



## Swing 응용 컴포넌트 활용하기



한국기술교육대학교  
온라인평생교육원

## 학습내용

- 유용한 Swing 컴포넌트
- 유용한 Swing 컨테이너

## 학습목표

- Swing에서만 제공하는 유용한 컴포넌트들을 이용하여 프로그램을 작성할 수 있다.
- Swing에서만 제공하는 컨테이너와 관련된 컴포넌트들을 이용하여 프로그램을 작성할 수 있다.

# 유용한 Swing 컴포넌트

## 1 JProgressBar 컴포넌트

### ① JProgressBar 개요

- 1 진행 상황을 시각적으로 표시하는 컴포넌트
- 2 진행률은 백분율로 표시됨

생성자 메소드	설 명
JProgressBar()	기본 생성자 메소드
JProgressBar(int orient)	진행 바의 방향을 지정하여 객체 생성 SwingConstants.HORIZONTAL SwingConstants.VERTICAL

## 1 JProgressBar 컴포넌트

### ① JProgressBar 개요

- 1 진행 상황을 시각적으로 표시하는 컴포넌트
- 2 진행률은 백분율로 표시됨

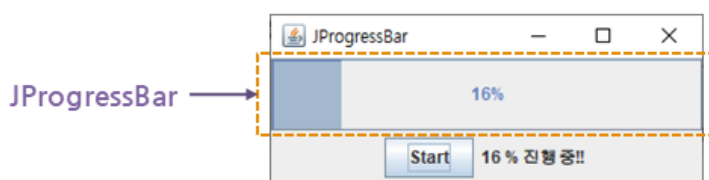
생성자 메소드	설 명
JProgressBar(int min, int max)	진행 바의 최솟값과 최댓값을 지정하여 객체 생성
JProgressBar(int orient, int min, int max)	진행 바의 방향, 최솟값, 최댓값을 지정하여 객체 생성

# 유용한 Swing 컴포넌트

## 1 JProgressBar 컴포넌트

### ② 주요 메소드

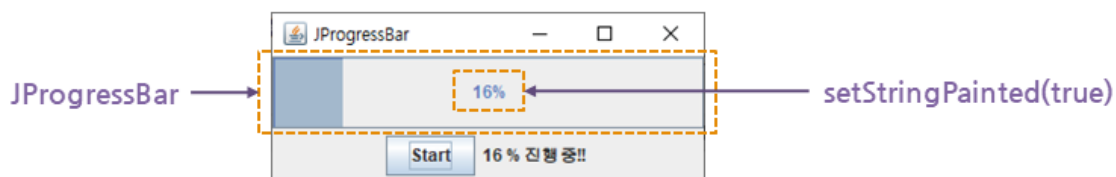
주요 메소드	설 명
<code>void addChangeListener(ChangeListener l)</code>	ChangeEvent를 등록
<code>int getValue()</code>	현재 진행 바의 진행률을 반환
<code>void setValue(int n)</code>	진행 바의 현재 값을 지정



## 1 JProgressBar 컴포넌트

### ② 주요 메소드

주요 메소드	설 명
<code>void addChangeListener(ChangeListener l)</code>	ChangeEvent를 등록
<code>int getValue()</code>	현재 진행 바의 진행률을 반환
<code>void setValue(int n)</code>	진행 바의 현재 값을 지정
<code>void setStringPainted(boolean b)</code>	true 지정시 진행 바에 진행률을 문자열로 표시



## 유용한 Swing 컴포넌트

### 2 JSlider 컴포넌트

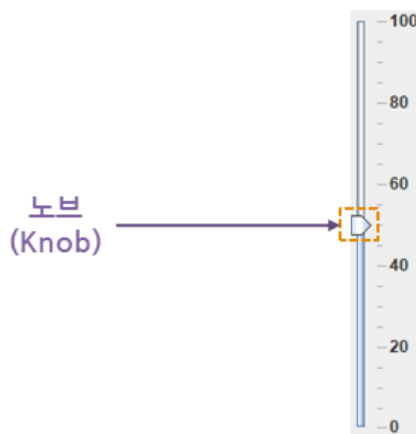
#### ① JSlider 개요

- 1 노브(Knob)를 움직여서 값을 선택할 수 있는 컴포넌트
- 2 구성요소 : 노브, 큰 눈금, 작은 눈금, 레이블(Label)

### 2 JSlider 컴포넌트

#### ① JSlider 개요

- 1 노브(Knob)를 움직여서 값을 선택할 수 있는 컴포넌트
- 2 구성요소 : 노브, 큰 눈금, 작은 눈금, 레이블(Label)

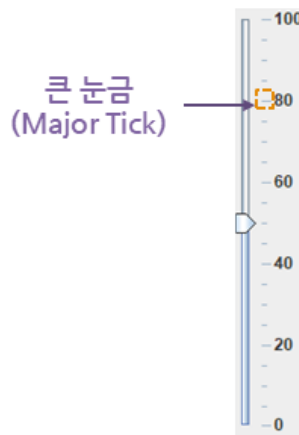


## 유용한 Swing 컴포넌트

### 2 JSlider 컴포넌트

#### ① JSlider 개요

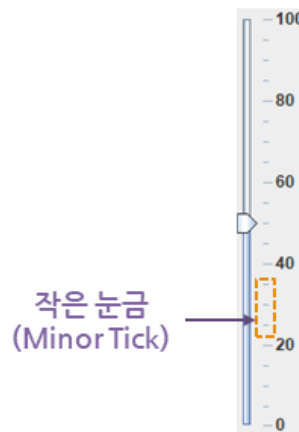
- 1 노브(Knob)를 움직여서 값을 선택할 수 있는 컴포넌트
- 2 구성요소 : 노브, 큰 눈금, 작은 눈금, 레이블(Label)



