

자바의 이벤트 처리

1. 이벤트 기반 프로그래밍이란?

이벤트의 발생에 의해 프로그램 실행 흐름이 결정되는 방식의 프로그래밍.

2. 이벤트 발생:

- 키 입력, 마우스 클릭, 마우스 드래그 등 사용자의 액션이나, 센서 등 **외부 장치로부터**의 입력 시 발생.
- **네트워크를 통한 다른 컴퓨터로부터의 데이터 수신 시** 발생.
- **다른 스레드나 프로그램으로부터의 메시지 수신 등에 의해** 발생.

3. 이벤트 처리 과정에서 이해 해야 할 요소들?

-이벤트 소스: 이벤트를 발생시킨 GUI컴포넌트임.

예) 버튼, 텍스트필드, 메뉴아이템, 리스트 등.....

-이벤트 객체:

이벤트 발생시 이벤트 종류,

이벤트 소스,

이벤트가 발생한 화면의 좌표,

마우스가 버튼 종류, 눌려진 키의 코드 값 등.....

이벤트에 대한 여러 속성 값을 가진 객체를 말함.

-이벤트 리스너 (Event Listener) :

#각각의 이벤트를 처리하는 이벤트 리스너로..

이벤트가 발생할 때마다 해당 종류의 이벤트 리스너가 실행됨.

#이벤트를 처리하는 클래스 코드로서

컴포넌트 (버튼, 텍스트필드 등..)에 등록되어야 작동 가능함.

-이벤트 분배 스레드:

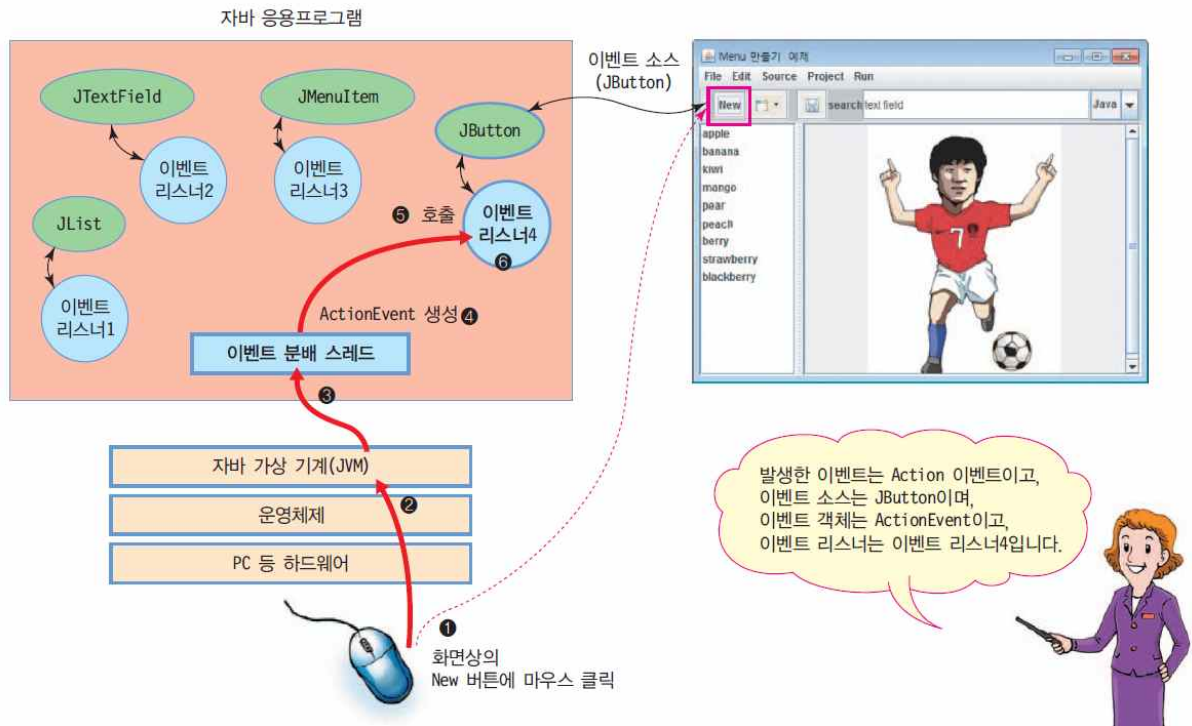
자바가상 기계로부터 이벤트의 발생을 통지 받으면 이벤트 소스와 이벤트 종류를 결정하고, 이에 따라 이벤트 객체를 생성하며, 이벤트 리스너를 찾아 호출한다.

