클릭하세요 자바

! 7 4

최 문 환

26장 컴포넌트의 이벤트 처리

1. 이벤트 처리 모델
2. 이벤트 처리
3. 리스너 인터페이스와 어댑터 클래스

1. 이벤트 처리 모델

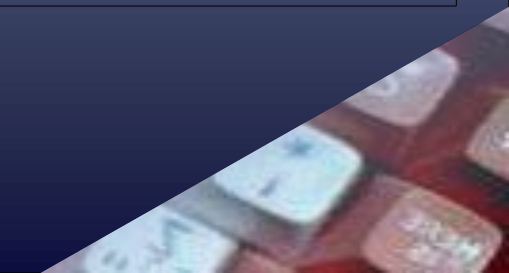
이벤트(event)란?
프로그램과 사용자간의 상호작용을 위해서 사용자가 키보드나 마우스 등의 장치를 통해서 응용 프로그램에 어떤 요구를 하는 사건을 말합니다.

1. 이벤트 처리 모델

EVENTLISTENE

이벤트 소스(event source)란?

버튼(Button)이나 스크롤바(Scrollbar)와 같이 이벤트가 발생한 컴포넌트 객체를 말합니다.



. O

2. 이벤트 처리

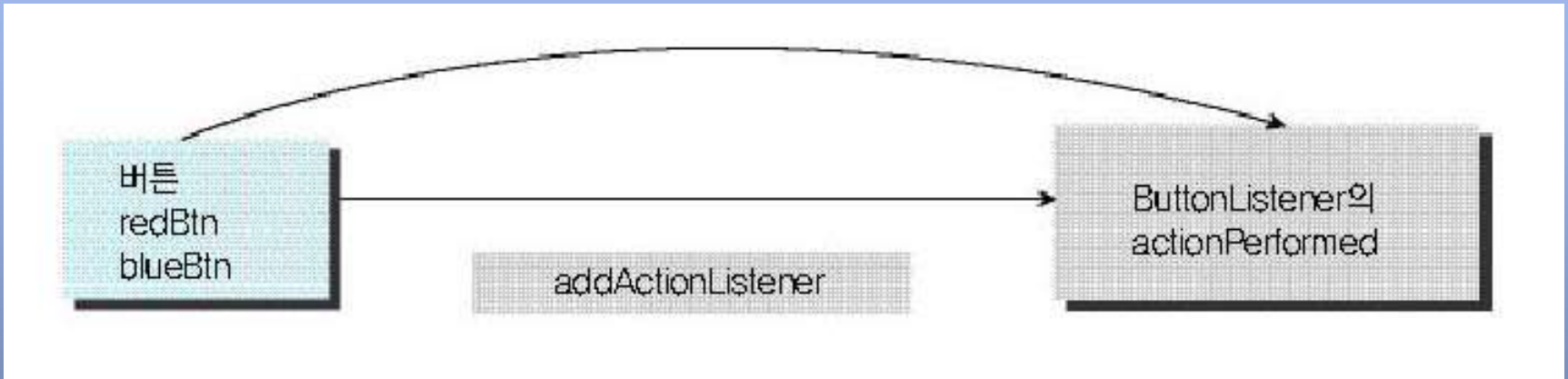
```
                                                ;                VENT4EST    AVA=

IMPO T AVA AWT

IMPO T AVA AWTEVENT

CLASS   AME  VENTE  TENDS   AME [
                                                                 AME  VENT
  "   TTON  ED" TN  BL E" TN

  P  BLLC   AME  VENT   [
    SET, A  O  T NEW   LOW, A  O  T


     ED" TN    NEW "   TTON
    BL E" TN    NEW "   TTON
    ADD  ED" TN
    ADD BL E" TN


   "   TTON, ISTENE    ANDLE    NEW "   TTON, ISTENE

    ED" TN ADD!  CTON, ISTENE   ANDLE
   BL E" TN ADD!  CTON, ISTENE   ANDLE
```

. O

2. 이벤트 처리

```
024:     setSize(300,200);
025:     setVisible(true);
026:
027:     addWindowListener(new WindowAdapter( ){
028:       public void windowClosing(WindowEvent e) {
029:           dispose();
030:           System.exit(0);
031:       }//windowClosing 메서드 끝
032:     }//내부 무명 클래스 정의 끝
033:     );//addWindowListener 메서드 끝
034:   }//FrameEvent  생성자 끝
035:}//FrameEvent  프레임 클래스 끝
036:
037:class ButtonListener implements ActionListener{
038:   public void actionPerformed(ActionEvent e) {
039:     System.out.println("버튼이 눌렸습니다.");
040:   } //actionPerformed 메서드 끝
041:}//ButtonListener 리스너 클래스 끝
042:
043:public class EventTest01{
044:   public static void main(String[] args) {
045:    new FrameEvent( );
046:   }
047:}
```

