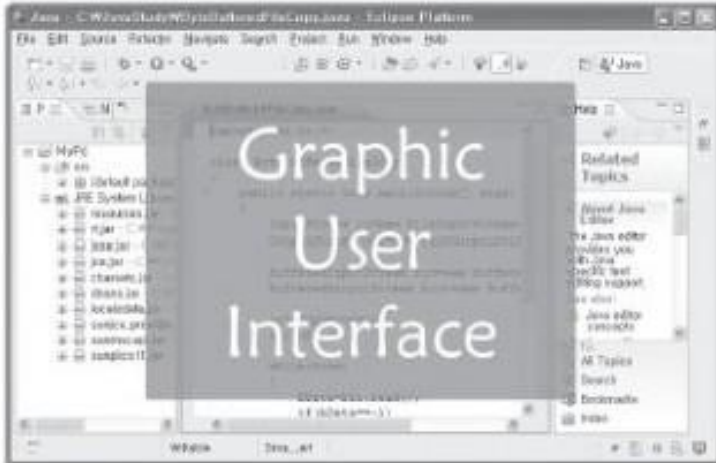


■ Swing



✓ UI와 기능의 구현을 구분하는 형태로 소프트웨어의 개발방법은 발전해 나갈 것이다.

✓ AWT와 Swing은 UI와 기능의 구현을 구분한 모델은 아니다. 하지만, 이에 대한 학습을 통해서 UI 구현에 대한 전반적인 이해를 갖출 수 있다.

✓ Swing을 사용하지 않더라도 Swing의 학습은 다른 프레임워크의 이해에 많은 도움이 된다.

✓ AWT와 Swing을 각각 별도로 공부할 필요는 없다. 둘 중 하나를 선택해서 UI 구현에 대한 경험을 갖는 것에 의미를 두자!

```

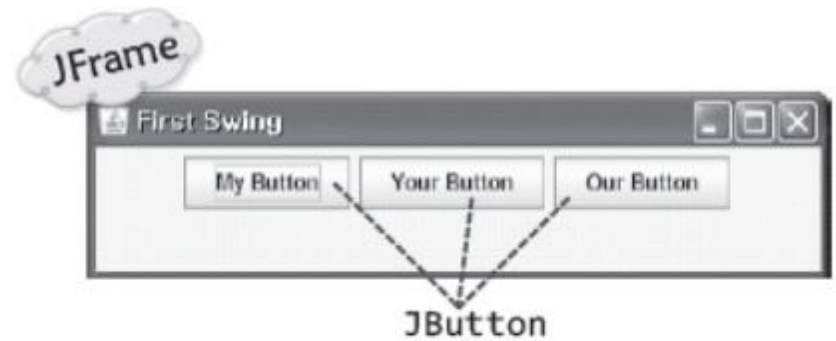
import java.awt.*;
import javax.swing.*;

class FirstSwing
{
    public static void main(String[] args)
    {
        JFrame frm=new JFrame("First Swing");
        frm.setBounds(120, 120, 400, 100);
        frm.setLayout(new FlowLayout());

        JButton btn1=new JButton("My Button");
        JButton btn2=new JButton("Your Button");
        JButton btn3=new JButton("Our Button");

        frm.add(btn1);
        frm.add(btn2);
        frm.add(btn3);
        frm.setVisible(true);
    }
}

```



골격이 되는 창은 **JFrame**의 인스턴스

JFrame이 상속하는 클래스중 하나는 **java.awt.Container** 클래스! 이 클래스를 상속하는 컴포넌트는 다른 Swing 컴포넌트를 엮을 수 있다.
Container 클래스는 배치와 관련된 클래스이다. 따라서 **setLayout** 메소드 역시 **Container** 클래스의 메소드이다.

FlowLayout 인스턴스가 장착되어 컴포넌트의 배치를 관리하게 된다. 단순히 **add** 메소드의 호출을 통해서 컴포넌트를 엮으면 **FlowLayout** 인스턴스가 적절히 배치한다.

setVisible 메소드 호출을 통해서 그간 작성한 **JFrame**을 눈에 보이게 한다.

■ Swing & AWT 코드 비교

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

class FirstSwing
{
    public static void main(String[] args)
    {
        JFrame frm=new JFrame("First Swing");
        frm.setBounds(120, 120, 400, 100);
        frm.setLayout(new FlowLayout());

        JButton btn1=new JButton("My Button");
        JButton btn2=new JButton("Your Button");
        JButton btn3=new JButton("Our Button");

        frm.add(btn1);
        frm.add(btn2);
        frm.add(btn3);
        frm.setVisible(true);
    }
}
```

Swing 코드

```
import java.awt.*;

class FirstAWT
{
    public static void main(String[] args)
    {
        Frame frm=new Frame("First Swing");
        frm.setBounds(120, 120, 400, 100);
        frm.setLayout(new FlowLayout());