### 2 JSlider 컴포넌트

#### ① JSlider 개요

- 1 노브(Knob)를 움직여서 값을 선택할 수 있는 컴포넌트
- 2 구성요소 : 노브, 큰 눈금, 작은 눈금, 레이블(Label)



# 유용한 Swing 컴포넌트

## 2 JSlider 컴포넌트

### ① JSlider 개요

- 1 노브(Knob)를 움직여서 값을 선택할 수 있는 컴포넌트
- 2 구성요소 : 노브, 큰 눈금, 작은 눈금, 레이블(Label)



## 2 JSlider 컴포넌트

### ② 생성자 메소드

생성자 메소드	설 명
<code>JSlider()</code>	기본 생성자 메소드
<code>JSlider(int orient)</code>	슬라이더의 방향을 지정하여 객체 생성 <code>SwingConstants.HORIZONTAL</code> <code>SwingConstants.VERTICAL</code>
<code>JSlider(int min, int max)</code>	최솟값과 최댓값을 지정하여 객체 생성
<code>JSlider(int min, int max, int value)</code>	최솟값, 최댓값, 현재 위치 값을 지정하여 객체 생성
<code>JSlider(int orient, int min, int max, int value)</code>	방향, 최솟값, 최댓값, 현재 위치 값을 지정하여 객체 생성

# 유용한 Swing 컴포넌트

## 2 JSlider 컴포넌트

### ③ 주요 메소드

주요 메소드	설 명
<code>void addChangeListener(ChangeListener l)</code>	ChangeEvent를 등록
<code>int getValue()</code>	현재 슬라이더의 값을 반환
<code>void setValue(int n)</code>	현재 위치 값을 지정
<code>void setMajorTickSpacing(int n)</code>	큰 눈금의 간격을 지정

## 2 JSlider 컴포넌트

### ③ 주요 메소드

주요 메소드	설 명
<code>void setMinorTickSpacing(int n)</code>	작은 눈금의 간격을 지정
<code>void setPaintTicks(boolean b)</code>	true를 지정하면 눈금을 화면에 표시
<code>void setPaintLabels(boolean b)</code>	true를 지정하면 레이블을 화면에 표시
<code>void setSnapToTick(boolean b)</code>	true를 지정하면 노브가 눈금 단위로 이동



## 유용한 Swing 컴포넌트

### 3 JSpinner 컴포넌트

#### ① JSpinner 개요

- 1 순서가 정해진 숫자 또는 객체를 선택할 수 있는 한 줄 입력 컴포넌트
- 2 항목을 선택할 수 있는 한 쌍의 화살표
- 3 키보드의 위/아래 화살표를 이용하여 항목을 선택



### 3 JSpinner 컴포넌트

#### ① JSpinner 개요

- 4 항목에 사용 가능한 객체

SpinnerModel 인터페이스를 구현한 클래스의 객체

# 유용한 Swing 컴포넌트

## 3 JSpinner 컴포넌트

### ① JSpinner 개요

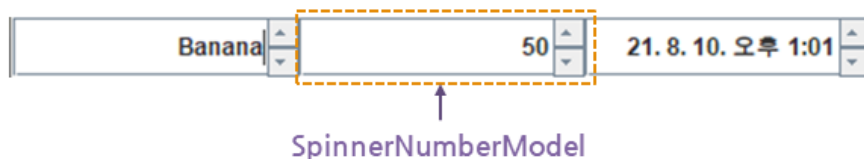
Model 클래스의 종류	설 명
SpinnerListModel	List 객체를 이용하여 JSpinner의 선택 항목을 생성



## 3 JSpinner 컴포넌트

### ① JSpinner 개요

Model 클래스의 종류	설 명
SpinnerListModel	List 객체를 이용하여 JSpinner의 선택 항목을 생성
SpinnerNumberModel	Number 객체를 이용하여 JSpinner의 선택 항목을 생성



# 유용한 Swing 컴포넌트

## 3 JSpinner 컴포넌트

### ① JSpinner 개요

Model 클래스의 종류	설 명
SpinnerListModel	List 객체를 이용하여 JSpinner의 선택 항목을 생성
SpinnerNumberModel	Number 객체를 이용하여 JSpinner의 선택 항목을 생성
SpinnerDateModel	Date 객체를 이용하여 JSpinner의 선택 항목을 생성



## 3 JSpinner 컴포넌트

### ② 생성자 메소드

생성자 메소드	설 명
JSpinner(SpinnerModel model)	SpinnerModel을 지정하여 객체 생성
SpinnerListModel(List<?> values)	List를 지정하여 객체를 생성
SpinnerNumberModel(int value, int min, int max, int stepSize)	숫자를 지정하여 객체를 생성 <ul style="list-style-type: none"> <li>• value : 처음 선택되는 값</li> <li>• min, max : 선택 가능한 숫자의 최솟값과 최댓값</li> <li>• stepSize : 숫자의 증감 크기</li> </ul>

# 유용한 Swing 컴포넌트

## 3 JSpinner 컴포넌트

### ② 생성자 메소드

생성자 메소드	설 명
<code>SpinnerDateModel( Date value, Comparable&lt;Date&gt; start, Comparable&lt;Date&gt; end, int calendarField )</code>	Date를 지정하여 객체를 생성 • value : 처음 선택되는 날짜 • start, end : 선택 가능한 시작 날짜와 마지막 날짜 • calendarField : 날짜의 항목 Calendar 클래스의 상수를 지정

