응용 애플리케이션 개발을 위한 자바 프로그래밍

Swing 패키지 활용하기

한국기술교육대학교
온라인평생교육원

학습내용

- Swing 이해하기
- Swing 레이아웃

학습목표

- Swing을 이해하고, Swing의 기능을 이용하여 프로그램을 작성할 수 있다.
- Swing 레이아웃을 이용하여 프로그램을 작성할 수 있다.

Swing 이해하기

1 Swing의 개요

① Swing 이란?

1 AWT와 마찬가지로 GUI를 구현하기 위한 컴포넌트와 컨테이너

AWT보다 다양한 컴포넌트와 메소드 제공

2 패키지 : javax.swing

3 컴포넌트 이름 : 'J'로 시작

1 Swing의 개요

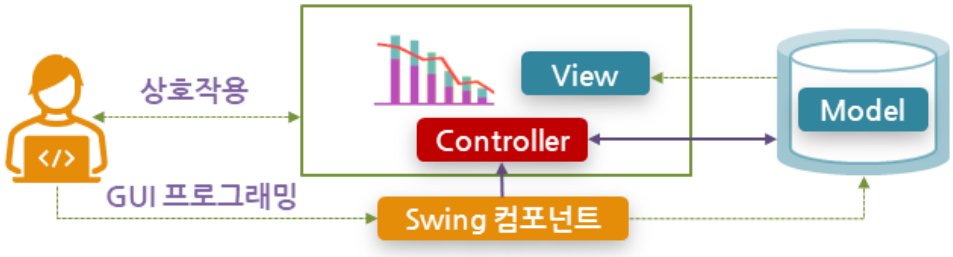
② MVC(Model View Controller) 모델

1 Swing 컴포넌트의 설계 모델

모델(Model) : 컴포넌트가 가지고 있는 데이터

뷰(View) : 모델을 시각화

컨트롤러(Controller) : 사용자와 상호 작용하여 모델의 변경



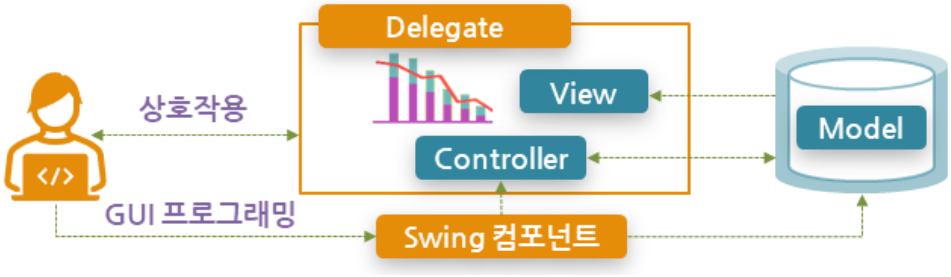
Swing 이해하기

1 Swing의 개요

〈2〉 MVC(Model View Controller) 모델

3 델리케이트(Delegate)로 구현

뷰+컨트롤러

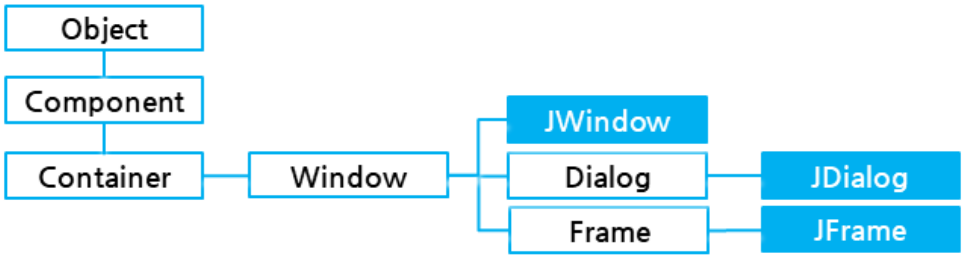


2 Swing 컴포넌트의 구조

〈1〉 독립 창으로 실행이 가능한 컨테이너

1 Window 클래스의 하위 클래스로 구현

JWindow, JDialog, JFrame



Swing 이해하기

2 Swing 컴포넌트의 구조

① 독립 창으로 실행이 가능한 컨테이너

3 창 닫기 이벤트 처리

Window 이벤트를 처리하지 않아도 됨

메소드를 호출하여 처리

창 닫기 이벤트 처리 메소드	void setDefaultCloseOperation(int operation)
설 명	이벤트 핸들러 클래스의 객체 지정

