

GUI 프로그래밍 4 이벤트



이벤트 기반 프로그래밍

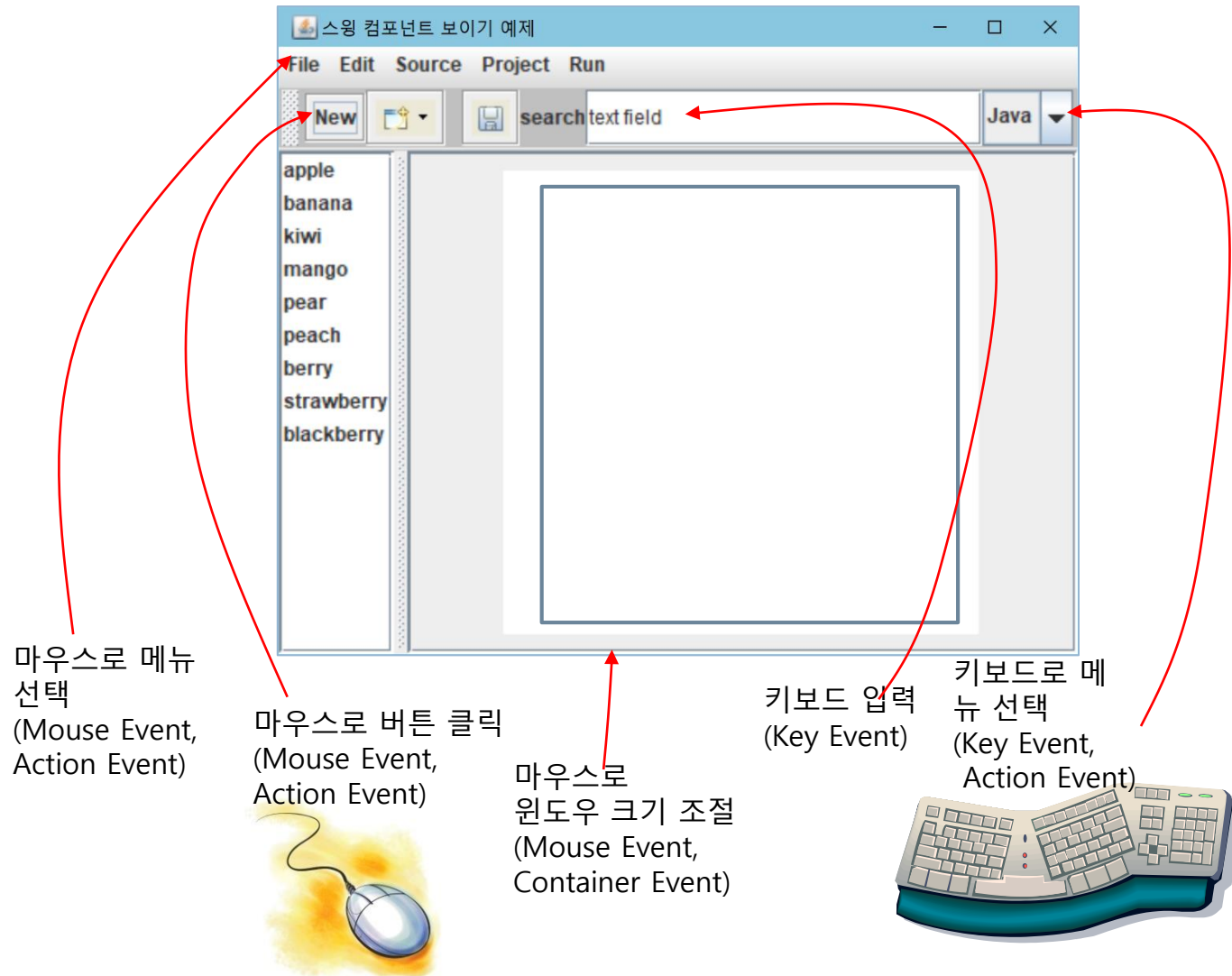
2

- 이벤트 기반 프로그래밍(Event Driven Programming)
 - 이벤트 종류
 - 사용자의 입력 : 마우스 드래그, 마우스 클릭, 키보드 누름 등
 - 센서로부터의 입력, 네트워크로부터 데이터 송수신
 - 다른 응용프로그램이나 다른 스레드로부터의 메시지
 - 이벤트의 발생에 의해 프로그램 흐름이 결정되는 방식
 - 이벤트가 발생하면 이벤트를 처리하는 루틴(이벤트 리스너) 실행
 - 프로그램 내의 어떤 코드가 언제 실행될 지 이벤트 발생에 의해 전적으로 결정
 - 반대되는 개념 : 배치 실행(batch programming)
 - 프로그램의 개발자가 프로그램의 흐름을 결정하는 방식
- 이벤트 기반 프로그램의 구조
 - 이벤트 리스너 들의 집합
- 이벤트 처리 순서
 - 이벤트 발생(예 :마우스나 키보드의 움직임 혹은 입력)
 - 이벤트 객체 생성
 - 현재 발생한 이벤트에 대한 정보를 가진 객체
 - 이벤트 리스너 찾기
 - 이벤트 리스너 호출
 - 이벤트 객체가 리스너에 전달됨
 - 이벤트 리스너 실행