## 3 JSpinner 컴포넌트

### ③ 주요 메소드

주요 메소드	설 명
<code>void addChangeListener(ChangeListener l)</code>	ChangeEvent를 등록
<code>Object getValue()</code>	선택된 항목을 Object 객체로 반환
<code>void setValue( Object value )</code>	Object 객체를 지정하여 항목을 선택

## 유용한 Swing 컴포넌트

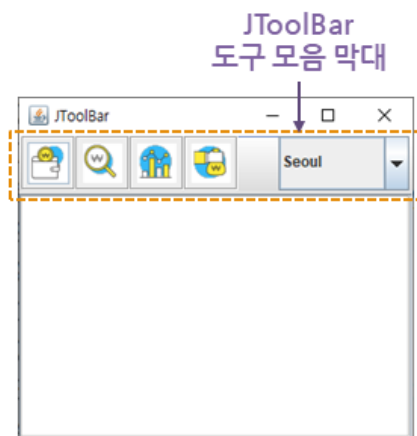
### 4 JToolBar 컴포넌트

#### ① JToolBar 개요

- 1 도구 모음 막대와 같은 형태로 표시되는 컴포넌트
- 2 다양한 Swing 컴포넌트 배치 가능
- 3 툴바의 위치
 

위쪽 : BorderLayout.PAGE\_START

아래쪽 : BorderLayout.PAGE\_END
- 4 드래그하여 툴바의 위치 변경 가능



### 4 JToolBar 컴포넌트

#### ② 생성자 메소드

생성자 메소드	설 명
JToolBar()	기본 생성자 메소드
JToolBar(int orientation)	방향을 지정하여 객체를 생성 SwingConstants.VERTICAL SwingConstants.HORIZONTAL

## 유용한 Swing 컴포넌트

### 4 JToolBar 컴포넌트

#### ③ 주요 메소드

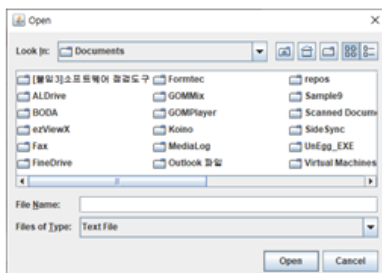
주요 메소드	설 명
<code>void addSeparator()</code>	툴바에 빈 공간을 추가
<code>void setFloatable(boolean b)</code>	true를 지정하여 투바를 드래그하여 옮길 수 있도록 설정

### 5 JFileChooser 컴포넌트

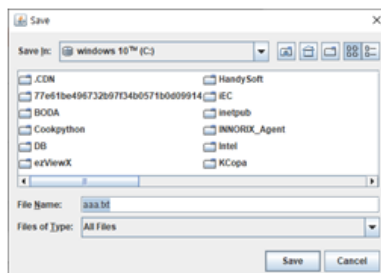
#### ① JFileChooser 개요

- 1 파일 열기 및 저장 다이얼로그
- 2 파일 확장자 필터 설정 가능

클래스 : `javax.swing.filechooser.FileNameExtensionFilter`



↑ 파일 열기



↑ 파일 저장하기

## 유용한 Swing 컴포넌트

### 5 JFileChooser 컴포넌트

#### ① JFileChooser 개요

- 1 파일 열기 및 저장 다이얼로그
- 2 파일 확장자 필터 설정 가능

클래스 : `javax.swing.filechooser.FileNameExtensionFilter`

FileNameExtensionFilter 생성자 메소드	설 명
<code>FileNameExtensionFilter(String description, String ... extensions)</code>	<ul style="list-style-type: none"> <li>• <code>description</code> : 지정한 파일 확장자에 대한 설명문</li> <li>• <code>extensions</code> : 필터를 적용할 확장자</li> </ul>

### 5 JFileChooser 컴포넌트

#### ② 생성자 메소드

생성자 메소드	설 명
<code>JFileChooser()</code>	기본 생성자 메소드
<code>JFileChooser(File currentDirectory)</code>	현재 폴더를 File 객체로 지정하여 객체 생성
<code>JFileChooser(String currentDirectory)</code>	현재 폴더를 문자열로 지정하여 객체 생성

## 유용한 Swing 컴포넌트

### 5 JFileChooser 컴포넌트

#### ③ 주요 메소드

주요 메소드	설 명
File getCurrentDirectory()	선택된 폴더를 File 객체로 반환
File getSelectedFile()	선택된 파일을 File 객체로 반환
void setCurrentDirectory(File dir)	현재 폴더를 File 객체로 지정
void setSelectedFile(File file)	현재 선택된 파일을 File 객체로 지정

### 5 JFileChooser 컴포넌트

#### ③ 주요 메소드

주요 메소드	설 명
void setFileFilter(FileFilter filter)	파일 필터를 지정 * FileFilter는 FileNameExtensionFilter의 부모 클래스
int showOpenDialog(Component parent)	파일 열기 다이얼로그 실행 반환값 : JFileChooser.APPROVE_OPTION (파일 선택시) JFileChooser.CANCEL_OPTION(취소 선택시)
int showSaveDialog(Component parent)	파일 저장하기 다이얼로그 실행 반환값 : JFileChooser.APPROVE_OPTION (파일 선택시) JFileChooser.CANCEL_OPTION(취소 선택시)

