KH정보교육원

이벤트처리

이벤트처리

KH정보교육원

이벤트 기반 프로그래밍

이벤트의 발생에 의해 프로그램 흐름이 결정된다. 이벤트가 발생하면 이벤트를 처리하는 루틴(이벤트 리스너)이 실행된다.

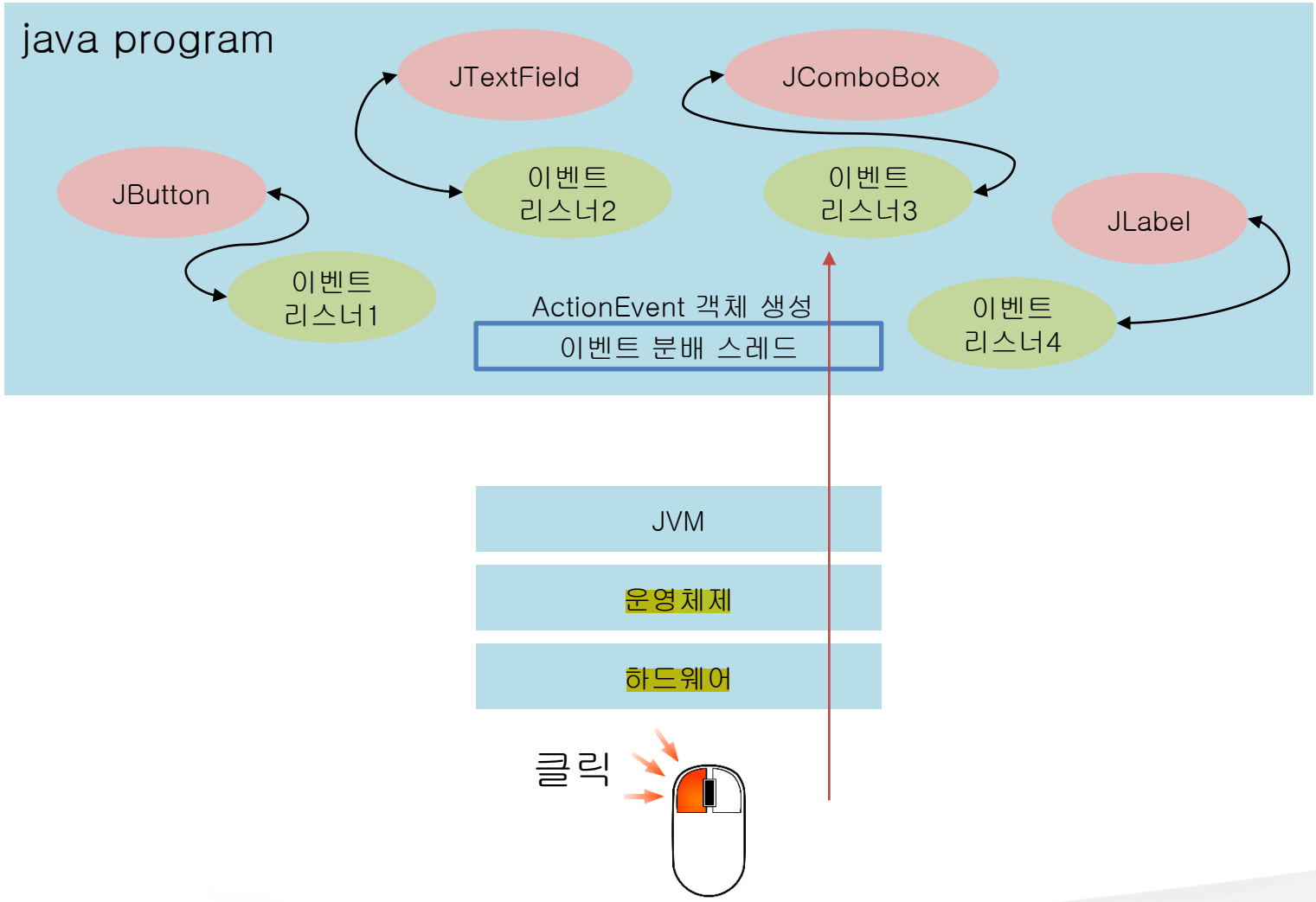
이벤트의 종류

1. 사용자의 입력 : 마우스 드래그, 마우스 클릭, 키보드 누름 등
2. 센서의 입력, 네트워크로부터 데이터 송수신
3. 다른 응용프로그램이나 다른 스레드로부터의 메시지

