



Power Java

제14장 배치 관리자





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.





배치 관리자의 개요

- 컨테이너 안의 각 컴포넌트의 위치와 크기를 결정하는 작업

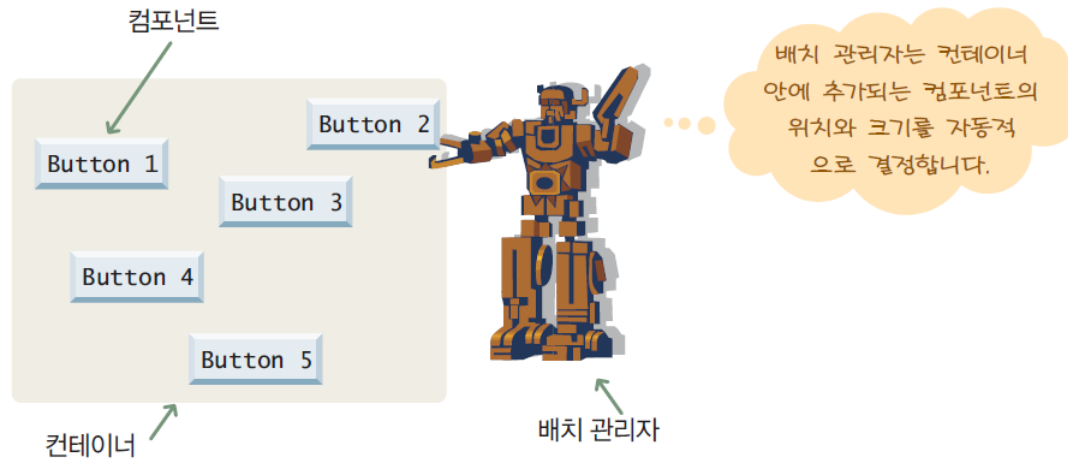


그림14-1. 배치 관리자의 개념

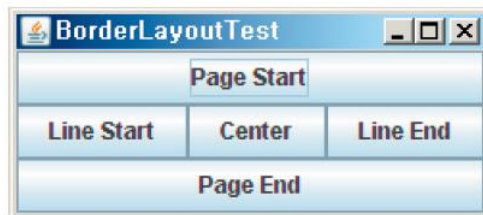


배치 관리자의 개요

- FlowLayout



- BorderLayout



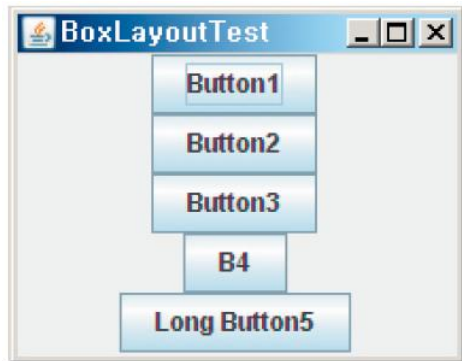
- GridLayout





배치 관리자의 개요

- BorderLayout



- CardLayout





배치 관리자를 사용하는 방법

```
panel.setLayout(new BorderLayout()); // 패널의 배치 관리자를 설정한다.
```

- 크기와 정렬 힌트 제공하기

```
button.setMaximumSize(new Dimension(300, 200)); // 최대 크기 힌트  
button.setAlignmentX(JComponent.CENTER_ALIGNMENT); // 중앙 정렬 힌트
```

- 배치 방향 설정

```
panel.applyComponentOrientation(ComponentOrientation.RIGHT_TO_LEFT);
```





FlowLayout 클래스

생성자	설명
<code>FlowLayout()</code>	새로운 <code>FlowLayout</code> 객체를 생성한다. 기본 설정은 중앙(center) 배치이며 간격은 세로, 가로 각각 5 픽셀이다.
<code>FlowLayout(int align)</code>	지정된 정렬 방식을 가진 새로운 <code>FlowLayout</code> 객체를 생성한다. 기본 설정은 중앙(center) 배치이며 간격은 세로, 가로 각각 5 픽셀이다. 정렬 매개변수는 다음 중 하나이다. <code>FlowLayout.LEADING</code> , <code>FlowLayout.CENTER</code> , <code>FlowLayout.TRAILING</code> .
<code>FlowLayout (int align, int hgap, int vgap)</code>	지정된 정렬 방식과 수평 간격 <code>hgap</code> 과 수직 간격 <code>vgap</code> 을 가진 새로운 <code>FlowLayout</code> 객체를 생성한다.



