



## AWT 컴포넌트 활용하기



한국기술교육대학교  
온라인평생교육원

## 학습내용

- 하이 레벨(high-level) 이벤트 활용
- 로우 레벨(low-level) 이벤트 활용

## 학습목표

- **AWT 컴포넌트**에 **하이 레벨 이벤트**를 처리하여 프로그래밍을 할 수 있다.
- **AWT 컴포넌트**에 **로우 레벨 이벤트**를 처리하여 프로그래밍을 할 수 있다.

# 하이 레벨(high-level) 이벤트 활용

## 1 하이 레벨 이벤트 프로그래밍

### ① ActionEvent

- 1 마우스를 클릭하여 선택하거나 엔터 키를 입력하는 경우 발생
- 2 컴포넌트 : Button, TextField, MenuItem, List
- 3 이벤트 관련 메소드

이벤트 관련 메소드	설 명
<code>void addActionListener(ActionListener l)</code>	이벤트 핸들러 클래스의 객체 지정
<code>void setActionCommand(String cmd)</code>	<ul style="list-style-type: none"> <li>문자열 명령어 지정(기본: 컴포넌트의 문자열)</li> <li>Button, MenuItem에 있는 메소드</li> </ul>
<code>String getActionCommand()</code>	<ul style="list-style-type: none"> <li>문자열 명령어를 반환</li> <li>Button, MenuItem에 있는 메소드</li> </ul>

## 1 하이 레벨 이벤트 프로그래밍

### ① ActionEvent

- 1 마우스를 클릭하여 선택하거나 엔터 키를 입력하는 경우 발생
- 2 컴포넌트 : Button, TextField, MenuItem, List
- 3 이벤트 관련 메소드

이벤트 관련 메소드	설 명
<code>void actionPerformed(ActionEvent e)</code>	이벤트 핸들러 메소드 <ul style="list-style-type: none"> <li>ActionEvent 객체가 매개 변수로 넘어옴</li> </ul>
<code>Object getSource()</code>	ActionEvent 객체에서 이벤트가 발생한 컴포넌트의 객체를 반환

# 하이 레벨(high-level) 이벤트 활용



AWT 컴포넌트 활용하기

하이 레벨(high-level) 이벤트 활용



## 실습하기



AWT 컴포넌트 활용하기

하이 레벨(high-level) 이벤트 활용

### 실습순서

1. ActionEvent를 이용한 프로그래밍 실습
  - 1) Button, TextField, List 컴포넌트 프로그래밍
  - 2) MenuItem 컴포넌트 프로그래밍



### 유의사항

- JDK와 이클립스를 설치 완료된 후에 실습이 가능
- 본인이 원하는 작업 폴더를 미리 정해 놓은 다음 실습하기
- 작업 폴더는 C드라이브에 지정하기 보다는 D드라이브나 외장하드디스크를 활용하기를 추천



※ 제공되는 실습 코드를 다운받아 실습해보시기 바랍니다.

# 하이 레벨(high-level) 이벤트 활용

## 1 하이 레벨 이벤트 프로그래밍

### ② AdjustmentEvent

- 1 Scrollbar 컴포넌트의 값을 조절하는 경우 발생
- 2 컴포넌트 : Scrollbar
- 3 이벤트 관련 메소드

이벤트 관련 메소드	<code>void addAdjustmentListener(AdjustmentListener l)</code>
설 명	이벤트 핸들러 클래스의 객체 지정
이벤트 관련 메소드	<code>void adjustmentValueChanged(AdjustmentEvent e)</code>
설 명	이벤트 핸들러 메소드 AdjustmentEvent 객체가 매개 변수로 넘어옴

## 1 하이 레벨 이벤트 프로그래밍

### ② AdjustmentEvent

- 1 Scrollbar 컴포넌트의 값을 조절하는 경우 발생
- 2 컴포넌트 : Scrollbar
- 3 이벤트 관련 메소드

이벤트 관련 메소드	<code>int getAdjustmentType()</code>
설 명	AdjustmentEvent 객체에서 값을 변경하는 유형을 반환
이벤트 관련 메소드	<code>int getValue()</code>
설 명	AdjustmentEvent 객체에서 현재 값을 반환