4. Swing 응용프로그램에서의 이벤트 발생사례

이벤트의 실제 예



5. Swing 응용프로그램에서의 이벤트 처리화면 설명.



위그림 설명

컴포넌트 설명 :

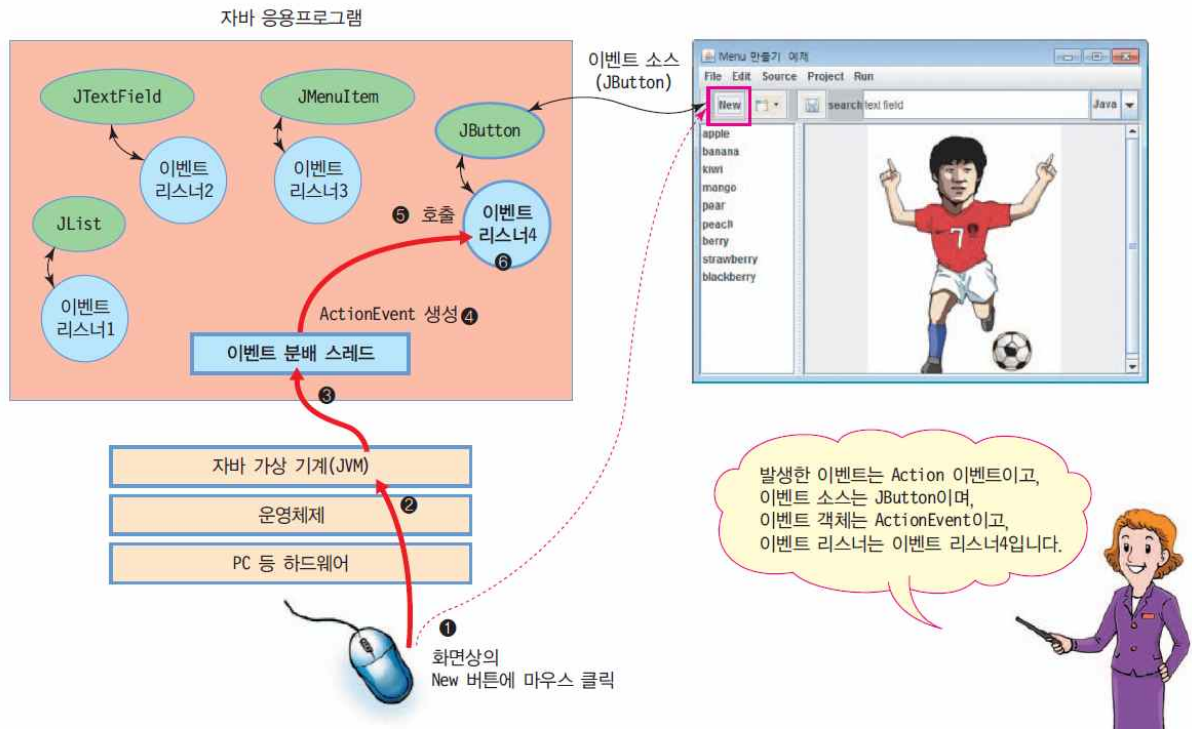
- 오른쪽 화면의 New버튼은 왼쪽 그림의 JButton 컴포넌트가 출력된 모양임.
- 오른쪽 화면의 'apple', 'banana', 'kiwi' 등의 리스트는 왼쪽 그림의 JList 컴포넌트가 출력 모양임.
- 그밖의 컴포넌트가 출력된 모양을 된다.

이벤트 리스너 설명 :

- 이벤트 리스너는 컴포넌트에 발생한 이벤트를 처리하는 프로그램코드로서 컴포넌트에 연결되어 작동 된다.
- 왼쪽 그림은 각컴포넌트들이 이벤트 리스너를 하나씩 가지고 있는 것을 보여 준다.

