응용 애플리케이션 개발을 위한 자바 프로그래밍

이벤트 이해하기

한국기술교육대학교
온라인평생교육원

학습내용

- 이벤트 개요
- 이벤트 구현하기

학습목표

- 이벤트 처리 방법을 이해하고 이벤트와 관련된 클래스와 인터페이스들을 설명할 수 있다.
- 이벤트를 처리하는 방법을 이해하고 다양한 방법으로 이벤트를 구현할 수 있다.

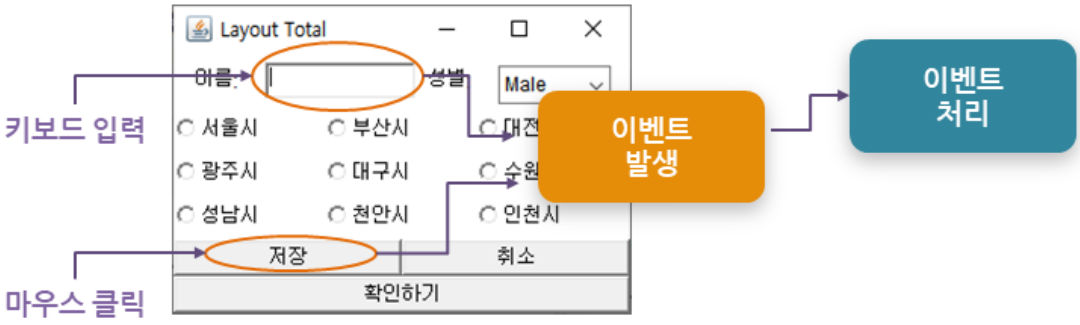
이벤트 개요

1 이벤트(Event)의 이해

① 이벤트란?

1 사용자와 프로그램의 상호작용에 발생하는 사용자의 행동

2 키보드, 마우스등의 동작에 따라 컴포넌트들에게 발생됨



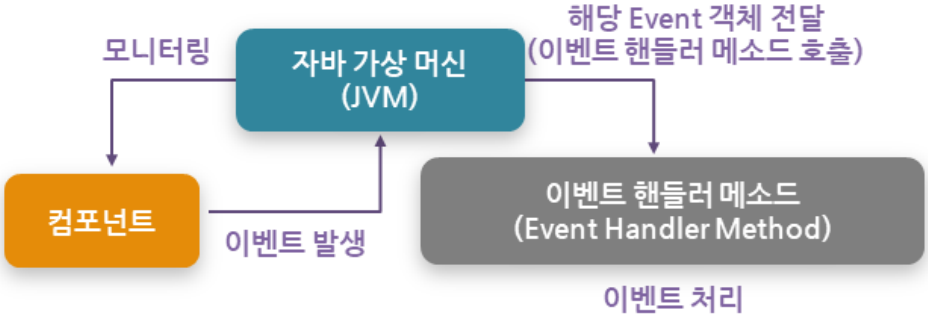
1 이벤트(Event)의 이해

② 이벤트 처리 방법

4 Event 객체를 이벤트 핸들러 메소드(Event Handler Method)에 인수로 전달

이벤트 핸들러 메소드 호출

5 이벤트 핸들러 메소드에서 이벤트를 처리



이벤트 개요

1 이벤트(Event)의 이해

〈3〉 이벤트의 종류

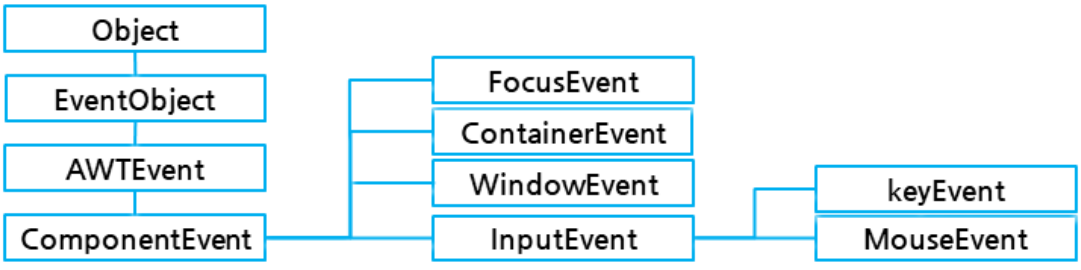
2 하이 레벨(high-level) 이벤트

- 의미적(semantic) 이벤트
- 컴포넌트의 종류에 따라 발생될 수 있는 이벤트
- 버튼 클릭, 메뉴 또는 항목 선택, 스크롤바 조정 등

1 이벤트(Event)의 이해

〈4〉 이벤트 클래스 구조

1 로우 레벨(low-level) 이벤트 클래스



이벤트 개요

1 이벤트(Event)의 이해

④ 이벤트 클래스 구조

1 로우 레벨(low-level) 이벤트 클래스

로우 레벨 이벤트 클래스	이벤트 발생되는 경우
ComponentEvent	컴포넌트의 이동, 크기변경, 화면 표시, 화면 숨기기
FocusEvent	컴포넌트가 포커스를 획득하거나 상실
ContainerEvent	컨테이너에 컴포넌트가 추가되거나 삭제