2 Swing 컴포넌트의 구조

① 독립 창으로 실행이 가능한 컨테이너

3 창 닫기 이벤트 처리

javax.swing.WindowConstants 상수	설 명
static int DISPOSE_ON_CLOSE	기본 창 닫기(메모리 해제)
static int DO_NOTHING_ON_CLOSE	창 닫기 동작 중 아무것도 실행 안함
static int EXIT_ON_CLOSE	창 닫기 동작 중 프로그램을 빠져나감
static int HIDE_ON_CLOSE	창 닫기 동작 중 창을 숨김

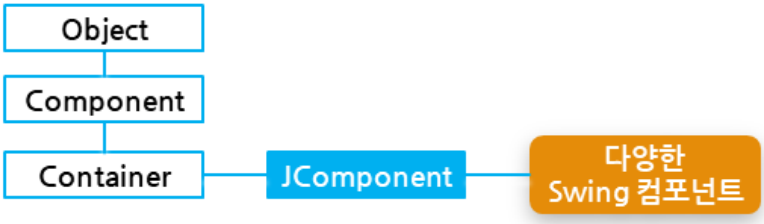
Swing 이해하기

2 Swing 컴포넌트의 구조

⑵ 독립 창으로 실행이 불가능한 컴포넌트

1 JComponent 클래스의 하위 클래스로 구현

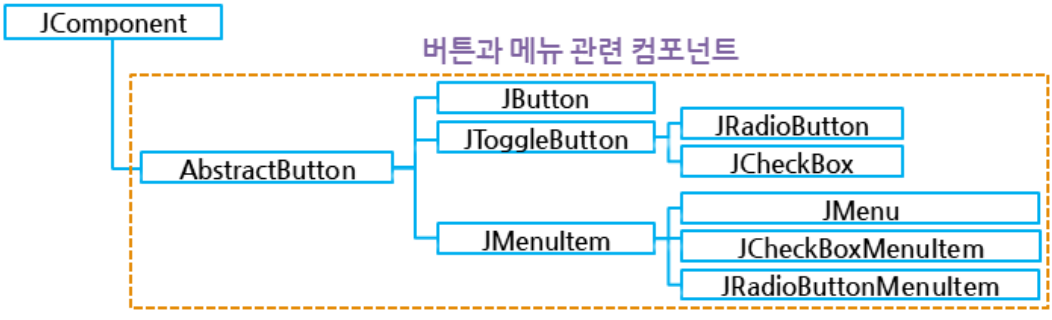
- Container 클래스를 상속한 JComponent 클래스
- 다양한 컴포넌트가 존재