# 하이 레벨 (high-level) 이벤트 활용

## 1 하이 레벨 이벤트 프로그래밍

### ② AdjustmentEvent

#### 4 값을 변경하는 유형

유형	상수	설명
1	AdjustmentEvent.UNIT_INCREMENT	기본값 1씩 증가



## 1 하이 레벨 이벤트 프로그래밍

### ② AdjustmentEvent

#### 4 값을 변경하는 유형

유형	상수	설명
1	AdjustmentEvent.UNIT_INCREMENT	기본값 1씩 증가
2	AdjustmentEvent.UNIT_DECREMENT	기본값 1씩 감소



# 하이 레벨 (high-level) 이벤트 활용

## 1 하이 레벨 이벤트 프로그래밍

### ② AdjustmentEvent

#### 4 값을 변경하는 유형

유형	상수	설명
1	AdjustmentEvent.UNIT_INCREMENT	기본값 1씩 증가
2	AdjustmentEvent.UNIT_DECREMENT	기본값 1씩 감소
3	AdjustmentEvent.BLOCK_DECREMENT	기본값 10씩 감소



## 1 하이 레벨 이벤트 프로그래밍

### ② AdjustmentEvent

#### 4 값을 변경하는 유형

유형	상수	설명
1	AdjustmentEvent.UNIT_INCREMENT	기본값 1씩 증가
2	AdjustmentEvent.UNIT_DECREMENT	기본값 1씩 감소
3	AdjustmentEvent.BLOCK_DECREMENT	기본값 10씩 감소
4	AdjustmentEvent.BLOCK_INCREMENT	기본값 10씩 증가



# 하이 레벨 (high-level) 이벤트 활용

## 1 하이 레벨 이벤트 프로그래밍

### ② AdjustmentEvent

#### 4 값을 변경하는 유형

유형	상수	설명
1	AdjustmentEvent.UNIT_INCREMENT	기본값 1씩 증가
2	AdjustmentEvent.UNIT_DECREMENT	기본값 1씩 감소
3	AdjustmentEvent.BLOCK_DECREMENT	기본값 10씩 감소
4	AdjustmentEvent.BLOCK_INCREMENT	기본값 10씩 증가
5	AdjustmentEvent.TRACK	조정바를 드래그하는 경우





# 하이 레벨(high-level) 이벤트 활용



AWT 컴포넌트 활용하기

하이 레벨(high-level) 이벤트 활용



## 실습하기



AWT 컴포넌트 활용하기

하이 레벨(high-level) 이벤트 활용

### 실습순서

1. AdjustmentEvent를 이용한 프로그래밍 실습  
1) Scrollbar 컴포넌트 프로그래밍



### 유의사항

- JDK와 이클립스를 설치 완료된 후에 실습이 가능
- 본인이 원하는 작업 폴더를 미리 정해 놓은 다음 실습하기
- 작업 폴더는 C드라이브에 지정하기 보다는 D드라이브나 외장하드디스크를 활용하기를 추천



※ 제공되는 실습 코드를 다운받아 실습해보시기 바랍니다.

# 하이 레벨(high-level) 이벤트 활용

## 1 하이 레벨 이벤트 프로그래밍

### ③ ItemEvent

- 1 항목을 선택/선택 해제하는 경우 발생
- 2 컴포넌트 : Checkbox, Choice, List, CheckboxMenuItem
- 3 이벤트 관련 메소드

이벤트 관련 메소드	설 명
<code>void addItemListener(ItemListener l)</code>	이벤트 핸들러 클래스의 객체 지정
<code>void itemStateChanged(ItemEvent e)</code>	이벤트 핸들러 메소드 • ItemEvent 객체가 매개 변수로 넘어옴

## 1 하이 레벨 이벤트 프로그래밍

### ③ ItemEvent

- 1 항목을 선택/선택 해제하는 경우 발생
- 2 컴포넌트 : Checkbox, Choice, List, CheckboxMenuItem
- 3 이벤트 관련 메소드

