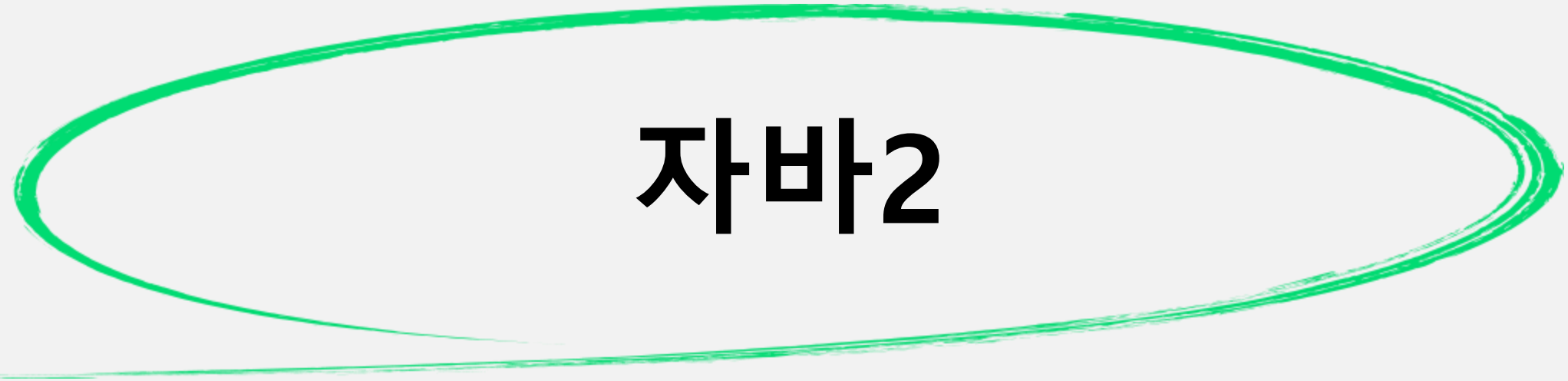


# 자바2

(GUI-1 : 사용자 그래픽 인터페이스)

# Chapter 06
# 사용자 그래픽 인터페이스

## Chapter06의 학습목표

- 자바의 GUI 구동 방법을 알아본다
- 사용자 그래픽 인터페이스를 구현해본다

## GUI

# GUI Program(Graphic user interface)

- 사용자가 그래픽을 통해 컴퓨터와 정보를 교환하는 작업 환경
- GUI는 컴포넌트들로 만들어지고 자바에서는 객체로 생성
- Component : 레이블, 버튼이나 텍스트 필드와 같은 GUI를 작성하는 기본적인 빌딩 블록. Control이라고도 부름

### AWT

- 초기 자바 버전에서 제공한 GUI
- 운영체제가 제공하는 자원을 이용하여 컴포넌트를 생성
- 컴포넌트가 플랫폼에 종속적 : 실행되는 플랫폼에 따라 컴포넌트의 모습이 달라짐

- package java.awt
- Frame, Window, Panel, Button, Label

### Swing

- 컴포넌트가 자바로 작성되어 있음
- 폴램폼에 독립적 : 플랫폼에 상관없이 일관된 화면을 보여줌
- AWT보다 다양하고 많은 컴포넌트 제공

- package javax.swing
- JFrame, JWindow, JPanel, JButton, JLabel

| 패키지명 | 설명 |
|---|---|
| java.awt | GUI 컴포넌트를 위한 부모 클래스 들을 제공하고 추가로 Color나 Point와 같은 유틸리티 타입의 클래스들을 포함 |
| java.awt.event | GUI 컴포넌트로부터 발생되는 이벤트 (ex: 버튼 클릭 이벤트)를 처리하기 위한 클래스와 인터페이스를 포함 |
| javax.swing | 버튼이나 텍스트 필드, 프레임, 패널과 같은 GUI 컴포넌트들을 포함 |