이벤트처리

KH정보교육원

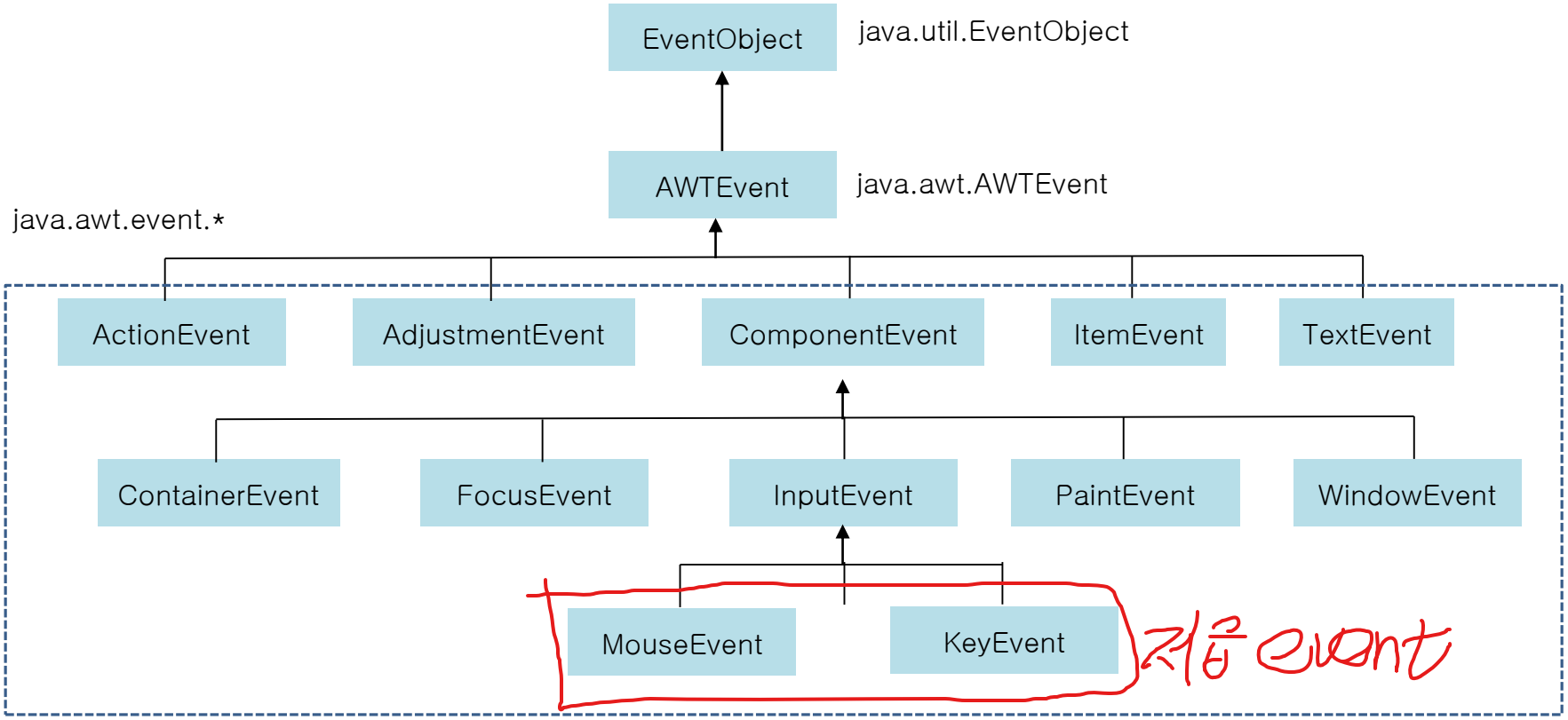
이벤트의 흐름



이벤트처리

KH정보교육원

이벤트 클래스 계층 구조



이벤트처리

KH정보교육원

이벤트객체 포함 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
- 눌러진 키의 코드 값과 문자 값
- 체크박스, 라디오버튼 등과 같은 컴포넌트에 이벤트 발생시,
  체크 상태

이벤트처리

이벤트 객체와 컴포넌트

KH정보교육원

이벤트 객체	컴포넌트	이벤트 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, 클릭할 때, 컴포넌트 위에 마우스가 올라가거나 내려갈 때, 마우스가 드래그될 때, 마우스가 움직일 때
FocusEvent	Component	컴포넌트가 포커스를 받거나 잃을 때
TextEvent	TextField	텍스트가 변경될 때
	TextArea	텍스트가 변경될 때
WindowEvent	Window	Window를 상속받는 모든 컴포넌트에 대해 활성화, 비활성화, 아이콘화, 아이콘에서 복구, 윈도우 열기, 윈도우 닫기, 윈도우 종료 등이 발생할 때
AbjustmentEvent	JScrollbar	스크롤바를 움직일 때
ComponentEvent	Component	컴포넌트가 사라지거나, 나타나거나, 이동하거나, 크기가 변경될 때
ContainerEvent	Container	Container에 컴포넌트의 추가 혹은 삭제할 때

KH정보교육원

이벤트 리스너

클래스로 작성된 이벤트를 처리하는 코드를 말한다. 이벤트 처리를 위해서는 이벤트 발생을 시킬 컴포넌트에 이벤트 리스너를 연결해야 한다.