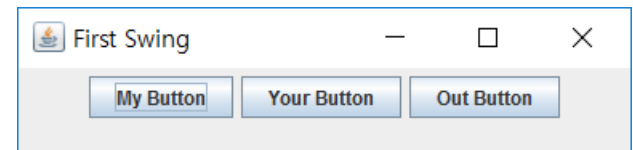
        Button btn1=new Button("My Button");
        Button btn2=new Button("Your Button");
        Button btn3=new Button("Our Button");

        frm.add(btn1);
        frm.add(btn2);
        frm.add(btn3);
        frm.setVisible(true);
    }
}
```

AWT 코드

■ Swing 기본 사용

```
public class SwingExam1 {  
    public static void main(String[] args) {  
        JFrame frm = new JFrame();  
        frm.setTitle("First Swing");  
  
        frm.setBounds(120, 120, 400, 100);  
        frm.setLayout(new FlowLayout());  
  
        JButton btn1 = new JButton("My Button");  
        JButton btn2 = new JButton("Your Button");  
        JButton btn3 = new JButton("Out Button");  
  
        frm.add(btn1);  
        frm.add(btn2);  
        frm.add(btn3);  
        frm.setVisible(true);  
    }  
}
```



■ Exit 메소드

```
class FirstAWTExitEvent
{
    public static void main(String[] args)
    {
        Frame frm=new Frame("First Swing");
        frm.setBounds(120, 120, 400, 100);
        frm.setLayout(new FlowLayout());

        WindowListener listen=new WindowAdapter()
        {
            public void windowClosing(WindowEvent ev)
            {
                System.exit(0);    // 프로그램의 종료 명령 메소드
            }
        };

        frm.addWindowListener(listen);

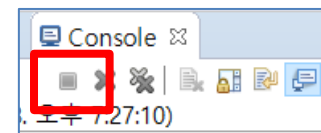
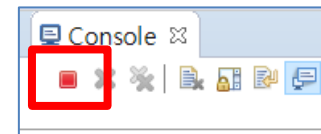
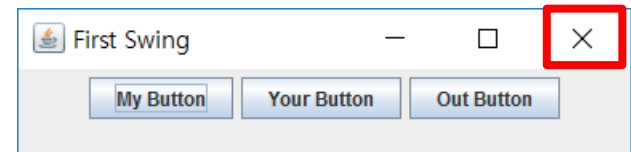
        Button btn1=new Button("My Button");
        Button btn2=new Button("Your Button");
        Button btn3=new Button("Our Button");

        frm.add(btn1);
        frm.add(btn2);
        frm.add(btn3);
        frm.setVisible(true);
    }
}
```

GUI 창의 우측 상단의 x 버튼을 눌러도 프로그램은 종료되지 않는다. x 버튼이 눌렀을 때 프로그램이 종료되기를 원한다면 그에 따른 이벤트 핸들링이 필요하다!

■ Exit – 1

```
public class SwingExam2 {  
    public static void main(String[] args) {  
        JFrame frm = new JFrame("First Swing");  
        frm.setBounds(120, 120, 400, 100);  
        frm.setLayout(new FlowLayout());  
  
        JButton btn1 = new JButton("My Button");  
        JButton btn2 = new JButton("Your Button");  
        JButton btn3 = new JButton("Out Button");  
  
        frm.add(btn1);  
        frm.add(btn2);  
        frm.add(btn3);  
        frm.setVisible(true);  
  
        frm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
    }  
}
```



■ Exit – 2

```
public class SwingExam2 {  
    public static void main(String[] args) {  
        JFrame frm = new JFrame("First Swing");  
        frm.setBounds(120, 120, 400, 100);  
        frm.setLayout(new FlowLayout());  
  
        frm.setVisible(true);  
  
        WindowListener listen = new WindowAdapter() {  
            public void windowClosing(WindowEvent e) {  
                System.exit(0);  
            }  
        };  
  
        frm.addWindowListener(listen);  
    }  
}
```



```

class MouseEventHandler implements MouseListener
{
    /* 마우스 버튼이 클릭되었을 때(눌렀다 풀렸을 때) 호출됩니다. */
    public void mouseClicked(MouseEvent e)
    {
        JButton button=(JButton)e.getComponent();
        button.setText("Clicked");
        System.out.println("Clicked Button"+e.getButton());
        System.out.println("마우스 버튼 눌렀다 풀림");
    }

    /* 마우스 커서가 버튼 위에 올라가면 호출됩니다. */
    public void mouseEntered(MouseEvent e)
    {
        System.out.println("커서 버튼 위 진입");
    }

    /* 마우스 커서가 버튼을 빠져나가면 호출됩니다. */
    public void mouseExited(MouseEvent e)
    {
        System.out.println("커서 버튼 위 탈출");
    }

    /* 마우스 버튼이 눌리는 순간 호출됩니다. */
    public void mousePressed(MouseEvent e)
    {
        System.out.println("마우스 버튼 눌림");
    }