GUI

# 컨테이너(Container)

- 다른 컴포넌트를 포함할 수 있는 컴포넌트
- 모든 컴포넌트는 컨테이너에 포함 되어야 화면에 출력 가능
- 최상위 컨테이너는 다른 컨테이너에 포함될 수 없음

| 종류 | 컴포넌트명 |
|---|---|
| 단순 컴포넌트 | JButton, JLabel, JCheckbox, JChoice, JList, JMenu, JTextField, JScrollbar, JTextArea, JCanvas |
| 컨테이너 컴포넌트 | JFrame, JDialog, JApplet, JPanel, JScrollPanel |
| 최상위 컨테이너 | JFrame, JDialog, JApplet |
| 일반 컨테이너 | JPenel, JScrollPanel |

GUI

# GUI 작성 절차

- JFrame : 윈도우와 메뉴를 가지는 일반적인 데스크탑 어플리케이션
- JDialog : 메뉴가 없는 대화 상자 형식의 간단한 어플리케이션
- JApplet : 애플릿을 작성

**컨테이너 생성** → **컴포넌트 추가**

## 프레임 생성하기 예제-1 (JFrame 객체를 생성)

```
//title을 "Frame Test"로 설정하여 frame을 설정
JFrame frame = new JFrame("Frame Test");
frame.setSize(300, 200); //프레임의 크기를 300, 200로 설정
//사용자가 frame의 오른쪽 상단에 있는 close 버튼을 눌렀을 경우 전체 프로그램을 종료하도록 설정
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
frame.setVisible(true); //프레임을 화면에 나타나게 만듦
```

GUI

# GUI 작성 절차

## 프레임 생성하기 예제-2 (JFrame 클래스를 상속받아 사용)

```
import javax.swing.JFrame;

public class MyFrame extends JFrame{
    public static void main(String[] args) {
        //title을 "Frame Test"로 설정하여 frame을 설정
        JFrame frame = new JFrame("Frame Test");
        frame.setSize(300, 200); //프레임의 크기를 300, 200로 설정
        //사용자가 frame의 오른쪽 상단에 있는 close 버튼을 눌렀을 경우 전체 프로그램을 종료하도록 설정
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setVisible(true); //프레임을 화면에 나타나게 만듦
    }
}
```

GUI

# GUI 작성 절차

## 프레임 생성하기 예제-3 (JFrame 클래스를 상속받고 main과 분리하여 사용)

```
import javax.swing.JFrame;

public class MyFrame extends JFrame{
    public MyFrame() {
        //title을 "Frame Test"로 설정하여 frame을 설정
        JFrame frame = new JFrame("Frame Test");
        frame.setSize(300, 200); //프레임의 크기를 300, 200로 설정
        //사용자가 frame의 오른쪽 상단에 있는 close 버튼을 눌렀을 경우 전체 프로그램을 종료하도록 설정
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setVisible(true); //프레임을 화면에 나타나게 만듦
    }

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

GUI

# GUI 작성 절차

## 프레임 생성하기 예제-3 (JFrame 클래스를 상속받고 main과 분리하여 사용)

```
import javax.swing.*;

public class MyFrame extends JFrame{
    public MyFrame() {
        //title을 "Frame Test"로 설정하여 frame을 설정
        this.setTitle("Frame test");
        setSize(300, 200); //프레임의 크기를 300, 200로 설정
        //사용자가 frame의 오른쪽 상단에 있는 close 버튼을 눌렀을 경우 전체 프로그램을 종료하도록 설정
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        setVisible(true); //프레임을 화면에 나타나게 만듦
    }