1 이벤트(Event)의 이해

④ 이벤트 클래스 구조

1 로우 레벨(low-level) 이벤트 클래스

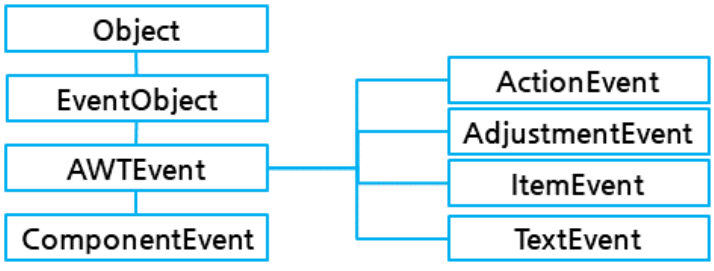
로우 레벨 이벤트 클래스	이벤트 발생되는 경우
WindowEvent	윈도우 창의 활성화/비활성화, 닫기/열기, 최소화/최대화 등
KeyEvent	키보드의 키 입력 동작
MouseEvent	마우스의 클릭 또는 이동 등의 동작

이벤트 개요

1 이벤트(Event)의 이해

⟨4⟩ 이벤트 클래스의 구조

2 하이 레벨(high-level) 이벤트 클래스



1 이벤트(Event)의 이해

⟨4⟩ 이벤트 클래스의 구조

2 하이 레벨(high-level) 이벤트 클래스

하이 레벨 이벤트 클래스	이벤트 발생되는 경우
ActionEvent	버튼 클릭, 메뉴 선택, 엔터키 입력 등
AdjustmentEvent	스크롤바의 값 조정 등
ItemEvent	항목 선택/해제 등
TextEvent	입력된 값의 변동

이벤트 개요

2 이벤트 관련 클래스와 인터페이스

〈1〉 패키지 : java.awt.event

〈2〉 이벤트 핸들러(Event Handler) 클래스란?

1 이벤트 핸들러 메소드를 가지고 있는 클래스

이벤트 핸들러 메소드는 이벤트 종류에 따라서 정해져 있음

2 컴포넌트 생성 후 이벤트 핸들러 클래스의 객체를 지정

이벤트 발생시 처리할 이벤트 핸들러 클래스의 객체 지정

2 이벤트 관련 클래스와 인터페이스

〈1〉 패키지 : java.awt.event

〈2〉 이벤트 핸들러(Event Handler) 클래스란?