예) JButton 컴포넌트는 '이벤트 리스너4'의 이름을 가진 이벤트 리스너를 가지고 있다.

6. New 버튼이 클릭되어 처리되는 과정을 통해 이벤트가 발생하여 실행되는 과정을 세밀하게 살펴보기.



순서1) 사용자가 마우스로 화면의 **New** 버튼을 클릭한다.

순서2) 버튼 클릭은 운영체제의 마우스 드라이버를 거쳐 자바 가상 기계에 전달된다.

순서3) 자바 가상 기계는 이벤트 분배 스레드에게 마우스 클릭에 관한 정보를 보낸다.

순서4) 이벤트 분배 스레드는 이벤트 객체를 생성한다.

이벤트를 발생시키게 된 버튼컴포넌트 객체의 주소를

이벤트 객체 내에 저장한다.

순서5) 이벤트 분배 스레드는 이벤트를 처리할 '**이벤트 리스너4**'를 찾고 실행한다.

순서6) 이벤트 분배 스레드는 '**이벤트 리스너4**'로부터 리턴한 후 다음 이벤트 발생을 기다린다.

7. 이벤트 객체 에 대하여...

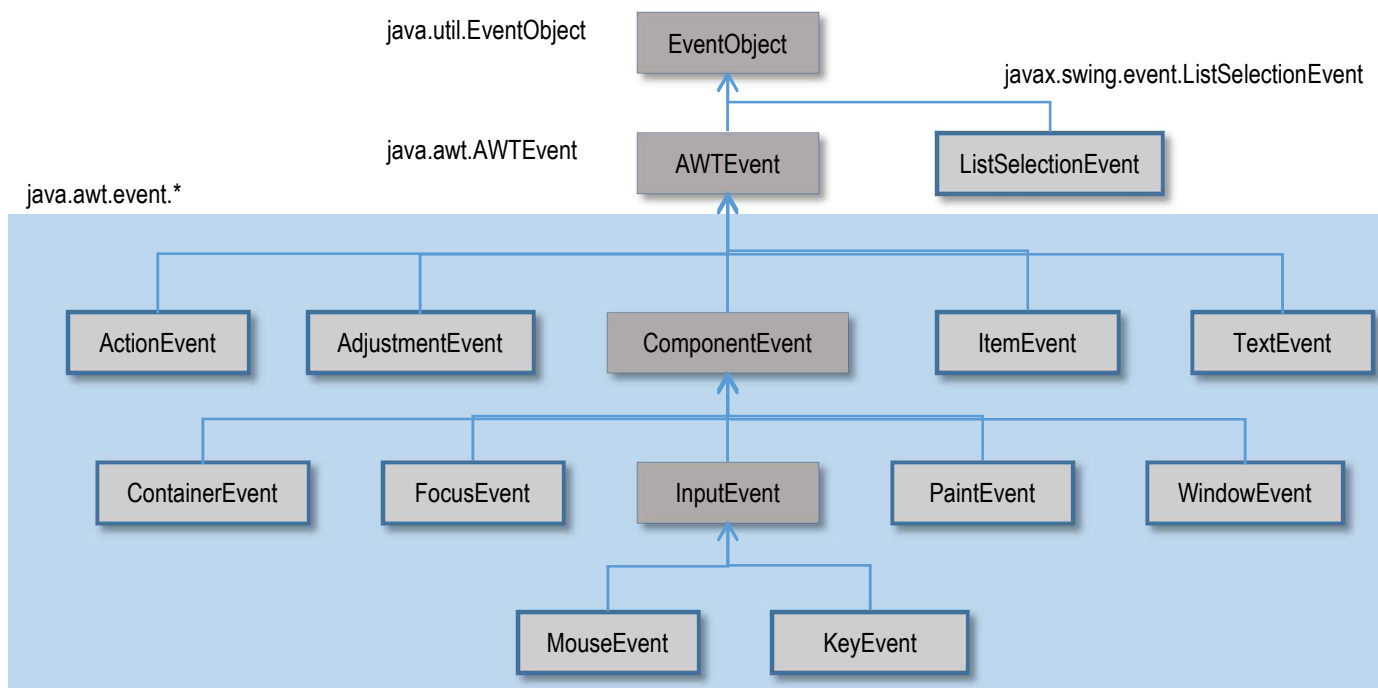
□ 이벤트 객체란?