2 Swing 컴포넌트의 구조

⑵ 독립 창으로 실행이 불가능한 컴포넌트

2 컴포넌트 종류

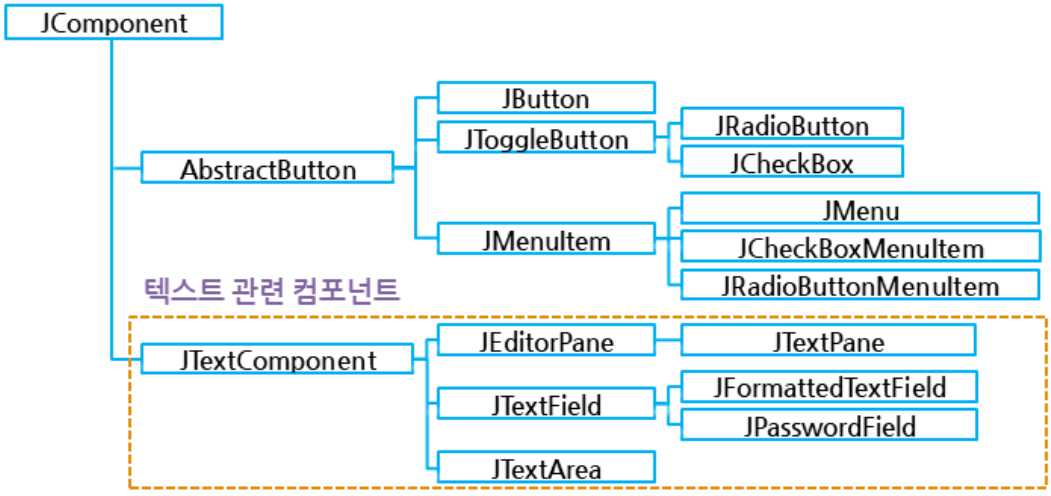


Swing 이해하기

2 Swing 컴포넌트의 구조

⑵ 독립 창으로 실행이 불가능한 컴포넌트

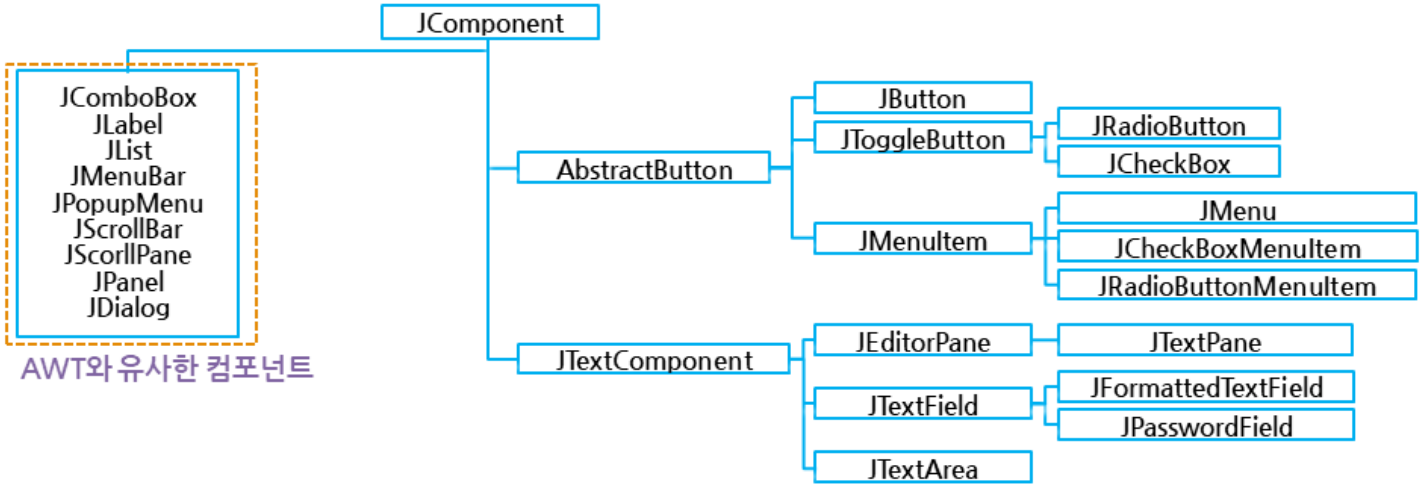
2 컴포넌트 종류



2 Swing 컴포넌트의 구조

⑵ 독립 창으로 실행이 불가능한 컴포넌트

2 컴포넌트 종류

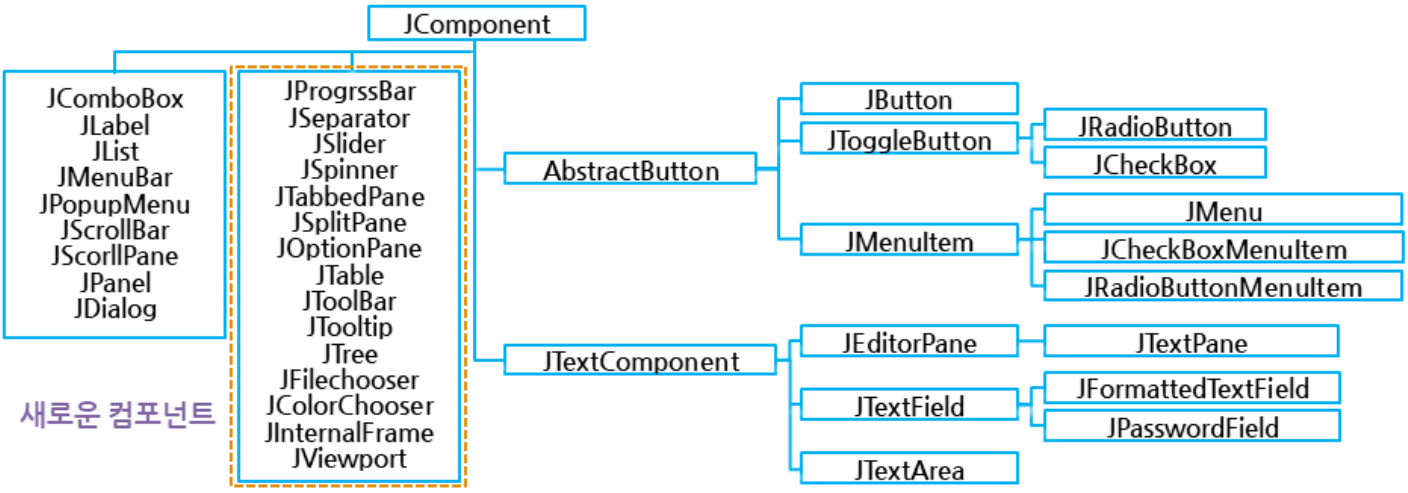


Swing 이해하기

2 Swing 컴포넌트의 구조

⟨2⟩ 독립 창으로 실행이 불가능한 컴포넌트

2 컴포넌트 종류



3 JComponent 이해하기

⟨1⟩ JComponent 개요

1 Swing 컨테이너를 제외한 모든 Swing 컴포넌트의 기본 클래스

Swing 컨테이너 : JFrame, JWindow, JDialog

2 제공하는 공통적인 기능

화면 외관(Look & Feel) 선택

풍선 도움말(Tooltip) 제공

더블 버퍼링(Double Buffering) 제공 : 이미지를 화면에 그릴 때 자연스러움

테두리(Border) 제공

Swing 이해하기

3 JComponent 이해하기

⟨2⟩ 주요 메소드

이벤트 관련 메소드	설 명
Graphics getGraphics( )	• 컴포넌트의 Graphics 객체를 반환 • 컴포넌트에 그림을 그릴 수 있음
void setPreferredSize(Dimension ps) Dimension getPreferredSize( )	컴포넌트의 기본 크기를 Dimension 객체로 지정하거나 반환
void paint(Graphics g) void update(Graphics g)	• 컴포넌트의 그래픽 관련 메소드 • 오버라이딩하여 사용