    /* 마우스 버튼이 풀리는 순간 호출됩니다. */
    public void mouseReleased(MouseEvent e)
    {
        System.out.println("마우스 버튼 풀림");
    }
}

```

```

class EventHandler
{
    public static void main(String[] args)
    {
        JFrame frm=new JFrame("First Swing");
        frm.setBounds(120, 120, 400, 100);
        frm.setLayout(new FlowLayout());
        JButton btn1=new JButton("My Button");
        MouseListener listener=new MouseEventHandler();
        btn1.addMouseListener(listener);

        JButton btn2=new JButton("Your Button");
        btn2.addMouseListener(listener);

        JButton btn3=new JButton("Our Button");
        btn3.addMouseListener(listener);

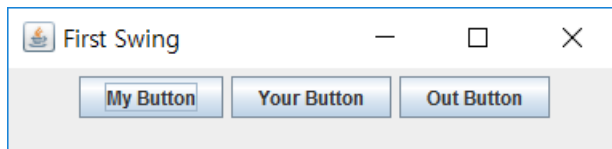
        frm.add(btn1);
        frm.add(btn2);
        frm.add(btn3);
        frm.setVisible(true);
    }
}

```

마우스 이벤트 리스너 등록

■ 이벤트 (1 / 2)

```
public class SwingExam3 {  
    public static void main(String[] args) {  
        JFrame frm = new JFrame("First Swing");  
        frm.setBounds(120, 120, 400, 100);  
        frm.setLayout(new FlowLayout());  
  
        JButton btn1 = new JButton("My Button");  
        JButton btn2 = new JButton("Your Button");  
        JButton btn3 = new JButton("Out Button");  
  
        frm.add(btn1);  
        frm.add(btn2);  
        frm.add(btn3);  
        frm.setVisible(true);
```



```
        frm.setDefaultCloseOperation(  
            JFrame.EXIT_ON_CLOSE);
```

```
        MouseEventHandler mouseEventHandler =  
            new MouseEventHandler();
```

```
        btn1.addMouseListener(  
            mouseEventHandler);
```

```
        btn2.addMouseListener(  
            mouseEventHandler);
```

```
        btn3.addMouseListener(  
            mouseEventHandler);
```

```
    }  
}
```

■ 이벤트 (2 / 2)

```
public class MouseEventHandler
```

```
    implements MouseListener {
```

```
        @Override
```

```
        public void mouseClicked(MouseEvent e) {  
            JButton btn = (JButton)e.getComponent();  
            btn.setText("Clicked");  
            System.out.println("마우스 클릭");  
        }
```

```
        @Override
```

```
        public void mouseEntered(MouseEvent e) {  
            System.out.println("마우스 커서 진입");  
        }
```

```
        @Override
```

```
        public void mouseExited(MouseEvent e) {  
            System.out.println("마우스 커서 탈출");  
        }
```

```
        @Override
```

```
        public void mousePressed(MouseEvent e) {  
            System.out.println("마우스 눌림");  
        }
```

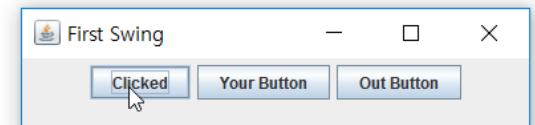
```
        @Override
```

```
        public void mouseReleased(MouseEvent e) {  
            System.out.println("마우스 풀림");  
        }
```

```
    }
```

SwingExam3 [Java Application] C:\Program Files\Java\jre1.8.0_201\bin\javaw.exe

마우스 커서 진입
마우스 커서 탈출
마우스 커서 진입
마우스 눌림
마우스 풀림
마우스 클릭



■ 이벤트 리스너 정보 획득

동일 위치에서 버튼이 눌렀다 놓일 때 호출되는 메소드

```
public void mouseClicked(MouseEvent e)
{
    JButton button=(JButton)e.getComponent();
    button.setText("Clicked");
    System.out.println("clicked Button"+e.getButton());
    System.out.println("마우스 버튼 눌렀다 풀림");
}
```

MouseEvent 인스턴스를 통해서 얻을 수 있는 정보들

- | | |
|------------------------|------------------|
| • 이벤트가 발생한 위치정보 | getX, getY 메소드 |
| • 이벤트가 발생한 인스턴스의 참조 값 | getComponent 메소드 |
| • 이벤트를 발생시킨 마우스 버튼의 종류 | getButton 메소드 |



버튼이 눌린 이후

mouseClicked 메소드 호출을 통해서 발생한 상태의 변화

FlowLayout 배치 관리자

- 왼쪽에서 오른쪽으로 배치한다.
- 중앙으로 정렬해가며 배치한다.
- 한 줄에 모든 컴포넌트를 배치할 수 없을 때에는 다음 줄에 이어서 배치를 한다.

배치의 기준