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

GUI

# 컴포넌트 추가하기

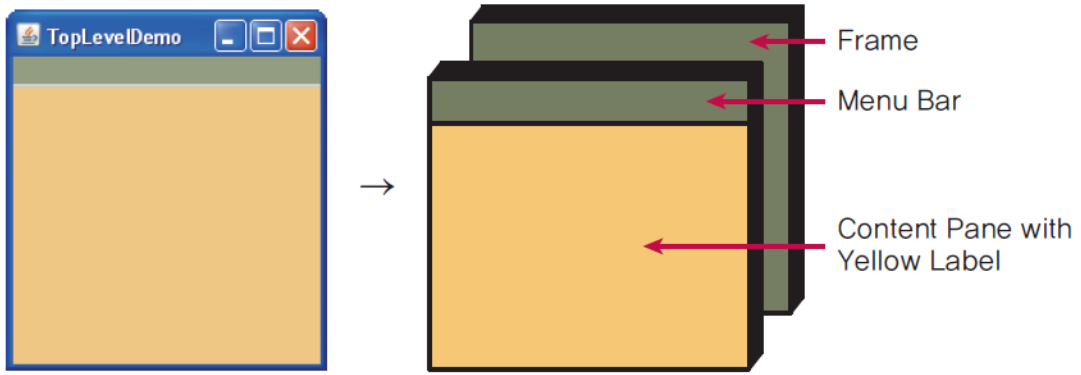
## Button컴포넌트 추가하기

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

public class MyFrame extends JFrame{
    public MyFrame() {
        //title을 "Frame Test"로 설정하여 frame을 설정
        this.setTitle("Frame test");
        setSize(300, 200); //프레임의 크기를 300, 200로 설정
        //사용자가 frame의 오른쪽 상단에 있는 close 버튼을 눌렀을 경우 전체 프로그램을 종료하도록 설정
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        //배치관리자 FlowLayout()객체를 setLayout()메소드를 사용하여 생성 후 배치
        setLayout(new FlowLayout());
        JButton button = new JButton("button");
        add(button);//JButton의 객체를 생성한 후 프레임에 추가

        setVisible(true); //프레임을 화면에 나타나게 만듦
    }

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

GUI

# 최상위 컨테이너(JFrame, JDialog, JApplet)

- 컨테이너는 컴포넌트들을 트리(tree)형태로 저장
- 최상위 컨테이너는 트리의 루트(root) 노드가 됨
- 각 컴포넌트들은 딱 한번만 컨테이너에 포함될 수 있음
  (다른 컨테이너에 있는 컴포넌트를 다른 컨테이너에 넣을 수 없음)
- 최상위 컨테이너는 다른 컨테이너에 포함될 수 없음
- 최상위 컨테이너는 내부에 콘텐트 페인(content pane)을 가짐
- 최상위 컨테이너에는 메뉴바 추가할 수 있음




| 메소드명 | 설명 |
|---|---|
| add(component) | 프레임에 컴포넌트를 추가 |
| setLocation(x, y) | 프레임의 위치를 설정 |
| setSize(width, height) | 프레임의 크기를 설정 |
| setIconImage(IconImage) | 윈도우 시스템에 표시할 아이콘을 설정 |
| setTitle() | 타이틀바의 제목을 설정 |
| setVisible(boolean) | 화면에 표시여부를 설정 |
| setResizable(boolean) | 사용자가 크기를 조절할 수 있는지 여부를 설정 |

GUI

# 컴포넌트 추가하기

## Button컴포넌트 추가하기

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

public class MyFrame extends JFrame{
    public MyFrame() {
        //title을 "Frame Test"로 설정하여 frame을 설정
        this.setTitle("Frame test");
        setSize(300, 200); //프레임의 크기를 300, 200로 설정
        //사용자가 frame의 오른쪽 상단에 있는 close 버튼을 눌렀을 경우 전체 프로그램을 종료하도록 설정
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        //현재 화면의 크기를 얻는다
        Toolkit kit = Toolkit.getDefaultToolkit();
        Dimension screenSize = kit.getScreenSize();
        //프레임의 위치를 현재 화면의 중앙으로 배치함
        setLocation(screenSize.width / 2, screenSize.height / 2);
        //프로젝트 폴더에 있는 icon.png파일으로 아이콘을 변경
        Image image = kit.getImage("icon.png");
        setIconImage(image);
        setLayout(new FlowLayout());

