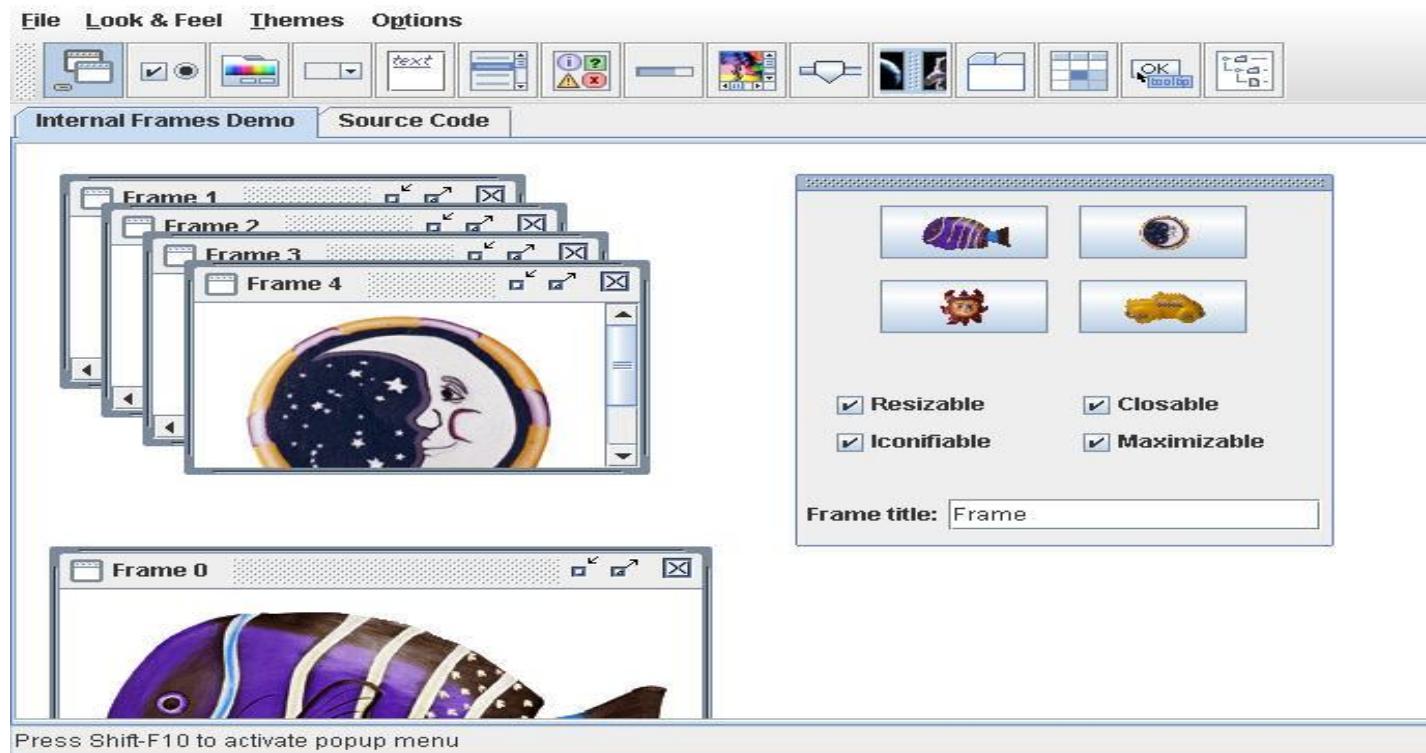


Swing

스윙의 특징

- 자바에의 JFC(Java Foundation Class)는 GUI 프로그래밍에 필요한 각종 툴킷을 모아놓은 것으로 현재는 GUI의 기능들을 구현할 수 있는 스윙, 2D, Drag&Drop 등을 지원한다.
- 스윙을 사용하는 방법은 AWT와 거의 유사하나 AWT보다는 많은 컴포넌트 및 기능을 지원하고 있다.
- 스윙은 AWT와 달리 자바 프로그래밍으로 자체적인 제작된 컴포넌트 이므로 플랫폼에 관계없이 모양이 동일하게 사용할 수 있다.



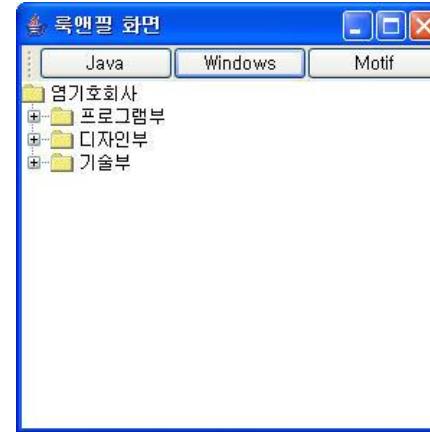
스윙의 특징

- 룩앤팔(Look & Feel)

- 스윙에서 가장 획기적으로 바뀐 것 중에 하나가 컴포넌트의 화려함이다.
- 이러한 외관(Look & Feel)을 프로그램을 실행하는 도중에 여러 가지 형태로 바꾸어 사용할 수 있는 기능을 제공한다.
- 스윙은 순수한 자바로만 만들어졌기 때문에 어떤 플랫폼에서라도 동일한 룩앤팔(Look & Feel)을 유지할 수가 있다.



Java Look & Feel



Windows Look & Feel



Motif Look & Feel

스윙의 특징

- 경량의 컴포넌트

- AWT에서 제공하는 컴포넌트들은 JVM이 기반으로 설치되어 있는 네 이티브 플랫폼에 의존하여 그 컴포넌트들을 그대로 가져다 사용하는 중량의 컴포넌트들이다.
- 스윙은 순수 자바로 구현되어 있는 컴포넌트들이기 때문에 어떤 플랫폼을 사용하더라도 거기에 의존하지 않고 독립적으로 사용할 수 있는 경량의 컴포넌트이다.