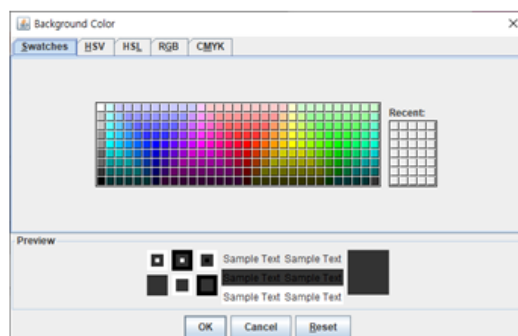


## 유용한 Swing 컴포넌트

### 6 JColorChooser 컴포넌트

#### ① JColorChooser 개요

- 1 색상을 선택할 수 있는 다이얼로그



### 6 JColorChooser 컴포넌트

#### ① JColorChooser 개요

- 1 색상을 선택할 수 있는 다이얼로그
- 2 주요 메소드

생성자 메소드	설 명
<pre>static Color showDialog(Component component, String title, Color initialColor)</pre>	<ul style="list-style-type: none"> <li>• component : 다이얼로그가 실행되는 객체 지정 일반적으로 this 지정</li> <li>• title : 다이얼로그의 제목</li> <li>• initialColor : 초기에 선택되는 색상 지정</li> </ul>

# 유용한 Swing 컴포넌트



Swing 응용 컴포넌트 활용하기

유용한 Swing 컴포넌트



Swing 응용 컴포넌트 활용하기

유용한 Swing 컴포넌트

## 실습순서

1. 유용한 Swing 컴포넌트를 이용한 프로그래밍 실습
  - 1) JProgressBar, JSlider, JSpinner, JToolBar, JFileChooser, JColorChooser 컴포넌트 프로그래밍



## 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.

# 유용한 Swing 컨테이너

## 1 JSplitPane 컴포넌트

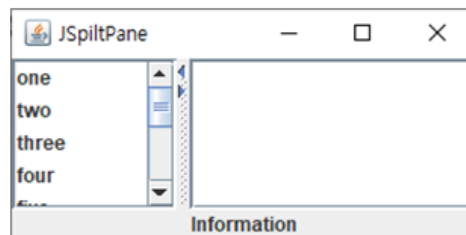
### ① JSplitPane 컴포넌트 개요

- 1 수평 또는 수직으로 분할되는 창을 제공
- 2 분할된 창의 크기 조정 가능

## 1 JSplitPane 컴포넌트

### ① JSplitPane 컴포넌트 개요

생성자 메소드	<code>JSplitPane(int newOrientation, Component newLeftComponent, Component newRightComponent)</code>
설 명	<ul style="list-style-type: none"> <li>• newOrientation : 분할 창의 방향을 지정  <code>JSplitPane.HORIZONTAL_SPLIT</code> : 수평 분할 상수  <code>JSplitPane.VERTICAL_SPLIT</code> : 수직 분할 상수 </li> </ul>

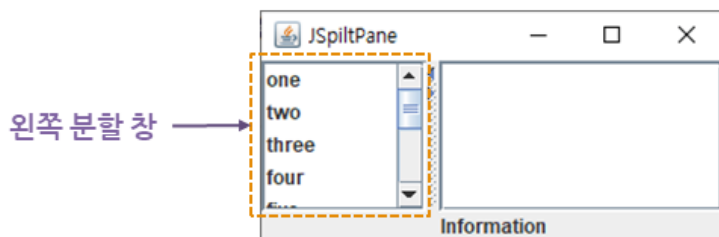


# 유용한 Swing 컨테이너

## 1 JSplitPane 컴포넌트

### ① JSplitPane 컴포넌트 개요

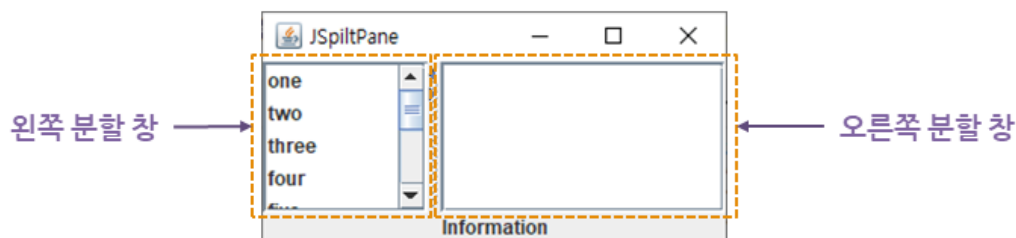
생성자 메소드	<code>JSplitPane(int newOrientation, Component newLeftComponent, Component newRightComponent)</code>
설 명	<ul style="list-style-type: none"> <li>• <code>newOrientation</code> : 분할 창의 방향을 지정  <code>JSplitPane.HORIZONTAL_SPLIT</code> : 수평 분할 상수  <code>JSplitPane.VERTICAL_SPLIT</code> : 수직 분할 상수</li> <li>• <code>newLeftComponent</code> : 왼쪽 분할 창의 컴포넌트 지정</li> </ul>



## 1 JSplitPane 컴포넌트

### ① JSplitPane 컴포넌트 개요

생성자 메소드	<code>JSplitPane(int newOrientation, Component newLeftComponent, Component newRightComponent)</code>
설 명	<ul style="list-style-type: none"> <li>• <code>newOrientation</code> : 분할 창의 방향을 지정  <code>JSplitPane.HORIZONTAL_SPLIT</code> : 수평 분할 상수  <code>JSplitPane.VERTICAL_SPLIT</code> : 수직 분할 상수</li> <li>• <code>newLeftComponent</code> : 왼쪽 분할 창의 컴포넌트 지정</li> <li>• <code>newRightComponent</code> : 오른쪽 분할 창의 컴포넌트 지정</li> </ul>