3 이벤트 핸들러 클래스 작성 방법

1) Listener 인터페이스를 구현하는 방법

2) Adapter 클래스를 상속하는 방법

이벤트 개요

2 이벤트 관련 클래스와 인터페이스

〈3〉 Listener 인터페이스

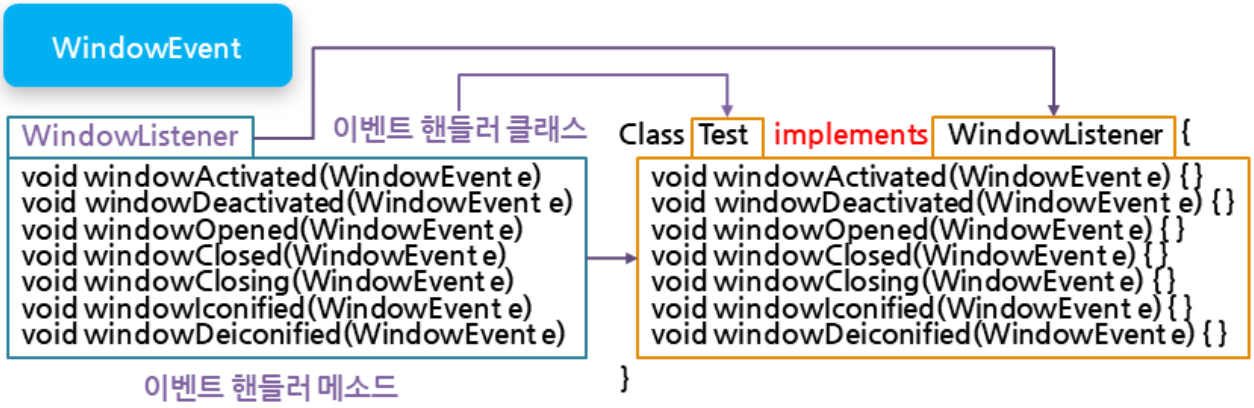
1 이벤트 종류별로 이벤트 핸들러 메소드가 정의되어 있는 인터페이스

2 Listener 인터페이스를 구현한 클래스는 이벤트 핸들러 클래스가 됨

3 이벤트 핸들러 클래스에서는 Listener 인터페이스에 정의된 모든 메소드들을 구현해야 함

2 이벤트 관련 클래스와 인터페이스

〈3〉 Listener 인터페이스

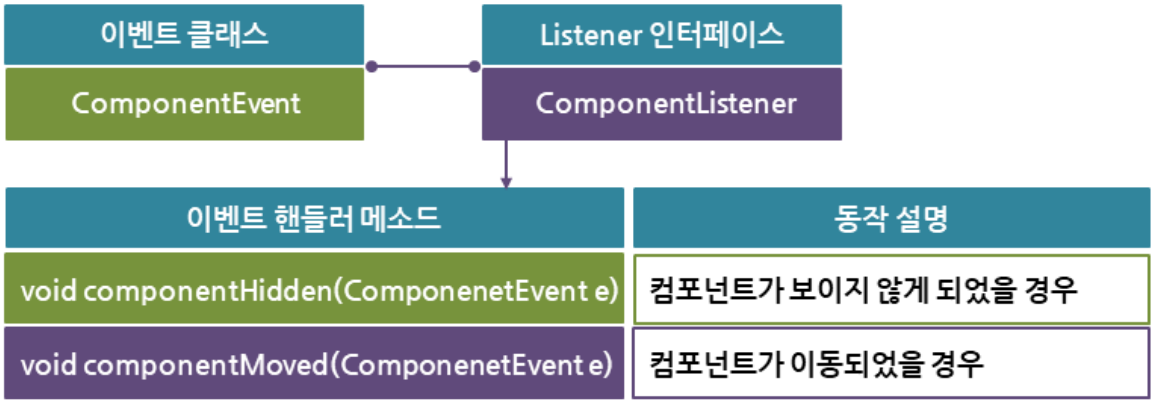


이벤트 개요

2 이벤트 관련 클래스와 인터페이스

〈3〉 Listener 인터페이스

4 Listener 인터페이스의 종류

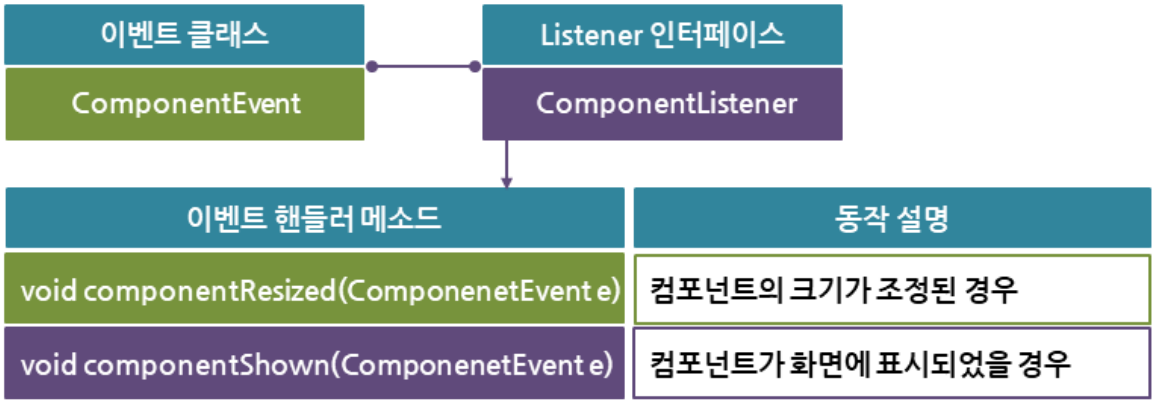


이벤트 핸들러 메소드	동작 설명
void componentHidden(ComponenetEvent e)	컴포넌트가 보이지 않게 되었을 경우
void componentMoved(ComponenetEvent e)	컴포넌트가 이동되었을 경우

2 이벤트 관련 클래스와 인터페이스

〈3〉 Listener 인터페이스

4 Listener 인터페이스의 종류



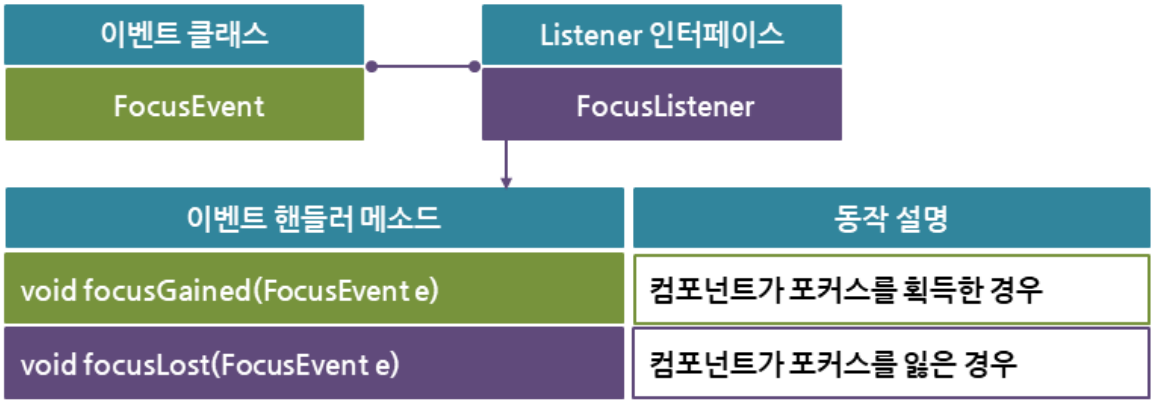
이벤트 핸들러 메소드	동작 설명
void componentResized(ComponenetEvent e)	컴포넌트의 크기가 조정된 경우
void componentShown(ComponenetEvent e)	컴포넌트가 화면에 표시되었을 경우

