayout(0, 3));           // 3개의 열과 필요한 만큼의 행
```

```
        add(new JButton("Button1"));
        add(new JButton("Button2"));
        add(new JButton("Button3"));
        add(new JButton("B4"));
        add(new JButton("Long Button5"));
```

```
        pack();
        setVisible(true);
```

```
    }
}
```

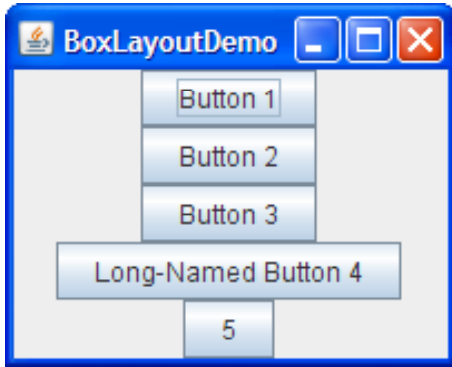


© 2009 인피니티박스 All rights reserved



# BoxLayout

- BoxLayout은 컴포넌트를 다른 컴포넌트 위에 쌓거나 컴포넌트를 하나의 행에 배치할 수 있다.
- BoxLayout은 FlowLayout의 하나의 버전으로 간주할 수 있으나 좀 더 기능이 많다.



© 2009 인피니티박스 All rights reserved



## BoxLayout 예제

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

class MyFrame extends JFrame {

    public MyFrame() {
        setTitle("BoxLayoutTest");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        JPanel panel = new JPanel();

        // Y축 방향으로 컴포넌트를 쌓는다.
        panel.setLayout(new BoxLayout(panel, BoxLayout.Y_AXIS));

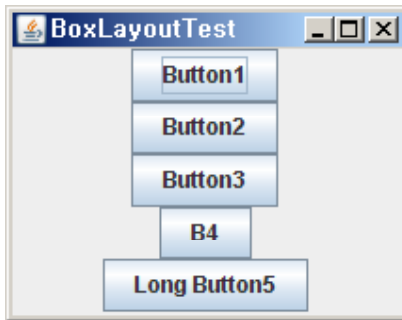
        makeButton(panel, "Button1");
        makeButton(panel, "Button2");
        makeButton(panel, "Button3");
        makeButton(panel, "B4");
        makeButton(panel, "Long Button5");
        add(panel);
        pack();
        setVisible(true);
    }
}
```





## GridLayout 예제

```
private void makeButton(JPanel panel, String text) {  
    JButton button = new JButton(text);  
    button.setAlignmentX(Component.CENTER_ALIGNMENT);  
    panel.add(button);  
}  
}
```

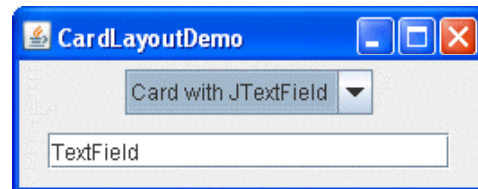


© 2009 인피니티박스 All rights reserved



## CardLayout

- CardLayout 클래스는 여러 개의 카드가 쌓여 있는 형태로 컴포넌트를 배치



© 2009 인피니티박스 All rights reserved



## CardLayout 예제

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

class MyFrame extends JFrame implements ActionListener {
    JPanel panel;
    Cards cards;

    public MyFrame() {
        setTitle("CardLayoutTest");
        setSize(400, 200);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        panel = new JPanel(new GridLayout(0, 5, 10, 0));
        addButton("<<", panel);
        addButton("<", panel);
        addButton(">", panel);
        addButton(">>", panel);
        addButton("종료", panel);
        add(panel, "North");
        cards = new Cards();
        add(cards, "Center");
        setVisible(true);
    }
}
```



## CardLayout 예제

