

## 제 47강

# Java GUI Programming 1

교재: p272~280

# 목차

## 1. Java GUI Programming1

1. GUI
2. AWT 와 Swing
3. Swing GUI Programming

# 1. GUI

## GUI(Graphical User Interface)

: 컴퓨터를 사용하는 사용자를 위해 만들어진 특정한 기능을 가진 그래픽 요소

### [1] JAVA GUI 프로그래밍

: 사용자가 프로그램을 쉽게 다룰 수 있도록 그래픽을 제공하도록 프로그래밍 하는 것

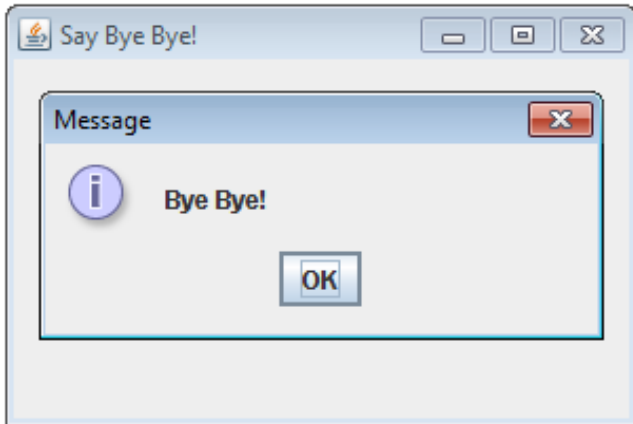
## 2. AWT 와 Swing

### [1] AWT란?

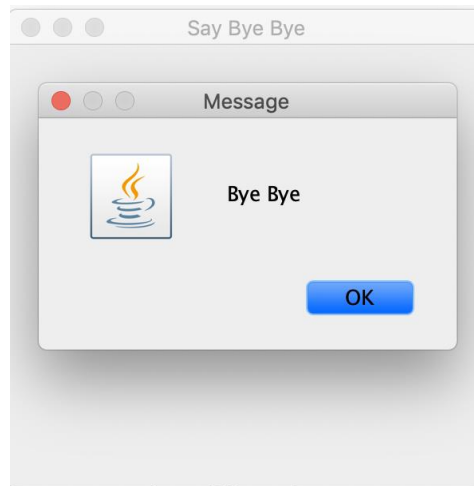
: 자바에서 GUI를 프로그래밍 하기 위해 처음으로 제공한 라이브러리

<특징>

- java.awt 패키지에 Frame, Window, Panel, Button, Label 등이 존재
- 운영체제의 자원 활용



[window]



[linux]

## 2. AWT 와 Swing

### [1] swing란?

: AWT를 대체하기 위해 자바로 작성된 GUI 객체

<특징>

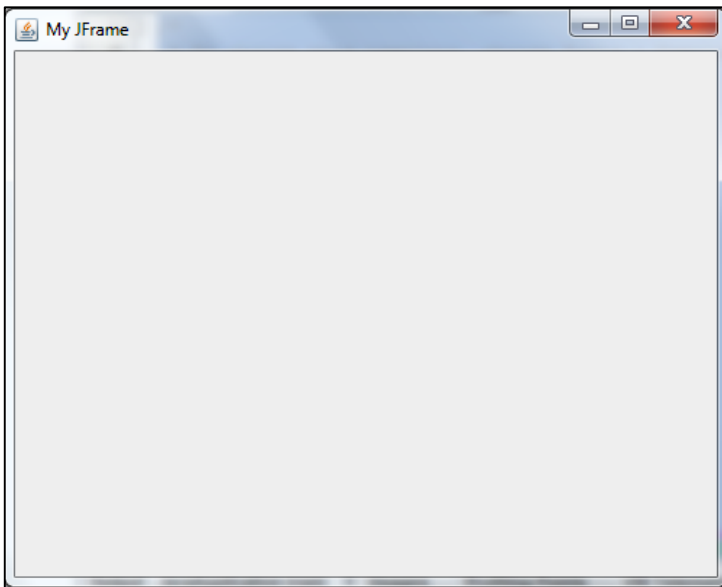
- javax.swing 패키지에 JFrame, JWindow, JPanel, JButton, JLabel 등이 존재
- 운영체제에 의존적이지 않으며, 운영체제와 관계없이 동일한 모양을 사용