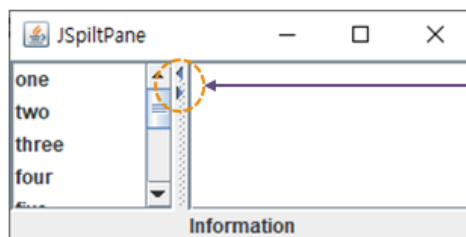


# 유용한 Swing 컨테이너

## 1 JSplitPane 컴포넌트

### ② 주요 메소드

주요 메소드	<code>void setOneTouchExpandable(boolean newValue)</code>
설 명	분할 창 중앙에 위치하는 구분선에 확장/축소 버튼 설정
주요 메소드	
설 명	

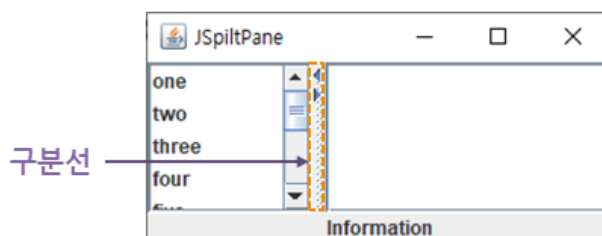


확장/축소 버튼

## 1 JSplitPane 컴포넌트

### ② 주요 메소드

주요 메소드	<code>void setOneTouchExpandable(boolean newValue)</code>
설 명	분할 창 중앙에 위치하는 구분선에 확장/축소 버튼 설정
주요 메소드	<code>void setDividerLocation(int location)</code>
설 명	분할 창의 구분선 위치를 지정



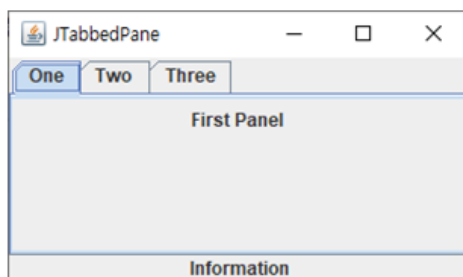
구분선

# 유용한 Swing 컨테이너

## 2 JTabbedPane 컴포넌트

### ① JTabbedPane 컴포넌트 개요

- 1 제목이 있는 탭을 클릭하여 컴포넌트 그룹간에 전환할 수 있는 컴포넌트



## 2 JTabbedPane 컴포넌트

### ① JTabbedPane 컴포넌트 개요

- 1 제목이 있는 탭을 클릭하여 컴포넌트 그룹간에 전환할 수 있는 컴포넌트

생성자 메소드	설 명
<code>JTabbedPane()</code>	기본 생성자 메소드 (탭의 위치 : 위쪽)
<code>JTabbedPane(int tabPlacement)</code>	<ul style="list-style-type: none"> <li>• 탭의 위치를 지정하여 객체 생성</li> <li>• <code>tabPlacement</code>: 탭의 위치 지정               <ul style="list-style-type: none"> <li><code>JTabbedPane.TOP</code>: 위쪽</li> <li><code>JTabbedPane.BOTTOM</code>: 아래쪽</li> <li><code>JTabbedPane.LEFT</code>: 왼쪽</li> <li><code>JTabbedPane.RIGHT</code>: 오른쪽</li> </ul> </li> </ul>

# 유용한 Swing 컨테이너

## 2 JTabbedPane 컴포넌트

### ② 주요 메소드

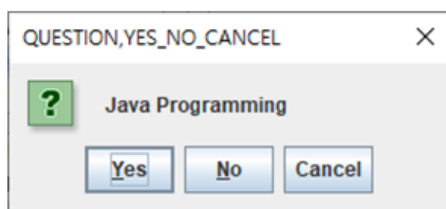
주요 메소드	설 명
<code>void addChangeListener(ChangeListener l)</code>	ChangeEvent를 등록
<code>void addTab(String title, Component component)</code>	탭을 등록 • title: 탭의 제목 • component: 탭을 클릭한 경우 표시되는 컴포넌트
<code>int getSelectedIndex()</code>	선택된 탭의 인덱스 값을 반환
<code>String getTitleAt(int index)</code>	지정한 탭의 제목을 문자열로 반환
<code>void removeTabAt(int index)</code>	지정한 탭을 삭제

## 3 JOptionPane 컴포넌트

### ① JOptionPane 컴포넌트 개요

- 1 사용자에게 값을 입력 받거나 선택할 수 있는 버튼을 제공하는 표준 대화 상자를 제공

표준 대화상자 종류	정적 메소드
Yes/No/Cancel 확인 질문	<code>showConfirmDialog()</code>





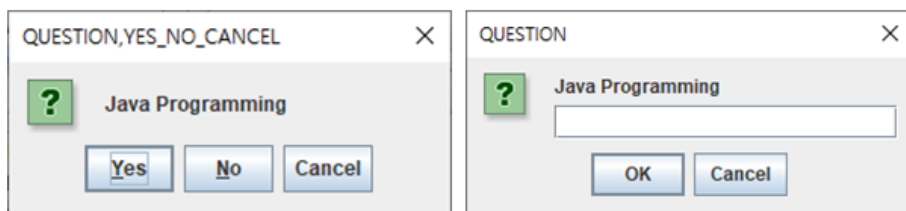
# 유용한 Swing 컨테이너

## 3 JOptionPane 컴포넌트

### ① JOptionPane 컴포넌트 개요

- 1 사용자에게 값을 입력 받거나 선택할 수 있는 버튼을 제공하는 표준 대화 상자를 제공

표준 대화상자 종류	정적 메소드
Yes/No/Cancel 확인 질문	<code>showConfirmDialog()</code>
입력 받기	<code>showInputDialog()</code>

