

어서와 *Java*는 처음이지!

제10장 이벤트 처리

- 이벤트 분류
- 액션 이벤트
 - 키 이벤트
 - 마우스 이동 이벤트
- 어댑터 클래스

이벤트를 처리한다는 것은
어떤 의미인가요?

이벤트는 사용자가 카를 누르
거나 버튼을 누를 때 발생합니다.
이 이벤트를 처리할 수 있어야
유용한 프로그램이 됩니다.





이벤트의 분류

- 스윙 컴포넌트에 의하여 지원되는 이벤트는 크게 두 가지의 카테고리로 나뉘어진다.



저수준 이벤트:

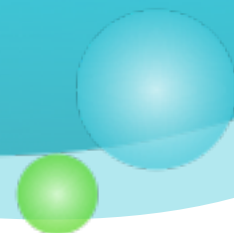
Mouse, MouseMotion, Key, Component, Container, Focus, Window

의미적 이벤트:

Action, Adjustment, Document, Item, Text



저수준 이벤트



이벤트 종류	설명
Component	컴포넌트의 크기나 위치가 변경되었을 경우 발생
Focus	키보드 입력을 받을 수 있는 상태가 되었을 때, 혹은 그반대의 경우에 발생
Container	컴포넌트가 컨테이너에 추가되거나 삭제될 때 발생
Key	사용자가 키를 눌렀을 때 키보드 포커스를 가지고 있는 객체에서 발생
Mouse	마우스 버튼이 클릭되었을 때, 또는 마우스가 객체의 영역으로 들어오거나 나갈 때 발생
MouseMotion	마우스가 움직였을 때 발생
MouseWheel	컴포넌트 위에서 마우스 휠을 움직이는 경우 발생
Window	윈도우에 어떤 변화가 있을 때 발생(열림, 닫힘, 아이콘화등)



의미적 이벤트



이벤트 종류	설 명
Action	사용자가 어떤 동작을 하는 경우에 발생
Caret	텍스트 삽입점이 이동하거나 텍스트 선택이 변경되었을 경우 발생
Change	일반적으로 객체의 상태가 변경되었을 경우 발생
Document	문서의 상태가 변경되는 경우 발생
Item	선택 가능한 컴포넌트에서 사용자가 선택을 하였을 때 발생
ListSelection	리스트나 테이블에서 선택 부분이 변경되었을 경우에 발생



액션 이벤트

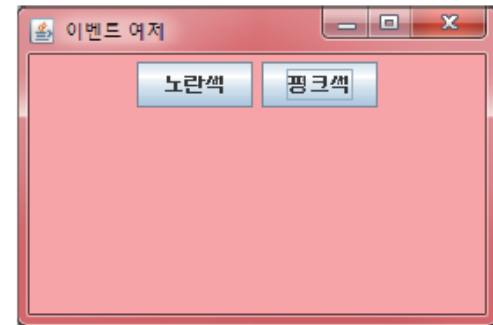
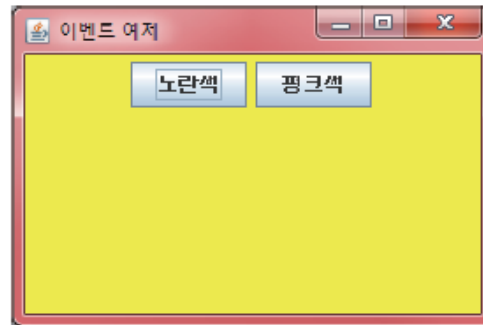
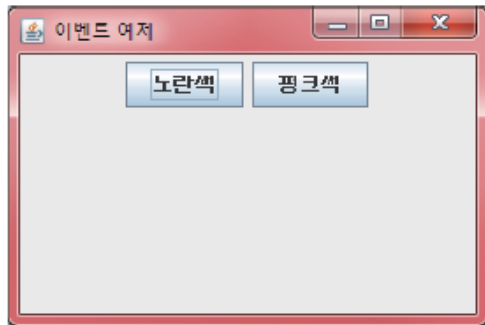
- 사용자가 버튼을 클릭하는 경우
- 사용자가 메뉴 항목을 선택하는 경우
- 사용자가 텍스트 필드에서 엔터키를 누르는 경우





액션 이벤트 예제

- 두 개의 버튼을 만들어서 패널의 배경 색을 변경하는 프로그램을 작성하여 보자.
- 이벤트 리스너는 하나만 생성한다.