○ DoubleBuffering 지원

- 더블버퍼링 기능은 그래픽의 성능을 향상시키기 위해 도입된 방식으로 AWT에서는 사용자에 의해 직접 구현해야 되지만 스윙에서는 자체적으로 더블버퍼링 기능을 제공한다.

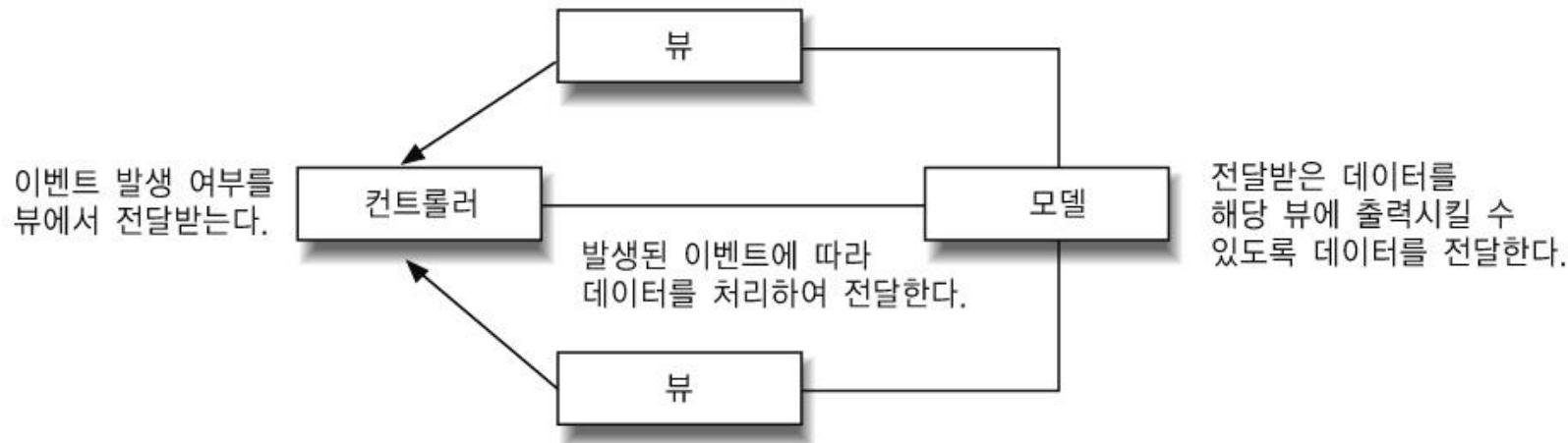
스윙의 특징

- ToolTip(풍선도움말)지원
 - 스윙에서 제공하는 컴포넌트들은 풍선 도움말을 지원한다.
 - javax.swing.JComponent 클래스의 메서드에서 setToolTipText(String text) 메서드를 이용해서 마우스를 컴포넌트 위에 위치하게 되면 풍선도움말을 지원할 수 있다.



스윙의 특징

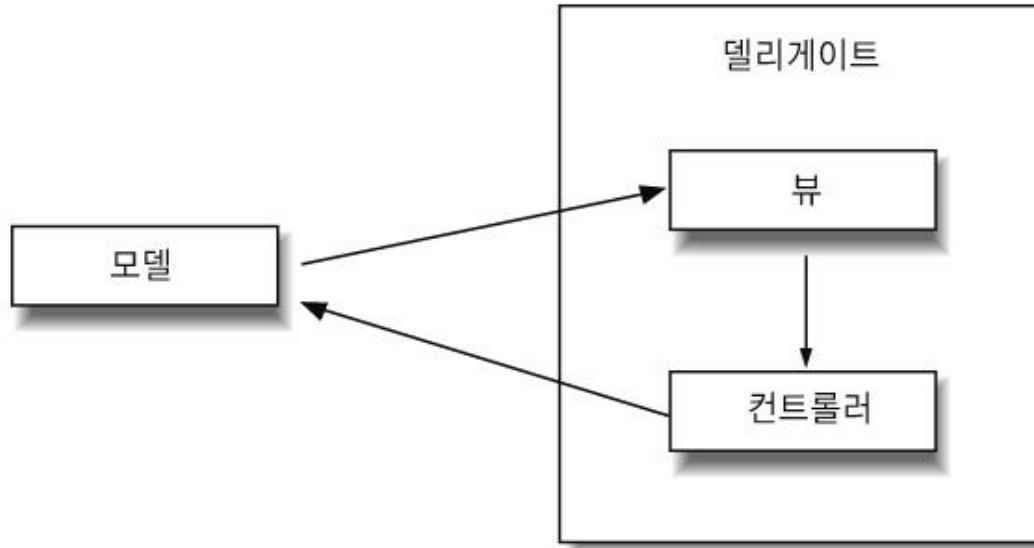
- Delegate 모델(MVC : Model-View-Controller)
 - MVC 모델은 GUI 설계 패턴 중 하나이다.
 - MVC모델은 3가지로 구성되어 있는데, 컴포넌트가 표현해야 하는 자료 구조를 추상화하고 뷰에게 필요한 데이터를 제공하는 모델(Model), 모델로부터 전달 받은 데이터를 화면에 표시하는 뷰(View), 모델과 뷰를 제어하면서 이벤트를 처리해 주는 컨트롤러(Controller)가 담당을 한다.