```
class FlowLayoutManager
{
    public static void main(String[] args)
    {
        JFrame frm=new JFrame("FlowLayout Test");
        frm.setBounds(120, 120, 400, 200);
        frm.setLayout(new FlowLayout());

        frm.add(new JButton("Hi!"));
        frm.add(new JButton("I like Swing"));
        frm.add(new JButton("I am a button"));

        frm.add(new LargeButton("Hi!"));
        frm.add(new LargeButton("I like Swing"));
        frm.add(new LargeButton("I am a button"));
        frm.setVisible(true);
    }
}
```

```
class LargeButton extends JButton
{
    public LargeButton(String str)
    {
        super(str);
    }
    public Dimension getPreferredSize()
    {
        Dimension largeBtnSz
            =new Dimension(
                super.getPreferredSize().width+30,
                super.getPreferredSize().height+15
            );
        return largeBtnSz;
    }
}
```

getPreferredSize 메소드는 JButton 클래스가 상속하는 상위 클래스에 정의된 메소드로써, GUI창에 그려질 컴포넌트의 적절한 크기정보를 반환한다. 그리고 FlowLayout 배치 관리자는 이 메소드를 호출해서 반환되는 값을 참조하여 컴포넌트를 배치한다.

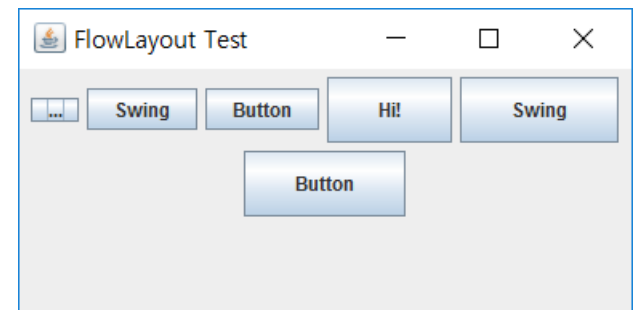
Dimension 클래스의 두 인스턴스 변수는 width와 height이며, 직접 접근이 가능하다.



■ FlowLayout

```
public class SwingExam4 {  
    public static void main(String[] args) {  
        JFrame frm = new JFrame(  
            "FlowLayout Test");  
        frm.setBounds(120, 120, 400, 200);  
        frm.setLayout(new FlowLayout());  
  
        JButton btn1 = new JButton("Hi!");  
        btn1.setPreferredSize(  
            new Dimension(30, 15));  
        frm.add(btn1);  
        frm.add(new JButton("Swing"));  
        frm.add(new JButton("Button"));  
        frm.add(new LargeButton("Hi!"));  
        frm.add(new LargeButton("Swing"));  
        frm.add(new LargeButton("Button"));  
        frm.setVisible(true);  
  
        frm.setDefaultCloseOperation(  
            JFrame.EXIT_ON_CLOSE);  
    }  
}
```

```
public class LargeButton extends JButton {  
    public LargeButton(String str) {  
        super(str);  
    }  
  
    public Dimension getPreferredSize() {  
        Dimension largeBtnSize = new Dimension(  
            super.getPreferredSize().width + 30,  
            super.getPreferredSize().height + 15);  
        return largeBtnSize;  
    }  
}
```



BorderLayout 배치 관리자

```
BorderLayout.NORTH    // 북(North)
BorderLayout.WEST     // 서(West)
BorderLayout.CENTER    // 중앙(Center)
BorderLayout.EAST     // 동(East)
BorderLayout.SOUTH    // 남(South)
```

배치의 기준

```
class BorderLayoutManager
{
    public static void main(String[] args)
    {
        JFrame frm=new JFrame("BorderLayout Test");
        frm.setBounds(120, 120, 300, 200);
        frm.setLayout(new BorderLayout());

        frm.add(new JButton("EAST"), BorderLayout.EAST);
        frm.add(new JButton("WEST"), BorderLayout.WEST);
        frm.add(new JButton("SOUTH"), BorderLayout.SOUTH);
        frm.add(new JButton("NORTH"), BorderLayout.NORTH);
        frm.add(new JButton("CENTER"), BorderLayout.CENTER);
        frm.setVisible(true);
    }
}
```

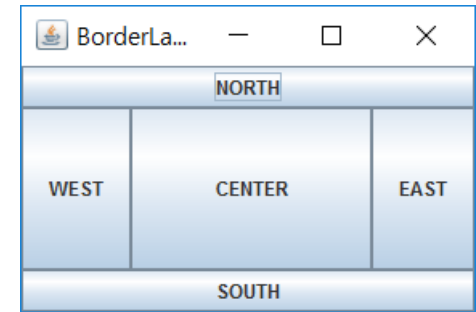


SOUTH, WEST 채우지 않았을 때의 실행결과

중앙을 제외한 나머지 영역이 비게 되면, 그만큼 다른 영역의 컴포넌트에 의해 빈 영역이 채워짐!

■ BorderLayout