        setVisible(true); //프레임을 화면에 나타나게 만듦
    }

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

GUI

# Panel

- 컴포넌트들을 포함하고 있도록 설계된 컨테이너

# Label

- 편집이 불가능한 텍스트를 생성

```
JLabel label = new JLabel(text);
```

# Text field

- 입력이 가능한 한 줄의 텍스트 필드를 생성

| 메소드명 | 설명 |
|---|---|
| JTextField | 기본적인 텍스트 필드 |
| JFormattedTextField | 사용자가 입력할 수 있는 문자를 제한한다. |
| JPasswordField | 사용자가 입력하는 내용이 보이지 않는다. |
| JComboBox | 사용자가 직접 입력할 수도 있지만 항목 중에서 선택할 수 있다. |
| JSpinner | 텍스트 필드와 버튼이 조합된 것으로 사용자는 이전 버튼과 다음 버튼을 이용하여 선택할 수 있다 |

GUI

# Button

- 사용자가 클릭했을 경우 이벤트를 발생하여 원하는 동작을 실행

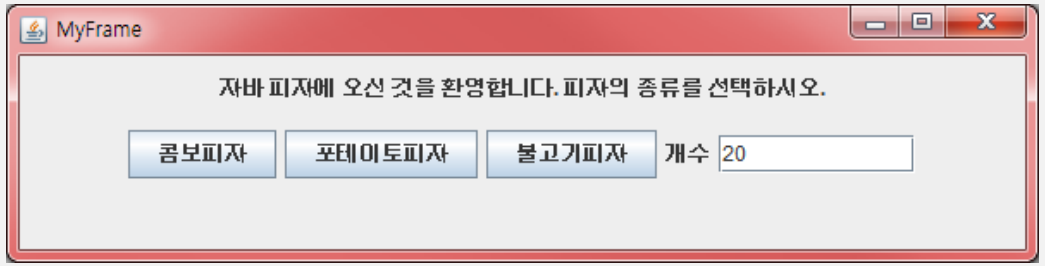
| 메소드명 | 설명 |
|---|---|
| JButton | 가장 일반적인 버튼이다 |
| JCheckBox | 체크박스 버튼 |
| JRadioButton | 라디오 버튼으로 그룹 중의 하나의 버튼만 체크할 수 있다. |