이벤트의 실제 예

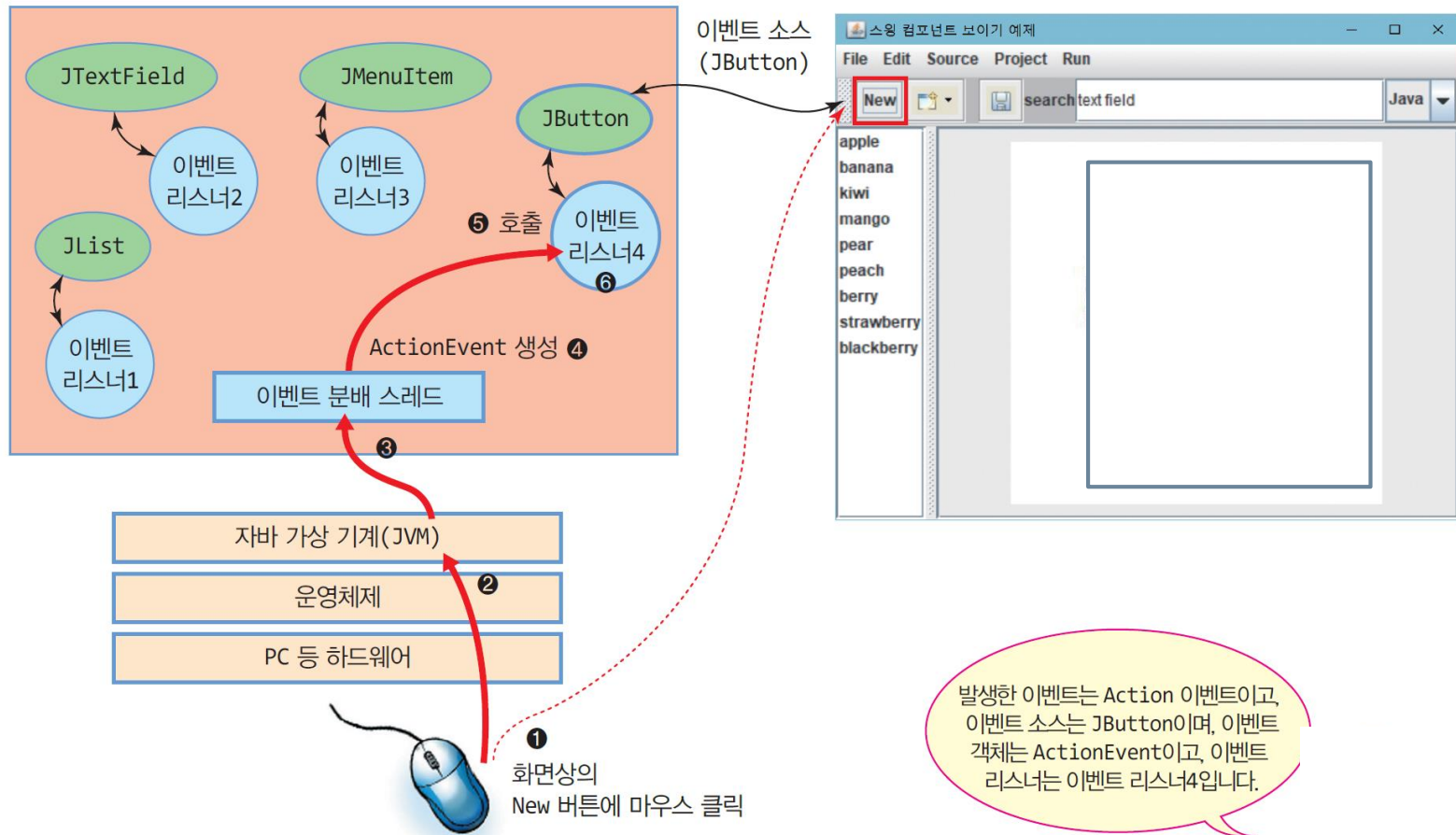
3



자바의 이벤트 기반 GUI 응용프로그램 구성

4

자바 응용프로그램



발생한 이벤트는 Action 이벤트이고, 이벤트 소스는 JButton이며, 이벤트 객체는 ActionEvent이고, 이벤트 리스너는 이벤트 리스너4입니다.



이벤트 관련 용어

5

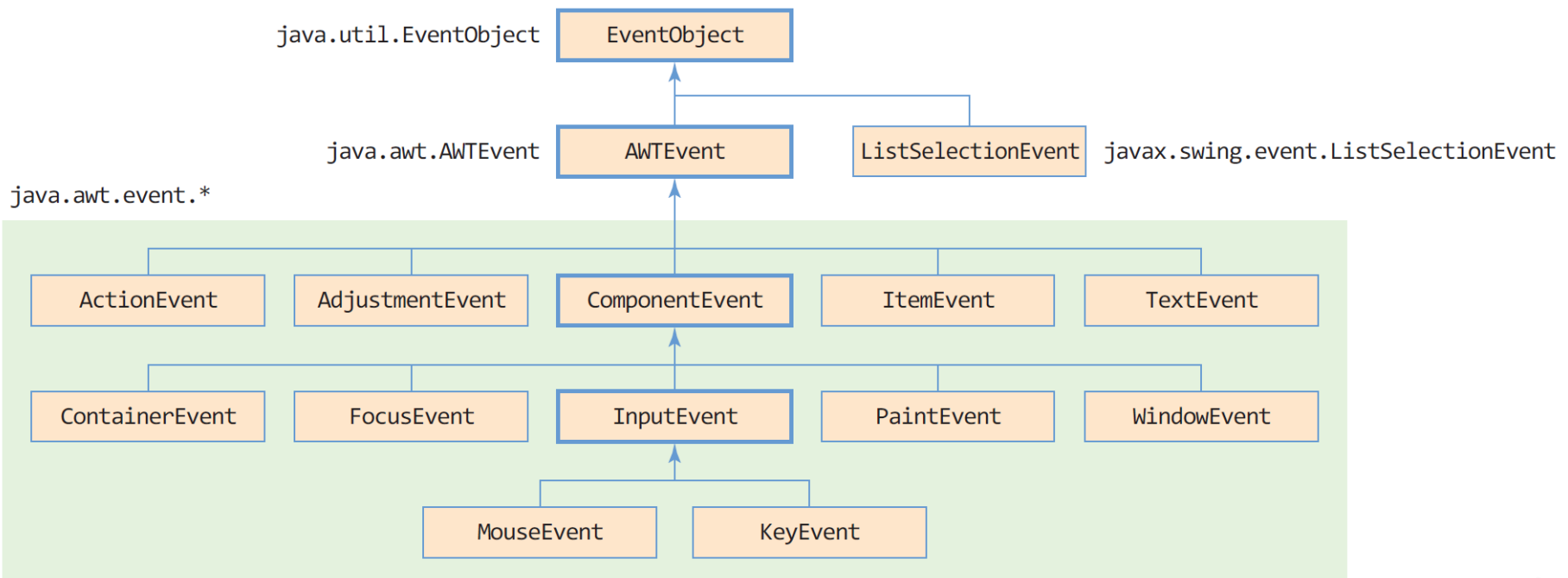
- 이벤트 소스
 - ▣ 이벤트를 발생시킨 GUI 컴포넌트
- 이벤트 객체
 - ▣ 발생한 이벤트에 대한 정보
 - 예) 이벤트 종류, 이벤트 소스, 화면 좌표, 마우스 버튼 종류, 눌려진 키
- 이벤트 리스너
 - ▣ 이벤트를 처리하는 코드
 - ▣ 컴포넌트에 등록되어야 작동 가능
- 이벤트 분배 스레드
 - ▣ 동작
 - 자바 가상 기계로부터 이벤트의 발생을 통지 받음
 - 이벤트 소스와 이벤트 종류 결정
 - 적절한 이벤트 객체 생성, 이벤트 리스너를 찾아 호출
 - ▣ 무한 루프를 실행하는 스레드