FlowLayout 예제

FlowTest.java

```
01 import java.awt.*;
02 import javax.swing.*;
03
04 class MyFrame extends JFrame {
05     public MyFrame() {
06
07         setTitle("FlowLayoutTest");
08         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
09
10         JPanel panel;
11         // 패널을 생성하고 배치 관리자를 FlowLayout으로 설정
12         panel = new JPanel();
13         panel.setLayout(new FlowLayout(FlowLayout.CENTER));
14         // 패널에 버튼을 생성하여 추가
15         panel.add(new JButton("Button1"));
16         panel.add(new JButton("Button2"));
17         panel.add(new JButton("Button3"));
18         panel.add(new JButton("B4"));
19         panel.add(new JButton("Long Button5"));
20         add(panel);
```

패널의 배치 관리자를
FlowLayout으로 설정

패널에 버튼 추가

패널을 프레임에 추가



FlowLayout 예제

```
21
22     pack();
23     setVisible(true);
24 }
25 }
26
27 public class FlowTest {
28     public static void main(String argv[]) {
29         MyFrame f = new MyFrame();
30     }
31 }
```

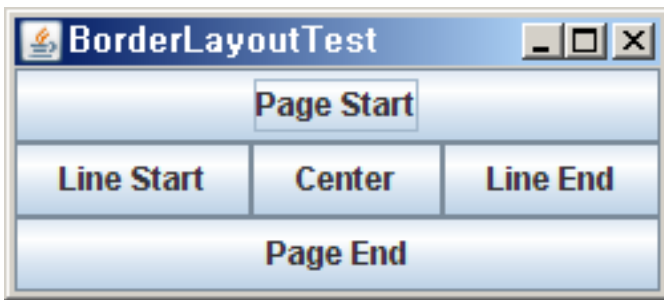
실행결과





BorderLayout 클래스

- 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

```
panel.add(aComponent, BorderLayout.PAGE_START); // 북쪽에 배치한다.
```



생성자

생성자 또는 메소드	설명
<code>BorderLayout(int hgap, int vgap)</code>	컴포넌트 사이의 수평 간격 hgap과 수직 간격 vgap을 가지는 BorderLayout 객체 생성
<code>setHgap(int)</code>	컴포넌트 사이의 수평 간격 설정(단위는 픽셀)
<code>setVgap(int)</code>	컴포넌트 사이의 수직 간격 설정



예제

BorderTest.java

```
01 import java.awt.*;
02 import javax.swing.*;
03
04 class MyFrame extends JFrame {
05     public MyFrame() {
06
07         setTitle("BorderLayoutTest");
08         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
09
10         // 프레임은 디폴트로 BorderLayout 이므로 사실은 불필요
11         setLayout(new BorderLayout());
12
13         // 버튼을 추가한다.
14         add(new JButton("Center"), BorderLayout.CENTER);
15         add(new JButton("Line Start"), BorderLayout.LINE_START);
16         add(new JButton("Line End"), BorderLayout.LINE_END);
17         add(new JButton("Page Start"), BorderLayout.PAGE_START);
18         add(new JButton("Page End"), BorderLayout.PAGE_END);
19     }
20 }
```

프레임에 버튼을 추가한다.
추가하는 위치를 지정한다.



예제

```
19
20     pack();
21     setVisible(true);
22 }
23 }
24
25 public class BorderTest {
26     public static void main(String argv[]) {
27         MyFrame f = new MyFrame();
28     }
29 }
```

실행결과





GridLayout 클래스

- 생성자

생성자	설명
<code>GridLayout(int rows, int cols)</code>	rows 행과 cols 열을 가지는 GridLayout 객체를 생성한다. 만약 rows나 cols가 0이면 필요한 만큼의 행이나 열이 만들어진다.
<code>GridLayout(int rows, int cols, int hgap, int vgap)</code>	rows 행과 cols 열을 가지는 GridLayout 객체를 생성한다. hgap과 vgap은 컴포넌트 사이의 수평 간격과 수직 간격으로 단위는 픽셀이다.

```
setLayout(new GridLayout (3,0));
```