2. 이벤트 처리


버튼
redBtn
blueBtn
addActionListener
ButtonListener의
actionPerformed

2.1 ActionEvent 클래스

<예제> 버튼이 눌리면 프레임의 배경 색을 변경-[파일 이름 : EventTest02java]

```
001:import java.awt.*;
002://1. 이벤트 클래스 사용하기 위해서 java.awt.event 패키지를 import
003:import java.awt.event.*;
004:
005:class FrameEvent extends Frame {
006:  //이벤트를 받아들인 컴포넌트(버튼) 객체는 FrameEvent의 멤버변수
007:  Button redBtn, blueBtn;
008:
009:  public FrameEvent( ) {
010:    setLayout(new FlowLayout());  //플로우 레이아웃을 배치관리자 지정
011:
012:    //이벤트를 받아들인 컴포넌트 객체(여기서는 버튼 객체) 생성
013:    redBtn = new Button("빨간 색");
014:    blueBtn = new Button("파란 색");
015:    add(redBtn);  //버튼 객체를 프레임에 등록
016:    add(blueBtn); //배치는 배치관리자가 한다.
. ○
```

2.1 ActionEvent 클래스

```
018:     //3. 이벤트 리스너 객체를 생성
019:     ButtonListener  handler = new ButtonListener(this);
020:     //4. 이벤트를 받아들인 컴포넌트 객체(여기서는 버튼 객체)에 리스너를 등록
021:     redBtn.addActionListener(handler);
022:     blueBtn.addActionListener(handler);
023:
024:     setSize(300,200);
025:     setVisible(true);
026:
027:     addWindowListener(new WindowAdapter( ){
028:       public void windowClosing(WindowEvent e) {
029:           dispose();
030:           System.exit(0);
031:       }//windowClosing 메서드 끝
032:     }  //클래스 정의 끝
033:     );  //addWindowListener 메서드 끝
034:   }//생성자 끝
035:}//프레임 클래스 끝
```

2.1 ActionEvent 클래스

```
037:class ButtonListener implements  ActionListener{
038:   Frame  frm=null;
039:   public ButtonListener(){
040:   }
041:   public ButtonListener(Frame value){
042:     frm=value;
043:   }
044:   public void actionPerformed(ActionEvent e){
045:     if(e.getActionCommand( )=="빨간 색")
046:       frm.setBackground(Color.red);
047:     else //파란 버튼이 눌리면 프레임의 배경 색을 파란색으로
048:       frm.setBackground(Color.blue);
049:     }
050:};
051:
052:public class EventTest02{
053:   public static void main(String[] args) {
054:    new FrameEvent( );
055:   }
056:}
```

. ○

2.3 ActionEvent의 getSource( ) 메서드

<예제> 버튼이 눌리면 프레임의 배경색 변경-[파일 이름 : EventTest03.java]

```
001:import java.awt.*;
002://1. 이벤트 클래스 사용하기 위해서 java.awt.event 패키지를 import
003:import java.awt.event.*;
004:
005://2. 이벤트 소스를 포함하고 있는 클래스 자체가 이벤트 리스너가 되도록 설계
006:class FrameEvent extends Frame implements ActionListener{
007:   //이벤트를 받아들인 컴포넌트(버튼) 객체는 FrameEvent의 멤버
008:   Button redBtn, blueBtn;
009:
010:   public FrameEvent( ) {
011:     setLayout(new FlowLayout());
012:
013:     //이벤트를 받아들인 컴포넌트 객체(여기서는 버튼 객체) 생성
014:     redBtn = new Button("빨간 색");
015:     blueBtn = new Button("파란 색");
016:     add(redBtn);  // 프레임에 등록
017:     add(blueBtn);
```

ED" TN ADD! CTON, ISTENE T IS AME VENT

BL E" TN ADD! CTON, ISTENE T IS T IS

SET3I E

SET6ISIBLE T E

. O

2.3 ActionEvent의 getSource( ) 메서드

```
    ADD7 INDOW, ISTENE  NEW 7 INDOW! DAPTE   [
      P BLLC VOD WINDOW LOSIN  7 INDOW VENTE [
        DISPOSE
        3 STEM E IT
      ]  WINDOW LOSIN
    ]
        ADD7 INDOW, ISTENE
  ]

                                  ACTON0E FO MED
  P BLLC VOD ACTON0E FO MED ! CTON VENTE [
      ET3O  OE
    IF E  ET3O  OE       ED" TN
      T IS SET" AC   O ND  OLO ED
    ELSE IF E  ET3O  OE      BL E" TN
      T IS SET" AC   O ND  OLO BL E
  ]  ACTON0E FO MED
]    AME VENT

P BLLC CLASS  VENT4EST [
  P BLLC STATC VOD M AIN 3TIN ;=A  S [
    NEW  AME VENT
  ]
]
```

. O

3. 리스너 인터페이스와 어댑터 클래스

리스너 인터페이스를 구현해 놓은 클래스에서는 리스너 인터페이스의 추상 메서드를 모두 오버라이딩해서 메서드의 몸체를 구현해야 합니다.