[그림 12-4] MVC 모델 구조

스윙의 특징

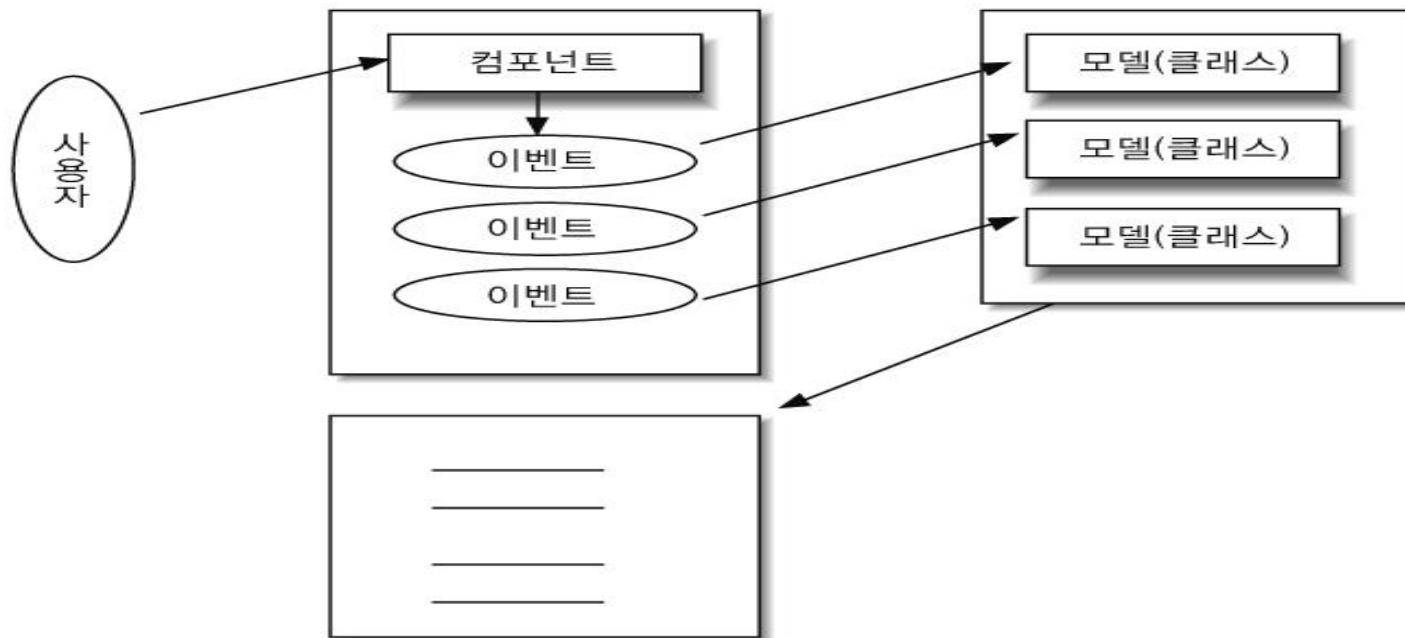
- 스윙에서는 MVC 모델에서 VC를 통합시켜 델리게이트(Delegate)라 불리는 컴포넌트로 묶어 사용하는 델리게이트 모델이다.
- 복잡한 구조를 단순화 시켜 프로그래밍을 좀 더 쉽게 할 수 있도록 하였다.
- 델리게이트는 컴포넌트의 뷰와 컨트롤러의 기능을 담당하므로 그 자체가 컴포넌트라고 할 수 있고, 이것을 UI Delegate 객체라고도 한다.



[그림 12-5] 델리게이트 모델 구조

스윙의 특징

- 실제적으로 스윙에 MVC 모델을 적용했을 때의 구조
 - 사용자가 컴포넌트에서 이벤트를 발생(Controller)시키게 되면 이벤트의 종류에 따라서 모델클래스(Model)를 선정하게 된다.
 - 선택된 모델클래스는 데이터를 처리하고 그 데이터를 다시 클라이언트에게 전송하여 출력(View)시킨다.



[그림 12-6] MVC 모델의 적용 구조

스윙의 특징

- 이미지 아이콘(ImageIcon) 지원
 - 텍스트뿐만 아니라 이미지를 화면에 출력시켜주는 기능을 가지고 있다.
 - 자바 플랫폼에서 지원하는 모든 그래픽을 사용하여 보여줄 수 있고 현재 사용할 수 있는 이미지 데이터 포맷은 JPEG와 GIF이다.



스윙의 특징

- 보더(Border) 지원
 - 보더라(Border)란 컴포넌트의 경계선을 의미한다.
 - 컴포넌트들은 경계선(Border)이 존재하지 않지만 스윙부터는 이 경계선을 어떻게 보여줄지를 결정할 수 있다.
 - 스윙에서 제공하는 Border는 8가지가 있는데 서로 조합이 가능하기 때문에 상당히 많은 형태가 존재한다고 말할 수 있다.
 - 보더를 지원해주는 패키지는 javax.swing.border 패키지에 포함되어 있다.