이벤트 객체

6

- 이벤트 객체란?
 - 이벤트가 발생할 때, 발생한 이벤트에 관한 정보를 가진 객체
 - 이벤트 리스너에 전달됨
 - 이벤트 리스너 코드에서 이벤트가 발생한 상황을 파악할 수 있게 함
- 이벤트 객체의 종류



이벤트 객체에 포함된 정보

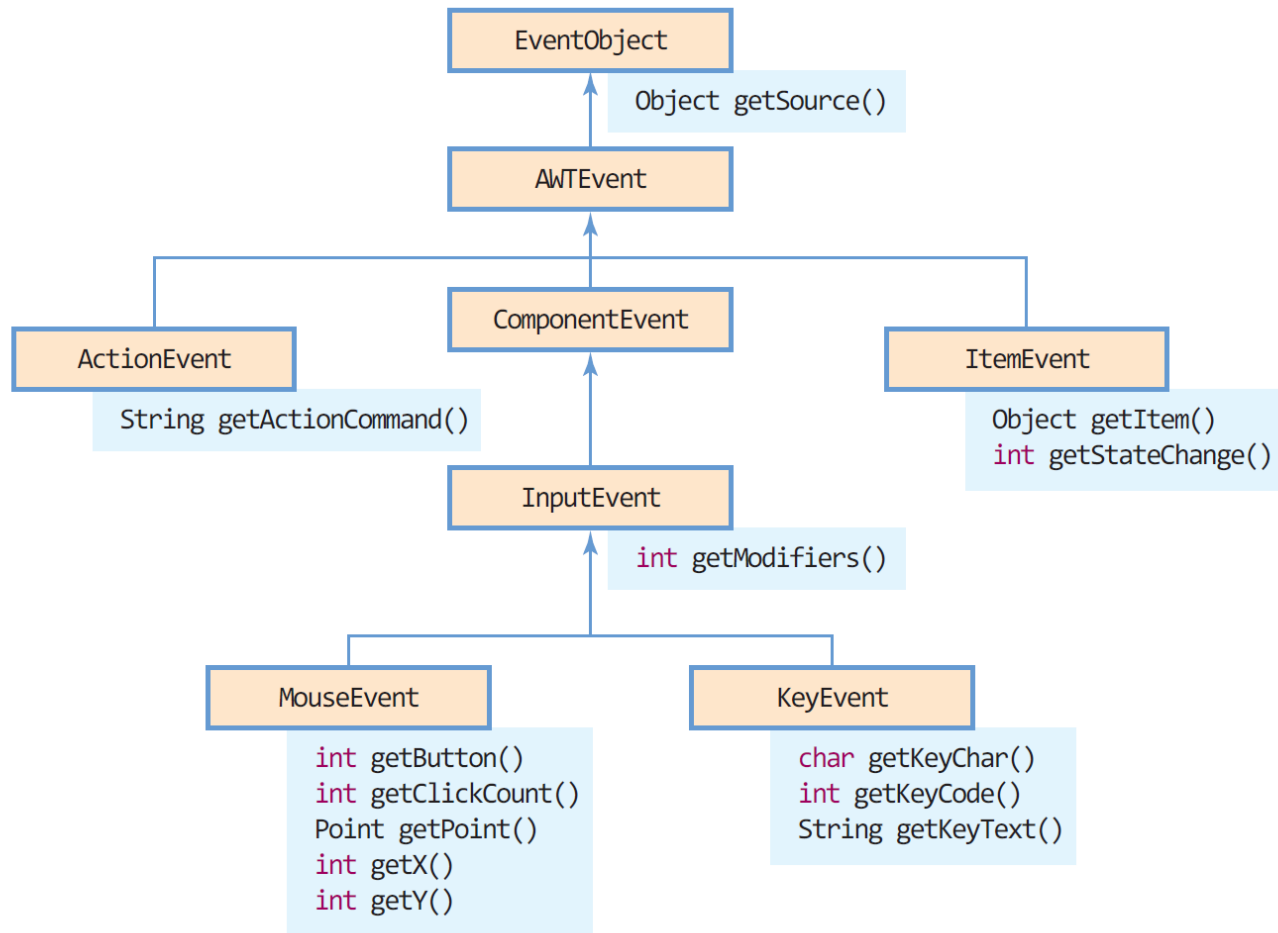
7

- 이벤트 객체가 포함하는 정보
 - ▣ 이벤트 종류
 - ▣ 이벤트 소스
 - ▣ 이벤트가 발생한 화면 좌표
 - ▣ 이벤트가 발생한 컴포넌트 내 좌표
 - ▣ 버튼이나 메뉴 아이템에 이벤트가 발생한 경우 버튼이나 메뉴 아이템의 문자열
 - ▣ 클릭된 마우스 버튼 번호
 - ▣ 마우스의 클릭 횟수
 - ▣ 키가 눌려졌다면 키의 코드 값과 문자 값
 - ▣ 체크박스, 라디오버튼 등과 같은 컴포넌트에 이벤트가 발생하였다면 체크 상태
- 이벤트에 따라 조금씩 다른 정보 포함
 - `ActionEvent` 객체 : 액션 문자열
 - `MouseEvent` 객체 : 마우스의 위치 정보, 마우스 버튼, 함께 눌려진 키 정보 등
 - `ItemEvent` 객체 : 아이템의 체크 상태
- 이벤트 소스 알아 내기
 - ▣ `Object EventObject.getSource()`
 - 발생한 이벤트의 소스 컴포넌트 리턴
 - `Object` 타입으로 리턴하므로 캐스팅하여 사용
 - 모든 이벤트 객체에 대해 적용



이벤트 객체의 메소드

8



이벤트 객체와 이벤트 소스

9

이벤트 객체	이벤트 소스	이벤트가 발생하는 경우
ActionEvent	JButton	마우스나 <Enter> 키로 버튼 선택
	JMenuItem	메뉴 아이템 선택
	TextField	텍스트 입력 중 <Enter> 키 입력
ItemEvent	JCheckBox	체크박스의 선택 혹은 해제
	JRadioButton	라디오버튼의 선택 상태가 변할 때
	JCheckBoxMenuItem	체크박스 메뉴 아이템의 선택 혹은 해제
ListSelectionEvent	JList	리스트에서 선택된 아이템이 변경될 때
KeyEvent	Component	키가 눌러지거나 눌러진 키가 떼어질 때
MouseEvent	Component	마우스 버튼이 눌러지거나 떼어질 때, 마우스 버튼이 클릭될 때, 컴포넌트 위에 마우스가 올라갈 때, 올라간 마우스가 내려올 때, 마우스가 드래그될 때, 마우스가 단순히 움직일 때