3 JComponent 이해하기

⟨2⟩ 주요 메소드

이벤트 관련 메소드	설 명
void repaint(long tm, int x, int y, int width, int height) void repaint(Rectangle r)	그래픽을 다시 그리기 위해 호출하는 메소드 • tm : 사용하지 않음 • x, y, width, height : 다시 그리기를 원하는 좌표와 크기 • r : 다시 그리기를 원하는 좌표와 크기를 가지는 Rectangle 객체
void setBorder(Boder border) Boder getBorder( )	컴포넌트의 테두리를 Border 객체로 지정하거나 반환

Swing 이해하기

3 JComponent 이해하기

⟨2⟩ 주요 메소드

주요 메소드	설 명
void setDoubleBuffered(boolean aFlag)	더블 버퍼링 여부를 지정 • aFlag = true : 더블 버퍼링 설정 • aFlag = false : 더블 버퍼링 해제
void setToolTipText(String text)	컴포넌트에 풍선 도움말을 지정

3 JComponent 이해하기

⟨3⟩ 룩앤필(Look & Feel) 설정

1 주요 룩앤필 클래스

룩앤필 클래스	설 명
MetalLookAndFeel	기본 룩앤필
NimbusLookAndFeel	Nimbus(비구름, 후광) 스타일의 룩앤필

Swing 이해하기

3 JComponent 이해하기

〈3〉 룩앤필(Look & Feel) 설정

2 룩앤필 설정 순서

- UIManager 클래스의 setLookAndFeel() 메소드로 룩앤필 클래스 지정
- SwingUtilities 클래스의 updateComponentTreeUI() 메소드로 룩앤필을 적용
- validate( ) 메소드로 그래픽의 변경을 유효화
- repaint( ) 메소드로 그래픽을 그림