□ 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 :



8. 이벤트 객체가 포함하는 정보 알아보기

□ 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

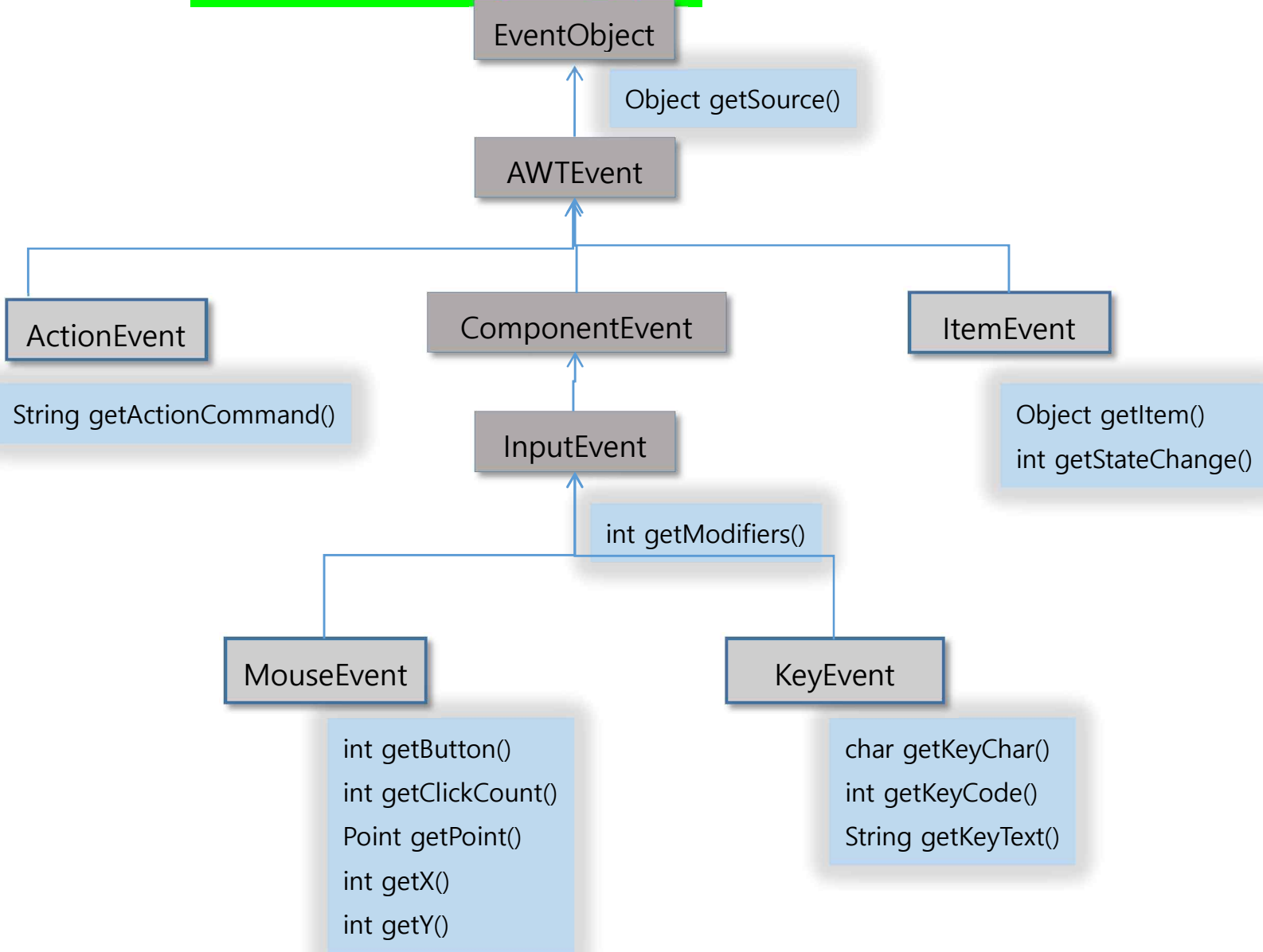
- `Object EventObject.getSource()`

- 발생한 이벤트의 소스 컴포넌트 리턴

- `Object` 타입으로 리턴하므로 캐스팅하여 사용

- 모든 이벤트 객체에 대해 적용

9. 이벤트 객체들의 메소드 알아보기



이벤트 객체는 메소드를 통해 이벤트 정보를 제공한다.

MouseEvent 객체의 메소드 살펴보자

int getButton() : 눌러진 마우스 버튼 알려줌.

int getClickCount() : 마우스 클릭 횟수 알려줌.

Point getPoint() : 마우스 포인터 x 좌표 y 좌표 값을 알려줌.

int getX()

int getY()

모든 이벤트 객체는 EventObject 객체의 getSource()메소드를 공통적으로 상속 받는다.

getSource()메소드 : 이벤트를 발생시키게 된 버튼컴포넌트 객체의 주소를 리턴

10. 이벤트가 발생하는 경우..

이벤트를 발생시킬 이벤트소스정보 (버튼... 등) 들을...

이벤트 객체의 메소드를 통해서 알아 낼 수 있다.

이벤트 객체와 이벤트 소스

이벤트 객체	이벤트 소스	이벤트가 발생하는 경우
ActionEvent	JButton	마우스나 키로 버튼 선택
	JList	리스트 아이템을 더블클릭하여 리스트 아이템 선택
	JMenuItem	메뉴 아이템 선택
	JTextField	텍스트 입력 중 <Enter> 키 입력
ItemEvent	JCheckBox	체크박스의 선택 혹은 해제
	JCheckBoxMenuItem	체크박스 메뉴 아이템의 선택 혹은 해제
	JList	리스트 아이템 선택
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
TextEvent	TextField	텍스트 변경
	TextArea	텍스트 변경
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, 이동하거나, 크기가 변경될 때