이벤트 개요

2 이벤트 관련 클래스와 인터페이스

〈3〉 Listener 인터페이스

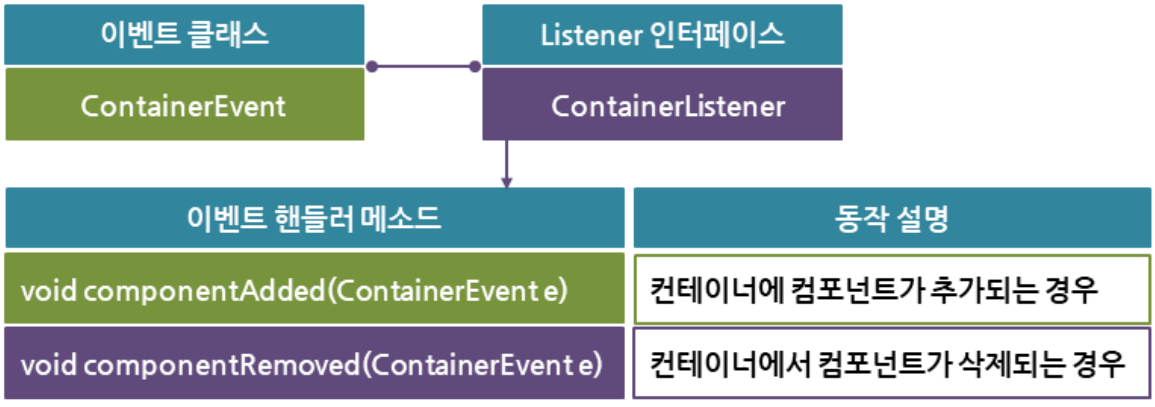
4 Listener 인터페이스의 종류



2 이벤트 관련 클래스와 인터페이스

〈3〉 Listener 인터페이스

4 Listener 인터페이스의 종류

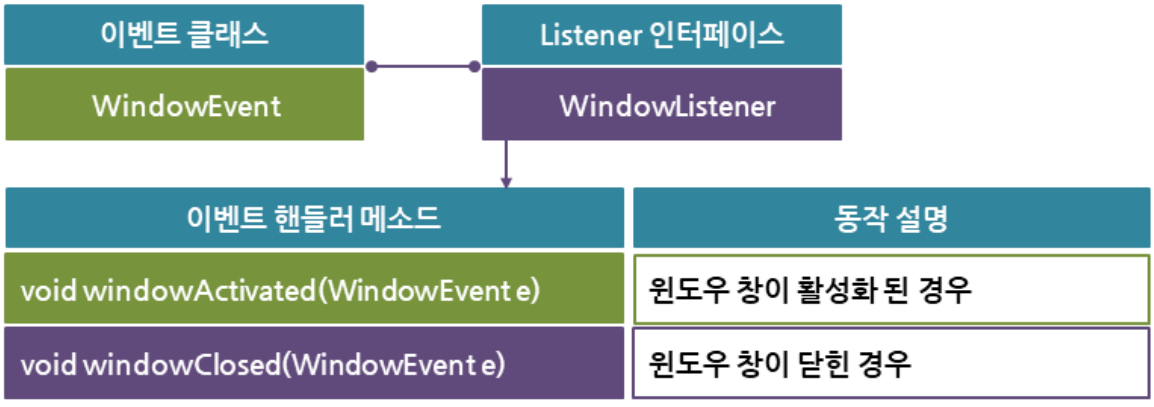


이벤트 개요

2 이벤트 관련 클래스와 인터페이스

〈3〉 Listener 인터페이스

4 Listener 인터페이스의 종류



이벤트 핸들러 메소드	동작 설명
void windowActivated(WindowEvent e)	윈도우 창이 활성화 된 경우
void windowClosed(WindowEvent e)	윈도우 창이 닫힌 경우

2 이벤트 관련 클래스와 인터페이스

〈3〉 Listener 인터페이스

4 Listener 인터페이스의 종류

이벤트 핸들러 메소드	동작 설명
void windowClosing(WindowEvent e)	윈도우 창이 닫히고 있는 중인 경우
void windowDeactivated(WindowEvent e)	윈도우 창이 비활성화 된 경우
void windowActivated(WindowEvent e)	윈도우 창이 아이콘화 된 상태에서 복귀하는 경우
void windowDeiconified(WindowEvent e)	윈도우 창이 아이콘화 된 경우
void windowOpened(WindowEvent e)	윈도우 창이 화면에 표시되는 경우

이벤트 개요

2 이벤트 관련 클래스와 인터페이스

〈3〉 Listener 인터페이스

4 Listener 인터페이스의 종류

이벤트 클래스	Listener 인터페이스
KeyEvent	KeyListener

이벤트 핸들러 메소드	동작 설명
void keyPressed(KeyEvent e)	키보드의 키를 누르는 경우
void keyReleased(keyEvent e)	키보드의 키를 떼는 경우
void keyTyped(keyEvent e)	키보드의 키를 눌렀다가 떼는 경우