[표 12-1] 스윙에서 제공하는 보더와 모양

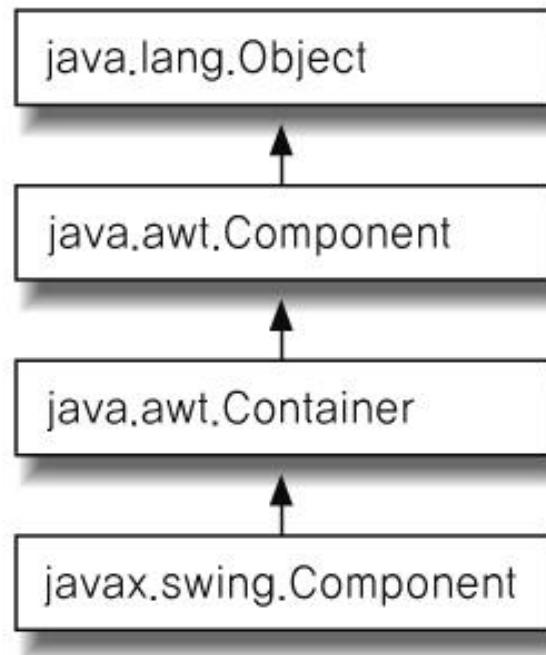
보더 종류	모양
EmptyBorder	아무런 선이나 그림이 없는 빈 공간만 차지하는 경계선이다. 빈 공간의 값은 객체를 생성할 때 Inset의 값으로 각 방향의 공간을 결정하면 된다.
EtchedBorder	평면을 마치 날카로운 끌로 판듯이 선을 보여주는 경계선이다. 경계선이 들어간 모양과 나온 모양으로 보여지게 할 수 있고 그림자나 선의 색상도 지정할 수 있다.
LineBorder	단색으로 경계선을 보여주는 경계선이다. 선의 색과 두께를 지정할 수도 있고 모서리를 둥글게 지정할 수도 있다.
BevelBorder	약간 경사지게 3차원적인 효과를 낼 수 있는 경계선이다.
SoftBevelBorder	BevelBorder와 유사하며 모서리를 부드럽게 처리한 경계선이다.
MatteBorder	이미지나 여러 가지 색상으로 매트 효과를 경계선에 보여줄 수 있는 경계선이다.
TitledBorder	경계선에 문자열을 추가하여 타이틀 형태로 보여줄 수 있는 경계선으로 문자열의 위치나 폰트를 지정할 수 있다.
CompoundBorder	두 개의 경계선을 조합해서 보여주는 경계선이다. 바깥쪽의 경계선과 안쪽의 경계선을 만든 후 CompoundBorder를 이용해서 합성하면 된다.

스윙의 특징



스윙의 기본구조

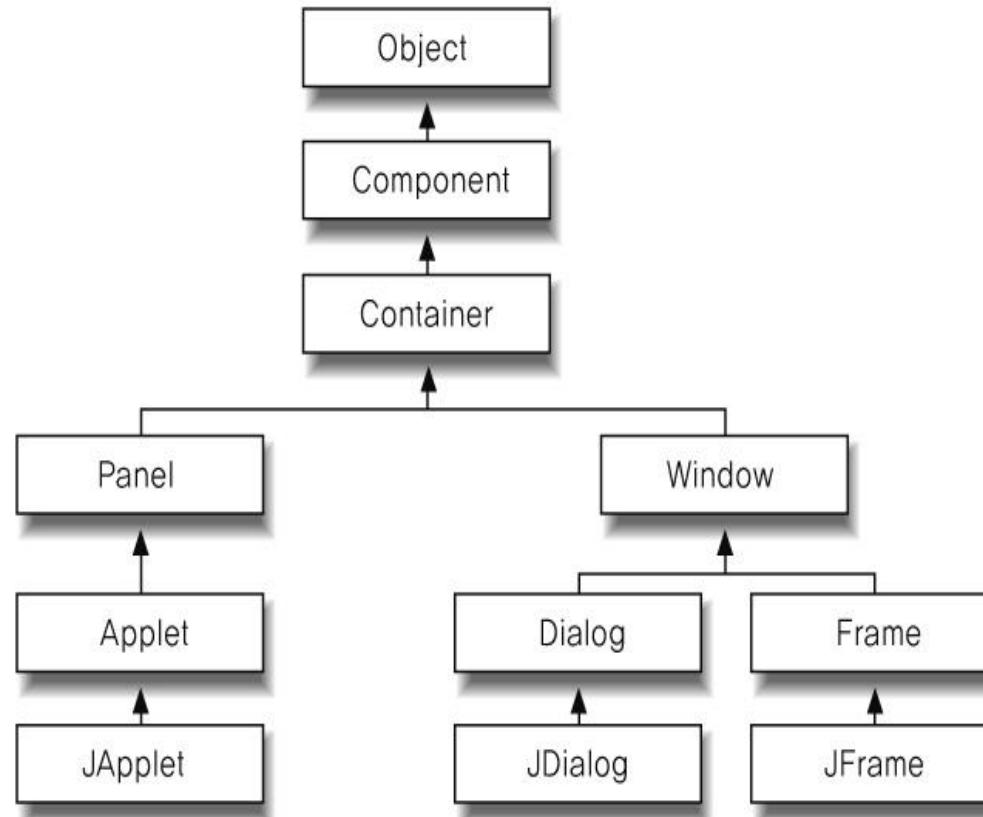
- 스윙 컴포넌트는 javax.swing.JComponent 클래스로부터 상속받은 Sub 클래스들이다.
- 컴포넌트들은 경계선(Border)이 존재하지 않지만 스윙부터는 이 경계선을 어떻게 보여줄지를 결정할 수 있다.