## 2. AWT 와 Swing

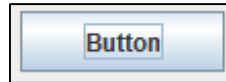
### [3] 컴포넌트(Component)와 컨테이너(Container)

- 컴포넌트: JAVA에서 GUI를 구성하는 요소
- 컨테이너: 다른 컴포넌트를 포함할 수 있는 컴포넌트

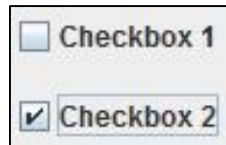
JFrame



JButton



JCheckBox



JSlider



JTextField



JComboBox



JMenu



## 3. Swing GUI Programming

실제 GUI 프로그래밍을 하기 위해서 사용하는 메서드

| 메서드   | 기능                     |
|---|------------------------|
| <code>setLocation(int x, int y)</code>      | 프레임의 위치를 지정            |
| <code>setSize(int width, int height)</code> | 프레임의 크기를 지정            |
| <code>setIconImage(Image IconImage)</code>  | 프레임의 타이틀 바에 보여질 아이콘 설정 |
| <code>setTitle(String title)</code>         | 타이틀바의 제목을 설정           |
| <code>setVisible(boolean b)</code>          | 화면에 표시여부를 설정           |

# 3. Swing GUI Programming

## GUI 프로그래밍 순서

### 1. extends JFrame

- JFrame 클래스를 상속받아, 나만의 프레임 클래스 작성
- 생성자에서 컴포넌트 및 프레임의 디자인 구현하기

```
public class AddFrame extends JFrame {
```

```
    public AddFrame() {
```

### 2. 기본설정 - 생성자에서 하기

: title, size, visible 설정하기

- title 설정 : setTitle("타이틀명");
- size 설정: setSize(가로길이,세로길이);
- visible 설정: setVisible(true);

```
        setTitle("JFrame");  
        setSize(300,300);  
        setVisible(true);
```



### 3. Swing GUI Programming

#### <실습> Exam-102.java

JFrame 클래스를 만들 때 위의 메서드들을 사용해서 기본적인 설정 가능

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

public class AddFrame extends JFrame{
    public AddFrame() {
        setTitle("JFrame");
        setSize(300,300);
        setVisible(true);

        this.setLayout(new FlowLayout());
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }

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

#### <실행 화면>



# 3. Swing GUI Programming

## <실습> Exam-103.java

JFrame에 컴포넌트 추가하는 프로그램

```
class AddFrame2 extends JFrame{
    public AddFrame2() {
        setTitle("JFrame");
        setSize(300,300);

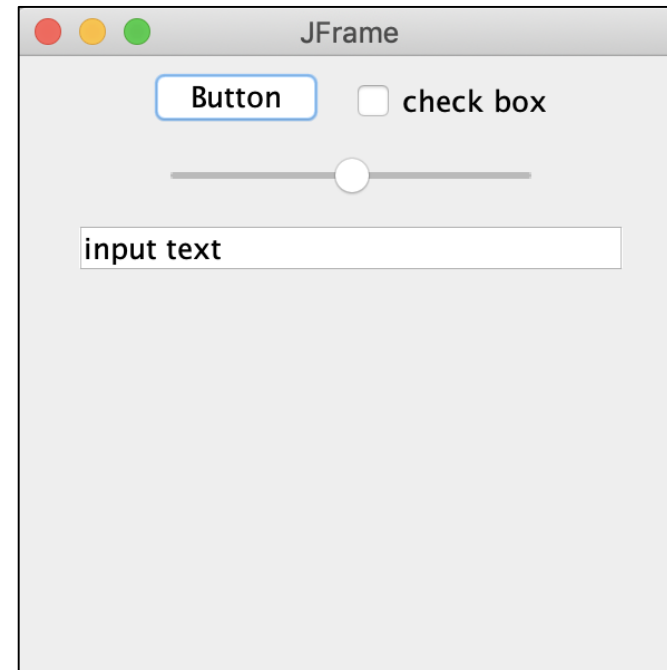
        //컴포넌트 객체 생성
        JButton button = new JButton("Button");
        JCheckBox box = new JCheckBox("check box");
        JSlider slide = new JSlider();
        JTextField tf = new JTextField("input text",20);
        this.setLayout(new FlowLayout());

        //컴포넌트 객체추가
        this.add(button);
        this.add(box);
        this.add(slide);
        this.add(tf);

        setVisible(true);
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}
```

```
public class SwingFrame {
    public static void main(String[] args) {
        new AddFrame2();
    }
}
```