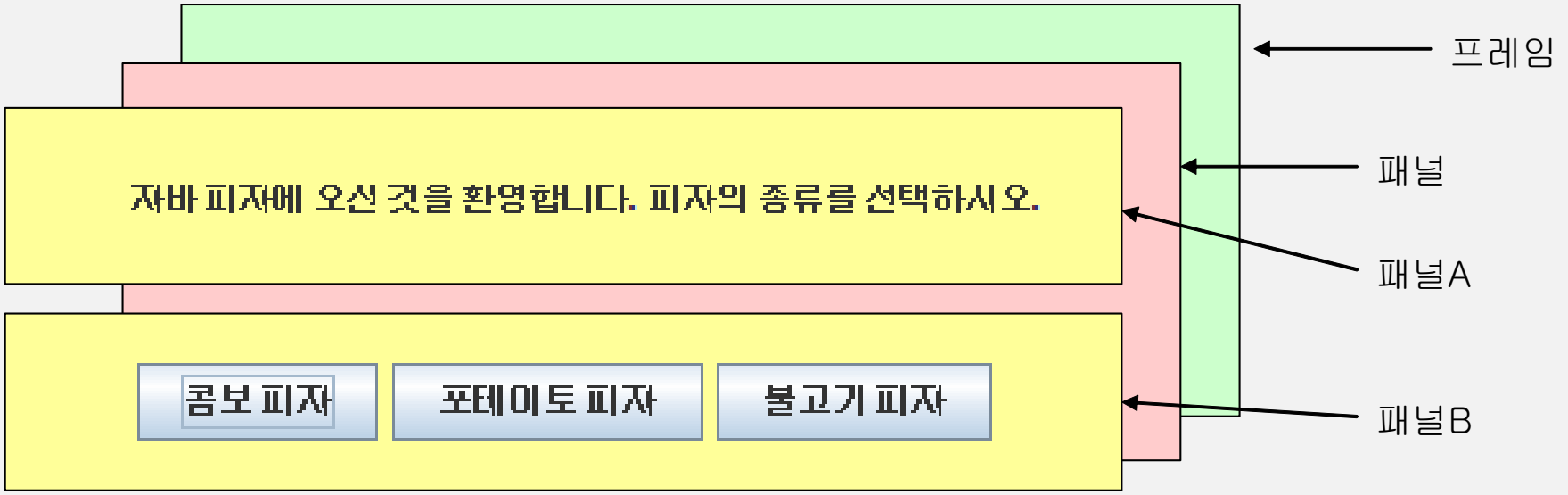
GUI

# 피자 주문 화면 제작

## 다음 그림처럼 프로그램 화면을 디자인하시오.

- 프레임 – (패널 – (패널 A – 패널 B)) 구조를 따름
- 패널 A에는 라벨 컴포넌트 1개가 포함되어 있음
- 패널 B에는 버튼 컴포넌트3개가 포함되어 있음



GUI

# 피자 주문 화면 제작

```
class MyFrame extends JFrame{
    public MyFrame(){
        setSize(600, 150);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setTitle("피자 주문");

        //가장 상위 패널을 생성
        JPanel panel_Parent = new JPanel();
        JPanel panelA = new JPanel();//패널 A를 생성
        JPanel panelB = new JPanel();//패널 B를 생성

        //가장 상위 패널에 패널A와 패널B를 추가
        panel_Parent.add(panelA);
        panel_Parent.add(panelB);

        //텍스트 라벨을 생성
        JLabel label1 = new JLabel("자바 피자에 오신 것을 환영합니다. 피자의 종류를 선택하시오.");
        panelA.add(label1); // 패널 A에 라벨을 추가
```

GUI

# 피자 주문 화면 제작

```
        //버튼 3개를 생성
        JButton button1 = new JButton("콤보피자");
        JButton button2 = new JButton("포테이토피자");
        JButton button3 = new JButton("불고기피자");
        //패널 B에 버튼 3개를 추가
        panelB.add(button1);
        panelB.add(button2);
        panelB.add(button3);

        //개수 라벨을 생성
        JLabel label2 = new JLabel("개수");
        //텍스트필드를 생성 (크기를 10으로)
        JTextField field1 = new JTextField(10);
        //패널B에 라벨과 텍스트필드를 추가
        panelB.add(label2);
        panelB.add(field1);

        //가장 상위 패널을 프레임에 추가
        add(panel_Parent);
        setVisible(true);
    }
}
```

```
public class TEST{
    Run | Debug
    public static void main(String[] args) {
        MyFrame frame = new MyFrame();
    }
}
```

GUI

# 배치관리자 (Layout manager)

- 컨테이너 안의 각 컴포넌트의 위치와 크기를 설정해주는 관리자

## 1. 생성자를 사용

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

## 2. setLayout() 메소드를 사용

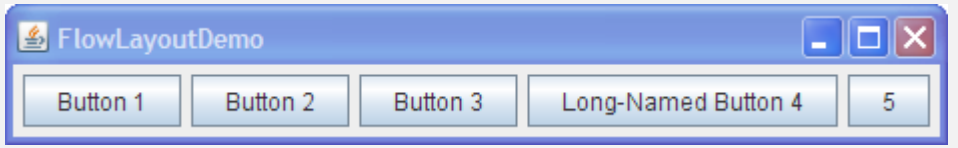
```
panel.setLayout(new FlowLayout());
```