액션 이벤트 예제

```
...  
class MyFrame extends JFrame {  
    private JButton button1;  
    private JButton button2;  
    private JPanel panel;  
  
    public MyFrame() {  
        this.setSize(300, 200);  
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        this.setTitle("이벤트 예제");  
        panel = new JPanel();  
        button1 = new JButton("노란색");  
        button1.addActionListener(new MyListener());  
        panel.add(button1);  
        button2 = new JButton("핑크색");  
        button2.addActionListener(new MyListener());  
        panel.add(button2);  
        this.add(panel);  
        this.setVisible(true);  
    }  
}
```



액션 이벤트 예제

```
private class MyListener implements ActionListener {
    public void actionPerformed(ActionEvent e) {
        if (e.getSource() == button1) {
            panel.setBackground(Color.YELLOW);
        } else if (e.getSource() == button2) {
            panel.setBackground(Color.PINK);
        }
    }
}

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



이벤트 발생원의 식별

- getSource() 메소드를 이용하여 이벤트를 발생시킨 객체를 식별한다.
- getId() 메소드를 이용하여 이벤트의 타입을 식별한다.
- getActionCommand() 메소드를 이용하여 이벤트를 발생시킨 컴포넌트 이름을 식별한다.

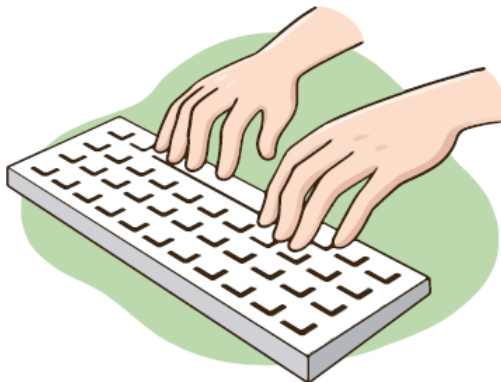
```
public void actionPerformed(ActionEvent e) {  
    if (e.getSource () == button1){  
        ...  
    }  
}
```



Key 이벤트

○ KeyListener 인터페이스 구현

메소드	설 명
<code>keyTyped(KeyEvent e)</code>	사용자가 글자를 입력했을 경우에 호출
<code>keyPressed(KeyEvent e)</code>	사용자가 키를 눌렀을 경우에 호출
<code>keyReleased(KeyEvent e)</code>	사용자가 키에서 손을 떼었을 경우에 호출



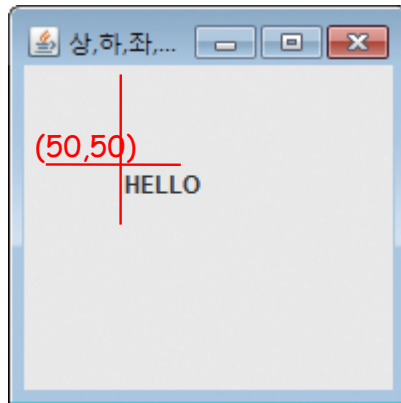
사용자가 키를 누르면 키에
해당되는 문자와 키가 눌렸다는
이벤트가 동시에 보내집니다.



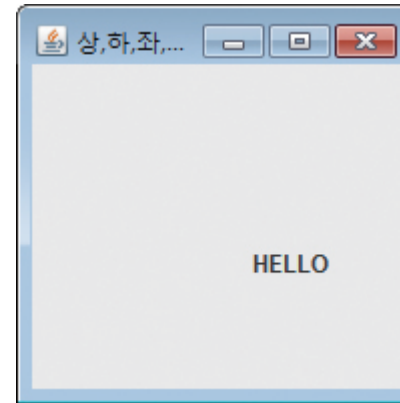


LAB#1: KeyListener 활용

상(↑), 하(↓), 좌(←), 우(→) 키를 입력하면, 다음 그림과 같이 "HELLO" 문자열이 10픽셀씩 이동하는 프로그램을 작성하라.



초기 상태



상, 하, 좌, 우 키를 여러 번 입력하여 "HELLO"를 움직인 상태

상,하,좌,우 키를 움직이면 한 번에 **10**픽셀씩 "HELLO" 텍스트는 상,하,좌,우로 이동한다. 이 텍스트는 프레임의 영역을 벗어나서 움직일 수 있다.



예제 9-7 정답

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

public class FlyingTextEx extends JFrame {
    JPanel contentPane = new JPanel();
    JLabel la = new JLabel("HELLO");

    FlyingTextEx() {
        super("상,하,좌,우 키를 이용하여 텍스트 움직이기");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        setContentPane(contentPane);
        contentPane.setLayout(null);
        contentPane.addKeyListener(new MyKeyListener());

        la.setLocation(50, 50);
        la.setSize(100, 20);
        contentPane.add(la);

        setSize(200, 200);
        setVisible(true);

        contentPane.requestFocus(); // 포커스 지정
    }
}
```