## <실행 화면>



# 3. Swing GUI Programming

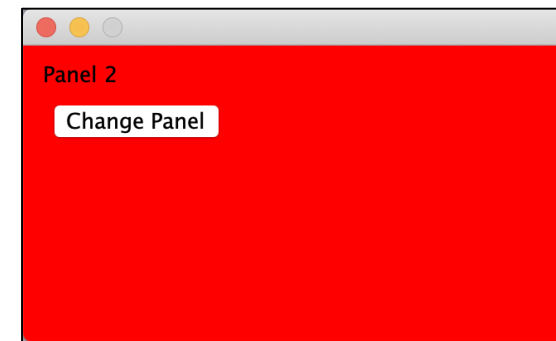
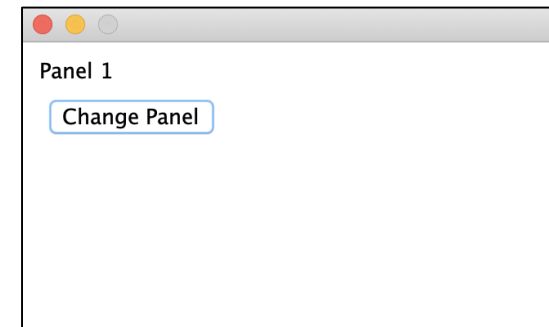
## [5] 배치 관리자(Layout Manager)

: 컨테이너 내부에 컴포넌트 추가 시 각각의 위치와 크기에 관해 결정

- 배치관리자의 종류

| 배치 관리자       | 특징   |
|--------------|--|
| FlowLayout   | 왼->오 순서대로 배치                                     |
| BorderLayout | 컨테이너 공간을 동서남북 중앙으로 나누고<br>지정한 영역에 배치             |
| GridLayout   | 컨테이너 공간을 동일 크기의 2차원 격자로<br>나누고 들어오는 순서대로 왼->오 배치 |
| CardLayout   | 카드를 쌓아놓은 듯이 컴포넌트를<br>쫓개어 배치                      |

### [CardLayout 예시]



# 3. Swing GUI Programming

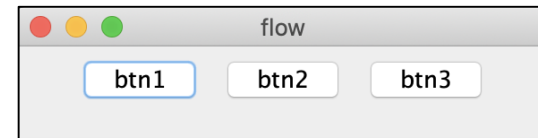
## <실습> LayoutEx.java

JFrame에 레이아웃을 사용해 컴포넌트 추가하기

### 1. FlowLayout

```
class FlowWindow extends JFrame {  
    FlowWindow() {  
        setTitle("flow");  
        setSize(300, 300);  
        JButton btn1 = new JButton("btn1");  
        JButton btn2 = new JButton("btn2");  
        JButton btn3 = new JButton("btn3");  
        this.add(btn1);  
        this.add(btn2);  
        this.add(btn3);  
        this.setLayout(new FlowLayout());  
        setVisible(true);  
    }  
}
```