이벤트 관련 메소드	설 명
<code>Object getItem()</code>	ItemEvent 객체에서 선택/선택 해제된 항목의 객체를 반환
<code>int getStateChange()</code>	ItemEvent 객체에서 선택된 상황을 반환 • ItemEvent.SELECTED : 항목이 선택된 상황 • ItemEvent.DESELECTED : 항목이 선택 해제된 상황

# 하이 레벨(high-level) 이벤트 활용



AWT 컴포넌트 활용하기

하이 레벨(high-level) 이벤트 활용



## 실습하기



AWT 컴포넌트 활용하기

하이 레벨(high-level) 이벤트 활용

### 실습순서

1. ItemEvent를 이용한 프로그래밍 실습
  - 1) Checkbox, Choice, List, CheckboxMenuItem 컴포넌트 프로그래밍



### 유의사항

- JDK와 이클립스를 설치 완료된 후에 실습이 가능
- 본인이 원하는 작업 폴더를 미리 정해 놓은 다음 실습하기
- 작업 폴더는 C드라이브에 지정하기 보다는 D드라이브나 외장하드디스크를 활용하기를 추천



※ 제공되는 실습 코드를 다운받아 실습해보시기 바랍니다.

# 하이 레벨 (high-level) 이벤트 활용

## 1 하이 레벨 이벤트 프로그래밍

### ④ TextEvent

- 1 텍스트 입력 값의 변화가 생길 경우 발생
- 2 컴포넌트 : TextField, TextArea
- 3 이벤트 관련 메소드

이벤트 관련 메소드	설 명
<code>void addTextListener(TextListener l)</code>	이벤트 핸들러 클래스의 객체 지정
<code>void textValueChanged(TextEvent e)</code>	이벤트 핸들러 메소드 TextEvent 객체가 매개 변수로 넘어옴



# 하이 레벨(high-level) 이벤트 활용



AWT 컴포넌트 활용하기

하이 레벨(high-level) 이벤트 활용



## 실습하기



AWT 컴포넌트 활용하기

하이 레벨(high-level) 이벤트 활용

### 실습순서

1. TextEvent를 이용한 프로그래밍 실습
  - 1) TextField, TextArea 컴포넌트 프로그래밍



### 유의사항

- JDK와 이클립스를 설치 완료된 후에 실습이 가능
- 본인이 원하는 작업 폴더를 미리 정해 놓은 다음 실습하기
- 작업 폴더는 C드라이브에 지정하기 보다는 D드라이브나 외장하드디스크를 활용하기를 추천



※ 제공되는 실습 코드를 다운받아 실습해보시기 바랍니다.

# 로우 레벨 (low-level) 이벤트 활용

## 1 로우 레벨 이벤트 프로그래밍

### ① ComponentEvent

- 1 컴포넌트의 이동, 크기 변경, 화면 표시, 화면 숨기기 등의 동작시 발생
- 2 컴포넌트 : 대부분의 컴포넌트에서 발생
- 3 이벤트 관련 메소드

이벤트 관련 메소드	설 명
<code>void addComponentListener(ComponentListener l)</code>	이벤트 핸들러 클래스의 객체 지정
<code>void componentHidden(ComponentEvent e)</code> <code>void componentMoved(ComponentEvent e)</code> <code>void componentResized(ComponentEvent e)</code> <code>void componentShown(ComponentEvent e)</code>	이벤트 핸들러 메소드 TextEvent 객체가 매개 변수로 넘어옴