[그림 12-9] 스윙의 기본구조

스윙의 기본구조

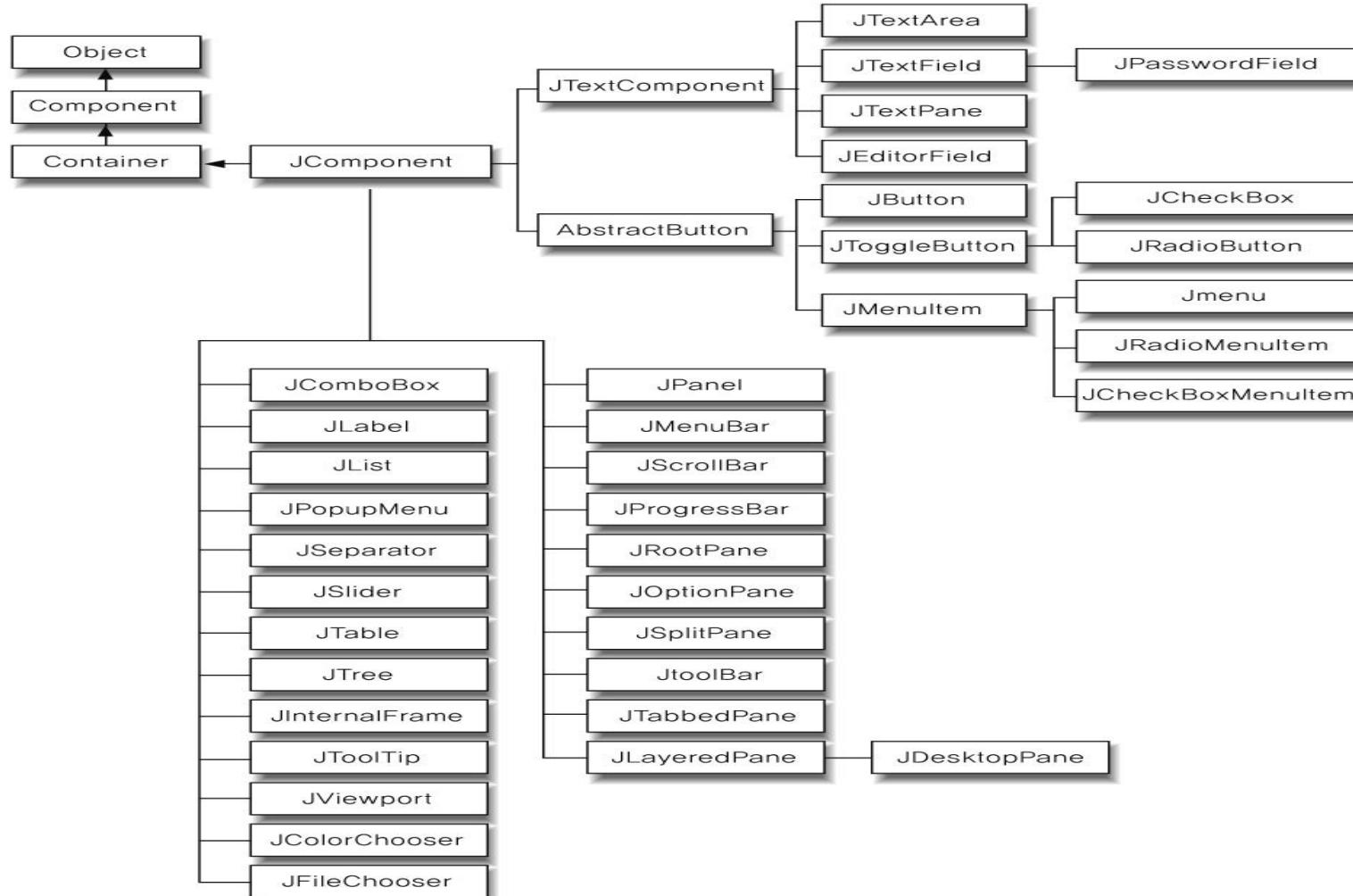
- 컨테이너 구조



[그림 12-10] 스윙 컨테이너 구조

스윙의 기본구조

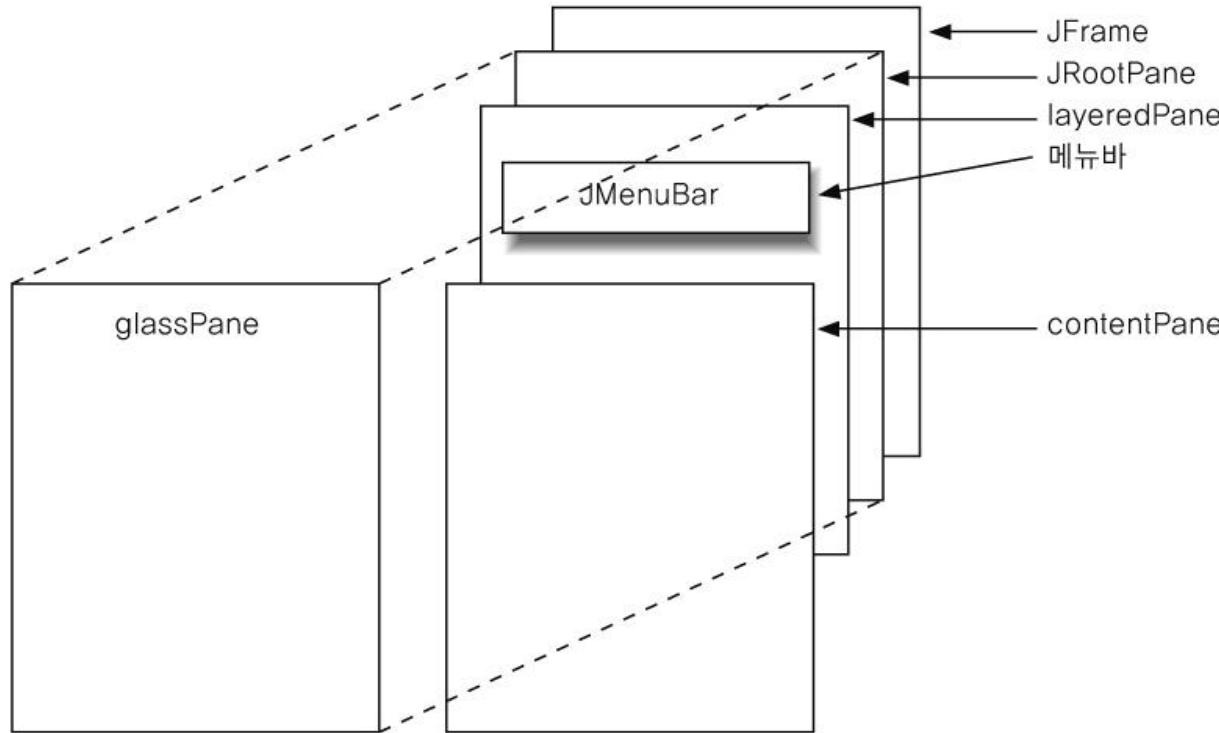
• 컴포넌트 구조



[그림 12-11] 스윙 컴포넌트 구조

스윙 컨테이너

- JFrame 클래스
 - 스윙의 JFrame은 AWT의 Frame과 달리 좀 복잡한 구조로 되어있다.
 - 프레임자체로 구성되어 있는 것이 아니라 그 안에 4개의 페인(pane)이 층으로 구성되어 있다. 다음 그림은 JFrame의 내부 구조이다.