```
public class SwingExam5 {  
    public static void main(String[] args) {  
        JFrame frm = new JFrame("BorderLayout Test");  
        frm.setBounds(120, 120, 300, 200);  
        frm.setLayout(new BorderLayout());  
  
        frm.add(new JButton("EAST"), BorderLayout.EAST);  
        frm.add(new JButton("WEST"), BorderLayout.WEST);  
        frm.add(new JButton("SOUTH"), BorderLayout.SOUTH);  
        frm.add(new JButton("NORTH"), BorderLayout.NORTH);  
        frm.add(new JButton("CENTER"), BorderLayout.CENTER);  
  
        frm.setVisible(true);  
  
        frm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
    }  
}
```




```

class GridLayoutManager
{
    public static void main(String[] args)
    {
        JFrame frm=new JFrame("GridLayout Test");
        frm.setBounds(120, 120, 300, 200);
        frm.setLayout(new GridLayout(3, 2));

        frm.add(new JButton("One")); frm.add(new JButton("Two"));
        frm.add(new JButton("Three")); frm.add(new JButton("Four"));
        frm.add(new JButton("Five")); frm.add(new JButton("Six"));
        frm.setVisible(true);
    }
}

```

세로 3, 가로 2로 분할

올리는 대로 왼쪽에서 오른쪽으로,
위에서 아래로



GridLayout 배치 관리자

```
public GridLayout(int rows, int cols, int hgap, int vgap)
```

위의 생성자를 통해서 세로와 가로의 분할정보 뿐만 아니라, 세로와 가로의 컴포넌트간 간격도 지정할 수 있다.



hgap과 vgap에 2를 전달했을때의
실행결과

■ GridLayout

```
public class SwingExam6 {  
    public static void main(String[] args) {  
        JFrame frm = new JFrame("GridLayout Test");  
        frm.setBounds(120, 120, 300, 200);  
        frm.setLayout(new GridLayout(3, 2));  
  
        frm.add(new JButton("One"));  
        frm.add(new JButton("Two"));  
        frm.add(new JButton("Three"));  
        frm.add(new JButton("Four"));  
        frm.add(new JButton("Five"));  
        frm.add(new JButton("Six"));  
  
        frm.setVisible(true);  
        frm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
    }  
}
```

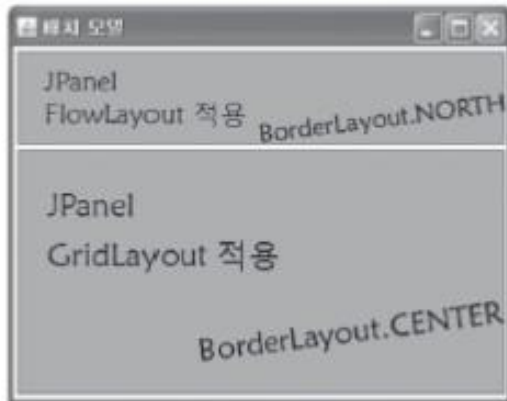


둘 이상의 배치 관리자 적용

JPanel 컴포넌트

- ✓ Jpanel은 눈에 보이는 성격의 컴포넌트가 아니다.
- ✓ JFrame처럼 다른 컴포넌트를 엮을 수 있고, 또 배치 관리자의 지정도 가능하다!

JFrame에는
BorderLayout 적용



적용의 예

JFrame에 배치 관리자 지정해서 두 개의 JPanel을 올린다.
각각의 Jpanel에 각각의 배치 관리자를 별도로 지정한다.



```
class MultiLayoutManager
{
    public static void main(String[] args)
    {
        JFrame frm=new JFrame("Multi Layout Manager");
        frm.setBounds(120, 120, 250, 150);
        frm.setLayout(new BorderLayout());

        JButton btm1=new JButton("B1");
        JButton btm2=new JButton("B2");
        JButton btm3=new JButton("B3");
        JButton btm4=new JButton("B4");
        JButton btm5=new JButton("B5");
        JButton btm6=new JButton("B6");
        JButton btm7=new JButton("B7");
        JButton btm8=new JButton("B8");
        JButton btm9=new JButton("B9");
        JButton btm10=new JButton("B10");

        JPanel panel1=new JPanel();
        panel1.setLayout(new FlowLayout());
        panel1.add(btm1); panel1.add(btm2);
        panel1.add(btm3); panel1.add(btm4);

        JPanel panel2=new JPanel();
        panel2.setLayout(new GridLayout(2, 3, 2, 2));
        panel2.add(btm5); panel2.add(btm6);
        panel2.add(btm7); panel2.add(btm8);
        panel2.add(btm9); panel2.add(btm10);

        frm.add(panel1, BorderLayout.NORTH);
        frm.add(panel2, BorderLayout.CENTER);

        frm.setVisible(true);
    }
}
```

■ JPanel

```
public class SwingExam7 {  
    public static void main(String[] args) {  
        JFrame frm = new JFrame(  
            "Multi Layout Manager");  
        frm.setBounds(120, 120, 250, 150);  
        frm.setLayout(new BorderLayout());  
  