FocusEvent	Component	컴포넌트가 포커스를 받거나 잃을 때
WindowEvent	Window	Window를 상속받는 모든 컴포넌트에 대해 윈도우 활성화, 비활성화, 아이콘화, 아이콘에서 복구, 윈도우 열기, 윈도우 닫기, 윈도우 종료
AdjustmentEvent	JScrollBar	스크롤바를 움직일 때
ComponentEvent	Component	컴포넌트가 사라지거나, 나타나거나, 이동, 크기 변경 시
ContainerEvent	Container	Container에 컴포넌트의 추가 혹은 삭제



이벤트 리스너(Event Listener)

10

- 이벤트 리스너란?
 - ▣ 이벤트를 처리하는 코드, 클래스로 작성
- JDK에서 이벤트 리스너 작성을 위한 인터페이스(interface) 제공
 - ▣ 개발자가 리스너 인터페이스의 추상 메소드 구현
 - 이벤트가 발생하면 자바 플랫폼은 리스너 인터페이스의 추상 메소드 호출
 - ▣ 예) ActionListener 인터페이스

```
interface ActionListener { // 아래 메소드를 개발자가 구현해야 함
    public void actionPerformed(ActionEvent e); // Action 이벤트 발생시 호출됨
}
```

- ▣ 예) MouseListener 인터페이스

```
interface MouseListener { // 아래의 5개 메소드를 개발자가 구현해야 함
    public void mousePressed(MouseEvent e); // 마우스 버튼이 눌러지는 순간 호출
    public void mouseReleased(MouseEvent e); // 눌려진 마우스 버튼이 떼어지는 순간 호출
    public void mouseClicked(MouseEvent e); // 마우스가 클릭되는 순간 호출
    public void mouseEntered(MouseEvent e); // 마우스가 컴포넌트 위에 올라가는 순간 호출
    public void mouseExited(MouseEvent e); // 마우스가 컴포넌트 위에서 내려오는 순간 호출
}
```