### <실행 화면>



# 3. Swing GUI Programming

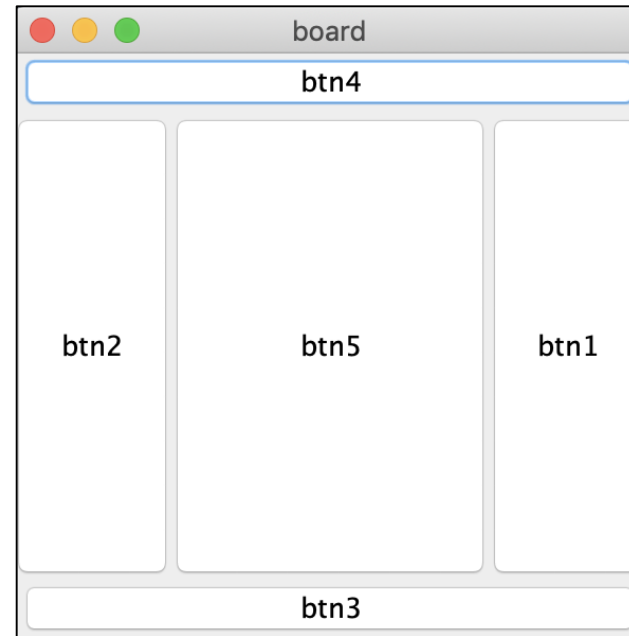
## <실습> LayoutEx.java

JFrame에 레이아웃을 사용해 컴포넌트 추가하기

### 2. BorderLayout

```
class BoardWindow extends JFrame {  
    BoardWindow() {  
        setTitle("board");  
        setSize(300, 300);  
        JButton btn1 = new JButton("btn1");  
        JButton btn2 = new JButton("btn2");  
        JButton btn3 = new JButton("btn3");  
        JButton btn4 = new JButton("btn4");  
        JButton btn5 = new JButton("btn5");  
        BorderLayout bl = new BorderLayout();  
        bl.addLayoutComponent(btn1, BorderLayout.EAST);  
        bl.addLayoutComponent(btn2, BorderLayout.WEST);  
        bl.addLayoutComponent(btn3, BorderLayout.SOUTH);  
        bl.addLayoutComponent(btn4, BorderLayout.NORTH);  
        bl.addLayoutComponent(btn5, BorderLayout.CENTER);  
        this.setLayout(bl);  
        this.add(btn1);  
        this.add(btn2);  
        this.add(btn3);  
        this.add(btn4);  
        this.add(btn5);  
        setVisible(true);  
    }  
}
```

### <실행 화면>



## 3. Swing GUI Programming

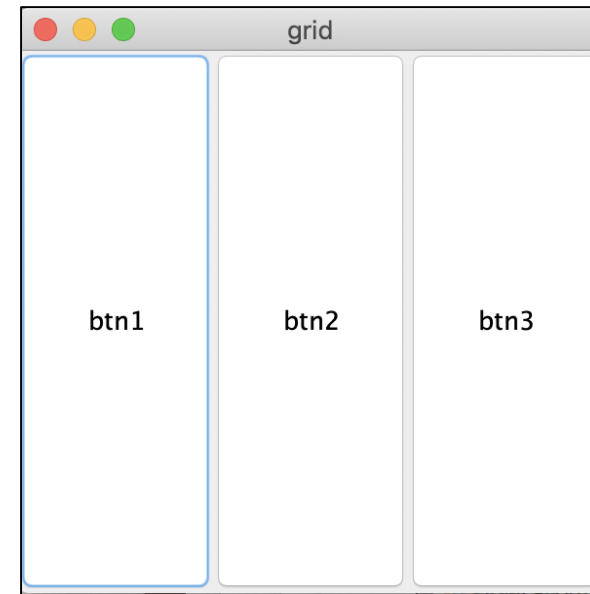
### <실습> LayoutEx.java

JFrame에 레이아웃을 사용해 컴포넌트 추가하기

#### 3. GridLayout

```
class GridWindow extends JFrame {  
    GridWindow() {  
        setTitle("grid");  
        setSize(300, 300);  
        JButton btn1 = new JButton("btn1");  
        JButton btn2 = new JButton("btn2");  
        JButton btn3 = new JButton("btn3");  
        this.setLayout(new GridLayout());  
        this.add(btn1);  
        this.add(btn2);  
        this.add(btn3);  
        setVisible(true);  
    }  
}
```

#### <실행 화면>



## 3. Swing GUI Programming

### <실습> LayoutEx.java

JFrame에 레이아웃을 사용해 컴포넌트 추가하기

#### 3. GridLayout

```
class GridWindow extends JFrame {  
    GridWindow() {  
        setTitle("grid");  
        setSize(300, 300);  
        JButton btn1 = new JButton("btn1");  
        JButton btn2 = new JButton("btn2");  
        JButton btn3 = new JButton("btn3");  
        this.setLayout(new GridLayout(2,2));  
        this.add(btn1);  
        this.add(btn2);  
        this.add(btn3);  
        setVisible(true);  
    }  
}
```

#### <실행 화면>