P BLIC VOID WINDOW! CTIVATED 7 INDOW VENTE P BLIC VOID WINDOW LOSED 7 INDOW VENTE P BLIC VOID WINDOW LOSIN 7 INDOW VENTE P BLIC VOID WINDOW EACTIVATED 7 INDOW VENTE P BLIC VOID WINDOW EICONIFIED 7 INDOW VENTE P BLIC VOID WINDOW)CONIFIED 7 INDOW VENTE P BLIC VOID WINDOW/ PENED 7 INDOW VENTE	

. O

<예제> 프레임 창 닫기 위한 이벤트 처리하기

```
<예제>-프레임 창 닫기 위한 이벤트 처리하기-[파일 이름 : EventTest04.java]
001:import java.awt.*;
002://1. 이벤트 클래스 사용하기 위해서 java.awt.event 패키지를 import
003:import java.awt.event.*;
004://2. 이벤트 소스를 포함하고 있는 클래스 자체가 이벤트 리스너가 되도록 설
005:class ExitEventFrame extends Frame implements WindowListener{
006:  public ExitEventFrame ( ){ //생성자
007:
008:    //이벤트 소스와 리스너 객체 모두 프레임임
009:    this.addWindowListener(this);
010:    setSize(200,200);
011:    setVisible(true);
012:  } //생성자의 끝
013:
014:  public void windowClosing(WindowEvent e){
015:    dispose( );     //프레임 창을 닫는다.
016:    System.exit(0);//프로그램을 종료한다.
017:  }
. ○
```

<예제> 프레임 창 닫기 위한 이벤트 처리하기

```
  P  BLLC VOLD WNDOW! CTVATED 7 NDOW VENTE [ ]
   P  BLLC VOLD WNDOW LOSED 7 NDOW VENTE [ ]
  P  BLLC VOLD WNDOW EACTVATED 7 NDOW VENTE [ ]
  P  BLLC VOLD WNDOW EICONFED 7 NDOW VENTE [ ]
  P  BLLC VOLD WNDOW)CONFED 7 NDOW VENTE [ ]
  P  BLLC VOLD WNDOW/ PENED 7 NDOW VENTE [ ]
]
P  BLLC  CLASS  VENT4EST  [
  P  BLLC STATC VOLD MAN  3TN  A  S; = [
    NEW   T VENT  AME
  ]
]
```

. O

3.1 어댑터 클래스

IMPLEMENTS

. O

<예제> 프레임 창 닫기 위한 이벤트 처리를 어댑터 클래스로 하기

```
IMPO T AVA AWT

IMPO T AVA AWTEVENT
    AME                    IT  AME  VENT
CLASS   IT  AME  VENTE TENDS   AME[
 P BLLC   IT  AME  VENT  [
  SET3I E
  SET6ISIBLE T  E
                                    7 INDOW! DAPTE

  ADD7 INDOW, ISTENE   NEW 3  B  LASS
 ]
]    IT  AME  VENT
 7 INDOW! DAPTE                 E TENDS
```

<예제> 프레임 창 닫기 위한 이벤트 처리를 어댑터 클래스로 하기

```
014:class SubClass extends WindowAdapter{
015:    //[닫기]버튼이 눌렸을 때 호출되는 메서드 오버라이딩
016:   public void windowClosing(WindowEvent e) {
017:     //dispose( );   //자원 해제
018:     System.exit(0); //프로세스를 강제 종료함, 윈도를 종료하기 위한 코딩
019:   }//windowClosing 메서드 끝
020:}//SubClass클래스 정의 끝
021:
022:public class EventTest05{
023:   public static void main(String[] args) {
024:     new ExitFrameEvent( );
025:   }
026:}
```

. ○

3.3 내부 무명 클래스

```
IMPO T AVA AWT

IMPO T AVA AWTEVENT
    AME                    IT AME VENT
CLASS   IT AME VENTE TENDS   AME[
 P BLIC   IT AME VENT  [
  SET3I E
  SET6ISIBLE T E

  ADD7 INDOW,ISTENE NEW7 INDOW! DAPTE   [
     ;     =
   P BLIC VOID WINDOW LOSIN 7 INDOW VENTE [
    DISPOSE
    3 STEM E IT
   ]  WINDOW LOSIN
  ]
      ADD7 INDOW,ISTENE
 ]
]

P BLIC CLASS  VENT4EST  [
 P BLIC STATIC VOID MAIN 3TIN ;=A  S [
  NEW   IT AME VENT
 ]
]
```

. O