# 로우 레벨(low-level) 이벤트 활용



AWT 컴포넌트 활용하기

로우 레벨(low-level) 이벤트 활용



+

## 실습하기

+



AWT 컴포넌트 활용하기

로우 레벨(low-level) 이벤트 활용

실습순서

1. ComponentEvent를 이용한 프로그래밍 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.

# 로우 레벨 (low-level) 이벤트 활용

## 1 로우 레벨 이벤트 프로그래밍

### ② FocusEvent

- 1 컴포넌트가 포커스를 획득하거나 잃는 경우 발생
- 2 대부분의 컴포넌트에서 발생
- 3 이벤트 관련 메소드

이벤트 관련 메소드	설 명
<code>void addFocusListener(FocusListener l)</code>	이벤트 핸들러 클래스의 객체 지정
<code>void focusGained(FocusEvent e)</code> <code>void focusLost(FocusEvent e)</code>	이벤트 핸들러 메소드 FocusEvent 객체가 매개 변수로 넘어옴



# 로우 레벨(low-level) 이벤트 활용



AWT 컴포넌트 활용하기

로우 레벨(low-level) 이벤트 활용



## 실습하기



AWT 컴포넌트 활용하기

로우 레벨(low-level) 이벤트 활용

실습순서

1. FocusEvent를 이용한 프로그래밍 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.

# 로우 레벨 (low-level) 이벤트 활용

## 1 로우 레벨 이벤트 프로그래밍

### ③ ContainerEvent

- 1 컨테이너에 컴포넌트가 추가 되거나 삭제되는 경우 발생
- 2 대부분의 컨테이너에서 발생
- 3 이벤트 관련 메소드

이벤트 관련 메소드	설 명
<code>void addContainerListener(ContainerListener l)</code>	이벤트 핸들러 클래스의 객체 지정
<code>void componentAdded(ContainerEvent e)</code> <code>void componentRemoved(ContainerEvent e)</code>	이벤트 핸들러 메소드 ContainerEvent 객체가 매개 변수로 넘어옴
<code>void validate()</code>	컨테이너에 컴포넌트가 추가 또는 삭제되는 경우 컨테이너를 유효화하여 반영함.

# 로우 레벨(low-level) 이벤트 활용



AWT 컴포넌트 활용하기

로우 레벨(low-level) 이벤트 활용



AWT 컴포넌트 활용하기

로우 레벨(low-level) 이벤트 활용

## 실습순서

1. ContainerEvent를 이용한 프로그래밍 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.

# 로우 레벨(low-level) 이벤트 활용

## 1 로우 레벨 이벤트 프로그래밍

### ④ KeyEvent

- 1 키보드의 키를 누르거나 떼는 경우 발생
- 2 대부분의 컴포넌트에서 발생
- 3 이벤트 관련 메소드

이벤트 관련 메소드	설 명
<code>void addKeyListener(KeyListener l)</code>	이벤트 핸들러 클래스의 객체 지정
<code>void keyPressed(KeyEvent e)</code> <code>void keyReleased(KeyEvent e)</code> <code>void keyTyped(KeyEvent e)</code>	이벤트 핸들러 메소드 KeyEvent 객체가 매개 변수로 넘어옴

## 1 로우 레벨 이벤트 프로그래밍

### ④ KeyEvent

- 1 키보드의 키를 누르거나 떼는 경우 발생
- 2 대부분의 컴포넌트에서 발생
- 3 이벤트 관련 메소드

이벤트 관련 메소드	설 명
<code>char getKeyChar()</code>	입력한 문자를 반환
<code>int getKeyCode()</code>	입력한 문자에 대한 코드 값 반환
<code>static String getKeyText(int keycode)</code>	문자 코드 값에 대한 문자열 반환("HOME", "F1", "A"등)

# 로우 레벨(low-level) 이벤트 활용

## 1 로우 레벨 이벤트 프로그래밍

### ④ KeyEvent

#### ④ 이벤트 발생 순서

키보드의 문자를 입력한 경우

keyPressed() → keyTyped() → keyReleased()

키보드의 특수키를 입력한 경우(ESC, F1, Alt, Ctrl등)

keyPressed() → keyReleased()