[그림 12-12] JFrame의 내부구조

스윙 컨테이너

- JRootPane : 실질적인 윈도우 기능을 수행하는 경량의 컨테이너이다. 이러한 구조를 사용하는 것은 모든 윈도우들의 동작이나 사용방법등이 같아지고 다른 컴포넌트들과 잘 조화가 될 수 있도록 도와주기 때문이다. JRootPane은 GlassPane과 layeredPane으로 구성되어 있고, layeredPane은 JMenuBar와 contentPane을 포함한다.
- layeredPane : 루트 페인에 대해 레이어를 할 수 있도록 여러 층의 패널을 포함 할 수 있는 패널로 여러 컴포넌트들을 서로 오버랩(컴포넌트위에 다른 컴포넌트를 붙이는 작업) 할 수 있도록 한다. 윗부분은 JMenuBar와 아래부분은 contentPane으로 이루어져 있다.
- contentPane : 일반적인 컴포넌트들을 가질 수 있는 패널이다. 프레임 객체의 getContentPane()메서드를 이용해서 얻을 수 있다.
- glassPane : 기본적으로 숨겨진 상태로 되어 있으며, 다른 패널 위에 존재하는 패널이다. 주로 마우스 이벤트를 처리하기 위해 가장 먼저 루트 페인에 추가된다.
- JMenuBar : 윈도우의 메뉴를 제공하는 역할로 생략이 가능한 선택항목이다. JMenu, JMenuItem 등을 이용해서 메뉴를 구성하여 setJMenuBar()메서드를 이용해서 등록할 수 있다.

스윙 컨테이너

- JFrame 클래스의 주요메서드

[표 12-2] JFrame 클래스의 주요 메서드

반환형	메서드	설명
Container	getContentPane()	프레임의 contentPane 객체를 얻어온다.
void	setDefaultCloseOperation(int operation)	프레임의 닫기 버튼을 눌렀을 때의 기본 동작을 정한다.
	setJMenuBar(JMenuBar menubar)	프레임의 메뉴바를 주어진 메뉴바로 지정한다.

- JDK 1.4까지는 반드시 getContentPane()메서드를 통해 contentPane을 얻어 컴포넌트를 불일 수 있도록 하였지만 JDK 1.5버전부터는 AWT에서 사용했던 것 처럼 프레임에 바로 add()함수를 이용해서 불일 수 있도록 지원하고 있다.
- JFrame은 사용자가 직접 이벤트를 처리하지 않아도 종료버튼을 클릭하면 프레임이 없어진다. 이것은 기본적으로 setDefaultCloseOperation(HIDE_ON_CLOSE)라는 메서드가 실행되었기 때문이다.
- 이것은 단지 프레임을 숨기는 것으로 실제적으로 프레임의 자원을 해제하지는 않는다. 그러므로 명시적으로 종료하기 위해서는 setDefaultCloseOperation(EXIT_ON_CLOSE)를 이용해서 완전히 프레임의 자원을 해제하면서 종료시키는 것이 좋다.

스윙 컨테이너

[표 12-3] WindowConstants 인터페이스에는 사용되는 4가지 상수

상수	설명
DISPOSE_ON_CLOSE	윈도우를 종료할 때 모든 자원을 반납한다.
DO_NOTHING_ON_CLOSE	윈도우를 종료할 때 아무 일도 하지 않는다.
EXIT_ON_CLOSE	윈도우를 종료할 때 강제로 종료한다.
HIDE_ON_CLOSE	윈도우를 종료할 때 윈도우를 숨긴다.

스윙 컨테이너

- JEditorPane
 - 여러가지 형태의 포맷의 문서를 처리할 수 있도록 설계된 컴포넌트이다.
 - 현재 사용할 수 있는 텍스트 포맷은 일반 TEXT, HTML, RTF이다.

[표 12-4] JEditorPane 클래스의 생성자

생성자	설명
JEditorPane()	새로운 JEditorPane 객체를 생성한다.
JEditorPane(String url)	지정한 url의 내용을 보여주는 JEditorPane 객체를 생성한다.
JEditorPane(String type, String text)	지정한 type과 text의 내용을 보여주는 JEditorPane 객체를 생성한다.
JEditorPane(URL initialPage)	지정한 url을 보여주는 JEditorPane 객체를 생성한다.

스윙 컨테이너

[표 12-5] JEditorPane 클래스의 주요 메서드

반환형	메서드	설명
void	addHyperlinkListener (HyperlinkListener listener)	JEditorPane에서 HTML 문서 중 하이퍼링크를 클릭할 때 발생하는 이벤트를 받기 위해 지정된 HyperlinkListener를 추가한다.
URL	getPage()	현재 보여주고 있는 페이지의 URL을 얻어온다.
String	getText()	현재 보여주고 있는 페이지의 내용을 얻어온다.
void	setPage(String url)	지정한 url을 보여줄 수 있도록 현재 페이지로 지정한다.
	setPage(URL page)	지정한 url을 보여줄 수 있도록 현재 페이지로 지정한다.
	replaceSelection(String content)	현재 선택한 영역을 지정한 content 내용으로 바꾼다.
	setEditorKit(EditorKit kit)	지정한 에디터 킷으로 사용한다.