        JPanel panel1 = new JPanel();  
        panel1.setLayout(new FlowLayout());  
        panel1.add(new JButton("B1"));  
        panel1.add(new JButton("B2"));  
        panel1.add(new JButton("B3"));  
        panel1.add(new JButton("B4"));  
  
        JPanel panel2 = new JPanel();  
        panel2.setLayout(  
            new GridLayout(2, 3, 2, 2));  
    }  
}
```

```
        panel2.add(new JButton("B5"));  
        panel2.add(new JButton("B6"));  
        panel2.add(new JButton("B7"));  
        panel2.add(new JButton("B8"));  
        panel2.add(new JButton("B9"));  
        panel2.add(new JButton("B10"));
```

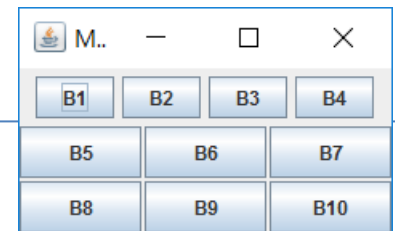
```
        frm.add(panel1, BorderLayout.NORTH);  
        frm.add(panel2, BorderLayout.CENTER);
```

```
        frm.setVisible(true);
```

```
        frm.setDefaultCloseOperation(  
            JFrame.EXIT_ON_CLOSE);
```

```
    }
```

```
}
```



■ 이벤트 종류

• MouseEvent	MouseListener	마우스 관련 이벤트
• MouseEvent	MouseMotionListener	마우스 움직임 관련 이벤트
• TextEvent	TextListener	텍스트 관련 컴포넌트의 문자 편집 이벤트
• ItemEvent	ItemListener	선택 관련 이벤트
• AdjustmentEvent	AdjustmentListener	스크롤 바 이벤트
• WindowEvent	WindowListener	GUI 프레임 창 관련 이벤트
• ActionEvent	ActionListener	컴포넌트 별 특정 행위 관련 이벤트

■ Adapter

```
class MouseEventHandler implements MouseListener
{
    public void mouseClicked(MouseEvent e)
    {
        System.out.println("마우스 버튼 눌렀다 폴림");
        . . . . .
    }
    public void mouseEntered(MouseEvent e) { }
    public void mouseExited(MouseEvent e) { }
    public void mousePressed(MouseEvent e) { }
    public void mouseReleased(MouseEvent e) { }
}
```



Adapter 클래스 기반의 구현

```
class MouseEventHandler extends MouseAdapter
{
    public void mouseClicked(MouseEvent e)
    {
        System.out.println("마우스 버튼 눌렀다 폴림");
        . . . . .
    }
}
```

• MouseListener	MouseAdapter
• MouseMotionListener	MouseMotionAdapter
• TextListener	어댑터 클래스 없음
• ItemListener	어댑터 클래스 없음
• AdjustmentListener	어댑터 클래스 없음
• WindowListener	WindowAdapter
• ActionListener	어댑터 클래스 없음

인터페이스의 모든 메소드를 빈 상태로 구현해 놓은 것이 Adapter 클래스이다.

■ Anonymous 클래스 활용

```
class AdapterAnonymousHandling
{
    public static void main(String[] args)
    {
        JFrame frm=new JFrame("Mouse Motion");
        frm.setBounds(120, 120, 250, 150);
        frm.addMouseListener(
            new MouseAdapter()
            {
                public void mouseClicked(MouseEvent e)
                {
                    System.out.println("마우스 버튼 눌림");
                }
            }
        );

        frm.setVisible(true);
        frm.setDefaultCloseOperation(WindowConstants.DISPOSE_ON_CLOSE);
    }
}
```


■ JLabel & JTextField

```
class PWHandler implements ActionListener
{
    JTextField id;
    JPasswordField pw;

    public PWHandler(JTextField id, JPasswordField pw)
    {
        this.id=id;
        this.pw=pw;
    }

    public void actionPerformed(ActionEvent e)
    {
        System.out.println("ID : "+id.getText());
        System.out.println("Password : "+new String(pw.getPassword()));
        id.setText("");
        pw.setText("");
    }
}
```

JTextField와 JPasswordField의 텍스트상에서 엔터 키가 입력되면
ActionEvent가 발생



JLabel은 문자열 출력을 위한 컴포넌트
JTextField는 문자열 출력을 위한 컴포넌트
JPasswordField는 문자열을 가리며 출력

```
class JLabelAndJTextField
{
    public static void main(String[] args)
    {
        JFrame frm=new JFrame("JLabel & JTextField");
        frm.setBounds(120, 120, 180, 80);
        frm.setLayout(new GridLayout(2, 2));