3 JComponent 이해하기

〈3〉 룩앤필(Look & Feel) 설정

2 룩앤필 설정 순서

메소드	설 명
static void setLookAndFeel(String className)	className : 룩앤필 클래스 이름을 문자열로 지정 "javax.swing.plaf.metal.MetalLookAndFeel" "javax.swing.plaf.nimbus.NimbusLookAndFeel"
static void updateComponentTreeUI (Component c)	c : 룩앤필이 적용될 객체를 지정

Swing 이해하기

3 JComponent 이해하기

④ 컴포넌트 테두리 지정

1 컴포넌트에 테두리를 지정하기 위한 클래스

패키지 : javax.swing.border

2 주요 클래스

주요 클래스	설 명
BevelBorder	두 줄의 경사 테두리를 구현
SoftBevelBorder	모서리가 부드럽게 올라가거나 낮아진 경사 테두리를 구현
EtchedBorder	식각(Etching) 테두리를 구현

3 JComponent 이해하기

④ 컴포넌트 테두리 지정

3 주요 클래스의 생성자 메소드

주요 클래스	주요 생성자 메소드
BevelBorder	BevelBorder(int bevelType) BevelBorder(int bevelType, Color highlight, Color shadow) bevelType : BevelBorder.LOWERED BevelBorder.RAISED • highlight : 강조 표시 색상 • shadow : 그림자 색상
SoftBevelBorder	SoftBevelBorder(int bevelType) SoftBevelBorder(int bevelType, Color highlight, Color shadow)
EtchedBorder	EtchedBorder(int etchType) EtchedBorder(int etchType, Color highlight, Color shadow)

Swing 이해하기

3 JComponent 이해하기

⟨4⟩ 컴포넌트 테두리 지정

3 주요 클래스의 생성자 메소드

주요 클래스	주요 생성자 메소드
LineBorder	LineBorder(Color color) LineBorder(Color color, int thickness) LineBorder(Color color, int thickness, Boolean roundedCorners) • color : 선의 색상 • thickness : 선의 두께 • roundedCorners : true 인 경우 둥근 모서리 설정

3 JComponent 이해하기

⟨4⟩ 컴포넌트 테두리 지정

3 주요 클래스의 생성자 메소드

주요 클래스	주요 생성자 메소드
TitledBorder	TitledBorder(String title) TitledBorder(Border border) TitledBorder(Border border, String title) TitledBorder(Border border, String title, int titleJustification, int titlePosition) • title : 테두리 문자열 제목 • border : 테두리 클래스의 객체 • titleJustfication : 문자열 제목 정렬 방식 • titlePosition : 문자열 제목 위치

Swing 이해하기

3 JComponent 이해하기

⟨4⟩ 컴포넌트 테두리 지정

3 주요 클래스의 생성자 메소드

주요 클래스	주요 상수
TitledBorder	[제목 정렬 방식] • TitledBorder.LEFT : 테두리 왼쪽에 제목 표시 • TitledBorder.RIGHT : 테두리 오른쪽에 제목 표시 • TitledBorder.CENTER : 테두리 중앙에 제목 표시 [제목 위치] • TitledBorder.ABOVE_TOP : 테두리 윗줄의 위에 제목 표시 • TitledBorder.ABOVE_BOTTOM : 테두리 아랫줄의 위에 제목 표시 • TitledBorder.BELOW_TOP : 테두리 윗줄의 아래에 제목 표시 • TitledBorder.BELOW_BOTTOM : 테두리 아랫줄의 아래에 제목 표시 • TitledBorder.TOP : 테두리 윗줄의 가운데에 제목 표시 • TitledBorder.BOTTOM : 테두리 아랫줄의 가운데에 제목 표시

Swing 이해하기

실습하기

Swing 패키지 활용하기 Swing 이해하기

실습순서

1. Swing을 이용한 기본 프로그래밍 실습
   1) JFrame을 이용한 프로그래밍 실습
   2) 룩앤필(Look & Feel) 프로그래밍 실습
   3) 테두리를 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.