스윙 컨테이너

- JTextPane
 - 스타일이 가미된 텍스트를 보여주거나 편집할 수 있도록 제공하여 워드프로세서의 형태의 프로그램을 제작 가능한 컴포넌트이다.
 - 여기에는 다양한 폰트, 색상, 크기, 기울임, 정렬 등과 같은 서식을 사용할 수 있다.

[표 12-6] JTextPane 클래스의 생성자

생성자	설명
JTextPane()	비어있는 새로운 JTextPane 객체를 생성한다.
JTextPane(StyledDocument doc)	지정한 스타일 문서 모델을 사용하는 JTextPane 객체를 생성한다.

스윙 컨테이너

[표 12-7] JTextPane 클래스의 주요 메서드

반환형	메서드	설명
Style	addStyle(String nm, Style parent)	새로운 스타일을 논리적 스타일 계층구조에 더 한다.
	getLogicalStyle()	현재 캐럿 위치에서 사용하는 스타일을 얻어온다.
	getStyle(String nm)	지정한 이름을 가진 스타일 이전에 추가한 스타일을 얻어온다.
void	removeStyle(String url)	지정한 이름을 가진 스타일을 제거한다.
	setLogicalStyle(Style s)	현재 캐럿이 속한 문단이 사용할 스타일을 지정 한다.
StyledDocument	getStyledDocument()	현재 사용하는 스타일 문서 모델을 얻어온다.
void	insertComponent(Component c)	문서에서 현재 선택된 내용 대신에 컴포넌트를 삽입한다.
	insertIcon(Icon g)	문서에서 현재 선택된 내용 대신에 아이콘을 삽입한다.
	replaceSelection(String content)	현재 선택된 내용을 지정한 content 내용으로 바꾼다.
	setStyledDocument (StyledDocument doc)	지정한 doc 스타일 문서로 모델로 지정한다.
AttributeSet	getCharacterAttributes()	현재 캐럿이 위치한 글자의 특성을 얻어온다.
	getParagraphAttributes()	현재 캐럿이 속한 문단의 문단 특성을 얻어온다.
void	setCharacterAttributes (AttributeSet attr, boolean replace)	문자 특성을 지정한다. replace가 true이면 기존의 같은 문자 특성이 있는 경우, 새로운 특성으로 바꾼다.
	setParagraphAttributes (AttributeSet attr, boolean replace)	문단 특성을 지정한다. replace가 true이면 기존의 같은 문단 특성이 있는 경우, 새로운 특성으로 바꾼다.

스윙 컨테이너

- JScrollPane

- 스크롤을 이용해서 컴포넌트들을 보여주는 컴포넌트이다.
- 스크롤을 이용해서 보여주는 화면을 상하좌우로 이동하여 포함된 컴포넌트의 원래 크기를 유지시킬 수 있다.
- 스크롤이 필요한 컴포넌트(JList, JTextArea, JTextPane 등)는 Scrollable 인터페이스가 구현된 컴포넌트이기 때문에 이 스크롤 패널에 포함시켜서 사용한다.
- JScrollPane의 중앙을 뷰포트라고 하는데 포함된 객체를 보여주기 위한 영역을 의미한다.
- 포함된 컴포넌트의 크기가 뷰포트 영역보다 큰 경우 자동으로 스크롤 바가 생성되어 보여지게 되고, 이러한 정책을 가지고 있는 인터페이스는 ScrollPaneConstants 인터페이스에 정의되어 있다.

스윙 컨테이너

[표 12-8] ScrollPaneConstants 클래스의 주요 필드

자료형	필드명	설명
static int	HORIZONTAL_SCROLLBAR_ALWAYS	수평 스크롤바를 항상 보여주는 정책이다.
	VERTICAL_SCROLLBAR_ALWAYS	수직 스크롤바를 항상 보여주는 정책이다.
static int	HORIZONTAL_SCROLLBAR_AS_NEEDED	수평 스크롤바를 필요할 때에만 보여주는 정책이다.
	VERTICAL_SCROLLBAR_AS_NEEDED	수직 스크롤바를 필요할 때에만 보여주는 정책이다.
	HORIZONTAL_SCROLLBAR_NEVER	수평 스크롤바를 사용하지 않는 정책이다.
	VERTICAL_SCROLLBAR_NEVER	수직 스크롤바를 사용하지 않는 정책이다.

[표 12-9] JScrollPane 클래스의 주요 생성자

생성자	설명
JScrollPane()	비어있는 새로운 JScrollPane 객체를 생성한다. 기본적으로 필요할 때 수평, 수직 스크롤바가 나타난다.
JScrollPane(Component view)	지정한 뷰 객체를 보여주는 JScrollPane 객체를 생성한다. 뷰보다 컴포넌트의 내용이 크면 수평, 수직 스크롤바가 나타난다.
JScrollPane(Component view, int vsbPolicy, int hsbPolicy)	지정한 뷰 객체를 보여주는 JScrollPane 객체를 생성한다. 지정한 수직, 수평 스크롤바 표시 정책(책)에 따라 스크롤바가 보여진다.
JScrollPane(int vsbPolicy, int hsbPolicy)	지정한 수직, 수평 스크롤바 표시 정책에 따라 JScrollPane 객체를 생성한다.

스윙 컨테이너

- JTabbedPane

- JTabbedPane 클래스는 여러 패널을 담을 때 사용하는 컴포넌트이다.
- 일반적으로 기능별로 분류된 옵션들을 동시에 보여 줄 필요가 없고, 필요시 하나의 패널만 보여주기 위해서 사용을 하는 컴포넌트이다.
- 사용방법은 타이틀이나 아이콘을 가지는 탭을 클릭함으로써 여러 개의 패널 중에 선택된 탭으로 교체되면서 화면에 보여주는 컴포넌트이다.
- 탭의 위치는 상하좌우에 위치할 수 있는데 기본적으로는 패널의 왼쪽 위(**Top**)에 있다.

