

10

자바의 이벤트 처리

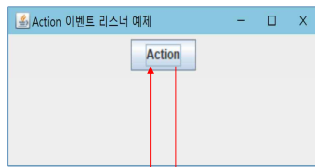
이벤트 리스너 작성 방법

2

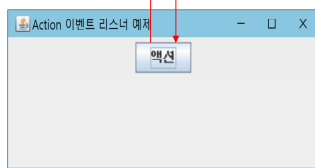
□ 3 가지 방법

1. 독립 클래스로 작성
 - 이벤트 리스너를 완전한 클래스로 작성
 - 이벤트 리스너를 여러 곳에서 사용할 때 적합
2. 내부 클래스(inner class)로 작성
 - 클래스 안에 멤버처럼 클래스 작성
 - 이벤트 리스너를 특정 클래스에서만 사용할 때 적합
3. 익명 클래스(anonymous class)로 작성
 - 클래스의 이름 없이 간단히 리스너 작성
 - 클래스 조차 만들 필요 없이 리스너 코드가 간단한 경우에 적합

예제 10-1 : 독립 클래스로 Action 이벤트의 리스너 작성



버튼 클릭



MyActionListener.java
파일로 작성하여도 됨

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

public class IndepClassListener extends JFrame {
    public IndepClassListener() {
        setTitle("Action 이벤트 리스너 예제");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        Container c = getContentPane();
        c.setLayout(new FlowLayout());
        JButton btn = new JButton("Action");
        btn.addActionListener(new MyActionListener()); // Action 리스너 달기
        c.add(btn);

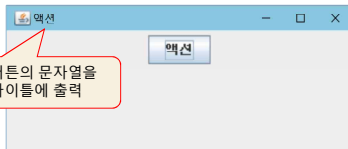
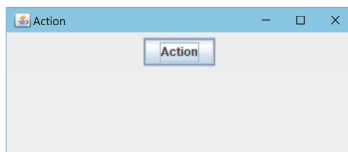
        setSize(350, 150);
        setVisible(true);
    }

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

class MyActionListener implements ActionListener {
    public void actionPerformed(ActionEvent e) {
        JButton b = (JButton)e.getSource();
        if(b.getText().equals("Action"))
            b.setText("액션");
        else
            b.setText("Action");
    }
}
```

3

예제 10-2 : 내부 클래스로 Action 이벤트 리스너 만들기



- Action 리스너를 내부 클래스로 작성
- private으로 선언하여 InnerClassListener 외부에서 사용할 수 없게 함
- 리스너에서 InnerClassListener의 멤버에 대한 접근 용이

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

public class InnerClassListener extends JFrame {
    public InnerClassListener() {
        setTitle("Action 이벤트 리스너 예제");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        Container c = getContentPane();
        c.setLayout(new FlowLayout());
        JButton btn = new JButton("Action");
        btn.addActionListener(new MyActionListener());
        c.add(btn);

        setSize(350, 150);
        setVisible(true);
    }

    private class MyActionListener implements ActionListener {
        public void actionPerformed(ActionEvent e) {
            JButton b = (JButton)e.getSource();
            if(b.getText().equals("Action"))
                b.setText("액션");
            else
                b.setText("Action");
            // InnerClassListener의 멤버나 JFrame의 멤버를 호출할 수 있음
            InnerClassListener.this.setTitle(b.getText()); // 프레임 타이틀에
                                                         버튼 문자열을 출력한다.
        }
    }

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

4

익명 클래스로 이벤트 리스너 작성

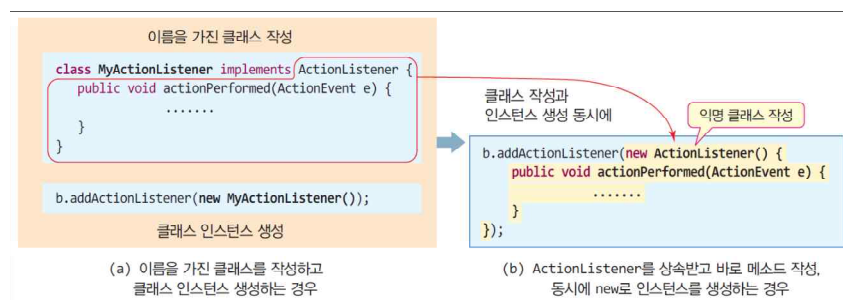
5

□ 익명 클래스란?

- (클래스 정의 + 인스턴스 생성)을 한번에 작성