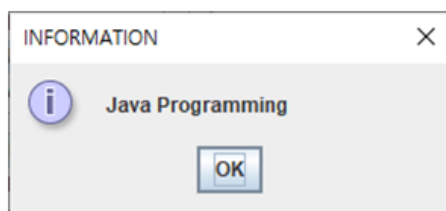


## 3 JOptionPane 컴포넌트

### ① JOptionPane 컴포넌트 개요

- 1 사용자에게 값을 입력 받거나 선택할 수 있는 버튼을 제공하는 표준 대화 상자를 제공

표준 대화상자 종류	정적 메소드
알림 메시지 출력	<code>showMessgaeDialog()</code>





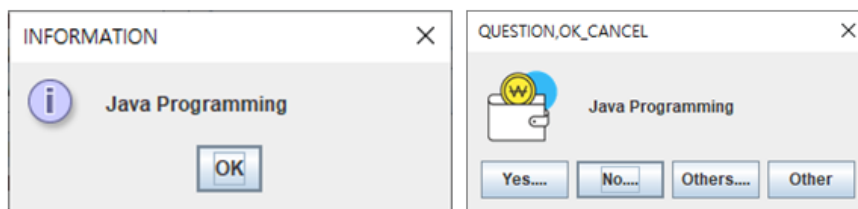
# 유용한 Swing 컨테이너

## 3 JOptionPane 컴포넌트

### ① JOptionPane 컴포넌트 개요

- 1 사용자에게 값을 입력 받거나 선택할 수 있는 버튼을 제공하는 표준 대화 상자를 제공

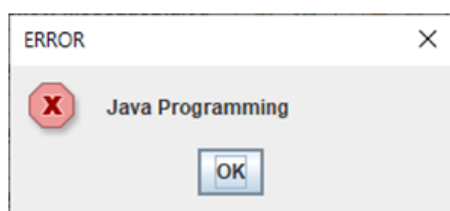
표준 대화상자 종류	정적 메소드
알림 메시지 출력	<code>showMessgaeDialog()</code>
다양한 선택	<code>showOptionDialog()</code>



## 3 JOptionPane 컴포넌트

### ② JOptionPane 상수

메시지 형태( Message Type )	설 명
<code>JOptionPane.ERROR_MESSAGE</code>	에러 메시지

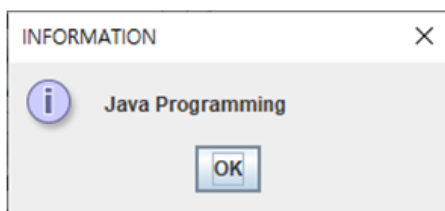
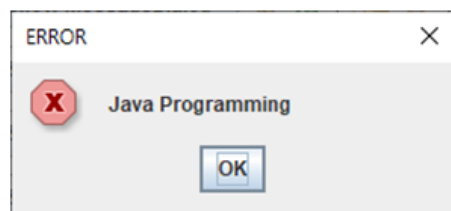


# 유용한 Swing 컨테이너

## 3 JOptionPane 컴포넌트

### ② JOptionPane 상수

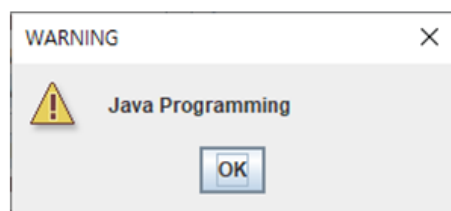
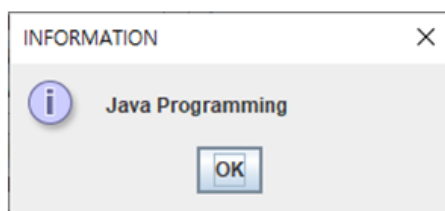
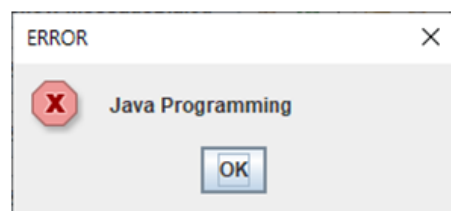
메시지 형태( Message Type )	설 명
JOptionPane.ERROR_MESSAGE	에러 메시지
JOptionPane.INFORMATION_MESSAGE	정보 알림 메시지



## 3 JOptionPane 컴포넌트

### ② JOptionPane 상수

메시지 형태( Message Type )	설 명
JOptionPane.ERROR_MESSAGE	에러 메시지
JOptionPane.INFORMATION_MESSAGE	정보 알림 메시지
JOptionPane.WARNING_MESSAGE	경고 메시지

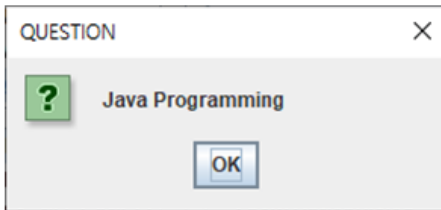


# 유용한 Swing 컨테이너

## 3 JOptionPane 컴포넌트

### ② JOptionPane 상수

메시지 형태( Message Type )	설 명
JOptionPane.QUESTION_MESSAGE	질문 메시지



## 3 JOptionPane 컴포넌트

### ② JOptionPane 상수

메시지 형태( Message Type )	설 명
JOptionPane.QUESTION_MESSAGE	질문 메시지
JOptionPane.PLAIN_MESSAGE	단순 메시지