2 이벤트 관련 클래스와 인터페이스

〈3〉 Listener 인터페이스

4 Listener 인터페이스의 종류

이벤트 클래스	Listener 인터페이스
MouseEvent	MouseListener

이벤트 핸들러 메소드	동작 설명
void mouseClicked(MouseEvent e)	마우스 버튼을 눌렀다가 떼는 경우(클릭)
void mouseEntered(MouseEvent e)	마우스 포인터가 컴포넌트 위로 들어오는 경우
void mouseExited(MouseEvent e)	마우스 포인터가 컴포넌트를 벗어나는 경우

이벤트 개요

2 이벤트 관련 클래스와 인터페이스

〈3〉 Listener 인터페이스

4 Listener 인터페이스의 종류

이벤트 클래스	Listener 인터페이스
MouseEvent	MouseListener

이벤트 핸들러 메소드	동작 설명
void mousePressed(MouseEvent e)	마우스의 버튼을 누르는 경우
void mouseReleased(MouseEvent e)	마우스의 버튼을 떼는 경우

2 이벤트 관련 클래스와 인터페이스

〈3〉 Listener 인터페이스

4 Listener 인터페이스의 종류

이벤트 클래스	Listener 인터페이스
MouseEvent	MouseMotionListener

이벤트 핸들러 메소드	동작 설명
void mouseDragged(MouseEvent e)	마우스를 클릭한 상태에서 포인터를 움직이는 경우
void mouseMoved(MouseEvent e)	마우스를 클릭하지 않은 상태에서 포인터만 움직이는 경우

이벤트 개요

2 이벤트 관련 클래스와 인터페이스

〈3〉 Listener 인터페이스

4 Listener 인터페이스의 종류

이벤트 클래스	Listener 인터페이스
ActionEvent	ActionListener

이벤트 핸들러 메소드	동작 설명
void actionPerformed(ActionEvent e)	버튼 클릭, 항목 또는 메뉴 선택, 엔터 키 입력등

2 이벤트 관련 클래스와 인터페이스

〈3〉 Listener 인터페이스

4 Listener 인터페이스의 종류

이벤트 클래스	Listener 인터페이스
AdjustmentEvent	AdjustmentListener

이벤트 핸들러 메소드	동작 설명
void adjustmentValueChanged(AdjustmentEvent e)	스크롤 바의 값을 변경하는 경우

