Swing 기초 2

기초 컴포넌트

- 프레임(frame)
- 이벤트 처리
- 패널(panel)
- 레이블(label)
- 버튼(button)
- 텍스트 필드(text field)

프레임

- 메뉴를 불일 수 있는 윈도우

생성자 또는 메소드	설명
void add(Component c)	지정된 컴포넌트를 프레임에 추가한다.
JMenuBar getJMenuBar()	이 프레임에 대한 메뉴를 얻는다.
void pack()	프레임을 크기를 추가된 컴포넌트들의 크기에 맞도록 조절한다.
void remove(Component c)	지정된 컴포넌트를 프레임에서 제거한다.
void setDefaultCloseOperation()	사용자가 프레임을 닫을 때 수행되는 동작을 설정한다. 일반적으로 JFrame.EXIT_ON_CLOSE으로 지정한다.
void setIconImage(Icon image)	프레임이 최소화되었을때의 아이콘 지정
void setLayout(LayoutManager layout)	프레임위에 놓이는 컴포넌트들을 배치하는 배치관리자 지정, 디폴트는 BorderLayout 배치관리자
void setLocation(int x, int y)	프레임의 x좌표와 y좌표를 지정한다.
void setResizeable(boolean value)	프레임의 크기 변경 허용 여부
void setSize(int width, int height)	프레임의 크기 설정
void setMenuBar(JMenuBar menu)	현재 프레임에 메뉴바를 붙인다.

프레임

- Toolkit 클래스 (java.awt.Toolkit)
  - 이미지를 불러오거나, 화면 정보 제공
  - 주요 메소드
    - Toolkit getDefaultToolkit()
      - 기본 툴킷 객체를 반환
    - Image getImage(String fileName)
      - fileName에 해당하는 이미지를 반환
    - Dimension getScreenSize()
      - 화면의 크기를 반환

```
Toolkit kit = Toolkit.getDefaultToolkit();
Dimension screenSize = kit.getScreenSize();
int screenHeight = screenSize.height;
int screenWidth = screenSize.width;
setSize(screenWidth/2, screenHeight/2);
setLocation( screenWidth/4, screenHeight/4);
Image img = kit.getImage("그림1.jpg");
setIconImage(img);
this.setTitle( "Frame 만들기 ");
```

프레임

- 버튼 클릭시 프레임 크기 변경하기

  java.awt.event.ActionListener;

```
JButton b= new JButton("프레임 크기 조절하기");
boolean sizeToggle = true;
b.addActionListener(this);
```

ActionListener 인터페이스 구현

```
public void actionPerformed(ActionEvent e)
{
        if ( sizeToggle == true ){
                this.setTitle( "작은 프레임");
                setSize(screenWidth/3, screenHeight/3);
                sizeToggle = false;
        }else{
                this.setTitle( "큰 프레임");
                setSize(screenWidth/2, screenHeight/2);
                sizeToggle = true;
        }
}
```

이벤트 처리 과정

이벤트 처리 과정

(1) 이벤트를 발생하는 컴포넌트를 생성하여야 한다.

```
public class MyFrame extends JFrame { // 프레임을 상속하여서 MyFrame 선언
    private JButton button;    // 버튼을 참조하는 변수를 선언
    ...
    public MyFrame() // 생성자에서 컴포넌트를 생성하고 추가한다.
    {
            JPanel panel = new JPanel();        // 패널 생성
            button = new JButton("동작");       // 버튼 생성
            panel.add(button);                  // 버튼을 패널에 추가
            add(panel);                         // 패널을 프레임에 추가
            ...
    }
    ...
}
```

이벤트 처리 과정

(2) 이벤트 리스너 클래스를 작성한다.

```
class MyListener implements ActionListener {
    public void actionPerformed(ActionEvent e) {
        ... // Action 이벤트를 처리하는 코드가 여기에 들어간다.
    }
}
```

```
public class MyFrame extends JFrame implements ActionListener{

    @Override
    public void actionPerformed(ActionEvent e) {

    }
}
```