Swing 레이아웃

1 Swing 레이아웃 관리자

⟨1⟩ 레이아웃 관리자 클래스

1 AWT에서 제공하는 레이아웃 관리자 클래스 사용 가능

FlowLayout, BorderLayout, GridLayout, GridbagLayout, CardLayout

2 Swing에서 제공하는 레이아웃 관리자 클래스

주요 레이아웃 관리자 클래스	설 명
OverlayLayout	컴포넌트를 겹쳐서 화면에 표시
BoxLayout	컴포넌트를 수평 또는 수직으로 배치

1 Swing 레이아웃 관리자

⟨1⟩ 레이아웃 관리자 클래스

1 AWT에서 제공하는 레이아웃 관리자 클래스 사용 가능

FlowLayout, BorderLayout, GridLayout, GridbagLayout, CardLayout

2 Swing에서 제공하는 레이아웃 관리자 클래스

주요 레이아웃 관리자 클래스	설 명
ScrollPaneLayout	JScrollPane 컴포넌트 전용
ViewportLayout	JViewport 컴포넌트 전용

Swing 레이아웃

1 Swing 레이아웃

〈2〉 OverlayLayout 관리자

1 생성자 메소드

생성자 메소드	설 명
OverlayLayout(Container target)	target : OverlayLayout 이 적용될 컨테이너의 객체를 지정

1 Swing 레이아웃

〈2〉 OverlayLayout 관리자

2 JComponent와 관련된 메소드

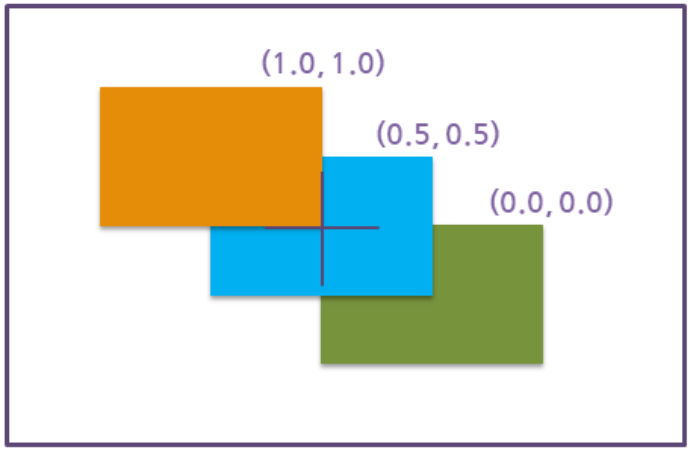
생성자 메소드	설 명
void setMaximumSize(Dimension maximum)	컴포넌트의 최대 크기를 Dimension 객체로 지정하여 설정
void setAlignmentX(float alignmenX)	컴포넌트의 수평 정렬 기준을 지정 (0.0 ~ 1.0 사이의 값 지정)
void setAlignmentY(float alignmenY)	컴포넌트의 수직 정렬 기준을 지정 (0.0 ~ 1.0 사이의 값 지정)

Swing 레이아웃

1 Swing 레이아웃

〈2〉 OverlayLayout 관리자

3 수평, 수직 정렬 기준



1 Swing 레이아웃

〈3〉 BoxLayout 관리자

1 생성자 메소드

생성자 메소드	설 명
BoxLayout(Container target, in axis)	• target : BoxLayout이 적용될 컨테이너의 객체를 지정 • axis : 컴포넌트가 배치되는 방식 지정

상수 (컴포넌트가 배치되는 방식)	설 명
BoxLayout.X_AXIS	왼쪽에서 오른쪽으로 수평 배치
BoxLayout.Y_AXIS	위에서 아래로 수직 배치

Swing 레이아웃

1 Swing 레이아웃

〈3〉 BoxLayout 관리자

1 생성자 메소드

생성자 메소드	설 명
BoxLayout(Container target, in axis)	• target : BoxLayout이 적용될 컨테이너의 객체를 지정 • axis : 컴포넌트가 배치되는 방식 지정

상수 (컴포넌트가 배치되는 방식)	설 명
BoxLayout.LINE	한 줄에 컴포넌트가 배치되는 방식 왼쪽에서 오른쪽, 오른쪽에서 왼쪽으로 배치 가능
BoxLayout.PAGE	텍스트 줄이 배치되는 방식, Y_AXIS와 유사

1 Swing 레이아웃

〈3〉 BoxLayout 관리자

2 BoxLayout.LINE 활용

메소드	void applyComponentOrientation(ComponentOrientation orientation)
설 명	• Component 클래스의 메소드로 컴포넌트의 배치 방향을 지정 • orientation : ComponentOrientation 클래스의 상수를 지정