        JLabel idLabel=new JLabel("ID ", SwingConstants.RIGHT);
        JTextField idText=new JTextField(10);

        JLabel pwLabel=new JLabel("Password ", SwingConstants.RIGHT);
        JPasswordField pwText=new JPasswordField(10);
        pwText.setEchoChar('*');

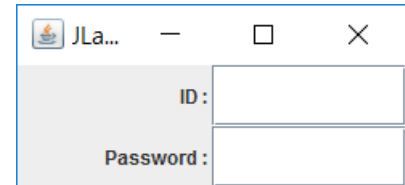
        pwText.addActionListener(new PWHandler(idText, pwText));

        frm.add(idLabel); frm.add(idText);
        frm.add(pwLabel); frm.add(pwText);

        frm.setVisible(true);
        frm.setDefaultCloseOperation(WindowConstants.DISPOSE_ON_CLOSE);
    }
}
```

■ JLabel & JTextField (1 / 2)

```
public class SwingExam8 {  
    public static void main(String[] args) {  
        JFrame frm = new JFrame(  
            "JLabel & JTextField");  
        frm.setBounds(120, 120, 180, 80);  
        frm.setLayout(new GridLayout(2, 2));  
  
        JLabel idLabel = new JLabel(  
            "ID : ", SwingConstants.RIGHT);  
        JTextField idText = new JTextField(10);  
        JLabel pwLabel = new JLabel(  
            "Password : ", SwingConstants.RIGHT);  
        JPasswordField pwText =  
            new JPasswordField(10);  
        pwText.setEchoChar('*');  
        pwText.addActionListener(  
            new PWHandler(idText, pwText));  
    }  
}
```



```
        frm.add(idLabel);  
        frm.add(idText);  
        frm.add(pwLabel);  
        frm.add(pwText);  
  
        frm.setVisible(true);  
  
        frm.setDefaultCloseOperation(  
            JFrame.EXIT_ON_CLOSE);  
    }  
}
```

■ JLabel & JTextField (2 / 2)

```
public class PWHandler implements ActionListener {  
    private JTextField id;  
    private JPasswordField pw;  
  
    public PWHandler(JTextField id, JPasswordField pw) {  
        this.id = id;  
        this.pw = pw;  
    }  
  
    @Override  
    public void actionPerformed(ActionEvent e) {  
        System.out.println("ID : " + id.getText());  
        System.out.println("PW : " + new String(pw.getPassword()));  
        id.setText("");  
        pw.setText("");  
    }  
}
```

■ JTextArea

```
class ButtonTextHandler implements ActionListener
{
    JTextArea textArea;
    public ButtonTextHandler(JTextArea area)
    {
        textArea=area;
    }
    public void actionPerformed(ActionEvent e)
    {
        textArea.setText("모두 지웠습니다. \n");
        textArea.append("원하는 내용 입력하세요. \n");
    }
}
```

JTextArea는 여러 줄의 문자열 입력을 위한 컴포넌트

```
class JTextAreaSimpleModel
{
    public static void main(String[] args)
    {
        JFrame frm=new JFrame("JTextArea");
        frm.setBounds(120, 120, 250, 270);
        frm.setLayout(new FlowLayout());
        JTextArea textArea=new JTextArea(10, 20);
        textArea.append("원하는 내용 입력하세요. \n");
        textArea.setCaretPosition(textArea.getText().length());
        textArea.setLineWrap(true);
        textArea.setWrapStyleWord(true);

        JButton btn=new JButton("Clear");
        btn.addActionListener(new ButtonTextHandler(textArea));
        frm.add(textArea); frm.add(btn);
        frm.setVisible(true);
        frm.setDefaultCloseOperation(WindowConstants.DISPOSE_ON_CLOSE);
    }
}
```

자동 줄 바꿈
단어 단위 줄 바꿈

세로 10, 가로 20의 JTextArea 생성

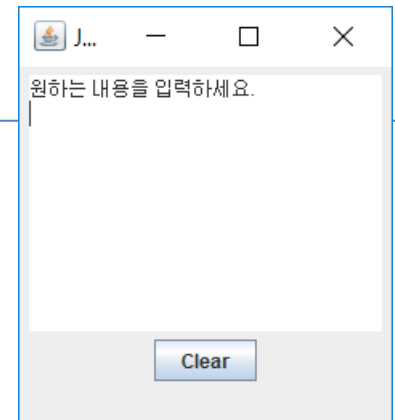
커서 다음 행으로 이동



■ JTextArea (1 / 2)

```
public class SwingExam9 {  
    public static void main(String[] args) {  
        JFrame frm = new JFrame("JTextArea");  
        frm.setBounds(120, 120, 250, 270);  
        frm.setLayout(new FlowLayout());  
  
        JTextArea textArea = new JTextArea(10, 20);  
        textArea.append(  
            "원하는 내용을 입력하세요.\n");  
        textArea.setCaretPosition(  
            textArea.getText().length());  
  