이벤트 처리 과정

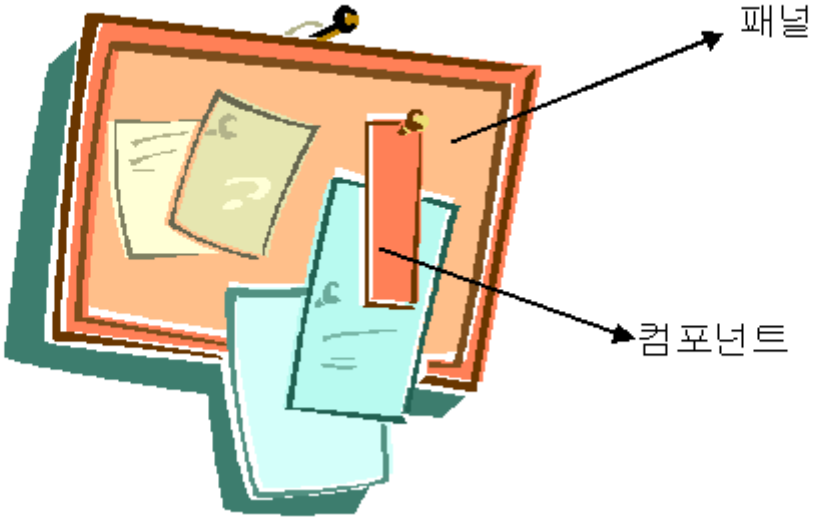
(3) 이벤트 리스너를 이벤트 소스에 등록한다.

```
public class MyFrame extends JFrame { // 프레임을 상속하여서 MyFrame 선언
    ...
    public MyFrame()// 생성자에서 컴포넌트를 생성하고 추가한다.
    {
            JPanel panel = new JPanel();        // 패널 생성
            button = new JButton("동작");       // 버튼 생성
            button.addActionListener(new MyListener());        // 이벤트 리스너 등록
            panel.add(button);                  // 버튼을 패널에 추가
            add(panel);                         // 패널을 프레임에 추가
            ...
    }
    ...
}
```

이벤트 리스너를 컴포넌트에 붙인다.

패널

- 패널(panel)은 컴포넌트들을 가질 수 있는 컨테이너



패널

- *생성자*

메소드	설명
JPanel()	새로운 패널을 생성한다.
JPanel(boolean isDoubleBuffered)	만약 매개변수가 참이면 더블 버퍼링을 사용한다.
JPanel(LayoutManager layout)	지정된 배치 관리자를 사용하는 패널을 생성한다.

- *메소드*

메소드	설명
void add(Component c)	지정된 컴포넌트를 패널에 추가한다.
void remove(Component c)	지정된 컴포넌트를 패널에서 제거한다.
void setLayout(LayoutManager layout)	배치 관리자를 지정한다. 디폴트는 FlowLayout이다.
void setLocation(int x, int y)	패널의 위치를 지정한다.
void setSize(int width, int height)	패널의 크기를 지정한다.
void setToolTipText(String text)	사용자가 마우스를 패널의 빈곳에 올려놓으면 툴팁을 표시한다.

레이블

- 레이블은 편집이 불가능한 텍스트를 표시.
  - (예) JLabel label = **new** JLabel("안녕하세요?");

메소드	설명
String getText()	레이블의 텍스트를 반환한다.
void setText(String text)	레이블의 텍스트를 설정한다.
void setToolTipText(String text)	사용자가 마우스를 레이블 위에 올려놓으면 툴팁을 표시한다.
void setVisible(boolean value)	레이블을 보이게 하거나 감춘다.