상 수	설 명
ComponentOrientation.LEFT_TO_RIGHT	왼쪽에서 오른쪽 방향
ComponentOrientation.RIGHT_TO_LEFT	오른쪽에서 왼쪽 방향

Swing 레이아웃

1 Swing 레이아웃

〈3〉 BoxLayout 관리자

3 Box 클래스

BoxLayout을 레이아웃 관리자로 사용하는 컨테이너

BoxLayout을 사용하기 위해 필요한 메소드 제공

Box 클래스의 주요 메소드	static Component createRigidArea(Dimension d)
설 명	• 컴포넌트 사이의 빈 공간을 위해서 보이지 않는 컴포넌트를 생성 • Dimension 객체로 크기 지정함

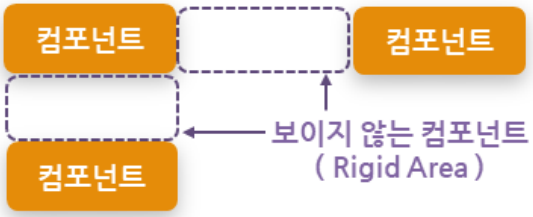
1 Swing 레이아웃

〈3〉 BoxLayout 관리자

3 Box 클래스

BoxLayout을 레이아웃 관리자로 사용하는 컨테이너

BoxLayout을 사용하기 위해 필요한 메소드 제공



Swing 레이아웃

1 Swing 레이아웃

〈3〉 BoxLayout 관리자

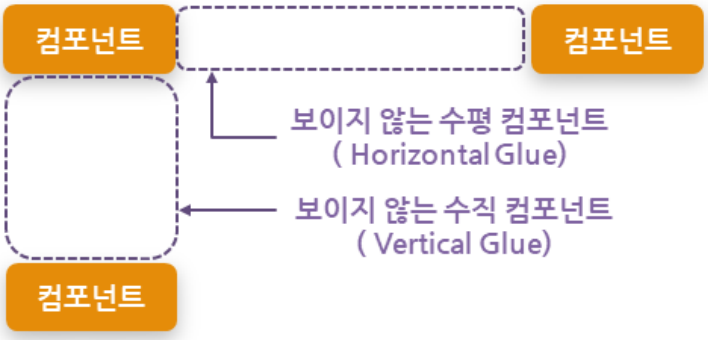
3 Box 클래스

Box 클래스의 주요 메소드	설 명
static Component createHorizontalGlue()	컨테이너 수평 크기에 비례하여 보이지 않는 컴포넌트를 생성, 크기를 지정하지 않음
static Component createVerticalGlue()	컨테이너 수직 크기에 비례하여 보이지 않는 컴포넌트를 생성, 크기를 지정하지 않음

1 Swing 레이아웃

〈3〉 BoxLayout 관리자

3 Box 클래스



Swing 레이아웃

Swing 패키지 활용하기 | Swing 레이아웃

실습하기

Swing 패키지 활용하기 | Swing 레이아웃

실습순서

1. Swing 레이아웃 프로그래밍
   1) OverlayLaout을 이용한 GUI 프로그래밍
   2) BoxLayout을 이용한 GUI 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.

응용문제

Swing 패키지 활용하기 응용문제

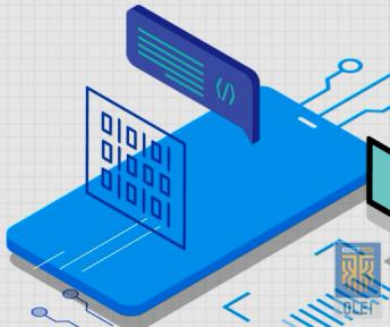
다음 실행화면과 조건에 맞게 프로그램을 작성하시오.

조건

1. 테두리 : EtchedBoder, TitledBorder 사용
2. 버튼 : JButton
   - 크기 지정 (80*30)
   - 사용 메소드 : setMaximumSize(new Dimension(80,30))
3. 레이아웃 : BoxLayout

클래스명 : SwingTotal

제공되는 실습 소스코드를 다운받아 실습해보시기 바랍니다.



Swing 패키지 활용하기 응용문제

실행화면