이벤트 개요

2 이벤트 관련 클래스와 인터페이스

〈3〉 Listener 인터페이스

4 Listener 인터페이스의 종류

이벤트 클래스	Listener 인터페이스
ItemEvent	ItemListener

이벤트 핸들러 메소드	동작 설명
void itemStateChanged(ItemEvent e)	항목을 선택 또는 해제하는 경우

2 이벤트 관련 클래스와 인터페이스

〈3〉 Listener 인터페이스

4 Listener 인터페이스의 종류

이벤트 클래스	Listener 인터페이스
TextEvent	TextListener

이벤트 핸들러 메소드	동작 설명
void textValueChanged(TextEvent e)	입력한 값의 내용이 변경되는 경우

이벤트 개요

2 이벤트 관련 클래스와 인터페이스

⟨4⟩ Adapter 클래스

1. Listener 인터페이스를 구현한 클래스

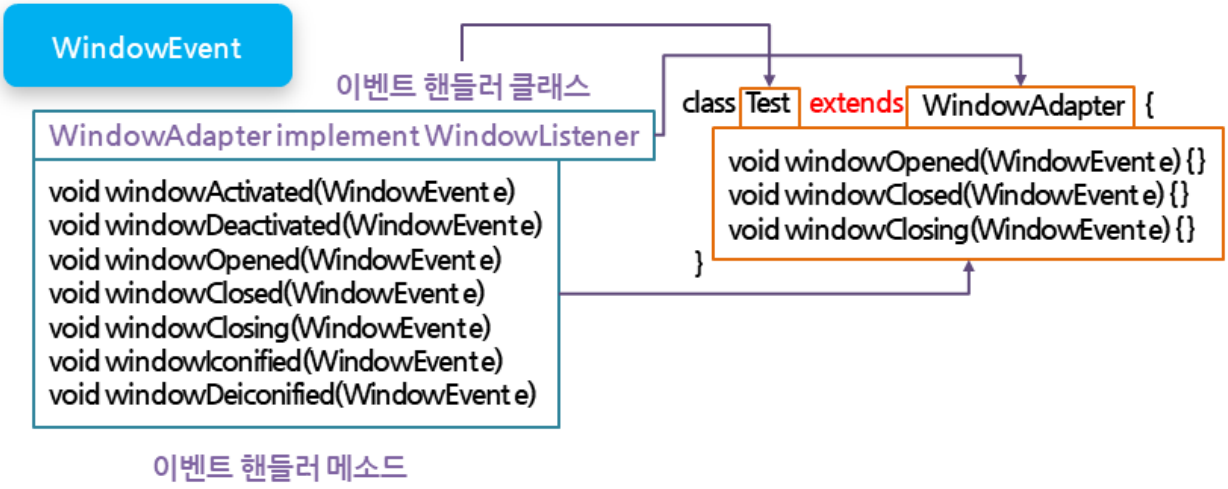
   이벤트 핸들러 메소드 내부에 구현된 코드는 없음

2. Adapter 클래스를 상속한 클래스는 이벤트 핸들러 클래스가 됨

3. 이벤트 핸들러 클래스에서 필요한 이벤트 핸들러 메소드만 오버라이딩(Overriding)하면 됨

2 이벤트 관련 클래스와 인터페이스

⟨4⟩ Adapter 클래스



이벤트 개요

2 이벤트 관련 클래스와 인터페이스

⟨4⟩ Adapter 클래스

4 Adapter 클래스의 종류

Listener 인터페이스에 해당하는 Adapter 클래스가 존재

Listener 인터페이스에 메소드가 1개인 경우 Adapter 클래스가 없음

4 Adapter 클래스의 종류

Adapter 클래스가 있는 Listener 인터페이스

Listener 인터페이스	Adapter 클래스
ComponentListener	ComponentAdapter
FocusListener	FocusAdapter
ContainerListener	ContainerAdapter
WindowListener	WindowAdapter
KeyListener	KeyAdapter
MouseListener	MouseAdapter
MouseMotionListener	MouseMotionAdapter

Adapter 클래스가 없는 Listener 인터페이스

Listener 인터페이스
ActionListener
AdjustmentListener
ItemListener
TextListener

이벤트 구현하기

1 이벤트 구현 개요

① 이벤트 발생 컴포넌트

1 대부분의 컴포넌트에서 발생되는 이벤트

ComponentEvent, FocusEvent, KeyEvent, MouseEvent, MouseMotion

2 컴포넌트에 따라 발생되는 이벤트

이벤트 클래스	대표적인 이벤트 발생 컴포넌트
ContainerEvent	Frame, Dialog, Panel, ScrollPane
WindowEvent	Frame, Dialog
ActionEvent	Button, TextField, MenuItem, List

1 이벤트 구현 개요

① 이벤트 발생 컴포넌트

1 대부분의 컴포넌트에서 발생되는 이벤트

ComponentEvent, FocusEvent, KeyEvent, MouseEvent, MouseMotion

2 컴포넌트에 따라 발생되는 이벤트

이벤트 클래스	대표적인 이벤트 발생 컴포넌트
AdjustmentEvent	Scrollbar
ItemEvent	Choice, List, Checkbox, CheckboxMenuitem
TextEvent	TextField, TextArea

이벤트 구현하기

1 이벤트 구현 개요

⟨1⟩ 이벤트 발생 컴포넌트

3 컴포넌트에 이벤트 핸들러 클래스의 객체 지정

(1) 컴포넌트에서 이벤트 발생시 이벤트 핸들러 클래스의 이벤트 핸들러 메소드 호출됨

(2) 컴포넌트에 해당 이벤트 처리 가능

1 이벤트 구현 개요

⟨1⟩ 이벤트 발생 컴포넌트

3 컴포넌트에 이벤트 핸들러 클래스의 객체 지정

(3) 메소드

이벤트 구현하기

3 컴포넌트에 이벤트 핸들러 클래스의 객체 지정

(3) 메소드

이벤트 클래스	메소드
ComponentEvent	void addComponentListener(ComponentListener l)
FocusEvent	void addFocusListener(FocusListener l)
ContainerEvent	void addContainerListener(ContainerListener l)
WindowEvent	void addWindowListener(WindowListener l)
KeyEvent	void addKeyListener(keyListener l)
MouseEvent	void addMouseListener(MouseListener l)
ActionEvent	void addActionListener(ActionListener l)
AdjustmentEvent	void addAdjustmentListener(AdjustmentListener l)
ItemEvent	void addItemListener(ItemListener l)
TextEvent	void addTextListener(TextListener l)

2 컴포넌트에 이벤트 핸들러 클래스의 객체 지정

(3) 메소드

이벤트 클래스	메소드
ComponentEvent	void addComponentListener(ComponentListener l)
FocusEvent	void addFocusListener(FocusListener l)
ContainerEvent	void addContainerListener(ContainerListener l)
WindowEvent	void addWindowListener(WindowListener l)
KeyEvent	void addKeyListener(keyListener l)
MouseEvent	void addMouseListener(MouseListener l)
ActionEvent	void addActionListener(ActionListener l)
AdjustmentEvent	void addAdjustmentListener(AdjustmentListener l)
ItemEvent	void addItemListener(ItemListener l)
TextEvent	void addTextListener(TextListener l)

이벤트 구현하기

2 컴포넌트에 이벤트 핸들러 클래스의 객체 지정

(3) 메소드

이벤트 클래스	메소드
ComponentEvent	void addComponentListener(ComponentListener l)
FocusEvent	void addFocusListener(FocusListener l)
ContainerEvent	void addContainerListener(ContainerListener l)
WindowEvent	void addWindowListener(WindowListener l)
KeyEvent	void addKeyListener(keyListener l)
MouseEvent	void addMouseListener(MouseListener l)
ActionEvent	void addActionListener(ActionListener l)
AdjustmentEvent	void addAdjustmentListener(AdjustmentListener l)
ItemEvent	void addItemListener(ItemListener l)
TextEvent	void addTextListener(TextListener l)