| 메소드명 | 설명 | 예 |
|---|---|---|
| **setMinimumSize()** | 컴포넌트의 최소 크기를 지정 | button.setMinimumSize(new Dimension(300, 200)); |
| **setMaximumSize()** | 컴포넌트의 최대 크기를 지정 | button.setMaximumSize(new Dimension(300, 200)); |
| **setAlignmentX()** | 컴포넌트의 정렬 방식을 지정 | button.setAlignmentX(JComponent.CENTER_ALIGNMENT); |
| **setPreferredSize()** | 컴포넌트의 사이즈를 적절하게 지정 | button. setPreferredSize(); |

GUI

# FlowLayout

- 컴포넌트들을 왼쪽에서 오른쪽으로 버튼을 배치하는 배치 관리자
- 패널의 디폴트 배치 관리자
- 컨테이너의 크기가 변경되면 자동으로 각 컴포넌트를 재배치




| 메소드명 | 설명 |
|---|---|
| **FlowLayout()** | 새로운 FlowLayout 객체를 생성함. 기본 설정은 중앙(center) 배치 |
| **FlowLayout(int align)** | 지정된 정렬 방식을 가진 새로운 FlowLayout 객체를 생성한다. 기본 설정은 중앙(center) 배치이며 간격은 세로, 가로 각각 5 픽셀이다.<br>정렬 매개 변수는 다음 중 하나이다.<br>FlowLayout.LEADING,<br>FlowLayout.CENTER,<br>FlowLayout.TRAILING. |
| **FlowLayout (int align, int hgap, int vgap)** | 지정된 정렬 방식과 수평 간격 hgap과 수직 간격 vgap을 가진 새로운 FlowLayout 객체를 생성한다. |

GUI

# BorderLayout

- 컴포넌트들이 5개의 영역인 North, South, East, West, Center 중 하나로 추가됨
- 프레임, 애플릿, 대화상자의 디폴트 배치 관리자
- BorderLayout에서 컴포넌트를 추가할 때는 어떤 영역에 추가할 것인지를 지정해야 함
  지정하지 않으면 디폴트는 Center

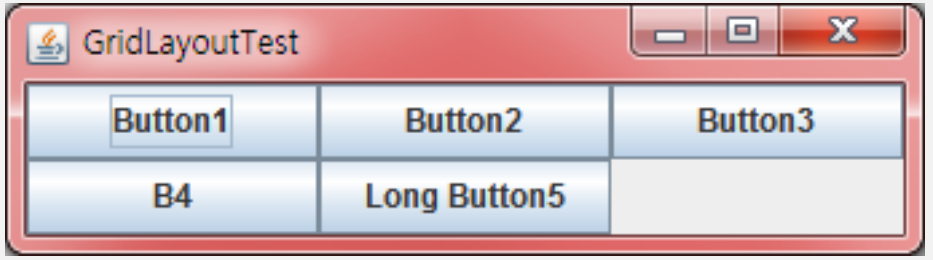
**PAGE_START (또는 NORTH)**
**PAGE_END (또는 SOUTH)**
**LINE_START (또는 WEST)**
**LINE_END (또는 EAST)**
**CENTER**


| 메소드명 | 설명 |
|---|---|
| **BorderLayout(int hgap, int vgap)** | 컴포넌트 사이의 수평 간격 hgap과 수직 간격 vgap을 을 가지는 BorderLayout 객체 생성 |
| **setHgap(int)** | 컴포넌트 사이의 수평 간격 설정(단위는 픽셀) |
| **setVgap(int)** | 컴포넌트 사이의 수직 간격 설정 |