예제

GridTest.java

```
01 import java.awt.*;
02 import javax.swing.*;
03
04 class MyFrame extends JFrame {
05     public MyFrame() {
06
07         setTitle("GridLayoutTest");
08         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
09
10         setLayout(new GridLayout(0, 3)); // 3개의 열과 필요한 만큼의 행
11
12         add(new JButton("Button1"));
13         add(new JButton("Button2"));
14         add(new JButton("Button3"));
15         add(new JButton("B4"));
16         add(new JButton("Long Button5"));
17
18         pack();
19         setVisible(true);
20     }
21 }
22
23 public class GridTest {
24     public static void main(String argv[]) {
25         MyFrame f = new MyFrame();
```

3개의 열을 가진 GridLayout, 행의 개수는
추가되는 컴포넌트에 의하여 결정된다.



GridLayout 예제

```
26     }  
27 }
```

실행결과





BoxLayout 클래스 예제

BoxTest.java

```
01 import java.awt.*;
02 import java.awt.event.*;
03 import javax.swing.*;
04
05 class MyFrame extends JFrame {
06
07     public MyFrame() {
08         setTitle("BoxLayoutTest");
09         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
10         JPanel panel = new JPanel();
11
12         // Y축 방향으로 컴포넌트를 쌓는다.
13         panel.setLayout(new BorderLayout(panel, BorderLayout.Y_AXIS));
14     }
15 }
```



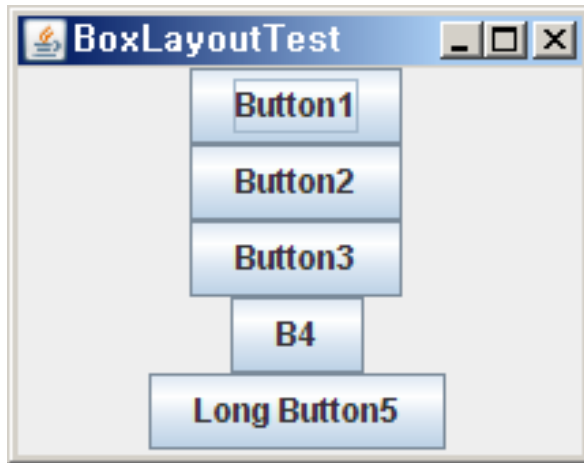
BoxLayout 예제

```
15     makeButton(panel, "Button1");
16     makeButton(panel, "Button2");
17     makeButton(panel, "Button3");
18     makeButton(panel, "B4");
19     makeButton(panel, "Long Button5");
20     add(panel);
21     pack();
22     setVisible(true);
23 }
24
25 private void makeButton(JPanel panel, String text) {
26     JButton button = new JButton(text);
27     button.setAlignmentX(Component.CENTER_ALIGNMENT);
28     panel.add(button);
29 }
30 }
31
32 public class BoxTest {
33     public static void main(String args[]) {
34         MyFrame f = new MyFrame();
35     }
36 }
```

BoxLayout의 중앙에
버튼들을 추가한다.



실행결과





CardLayout 클래스

- 생성자

생성자	설명
<code>CardLayout()</code>	디폴트 생성자
<code>CardLayout(int hgap, int vgap)</code>	hgap과 vgap은 컴포넌트 사이의 수평 간격과 수직 간격을 픽셀로 표시한다. 기본값은 0이다.

- 메소드

메소드	설명
<code>first (Container parent)</code>	첫 번째 카드를 선택한다.
<code>next (Container parent)</code>	다음 카드를 선택한다. 만약 현재 선택된 카드가 마지막이면 처음으로 간다.
<code>previous (Container parent)</code>	이전 카드를 선택한다.
<code>last (Container parent)</code>	마지막 카드를 선택한다.
<code>show (Container parent, String s)</code>	문자열 s로 지정한 카드를 선택 선택한다.



CardLayout 예제

CardTest.java

```
01 import java.awt.*;
02 import java.awt.event.*;
03 import javax.swing.*;
04
05 class MyFrame extends JFrame implements ActionListener {
06     JPanel panel;
07     Cards cards;
08
09     public MyFrame() {
10         setTitle("CardLayoutTest");
11         setSize(400, 200);
12         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
13         panel = new JPanel(new GridLayout(0, 5, 10, 0));
14         addButton("<<", panel);
15         addButton("<", panel);
16         addButton(">", panel);
17         addButton(">>", panel);
18         addButton("종료", panel);
19         add(panel, "North");
20         cards = new Cards();
21         add(cards, "Center");
22         setVisible(true);
23     }
```

GridLayout의 패널을 만들고
여기에 제어 버튼을 넣는다.