2 컴포넌트에 이벤트 핸들러 클래스의 객체 지정

(3) 메소드

이벤트 클래스	메소드
ComponentEvent	void addComponentListener(ComponentListener l)
FocusEvent	void addFocusListener(FocusListener l)
ContainerEvent	void addContainerListener(ContainerListener l)
WindowEvent	void addWindowListener(WindowListener l)
KeyEvent	void addKeyListener(keyListener l)
MouseEvent	void addMouseListener(MouseListener l)
ActionEvent	void addActionListener(ActionListener l)
AdjustmentEvent	void addAdjustmentListener(AdjustmentListener l)
ItemEvent	void addItemListener(ItemListener l)
TextEvent	void addTextListener(TextListener l)

이벤트 구현하기

1 이벤트 구현 개요

〈3〉 이벤트 구현 방식

1 Listener 인터페이스를 이용한 이벤트 구현

2 Adapter 클래스를 이용한 이벤트 구현

1 이벤트 구현 개요

〈3〉 이벤트 구현 방식

1 Listener 인터페이스를 이용한 이벤트 구현

- 이벤트가 발생하는 클래스를 이벤트 핸들러 클래스로 사용
- 내부(inner) 클래스를 이벤트 핸들러 클래스로 사용
- 별도의 클래스를 이벤트 핸들러 클래스로 사용

이벤트 구현하기

1 이벤트 구현 개요

〈3〉 이벤트 구현 방식

2 Adapter 클래스를 이용한 이벤트 구현

- 내부(inner) 클래스를 이벤트 핸들러 클래스로 사용
- 별도의 클래스를 이벤트 핸들러 클래스로 사용
- 내부(inner) 익명(Anonymous) 클래스를 이벤트 핸들러 클래스로 사용

2 이벤트 구현 방법

〈1〉 Listener 인터페이스를 이용한 이벤트 구현

1 이벤트가 발생하는 클래스를 이벤트 핸들러 클래스로 사용

- 별도의 이벤트 핸들러 클래스를 생성하지 않아도 됨
- 이벤트가 발생하는 클래스와 이벤트 핸들러 클래스가 동일함

이벤트 구현하기

2 이벤트 구현 방법

① Listener 인터페이스를 이용한 이벤트 구현

1 이벤트가 발생하는 클래스를 이벤트 핸들러 클래스로 사용

구현순서

[1] java.awt.event 패키지를 import
[2] xxxListener 인터페이스를 implements 하기
[3] xxxListener 인터페이스에 있는 이벤트 핸들러 메소드 구현
[4] 이벤트를 처리할 컴포넌트 생성
[5] 컴포넌트에 이벤트 핸들러 클래스의 객체 지정
: addxxxListener(this) 메소드 이용

2 이벤트 구현 방법

① Listener 인터페이스를 이용한 이벤트 구현

2 내부(inner) 클래스를 이벤트 핸들러 클래스로 사용

이벤트가 발생하는 클래스와 이벤트 핸들러 클래스가 별도임

이벤트 핸들러 클래스에서 외부 클래스
(이벤트가 발생하는 클래스)에 자유롭게 접근 가능

이벤트 구현하기

2 이벤트 구현 방법

① Listener 인터페이스를 이용한 이벤트 구현

2 내부(inner) 클래스를 이벤트 핸들러 클래스로 사용

구현순서

[1] java.awt.event 패키지를 import

[2] 이벤트 핸들러 클래스를 정의하고, xxxListener 인터페이스를 implements

[3] 이벤트 핸들러 클래스에서 xxxListener 인터페이스에 있는
이벤트 핸들러 메소드 구현

[4] 이벤트 핸들러 클래스의 객체 생성

[5] 이벤트를 처리할 컴포넌트 생성

[6] 컴포넌트에 이벤트 핸들러 클래스의 객체 지정

: addxxxListener(이벤트 핸들러 클래스의 객체) 메소드 이용

2 이벤트 구현 방법

① Listener 인터페이스를 이용한 이벤트 구현

3 별도의 클래스를 이벤트 핸들러 클래스로 사용

이벤트가 발생하는 클래스와 이벤트 핸들러 클래스가 별도임

이벤트 핸들러 클래스에서 외부 클래스
(이벤트가 발생하는 클래스)에 자유롭게 접근 불가능

다른 소스에서 이벤트 핸들러 클래스를 재사용 가능

이벤트 구현하기

2 이벤트 구현 방법

① Listener 인터페이스를 이용한 이벤트 구현

3 별도의 클래스를 이벤트 핸들러 클래스로 사용

구현순서

[1] java.awt.event 패키지를 import

[2] 이벤트 핸들러 클래스를 정의하고, xxxListener 인터페이스를 implements

[3] 이벤트 핸들러 클래스에서 xxxListener 인터페이스에 있는
이벤트 핸들러 메소드 구현

[4] 이벤트 핸들러 클래스의 객체 생성

[5] 이벤트를 처리할 컴포넌트 생성

[6] 컴포넌트에 이벤트 핸들러 클래스의 객체 지정

: addxxxListener(이벤트 핸들러 클래스의 객체) 메소드 이용

이벤트 구현하기

이벤트 이해하기 | 이벤트 구현하기

실습하기

이벤트 이해하기 | 이벤트 구현하기

실습순서

1. WindowListener 인터페이스를 이용한 이벤트 구현 실습
   1) 이벤트가 발생하는 클래스를 이벤트 헨들러 클래스로 사용하는 실습
   2) 내부(inner) 클래스를 이벤트 헨들러 클래스로 사용하는 실습
   3) 별도의 클래스를 이벤트 헨들러 클래스로 사용하는 실습

유의사항

- JDK와 이클립스를 설치한 후 실습이 가능함
- 본인이 원하는 작업 폴더를 미리 정해 놓은 다음 실습하기
- 작업 폴더는 C드라이브에 지정하기 보다는 D드라이브나 외장하드디스크를 활용하는 것을 추천함

※ 제공되는 실습 코드를 다운받아 실습해보시기 바랍니다.