리스너 인터페이스와 메소드

이벤트 종류	리스너 인터페이스	리스너의 추상 메소드	메소드가 호출되는 경우
Action	ActionListener	void actionPerformed(ActionEvent)	Action 이벤트가 발생하는 경우
Item	ItemListener	void itemStateChanged(ItemEvent)	Item 이벤트가 발생하는 경우
Key	KeyListener	void keyPressed(KeyEvent)	모든 키에 대해 키가 눌러질 때
		void keyReleased(KeyEvent)	모든 키에 대해 눌러진 키가 떼어질 때
		void keyTyped(KeyEvent)	유니코드 키가 입력될 때
Mouse	MouseListener	void mousePressed(MouseEvent)	마우스 버튼이 눌러질 때
		void mouseReleased(MouseEvent)	눌러진 마우스 버튼이 떼어질 때
		void mouseClicked(MouseEvent)	마우스 버튼이 클릭될 때
		void mouseEntered(MouseEvent)	마우스가 컴포넌트 위에 올라올 때
		void mouseExited(MouseEvent)	컴포넌트 위에 올라온 마우스가 컴포넌트를 벗어날 때
Mouse	MouseMotionListener	void mouseDragged(MouseEvent)	마우스를 컴포넌트 위에서 드래그할 때
		void mouseMoved(MouseEvent)	마우스가 컴포넌트 위에서 움직일 때
Focus	FocusListener	void focusGained(FocusEvent)	컴포넌트가 포커스를 받을 때
		void focusLost(FocusEvent)	컴포넌트가 포커스를 잃을 때
ListSelection	ListSelectionListener	void valueChanged(ListSelectionEvent)	JList에 선택된 아이템이 변경될 때
Window	WindowListener	void windowOpened(WindowEvent)	윈도우가 생성되어 처음으로 보이게 될 때
		void windowClosing(WindowEvent)	윈도우의 시스템 메뉴에서 윈도우 닫기를 시도할 때
		void windowIconified(WindowEvent)	윈도우가 아이콘화 될 때
		void windowDeiconified(WindowEvent)	아이콘 상태에서 원래 상태로 복귀할 때
		void windowClosed(WindowEvent)	윈도우가 닫혔을 때
		void windowActivated(WindowEvent)	윈도우가 활성화될 때
		void windowDeactivated(WindowEvent)	윈도우가 비활성화될 때
Adjustment	AdjustmentListener	void adjustmentValueChanged(AdjustmentEvent)	스크롤바를 움직일 때
Component	ComponentListener	void componentHidden(ComponentEvent)	컴포넌트가 보이지 않는 상태로 될 때
		void componentShown(ComponentEvent)	컴포넌트가 보이는 상태로 될 때
		void componentResized(ComponentEvent)	컴포넌트의 크기가 변경될 때
		void componentMoved(ComponentEvent)	컴포넌트의 위치가 변경될 때
Container	ContainerListener	void componentAdded(ContainerEvent)	컴포넌트가 컨테이너에 추가될 때
		void componentRemoved(ContainerEvent)	컴포넌트가 컨테이너에서 삭제될 때



Tip : 리스너 등록 메소드가 addXXXListener인 이유?

12

- 컴포넌트는 다른 이벤트에 대한 리스너를 동시에 가질 수 있다.
 - JButton.addActionListener(); // Action 리스너
 - JButton.addKeyListener(); // Key 리스너
 - JButton.addFocusListener(); // Focus 리스너
- 컴포넌트는 한 이벤트에 대해 여러 개의 리스너를 동시에 가질 수 있다.
 - JButton.addActionListener(new MyButtonListener1());
 - JButton.addActionListener(new MyButtonListener2());
 - JButton.addActionListener(new MyButtonListener3());
 - 이때, 리스너는 등록된 반대 순으로 모두 실행된다.