#### ⑤ 키보드의 키 코드 값에 대응하는 상수

VK\_A, VK\_F1, VK\_ALT 등



# 로우 레벨(low-level) 이벤트 활용



AWT 컴포넌트 활용하기

로우 레벨(low-level) 이벤트 활용



## 실습하기



AWT 컴포넌트 활용하기

로우 레벨(low-level) 이벤트 활용

실습순서

1. KeyEvent를 이용한 프로그래밍 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.

# 로우 레벨(low-level) 이벤트 활용

## 1 로우 레벨 이벤트 프로그래밍

### ⑤ MouseEvent

- 1 마우스의 버튼과 움직임이 있을 때 발생
- 2 대부분의 컴포넌트에서 발생
- 3 이벤트 관련 메소드

이벤트 관련 메소드	void addMouseListener(MouseListener) void addMouseMotionListener(MouseMotionListener)
설명	이벤트 핸들러 클래스의 객체 지정

## 1 로우 레벨 이벤트 프로그래밍

### ⑤ MouseEvent

- 1 마우스의 버튼과 움직임이 있을 때 발생
- 2 대부분의 컴포넌트에서 발생
- 3 이벤트 관련 메소드

이벤트 관련 메소드	void mouseClicked(MouseEvent e) void mouseEntered(MouseEvent e) void mouseExited(MouseEvent e) void mousePressed(MouseEvent e) void mouseReleased(MouseEvent e)
설명	MouseListener 인터페이스의 이벤트 핸들러 메소드 • MouseEvent 객체가 매개 변수로 넘어옴

# 로우 레벨(low-level) 이벤트 활용

## 1 로우 레벨 이벤트 프로그래밍

### ⑤ MouseEvent

- 1 마우스의 버튼과 움직임이 있을 때 발생
- 2 대부분의 컴포넌트에서 발생
- 3 이벤트 관련 메소드

이벤트 관련 메소드	void mouseDragged(MouseEvent e) void mouseMoved(MouseEvent e)
설 명	MouseMotionListener 인터페이스의 이벤트 핸들러 메소드 • MouseEvent 객체가 매개 변수로 넘어옴

## 1 로우 레벨 이벤트 프로그래밍

### ⑤ MouseEvent

- 4 이벤트 처리 메소드

이벤트 관련 메소드	설 명
int getButton()	클릭한 마우스 버튼의 종류를 반환 • MouseEvent.BUTTON1 : 마우스 왼쪽 버튼 • MouseEvent.BUTTON2 : 마우스 가운데 버튼 • MouseEvent.BUTTON3 : 마우스 오른쪽 버튼
int getX() int getY() Point getPoint()	마우스의 위치를 반환 (MouseEvent가 발생한 컴포넌트 기준)
int getXOnScreen() int getYOnScreen() Point getLocationOnScreen()	마우스의 위치를 반환 (화면 기준)



# 로우 레벨(low-level) 이벤트 활용



AWT 컴포넌트 활용하기

로우 레벨(low-level) 이벤트 활용



## 실습하기



AWT 컴포넌트 활용하기

로우 레벨(low-level) 이벤트 활용

### 실습순서

1. MouseEvent를 이용한 프로그래밍 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

### AWT 컴포넌트 활용하기 응용문제

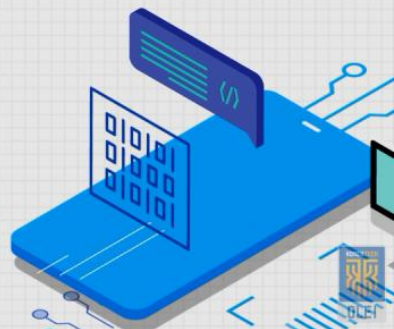
# 다음 실행화면과 조건에 맞게 프로그램을 작성하시오.

## 조건

- 1 “다이얼로그 열기” 버튼을 클릭하면 “Color Dialog” 실행
- 2 “Color Dialog”
  - (1) Dialog 클래스를 상속하여 클래스 생성
  - (2) 내부 클래스로 정의
  - (3) 라디오 버튼을 선택하면 해당 색상으로 변경하고 문자열 표시  
: setBackground() 메소드 이용
  - (4) “OK”버튼 클릭시 창 닫음

## 클래스명 : AwtDialog

제공되는 실습 소스코드를 다운받아 실습해보시기 바랍니다.



## 실행화면