이벤트 구현하기

2 이벤트 구현 방법

〈2〉 Adapter 클래스를 이용한 이벤트 구현

1 필요한 이벤트 핸들러 메소드만 구현하여 사용 가능함

2 내부(inner) 클래스를 이벤트 핸들러 클래스로 사용

2 이벤트 구현 방법

〈2〉 Adapter 클래스를 이용한 이벤트 구현

2 내부(inner) 클래스를 이벤트 핸들러 클래스로 사용

구현순서

[1] java.awt.event 패키지를 import

[2] 내부 클래스로 이벤트 핸들러 클래스를 정의하고, xxxAdapter 클래스를 상속

[3] 이벤트 핸들러 클래스에서 필요한 이벤트 핸들러 메소드만 오버라이딩(Overriding)

[4] 이벤트 핸들러 클래스의 객체 생성

[5] 이벤트를 처리할 컴포넌트 생성

[6] 컴포넌트에 이벤트 핸들러 클래스의 객체 지정

: addxxxListener(이벤트 핸들러 클래스의 객체) 메소드 이용

이벤트 구현하기

2 이벤트 구현 방법

〈2〉 Adapter 클래스를 이용한 이벤트 구현

3 별도의 클래스를 이벤트 핸들러 클래스로 사용

구현순서

[1] java.awt.event 패키지를 import

[2] 내부 클래스로 이벤트 핸들러 클래스를 정의하고, xxxAdapter 클래스를 상속

[3] 이벤트 핸들러 클래스에서 필요한 이벤트 핸들러 메소드만 오버라이딩(Overriding)

[4] 이벤트 핸들러 클래스의 객체 생성

[5] 이벤트를 처리할 컴포넌트 생성

[6] 컴포넌트에 이벤트 핸들러 클래스의 객체 지정

: addxxxListener(이벤트 핸들러 클래스의 객체) 메소드 이용

2 이벤트 구현 방법

〈2〉 Adapter 클래스를 이용한 이벤트 구현

4 내부(inner) 익명(Anonymous) 클래스를 이벤트 핸들러 클래스로 사용

이벤트 핸들러 클래스의 이름이 없음

컴포넌트의 이벤트 핸들러 클래스를 전용으로 사용

컴포넌트에 이벤트 핸들러 객체를 지정하는
addxxxListener() 메소드의 인수로 내부 익명 클래스를 정의

내부 익명 클래스 정의시 xxxAdapter 클래스의 객체
생성과 동시에 필요한 이벤트 핸들러 메소드를 오버라이딩

이벤트 구현하기

실습하기

실습순서

1.Adapter 클래스를 이용한 이벤트 구현 실습
 1) 내부(inner) 클래스를 이벤트 핸들러 클래스로 사용하는 실습
 2) 별도의 클래스를 이벤트 핸들러 클래스로 사용하는 실습
 3) 내부(inner) 익명(Anonymous) 클래스를 이벤트 핸들러 클래스로 사용하는 실습

유의사항

- JDK와 이클립스를 설치한 후 실습이 가능함
- 본인이 원하는 작업 폴더를 미리 정해 놓은 다음 실습하기
- 작업 폴더는 C드라이브에 지정하기 보다는 D드라이브나 외장하드디스크를 활용하는 것을 추천함

※ 제공되는 실습 코드를 다운받아 실습해보시기 바랍니다.

응용문제

이벤트 이해하기 응용문제

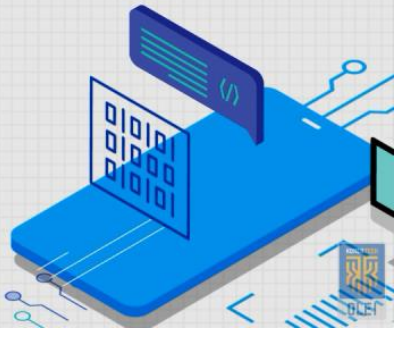
다음 실행화면과 조건에 맞게 프로그램을 작성하시오.

조건

1. 다음 실행화면과 같은 화면 구성
2. “ACTIVE” Checkbox는 창이 활성화 되면 자동 선택
3. “DEACTIVE” Checkbox는 창이 비활성화 되면 자동 선택
4. TextArea 컴포넌트에는 활성화/비활성화 메시지 자동 출력
5. WindowAdapter를 이용하여 내부 익명 클래스로 구현

클래스명 : EventTotal

제공되는 실습 소스코드를 다운받아 실습해보시기 바랍니다.



이벤트 이해하기 응용문제

실행화면