        JButton btn = new JButton("Clear");  
        btn.addActionListener(  
            new ButtonTextHandler(textArea));  
    }  
}
```

```
        frm.add(textArea);  
        frm.add(btn);  
  
        frm.setVisible(true);  
        frm.setDefaultCloseOperation(  
            JFrame.EXIT_ON_CLOSE);  
    }  
}
```



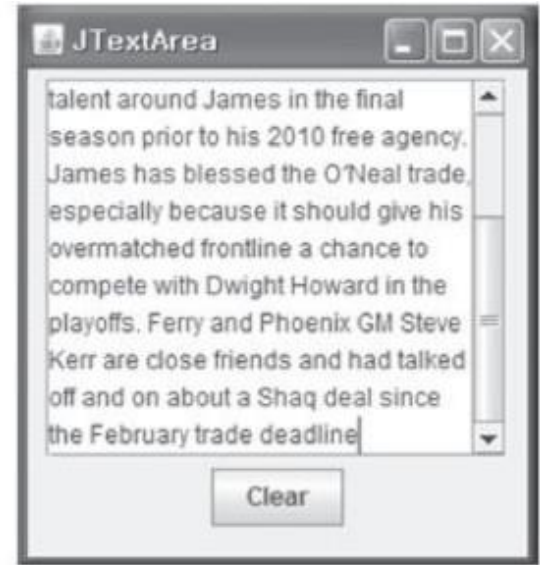
■ JTextArea (2 / 2)

```
public class ButtonTextHandler implements ActionListener {  
    private JTextArea textArea;  
  
    public ButtonTextHandler(JTextArea textArea) {  
        this.textArea = textArea;  
    }  
  
    @Override  
    public void actionPerformed(ActionEvent e) {  
        textArea.setText("모두 지웠습니다.\n");  
        textArea.append("원하는 내용을 입력하세요.\n");  
    }  
}
```

■ JScrollPane

```
JScrollPane simpleScroll=new JScrollPane(textArea);  
frm.add(simpleScroll); frm.add(btn);  
  
frm.setVisible(true);  
frm.setDefaultCloseOperation(WindowConstants.DISPOSE_ON_CLOSE);
```

스크롤 바가 필요한 시점에 생성됨



```
JScrollPane simpleScroll=  
    new JScrollPane(textArea,  
        JScrollPaneConstants.VERTICAL_SCROLLBAR_ALWAYS,    // 세로 스크롤 정책  
        JScrollPaneConstants.HORIZONTAL_SCROLLBAR_NEVER    // 가로 스크롤 정책  
    );
```

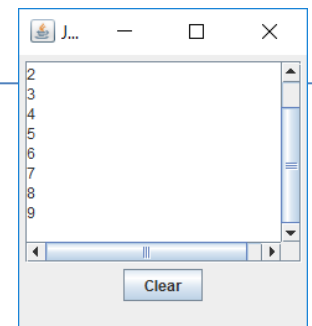
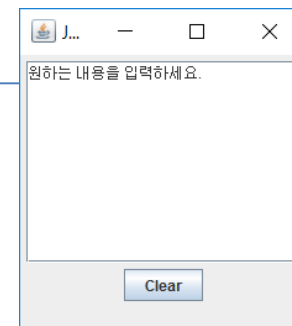
VERTICAL_SCROLLBAR_ALWAYS은 계속해서 스크롤 바를 표시함을 의미
HORIZONTAL_SCROLLBAR_NEVER는 스크롤 바를 표시하지 말라는 의미



■ JScrollPane (1 / 2)

```
public class SwingExam10 {  
    public static void main(String[] args) {  
        JFrame frm = new JFrame("JTextArea");  
        frm.setBounds(120, 120, 250, 270);  
        frm.setLayout(new FlowLayout());  
        JTextArea textArea = new JTextArea(10, 20);  
        textArea.append(  
            "원하는 내용을 입력하세요.\n");  
        textArea.setCaretPosition(  
            textArea.getText().length());  
  
        JScrollPane simpleScroll =  
            new JScrollPane(textArea);  
    }  
}
```

```
JButton btn = new JButton("Clear");  
btn.addActionListener(  
    new ButtonTextHandler2(textArea));  
frm.add(simpleScroll);  
frm.add(btn);  
  
frm.setVisible(true);  
frm.setDefaultCloseOperation(  
    JFrame.EXIT_ON_CLOSE);  
}
```



■ JScrollPane (2 / 2)

```
public class ButtonTextHandler2 implements ActionListener {  
    private JTextArea textArea;  
  
    public ButtonTextHandler2(JTextArea textArea) {  
        this.textArea = textArea;  
    }  
  
    @Override  
    public void actionPerformed(ActionEvent e) {  
        textArea.setText("모두 지웠습니다.\n");  
        textArea.append("원하는 내용을 입력하세요.\n");  
    }  
}
```