예제 9-7 정답

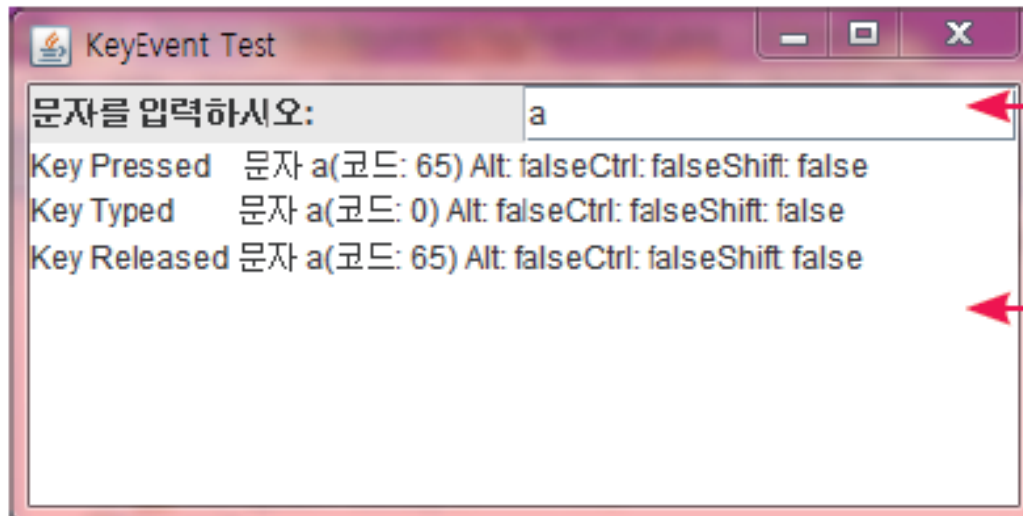
```
class MyKeyListener extends KeyAdapter {
    public void keyPressed(KeyEvent e) {
        int keyCode = e.getKeyCode(); // 입력된 키코드
        switch(keyCode) {
            case KeyEvent.VK_UP:
                la.setLocation(la.getX(), la.getY() - 10); break;
            case KeyEvent.VK_DOWN:
                la.setLocation(la.getX(), la.getY() + 10); break;
            case KeyEvent.VK_LEFT:
                la.setLocation(la.getX() - 10, la.getY()); break;
            case KeyEvent.VK_RIGHT:
                la.setLocation(la.getX() + 10, la.getY()); break;
        }
    }
}

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



LAB#2: 키이벤트 예제

- 키보드에서 문자가 입력되면 문자 코드와 키코드, ALT나 SHIFT 키의 상태를 텍스트 영역에 출력한다.



JTextField

텍스트 필드

JTextArea

텍스트 영역



키 이벤트 예제

○ `keyTyped(KeyEvent e)`

○ `e.isAltDown()`

○ `e.isControlDown()`

○ `e.isShiftDown()`



키 이벤트 예제

```
public class KeyEventTest extends JFrame implements KeyListener {
    private JPanel panel;
    private JTextField field;
    private JTextArea area;
    public KeyEventTest() {
        panel = new JPanel(new GridLayout(0, 2));
        panel.add(new JLabel("문자를 입력하시오: "));
        field = new JTextField(10);
        panel.add(field);
        area = new JTextArea(3, 30);
        add(panel, BorderLayout.NORTH);
        add(area, BorderLayout.CENTER);
        field.addKeyListener(this);
        setTitle("KeyEvent Test");
        setSize(400, 200);
        setVisible(true);
    }
    public static void main(String[] args) {
        new KeyEventTest();
    }
}
```



키 이벤트 예제

```
public void keyTyped(KeyEvent e) { // (3)
    display(e, "Key Typed ");
}
public void keyPressed(KeyEvent e) {
    display(e, "Key Pressed ");
}
public void keyReleased(KeyEvent e) {
    display(e, "Key Released ");
}
protected void display(KeyEvent e, String s) {
    char c = e.getKeyChar();
    int keyCode = e.getKeyCode();
    String modifiers = "Alt: " + e.isAltDown() + "Ctrl: "
        + e.isControlDown() + "Shift: " +
e.isShiftDown();
    area.append("" + s + "문자 " + c + "(코드: " + keyCode + ") " +
modifiers
        + "Wn");
}
}
```



액션 이벤트 예제

```
public void keyTyped(KeyEvent e) { // (3)
    display(e, "KeyTyped ");
}

public void keyPressed(KeyEvent e) {
    display(e, "KeyPressed ");
}

public void keyReleased(KeyEvent e) {
    display(e, "Key Released ");
}

protected void display(KeyEvent e, String s) {
    char c = e.getKeyChar();
    int keyCode = e.getKeyCode();
    String modifiers = e.isAltDown() + " " + e.isControlDown() + " "
        + e.isShiftDown();
    System.out.println(s + " " + c + " " + keyCode + " " + modifiers);
}
}
```



Q & A