Cards 객체를 생성하여 중앙에 배치한다.
Cards는 패널의 일종이다(뒤에 정의된다).



CardLayout 예제

```
24
25 void addButton(String str, Container target) {
26     JButton button = new JButton(str);
27     button.addActionListener(this);
28     target.add(button);
29 }
30
31 private class Cards extends JPanel {
32     CardLayout layout;
33
34     public Cards() {
35         layout = new CardLayout();
36         setLayout(layout);
37         for (int i = 1; i <= 10; i++) {
38             add(new JButton("현재 카드의 번호는 " + i + "입니다"), "Center");
39         }
40     }
41 }
```

← 카드 레이아웃 설정



CardLayout 예제

```
42
43 public void actionPerformed(ActionEvent e) {
44     if (e.getActionCommand().equals("종료")) {
45         System.exit(0);
46     } else if (e.getActionCommand().equals("<<")) {
47         cards.layout.first(cards);
48     } else if (e.getActionCommand().equals("<")) {
49         cards.layout.previous(cards);
50     } else if (e.getActionCommand().equals(">")) {
51         cards.layout.next(cards);
52     } else if (e.getActionCommand().equals(">>")) {
53         cards.layout.last(cards);
54     }
55 }
56 }
57
58 public class CardTest {
59     public static void main(String args[]) {
60         MyFrame f = new MyFrame();
61     }
62 }
```

이벤트를 처리하는 부분이다.
차후에 학습한다.

CardLayout 중에서
첫 번째 카드를 선택한다.



실행결과





절대 위치로 배치하기

- 배치 관리자를 null로 설정한다.

```
setLayout(null);
```

- add() 메소드를 사용하여 컴포넌트를 컨테이너에 추가한다.

```
Button b = Button("Absolute Position Button");  
add(b);
```

- setBounds() 메소드를 사용하여 절대 위치와 크기를 지정한다.

```
b.setBounds(x, y, w, h);
```

- 컴포넌트의 repaint() 메소드를 호출한다.

```
b.repaint();
```



예제

AbsoluteTest.javat

```
01 import java.awt.*;
02 import java.awt.event.*;
03 import javax.swing.*;
04
05 class MyFrame extends JFrame {
06     JButton b1;
07     private JButton b2, b3;
08
09     public MyFrame() {
10         setTitle("Absolute Position Test");
11         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
12         setSize(300, 200);
13         JPanel p = new JPanel();
14         p.setLayout(null);
15
16         b1 = new JButton("Button #1");
17         p.add(b1);
18         b2 = new JButton("Button #2");
19         p.add(b2);
20         b3 = new JButton("Button #3");
21         p.add(b3);
22
```

← 패널의 배치 관리자를 제거한다.



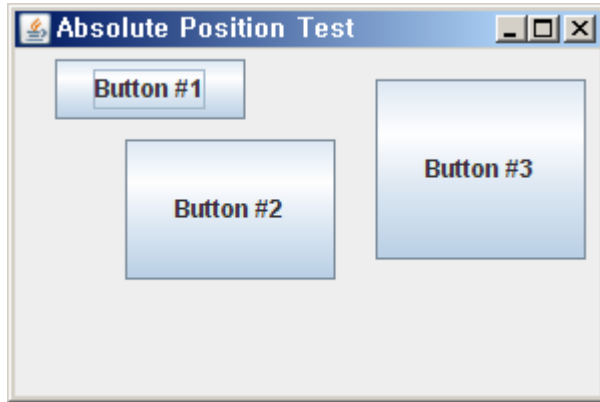
예제

```
23     b1.setBounds(20, 5, 95, 30);
24     b2.setBounds(55, 45, 105, 70);
25     b3.setBounds(180, 15, 105, 90);
26     add(p);
27     setVisible(true);
28 }
29 }
30
31 public class AbsoluteTest {
32     public static void main(String args[]) {
33         MyFrame f = new MyFrame();
34     }
35 }
```

버튼들의 절대 위치와 크기를 지정한다.



실행결과





Q & A