```
new 익명클래스의수퍼클래스이름(생성자의 인자들) {
    .....
    멤버 구현
    .....
};
```

- ActionListener를 구현하는 익명의 이벤트 리스너 작성 예



예제 10-3 : 익명 클래스로 Action 이벤트 리스너 만들기

익명 클래스로 Action 리스너 작성

AnonymousClassListener의 멤버나 JFrame의 멤버를 호출할 수 있음

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

public class AnonymousClassListener extends JFrame {
    public AnonymousClassListener() {
        setTitle("Action 이벤트 리스너 작성");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        Container c = getContentPane();
        c.setLayout(new FlowLayout());
        JButton btn = new JButton("Action");
        c.add(btn);

        btn.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                JButton b = (JButton)e.getSource();
                if(b.getText().equals("Action"))
                    b.setText("액션");
                else
                    b.setText("Action");
                setTitle(b.getText());
            }
        });

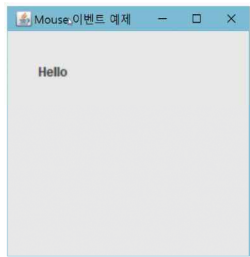
        setSize(350, 150);
        setVisible(true);
    }

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

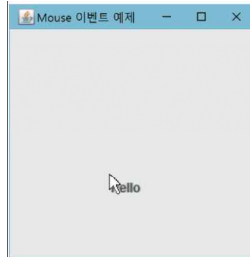
예제 10-4 : 마우스로 문자열 이동시키기 – 마우스 이벤트 연습

7

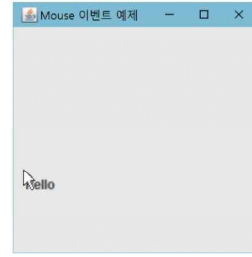
컨텐츠팬의 아무 위치에 마우스 버튼을 누르면 마우스 포인트가 있는 위치로 "hello" 문자열을 옮기는 스윙 응용프로그램을 작성하라.



초기화면



마우스 다른 곳에 클릭한 경우



마우스 다른 곳에 클릭한 경우

예제 10-4의 정답

8

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

public class MouseListenerEx extends JFrame {
    private JLabel la = new JLabel("Hello");

    public MouseListenerEx() {
        setTitle("Mouse 이벤트 예제");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        Container c = getContentPane();
        c.addMouseListener(new MyMouseListener());

        c.setLayout(null);
        la.setSize(50, 20);
        la.setLocation(30, 30);
        c.add(la);

        setSize(250, 250);
        setVisible(true);
    }
}
```

마우스 버튼이 눌러진 위치를 알아내어 la("hello" 문자열)를 그 위치로 옮긴다.

```
class MyMouseListener implements MouseListener {
    public void mousePressed(MouseEvent e) {
        int x = e.getX();
        int y = e.getY();
        la.setLocation(x, y);
    }

    public void mouseReleased(MouseEvent e) {}
    public void mouseClicked(MouseEvent e) {}
    public void mouseEntered(MouseEvent e) {}
    public void mouseExited(MouseEvent e) {}
}

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

어댑터(Adapter) 클래스

9

- 이벤트 리스너 구현에 따른 부담 해소를 위해
 - ▣ 리스너 작성시 추상 메소드들을 모두 구현해야 하는 부담
 - 마우스 리스너에서 마우스가 눌러지는 경우(mousePressed())만 처리하고자 하는 경우에도 나머지 4 개의 메소드를 모두 구현해야 하는 부담
- 어댑터 클래스
 - ▣ 리스너의 모든 메소드가 단순 리턴하도록 구현해 놓은 클래스
 - MouseAdapter 예
- ▣ 추상 메소드가 하나뿐인 리스너는 어댑터 클래스 없음
 - ActionListener, ItemAdapter 클래스는 존재하지 않음

```
class MouseAdapter implements MouseListener, MouseMotionListener,
                               MouseWheelListener {
    public void mousePressed(MouseEvent e) {}
    public void mouseReleased(MouseEvent e) {}
    public void mouseClicked(MouseEvent e) {}
    public void mouseEntered(MouseEvent e) {}
    public void mouseExited(MouseEvent e) {}
    public void mouseDragged(MouseEvent e) {}
    public void mouseMoved(MouseEvent e) {}
    public void mouseWheelMoved(MouseWheelEvent e) {}
}
```

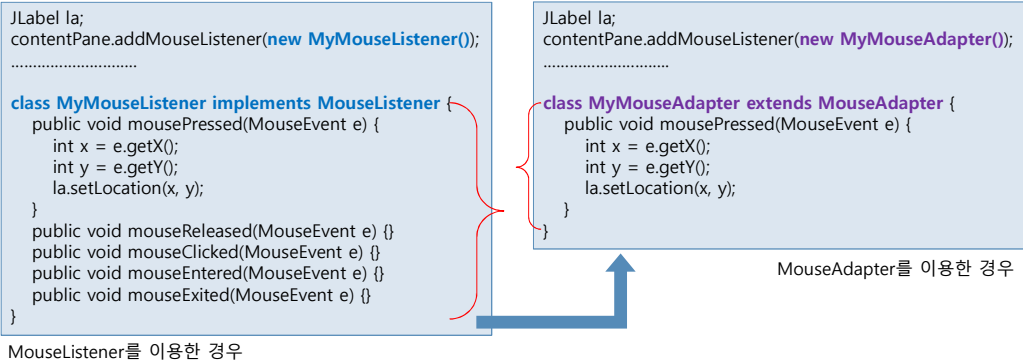
JDK에서 제공하는 어댑터 클래스

10

리스너 인터페이스	대응하는 어댑터 클래스
ActionListener	없음
ItemListener	없음
KeyListener	KeyAdapter
MouseListener	MouseAdapter
MouseMotionListener	MouseMotionAdapter 혹은 MouseAdapter
FocusListener	FocusAdapter
WindowListener	WindowAdapter
AdjustmentListener	없음
ComponentListener	ComponentAdapter
ContainerListener	ContainerAdapter

어댑터 사용 예(MouseAdapter)

11



예제 10-5 : MouseAdapter 사용하기

12

MouseAdapter를 이용하여 예제 10-4를 수정하라.

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

public class MouseAdapterEx extends JFrame {
    private JLabel la = new JLabel("Hello");

    public MouseAdapterEx() {
        setTitle("Mouse 이벤트 예제");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        Container c = getContentPane();
        c.addMouseListener(new MyMouseAdapter());

        c.setLayout(null);
        la.setSize(50, 20);
        la.setLocation(30, 30);
        c.add(la);

        setSize(250, 250);
        setVisible(true);
    }
}

class MyMouseAdapter extends MouseAdapter {
    public void mousePressed(MouseEvent e) {
        int x = e.getX();
        int y = e.getY();
        la.setLocation(x, y);
    }
}

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