JDK에서 이벤트 리스너 작성을 위한 인터페이스를 제공한다.

예) MouseListener 인터페이스

```
interface MouseListener {  //5개의 추상 메소드를 제공한다.
    public void mousePress(MouseEvent e);  //마우스 버튼 누르는 순간
    public void mouseReleased(MouseEvent e);  //눌린 버튼 떼는 순간
    public void mouseClicked(MouseEvent e);  //클릭되는 순간
    public void mouseEntered(MouseEvent e);  //컴포넌트 안에 들어갈 때
    public void mouseExited(MouseEvent e);  //컴포넌트 밖에 나갈 때
}
```

KH정보교육원

어댑터(Adapter) 클래스

리스너 인터페이스의 추상 메소드를 모두 구현해야 하는 부담을 줄이기 위해 제공된 클래스이다. 리스너의 모든 메소드가 단순 리턴하도록 구현한 클래스이다.

이벤트처리

KH정보교육원

리스너와 어댑터

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

이벤트처리

KH정보교육원

리스너 연결

컴포넌트 객체에 지정된 이벤트 리스너를 연결하면, 이벤트 리스너가 해당 이벤트를 감지하고 자동으로 이벤트 처리 객체로 연결되어 이벤트가 발생한 객체의 정보를 넘긴다.

예) 컴포넌트레퍼런스.add이벤트명Listener(이벤트핸들러);

이벤트처리용 클래스 방법

1. 별도의 클래스로 작성
2. 내부(Inner) 클래스로 작성

여러번 쓸때

3. 무명(익명 : Anoymous) 클래스로 작성

일회용

4. 프레임 구성 클래스에 인터페이스 상속하여 작성

이벤트처리

KH정보교육원

별도의 클래스로 작성하는 방법

해당 이벤트에 대한 이벤트리스너 인터페이스를 상속받아야 한다.

[표현식]

```
class 클래스명 implements 이벤트명Listener{
        컴포넌트 필드 선언;

        public 생성자(컴포넌트 전달받음){}

        //해당 인터페이스의 추상메소드들 모두 오버라이딩함

        @Override
        public 반환자료형 동작메소드명(이벤트명Event 레퍼런스) {
                //해당 이벤트가 발생한 객체를 골라내서
                        원하는 동작 처리 구문 작성함.
        }
}
```

이벤트처리

KH정보교육원

내부(Inner) 클래스로 작성하는 방법

현재 클래스 안에 이벤트 핸들러 클래스를 작성한다.

[표현식]

```
public class 클래스명 extends JFrame {
        //Field 선언

        //Constructor 작성

        class 이벤트핸들러클래스명 implements 이벤트인터페이스
        {
                @Override
                public 반환자료형 동작메소드명(이벤트클래스명 레퍼런스) {
                        //해당 이벤트가 발생된 객체의 정보를
                        //알아내서 원하는 동작처리 구문 작성함
                }
        }
}
```

이벤트처리

KH정보교육원

무명(익명, Anoymous) 클래스로 작성하는 방법

해당 이벤트에 대한 인터페이스를 상속받지 않는다.

[표현식]

```
컴포넌트레퍼런스.add이벤트Listener(new 이벤트인터페이스명(){
        @Override
        public 반환자료형 동작메소드명(이벤트클래스 레퍼런스){
                //해당 컴포넌트에 대한 동작 처리 구문 작성함
        }

        /** 사용하지 않는 메소드에 대해서도 다 재구현해야 함 **/
});
```

프레임 구성 클래스에 인터페이스 상속받는 방법

해당 이벤트에 대한 인터페이스를 상속받지 않는다.

[표현식]

```
public class 클래스명 extends JFrame implements 이벤트인터페이스 {
        //컴포넌트에 대한 필드 선언
        //생성자 작성
        public 생성자() {
                //프레임 구성, 컴포넌트 생성 및 배치
                //컴포넌트에 이벤트리스너 연결하기
                컴포넌트레퍼런스.add이벤트명Listener(this);
        }
        //상속받은 이벤트인터페이스의 추상메소드 오버라이딩함
        @Override
        public 반환자료형 동작메소드명(이벤트클래스명 레퍼런스) {
                //해당 이벤트가 발생한 객체를 골라내서 원하는
        }

}
```