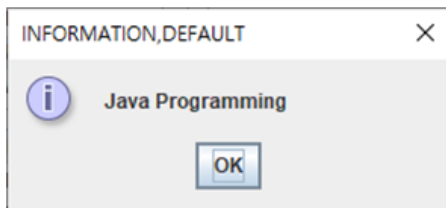


## 유용한 Swing 컨테이너

### 3 JOptionPane 컴포넌트

#### ② JOptionPane 상수

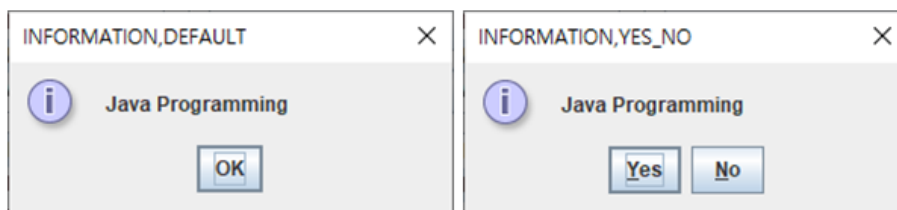
선택 버튼 형태 ( Option Type )	설 명
JOptionPane.DEFAULT_OPTION	“OK” 버튼



### 3 JOptionPane 컴포넌트

#### ② JOptionPane 상수

선택 버튼 형태 ( Option Type )	설 명
JOptionPane.DEFAULT_OPTION	“OK” 버튼
JOptionPane.YES_NO_OPTION	“YES”, “NO” 버튼

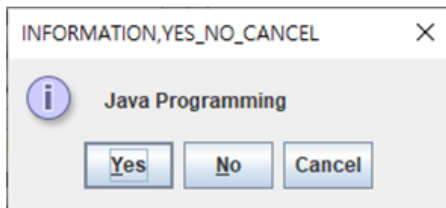


## 유용한 Swing 컨테이너

### 3 JOptionPane 컴포넌트

#### ② JOptionPane 상수

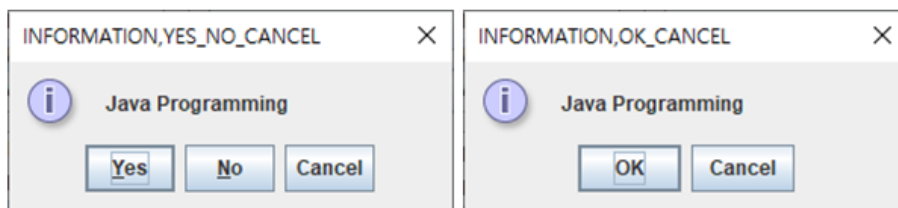
선택 버튼 형태 ( Option Type )	설 명
JOptionPane.YES_NO_CANCEL_OPTION	"YES", "NO", "CANCEL" 버튼



### 3 JOptionPane 컴포넌트

#### ② JOptionPane 상수

선택 버튼 형태 ( Option Type )	설 명
JOptionPane.YES_NO_CANCEL_OPTION	"YES", "NO", "CANCEL" 버튼
JOptionPane.OK_CANCEL_OPTION	"OK", "CANCEL" 버튼



# 유용한 Swing 컨테이너

## 3 JOptionPane 컴포넌트

### ② JOptionPane 상수

반환 값 ( Return Type )	설 명
JOptionPane.YES_OPTION	“YES” 버튼 선택시 반환
JOptionPane.NO_OPTION	“NO” 버튼 선택시 반환
JOptionPane.CANCEL_OPTION	“CANCEL” 버튼 선택시 반환
JOptionPane.OK_OPTION	“OK” 버튼 선택시 반환 ( “YES” 버튼과 값이 동일함 )
JOptionPane.CLOSED_OPTION	버튼 선택없이 대화상자 종료시 반환

## 3 JOptionPane 컴포넌트

### ③ 대화 상자 출력 메소드

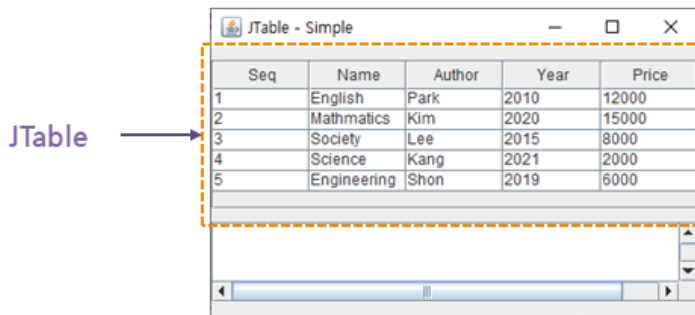
메소드	설 명
static void showConfirmDialog(Component parent, Object message, String title, int optionType, int messageType)	<ul style="list-style-type: none"> <li>parent : 대화상자를 표시하는 부모 컴포넌트</li> <li>message : 대화상자에 표시할 메시지</li> <li>title : 대화상자의 제목</li> <li>optionType : 선택 버튼의 형태</li> <li>messageType : 메시지 형태</li> </ul>
static void showInputDialog(Component parent, Object message, String title, int messageType)	
static void showMessageDialog(Component parent, Object message, String title, int messageType)	
static void showOptionDialog(Component parent, Object message, String title, int optionType, int messageType, Icon icon, Object[] options, Object initialValue)	<ul style="list-style-type: none"> <li>icon : 대화상자에 표시할 아이콘 이미지</li> <li>options : 선택 가능한 컴포넌트들</li> <li>initialValue : 컴포넌트들 중에 처음 선택될 컴포넌트</li> </ul>