레이블

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

class MyFrame extends JFrame {
        public MyFrame() {
                setSize(300, 200);
                setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
                setTitle("MyFrame");

                JPanel panel = new JPanel();          // 패널 생성
                JLabel label = new JLabel("안녕하세요?");     // 레이블 생성
                JButton button = new JButton("버튼");         // 버튼 생성
                panel.add(label);                     // 패널에 레이블 추가
                panel.add(button);                    // 패널에 버튼 추가
                add(panel);                           // 패널을 프레임에 추가
                setVisible(true);
        }
}

public class MyFrameTest3 {
        public static void main(String[] args) {
                MyFrame f = new MyFrame();
        }
}
```

버튼

- 버튼은 사용자가 클릭했을 경우, 이벤트를 발생하여 원하는 동작을 하게 하는데 이용된다

메소드	설명
String getText()	버튼의 현재 텍스트를 반환한다.
void setText(String text)	버튼의 텍스트를 설정한다.
void doClick()	사용자가 버튼을 누른 것처럼 이벤트를 발생한다.
void setBorderPainted(booelan value)	버튼의 경계를 나타내거나 감춘다.
void setContentAreaFilled(boolean value)	버튼의 배경을 채울 것인지를 지정한다.
void setEnabled(boolean value)	버튼을 활성화하거나 비활성화한다.
void setRolloverEnabled(boolean value)	마우스가 버튼 위에 있으면 경계를 진하게 하는 롤오버 효과를 설정
void setToolTipText(String text)	사용자가 마우스를 버튼 위에 올려놓으면 툴팁을 표시한다.
void setVisible(boolean value)	버튼을 보이게 하거나 감춘다.

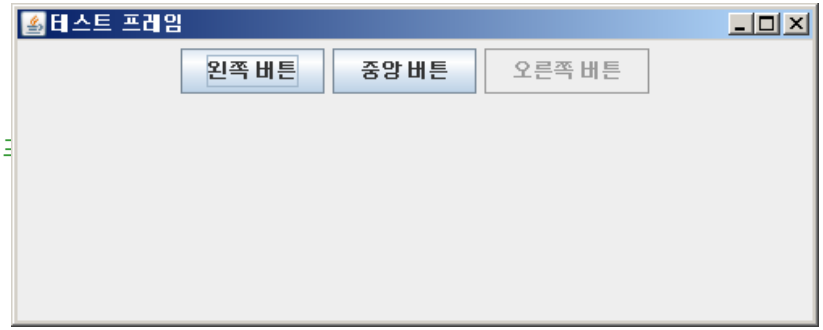
버튼

```
class MyFrame extends JFrame {
        public MyFrame() {
         setSize(500, 200);
         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
         setTitle("테스트 프레임");

         JPanel panel = new JPanel();
         JButton b1 = new JButton();
         b1.setText("왼쪽 버튼");
         JButton b2 = new JButton("중앙 버튼");
         JButton b3 = new JButton("오른쪽 버튼");
         b3.setEnabled(false);      // 세 번째 버튼을 불활성으로 설정

         // 컴포넌트를 패널에 추가
         panel.add(b1);
         panel.add(b2);
         panel.add(b3);

         add(panel);        // 패널을 프레임에
         setVisible(true);
        }
}
```



텍스트 필드

- 텍스트 필드(text field)는 입력이 가능한 한 줄의 텍스트 필드를 만드는 데 사용

  - 생성자

생성자	설 명
JTextField()	TextField를 생성한다.
JTextField(int columns)	지정된 칸수를 가지고 있는 TextField를 생성한다.
JTextField(String text)	지정된 문자열로 초기화된 TextField를 생성한다.

  - 메소드

메소드	설 명
void setText(String text)	지정된 문자열을 텍스트 필드에 쓴다.
String getText()	텍스트 필드에 입력된 문자열을 반환한다.
void setEditable(boolean) boolean isEditable()	사용자가 텍스트를 입력할 수 있는지 없는지를 설정하고 반환한다.

텍스트 필드

```
class MyFrame extends JFrame {
        public MyFrame() {
                setSize(500, 100);
                setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
                setTitle("테스트 프레임");

                JPanel panel = new JPanel();
                JTextField t1 = new JTextField(10);
                JTextField t2 = new JTextField(10);
                t2.setEditable(false);

                // 컴포넌트를 패널에 추가
                panel.add(t1);
                panel.add(t2);

                add(panel); // 패널을 프레임에 추가
                setVisible(true);
        }
}
```