이벤트 리스너 작성 방법

13

□ 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

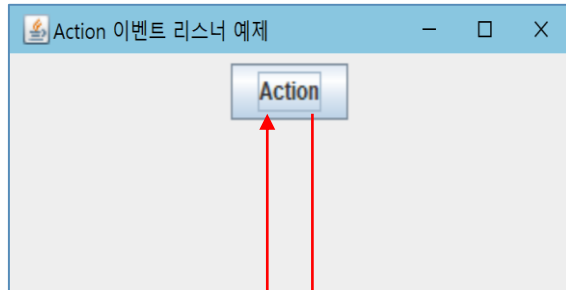
- 클래스 안에 멤버처럼 클래스 작성
- 이벤트 리스너를 특정 클래스에서만 사용할 때 적합

3. 익명 클래스(anonymous class)로 작성

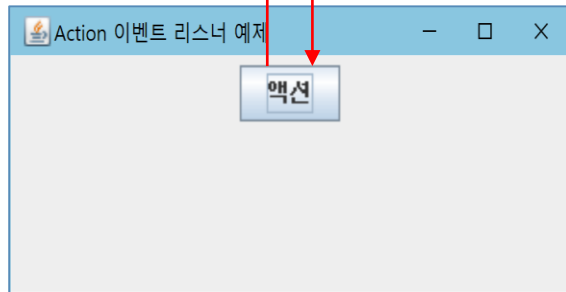
- 클래스의 이름 없이 간단히 리스너 작성
- 클래스 조차 만들 필요 없이 리스너 코드가 간단한 경우에 적합



예제 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);

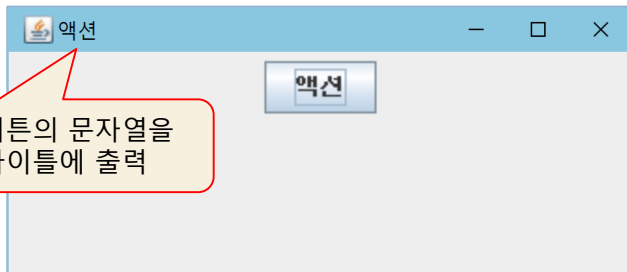
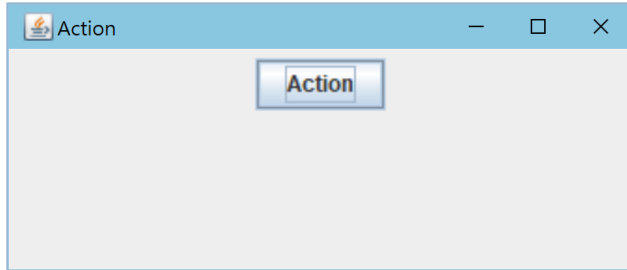
        setSize(350, 150);
        setVisible(true);
    }

    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");
    }
}
```



예제 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);
    }
}
```

```
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();
}
```



익명 클래스로 이벤트 리스너 작성

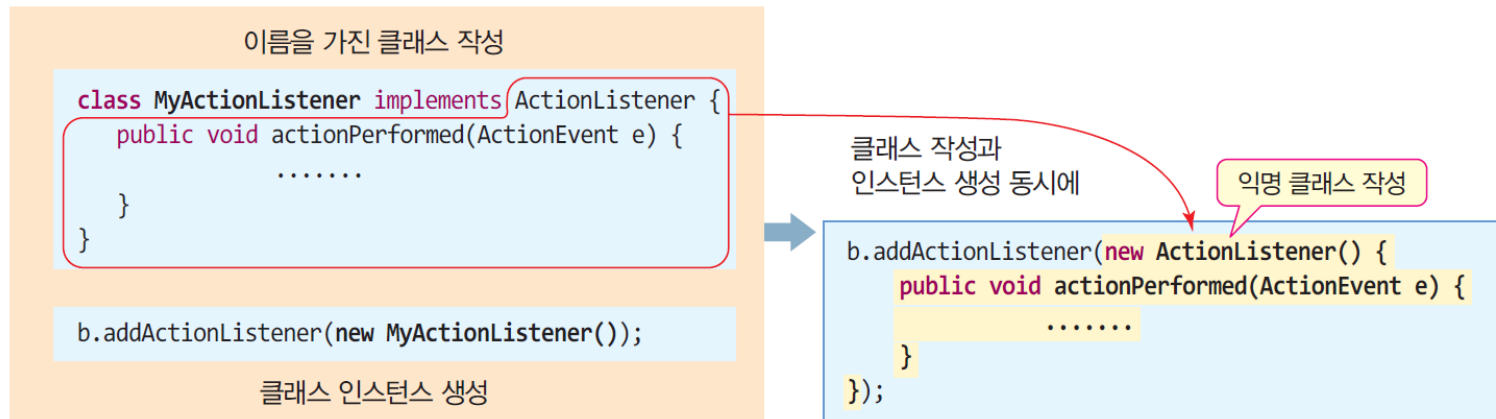
16

□ 익명 클래스란?