# 유용한 Swing 컨테이너

## 4 JTable 컴포넌트

### ① JTable 컴포넌트 개요

- 1 2차원 셀 테이블을 표시하고 편집하는 컴포넌트
- 2 모델(Model) 클래스를 사용
- 3 이벤트 : TableModelEvent



## 4 JTable 컴포넌트

### ① JTable 컴포넌트 개요

- 4 생성자 메소드

주요 생성자 메소드	설 명
<code>JTable(Object[][] rawData, Object[] columnData)</code>	<ul style="list-style-type: none"> <li>• <code>rawData</code> : 2차원 테이블의 데이터</li> <li>• <code>columnData</code> : 열의 제목</li> </ul>
<code>JTable(TableModel dm)</code>	<code>dm</code> : TableModel 객체

# 유용한 Swing 컨테이너

## 4 JTable 컴포넌트

### ① JTable 컴포넌트 개요

#### 5 주요 메소드

주요 메소드	설 명
<code>TableModel getModel()</code>	JTable의 TableModel 객체를 반환
<code>int getSelectedColumnCount()</code>	선택된 셀의 “행의 값”을 반환
<code>int getSelectedRowCount()</code>	선택된 셀의 “열의 값”을 반환

## 4 JTable 컴포넌트

### ② AbstractTableModel

#### 1 TableModel 인터페이스를 구현한 추상 클래스

#### 2 상속하여 JTable의 모델 클래스 작성

메소드들을 오버라이딩하여 구현

주요 오버라이딩 메소드	설 명
<code>int getColumnCount()</code>	모델의 “열의 개수”를 반환
<code>int getRowCount()</code>	모델의 “행의 개수”를 반환
<code>Object getValueAt(int rowIndex, int columnIndex)</code>	지정된 좌표의 데이터를 반환



# 유용한 Swing 컨테이너

## 4 JTable 컴포넌트

### ② AbstractTableModel

- 1 TableModel 인터페이스를 구현한 추상 클래스
- 2 상속하여 JTable의 모델 클래스 작성

메소드들을 오버라이딩하여 구현

주요 오버라이딩 메소드	설 명
<code>String getColumnName(int columnIndex)</code>	지정된 열의 제목을 문자열로 반환
<code>boolean isCellEditable(int rowIndex, int columnIndex)</code>	지정된 좌표의 셀에 대한 편집 가능 여부를 반환
<code>void setValueAt(Object aValue, int rowIndex, int columnIndex)</code>	지정된 좌표에 값을 지정

## 4 JTable 컴포넌트

### ③ TableModelEvent

- 1 JTable의 모델이 변경되었을 경우 발생하는 이벤트
- 2 JTable의 `getModel()` 메소드로 TableModel 객체를 가져와서 이벤트를 등록  
셀의 값이 변경된 경우 이벤트 리스너에게 알려주어야 함

TableModel 관련 메소드	설 명
<code>void addTableModelListener(TableModelListener l)</code>	TableModelEvent를 등록
<code>void fireTableCellUpdated(int row, int column)</code>	지정된 좌표의 “셀의 값”이 변경된 경우 이벤트 리스너에게 알려줌

# 유용한 Swing 컨테이너

## 4 JTable 컴포넌트

### ③ TableModelEvent

주요 메소드	설 명
<code>int getColumn()</code>	변경된 셀의 열 좌표 값을 반환
<code>int getFirstRow()</code>	변경된 셀의 행 좌표 값을 반환

## 4 JTable 컴포넌트

### ③ TableModelEvent

#### ③ TableModelListener 인터페이스

이벤트 핸들러 메소드	설 명
<code>void tableChanged( TableModelEvent e )</code>	셀의 변경이 있는 경우 호출

# 유용한 Swing 컨테이너



Swing 응용 컴포넌트 활용하기

유용한 Swing 컨테이너



## 실습하기



Swing 응용 컴포넌트 활용하기

유용한 Swing 컨테이너

### 실습순서

1. Swing 컨테이너 관련 프로그래밍 실습
  - JSplitPane을 이용한 프로그래밍 실습
  - JTabbedPane을 이용한 프로그래밍 실습
  - JOptionPane을 이용한 프로그래밍 실습
  - JTable을 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

### Swing 응용 컴포넌트 활용하기

#### 응용문제

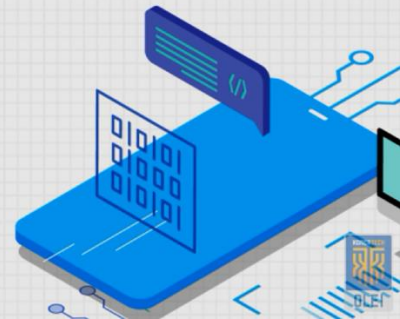
다음 실행화면과 조건에 맞게 프로그램을 작성하시오.

#### 조건

- 1 “Open” 버튼 클릭 시 파일이름으로 새로운 탭을 생성하기
- 2 “Close” 버튼 클릭 시 선택된 탭을 삭제하기

#### 클래스명 : SwingTotal

제공되는 실습 소스코드를 다운받아 실습해보시기 바랍니다.



### Swing 응용 컴포넌트 활용하기

#### 응용문제

#### 실행화면