GUI

# GridLayout

- 컴포넌트를 격자 모습으로 배치
- 모든 컴포넌트의 크기는 같고, 컨테이너의 모든 공간은 컴포넌트로 채워짐
- 윈도우의 크기에 따라 컴포넌트의 크기를 맞춤




| 메소드명 | 설명 |
|---|---|
| **GridLayout(int rows, int cols)** | rows 행과 cols 열을 가지는 GridLayout 객체를 생성<br>만약 rows나 cols가 0이면 필요한 만큼의 행이나 열이 만들어짐 |
| **GridLayout(int rows, int cols, int hgap, int vgap)** | rows 행과 cols 열을 가지는 GridLayout 객체를 생성<br>hgap과 vgap은 컴포넌트 사이의 수평 간격과 수직 간격으로 단위는 픽셀 |

GUI

# 배치관리자의 선택

- 컨테이너 안의 각 컴포넌트의 위치와 크기를 설정해주는 관리자

- **컴포넌트를 가능한 크게 나타내고 싶은 경우**
  -> GridLayout 혹은 BorderLayout을 사용

- **몇개의 컴포넌트를 자연스러운 크기로 한줄로 나타내고 싶은 경우**
  -> FlowLayout 혹은 BoxLayout을 사용

- **몇개의 컴포넌트를 행과 열로 동일한 크기로 나타내고 싶은 경우**
  -> GridLayout을 사용

- **몇개의 컴포넌트를 행과 열로 나타내는데 각 컴포넌트의 크기가 다르거나 간격, 정렬 방식을 다르게 내고 싶은 경우**
  -> BoxLayout을 사용

GUI

# 배치 관리자 사용 예제

**3열과 가변적인 행을 갖는 GridLayout을 설정하여 5개의 버튼을 배치하시오**

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

public class MyFrame extends JFrame{
    public MyFrame() {
        //title을 "Frame Test"로 설정하여 frame을 설정
        this.setTitle("Frame test");
        setSize(300, 200); //프레임의 크기를 300, 200로 설정
        //사용자가 frame의 오른쪽 상단에 있는 close 버튼을 눌렀을 경우
        //전체 프로그램을 종료하도록 설정
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        //3개의 열과 필요한 만큼의 행을 가지는 GridLayout생성
        //각 컴포넌트의 간격은 x = 2, y = 2만큼 설정
        setLayout(new GridLayout(0, 3, 2, 2));
```

GUI

# 배치 관리자 사용 예제

## 3열과 가변적인 행을 갖는 GridLayout을 설정하여 5개의 버튼을 배치하시오

```
        //3개의 열과 필요한 만큼의 행을 가지는 GridLayout생성
        //각 컴포넌트의 간격은 x = 2, y = 2만큼 설정
        setLayout(new GridLayout(0, 3, 2, 2));

        add(new JButton("button1"));
        add(new JButton("button2"));
        add(new JButton("button3"));
        add(new JButton("button4"));
        pack(); //각 컴포넌트의 크기를 알맞게 조정해줌
        setVisible(true); //프레임을 화면에 나타나게 만듦
    }

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

GUI

# 절대 위치로 컴포넌트들을 배치하기

- 배치 관리자를 null로 설정
- add() 메소드를 사용하여 컴포넌트를 추가하고 setSize()와 setLocation()을 사용하여 위치와 크기를 지정

```
public class MyFrame extends JFrame{
    public MyFrame() {
        this.setTitle("Frame test");
        setSize(300, 200);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        JPanel p = new JPanel();
        //배치관리자를 사용하지 않고 절대 위치를 사용
        p.setLayout(null);
        JButton b1 = new JButton("b1");
        JButton b2 = new JButton("b2");
        JButton b3 = new JButton("b3");
        p.add(b1);
        p.add(b2);
        p.add(b3);
        //(x위치, y위치, width, height)
        b1.setBounds(20, 5, 95, 30);
        b2.setBounds(40, 50, 80, 100);
        b3.setBounds(200, 70, 50, 70);

        add(p);
        setVisible(true);
    }

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

# 수고하셨습니다.