```
void addButton(String str, Container target) {
    JButton button = new JButton(str);
    button.addActionListener(this);
    target.add(button);
}

private class Cards extends JPanel {
    CardLayout layout;

    public Cards() {
        layout = new CardLayout();
        setLayout(layout);
        for (int i = 1; i <= 10; i++) {
            add(new JButton("현재 카드의 번호는 " + i + "입니다"), "Center");
        }
    }
}
```



## CardLayout 예제

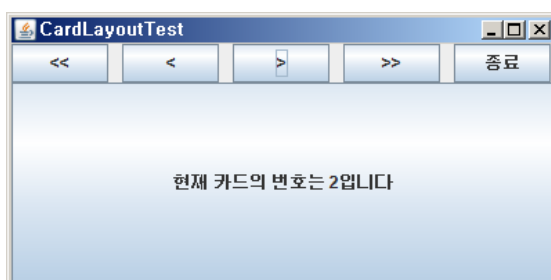
```
public void actionPerformed(ActionEvent e) {
    if (e.getActionCommand().equals("종료")) {
        System.exit(0);
    } else if (e.getActionCommand().equals("<<")) {
        cards.layout.first(cards);
    } else if (e.getActionCommand().equals("<")) {
        cards.layout.previous(cards);
    } else if (e.getActionCommand().equals(">")) {
        cards.layout.next(cards);
    } else if (e.getActionCommand().equals(">>")) {
        cards.layout.last(cards);
    }
}

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

© 2009 인피니티박스 All rights reserved



## GridLayout 예제



© 2009 인피니티박스 All rights reserved



## 어떤 배치 관리자를 선택할 것인가?

- 컴포넌트를 가능한 크게 나타내고 싶은 경우
  - GridLayout이나 BorderLayout을 사용
- 몇개의 컴포넌트를 자연스러운 크기로 한줄로 나타내고 싶은 경우
  - FlowLayout을 사용하던지 BoxLayout을 사용한다.
- 몇개의 컴포넌트를 행과 열로 동일한 크기로 나타내고 싶은 경우
  - GridLayout을 사용하여야 한다.
- 몇개의 컴포넌트를 행과 열로 나타내는데 각 컴포넌트의 크기가 다르거나 간격, 정렬 방식을 다르게 내고 싶은 경우
  - BoxLayout을 사용하면 된다.

© 2009 인피니티박스 All rights reserved



## 절대 위치로 배치하기

1. 배치 관리자를 null로 설정한다.
  - `setLayout(null);`
2. `add()` 메소드를 사용하여 컴포넌트를 컨테이너에 추가한다.
  - `Button b = Button("Absolute Position Button");`
  - `add(b);`
3. `setBounds()` 메소드를 사용하여 절대 위치와 크기를 지정한다.
  - `b.setBounds(x, y, w, h);`
4. 컴포넌트의 `repaint()` 메소드를 호출한다.
  - `b.repaint();`

© 2009 인피니티박스 All rights reserved



## CardLayout 예제

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

class MyFrame extends JFrame {
    JButton b1;
    private JButton b2, b3;

    public MyFrame() {
        setTitle("Absolute Position Test");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setSize(300, 200);
        JPanel p = new JPanel();
        p.setLayout(null);

        b1 = new JButton("Button #1");
        p.add(b1);
        b2 = new JButton("Button #2");
        p.add(b2);
        b3 = new JButton("Button #3");
        p.add(b3);
    }
}
```

© 2009 인피니티박스 All rights reserved

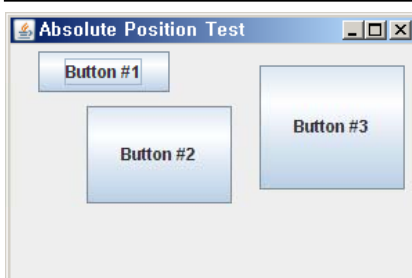


## CardLayout 예제

```

        b1.setBounds(20, 5, 95, 30);
        b2.setBounds(55, 45, 105, 70);
        b3.setBounds(180, 15, 105, 90);
        add(p);
        setVisible(true);
    }
}

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



© 2009 인피니티박스 All rights reserved



## Q & A