ContainerEvent	Container	Container에 컴포넌트의 추가 혹은 삭제

11. 이벤트 리스너..

□ 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

ActionListener 인터페이스는 JDK에 의해 제공되는 리스너 인터페이스이며 실제 다음과 같이 선언되어 있다.

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

설명)

Action 이벤트가 발생하면 ActionListener 인터페이스 내의 actionPerformed(ActionEvent e) 메소드가 호출되어 실행된다. 이때 ActionEvent 객체가 인자로 전달된다.

▣ 예) **MouseListener** 인터페이스

MouseListener 인터페이스는 다음과 같이 선언되어 있다.

```
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
}
```

설명)

5개의 메소드를 가지고 있으며, 각 메소드는 마우스의 각 작동 경우에 발생하는 이벤트를 처리하는 데 이용됨.

모든 메소드의 인자에 MouseEvent 객체가 전달된다.

12. 이벤트 리스너 등록 방법

□ 이벤트 리스너 등록

■ 이벤트를 받아 처리하고자 하는 컴포넌트에 이벤트 리스너 등록

□ 이벤트 리스너 등록 메소드

■ `Component.addXXXListener(listener)`

■ `xxx` : 이벤트 명

■ `listener` : 이벤트 리스너 객체

■ 예) `addMouseListener()`, `addActionListener()`, `addFocusListener()`
등

□ 이벤트 리스너가 등록된 컴포넌트에만 이벤트 전달

■ 이벤트 리스너가 등록된 컴포넌트만 이벤트 리스너 코드 작동

13. 이벤트 처리 방식 4단계

1. 프레임클래스와 다른 이벤트리스너 클래스를 이용 방법.

2. 프레임클래스 자신이 이벤트리스너 클래스역할을 하는 방법

3. 프레임클래스의 `inner`(내부) 클래스 형식으로

이벤트리스너 클래스를 이용 하는 방법.

4. 내부익명 클래스 형식의 이벤트리스너 클래스를 이용 방법.