- ▣ (클래스 정의 + 인스턴스 생성)을 한번에 작성

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

- ▣ ActionListener를 구현하는 익명의 이벤트 리스너 작성 예



(a) 이름을 가진 클래스를 작성하고
클래스 인스턴스 생성하는 경우

(b) ActionListener를 상속받고 바로 메소드 작성,
동시에 new로 인스턴스를 생성하는 경우



예제 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);
    }

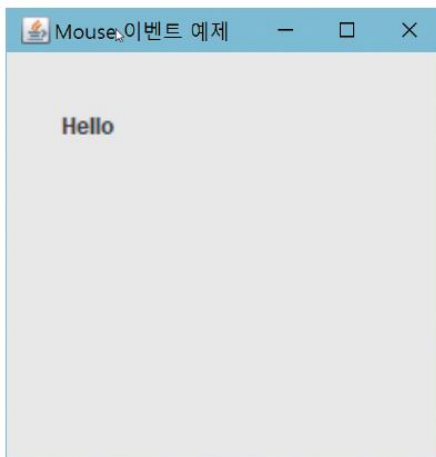
    public static void main(String [] args) {
        new AnonymousClassListener();
    }
}
```



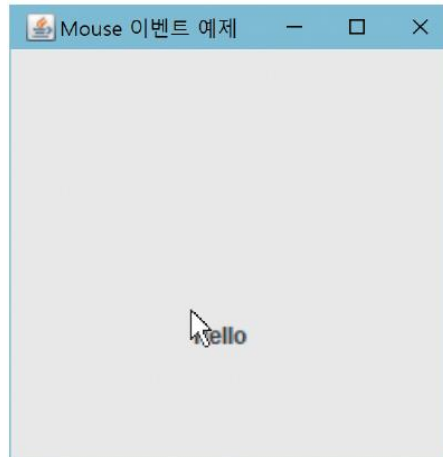
예제 4 : 마우스로 문자열 이동시키기 – 마우스 이벤트 연습

18

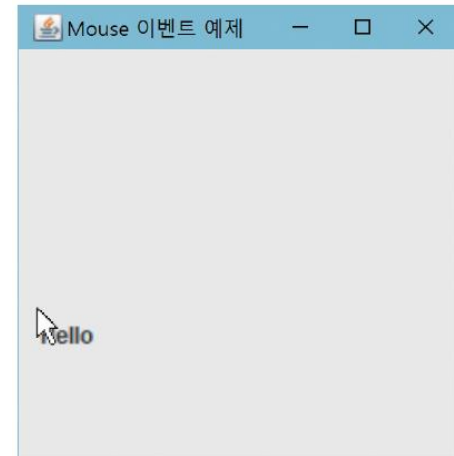
컨텐츠팬의 아무 위치에 마우스 버튼을 누르면 마우스 포인트가 있는 위치로 "hello" 문자열을 옮기는 스윙 응용프로그램을 작성하라.



초기화면



마우스 다른 곳에 클릭한 경우



마우스 다른 곳에 클릭한 경우



예제 4의 정답

19

```
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

20

- 이벤트 리스너 구현에 따른 부담 해소를 위해
 - ▣ 리스너 작성시 추상 메소드들을 모두 구현해야 하는 부담
 - 마우스 리스너에서 마우스가 눌리는 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

```
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) { }  
}
```

- ▣ 추상 메소드가 하나뿐인 리스너는 어댑터 클래스 없음
 - ActionAdapter, ItemAdapter 클래스는 존재하지 않음



JDK에서 제공하는 어댑터 클래스

21

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)

22

```
JLabel la;  
contentPane.addMouseListener(new MyMouseListener());  
.....
```

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) {}  
}
```

MouseListener를 이용한 경우

```
JLabel la;  
contentPane.addMouseListener(new MyMouseAdapter());  
.....
```

class MyMouseAdapter extends MouseAdapter {

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

MouseAdapter를 이용한 경우