[표 12-10] JTabbedPane 클래스의 주요 생성자

생성자	설명
JTabbedPane()	비어있는 새로운 탭 패널 객체를 생성한다. 기본적인 탭의 위치는 Top이다.
JTabbedPane(int tabPlacement)	지정된 탭의 위치를 가지는 새로운 탭 패널 객체를 생성한다. 탭의 위치는 LEFT, RIGHT, TOP, BOTTOM값을 가질 수 있다.

스윙 컨테이너

[표 12-11] JTabbedPane 클래스의 주요 메서드

반환형	메서드	설명
Component	add(Component component)	컴포넌트의 이름으로 탭 타이틀을 정하여 component를 붙인다.
	add(Component component, int index)	컴포넌트의 이름으로 탭 타이틀을 정하고 index로 탭의 위치를 정한 후에 component를 붙인다.
void	add(Component component, constraints)	component를 탭에 붙인다. 만약, Object constraints가 문자열이거나 Icon인 경우에는 이것을 이용하여 탭 타이틀로 정한다.
	add(Component component, Object constraints, int index)	index로 탭의 위치를 정하고 component를 탭에 붙인다. 만약, constraints가 문자열이거나 Icon인 경우에는 이것을 이용하여 탭 타이틀로 정한다.
	addTab(String title, Component component)	맨 마지막 탭 위치에 지정한 타이틀을 가지고 컴포넌트를 추가한다.
	addChangeListener(ChangeListener l)	탭 패널에서 선택된 탭의 인덱스가 바뀌었을 때 발생하는 이벤트를 받기 위해 지정된 ChangeListener를 추가한다.
	addTab(String title, Icon icon, Component component)	맨 마지막 탭 위치에 지정한 타이틀과 아이콘을 가지고 컴포넌트를 추가한다.
	insertTab(String title, Icon icon, Component component, String tip, int index)	index 위치에 컴포넌트를 추가한다. 지정한 타이틀과 아이콘을 탭에 보여준다.
	remove(Component component)	지정한 컴포넌트를 탭 패널에서 제거한다.
	removeTabAt(int index)	지정한 인덱스의 탭을 탭 패널에서 제거한다.
	setComponentAt(int index, Component component)	지정한 위치의 탭이 보여줄 컴포넌트를 지정한다.

스윙 컨테이너

- JSplitPane
 - 윈도우의 내부를 두개의 패널로 상하 또는 좌우로 나뉘어 사용할 수 있도록 보여주는 컴포넌트이다.
 - 두 패널 사이에는 분리자라고 불리는 가느다란 선이 있는데 이것을 움직이게 되면 패널의 크기를 조절할 수 있다.
 - 또한 JSplitPane을 중첨해서 사용하면 여러 개의 사각형으로 분할하여 사용 할 수 있다.
 - JSplitPane 클래스는 분할자에 한번의 클릭으로 각 패널을 전체 그리고 만들 수 있도록 하는 원터치 확장(OneTouchExpandable) 옵션이 있고, 분할자의 위치를 조절할 때 연속적으로 각 패널을 다시 그리는 연속 레이아웃(ContinuousLayout) 옵션이 있다.
 - 기본 설정은 분할자를 원하는 위치로 이동시킨후 마우스를 놓으면 그 때 두 패널이 다시 그려진다.

스윙 컨테이너

[표 12-12] JSplitPane 클래스의 주요 필드

자료형	필드명	설명
static int	HORIZONTAL_SPLIT	분할 패널을 수평으로 나누어 보여주는 정책이다.
	VERTICAL_SPLIT	분할 패널을 수직으로 나누어 보여주는 정책이다.

[표 12-13] JSplitPane 클래스의 주요 생성자

생성자	설명
JSplitPane()	기본 설정으로 수평 분할, ContinuousLayout 옵션은 사용하지 않는 JSplitPane 객체를 생성한다.
JSplitPane(int newOrientation)	지정한 방향으로 나누어지고 ContinuousLayout 옵션은 사용하지 않는 JSplitPane 객체를 생성한다.
JSplitPane(int newOrientation, boolean newContinuousLayout)	지정한 방향으로 나누어지고 ContinuousLayout 옵션을 사용하는 JSplitPane 객체를 생성한다.
JSplitPane(int newOrientation, boolean newContinuousLayout, Component newLeftComponent, Component newRightComponent)	지정한 방향으로 나누어지고 ContinuousLayout 옵션을 사용하는 JSplitPane 객체를 생성한다. newLeftComponent는 왼쪽 패널에, newRightComponent는 오른쪽 패널에 보여진다.
JSplitPane(int newOrientation, Component newLeftComponent, Component newRightComponent)	지정한 방향으로 나누어지는 JSplitPane 객체를 생성한다. newLeftComponent는 왼쪽 패널에, newRightComponent는 오른쪽 패널에 보여진다.

스윙 컨테이너

[표 12-14] JSplitPane 클래스의 주요 메서드

반환형	메서드	설명
void	setContinuousLayout(boolean newContinuousLayout)	ContinuousLayout 옵션을 사용할 것인가를 지정한다.
	setOrientation(int orientation)	분할시킬 방향을 지정한다.
int	getDividerLocation()	분할자의 현재 위치를 얻어온다.
void	resetToPreferredSizes()	분할 패널의 자식 컴포넌트들의 크기를 각각의 패널 크기에 근거하여 조절한다.
void	setDividerLocation(int location)	분할자의 위치를 지정한다.
	remove(Component component)	지정한 컴포넌트를 제거한다.
	setLeftComponent(Component comp)	왼쪽 패널에 컴포넌트를 붙이도록 지정한다.
	setRightComponent(Component comp)	오른쪽 패널에 컴포넌트를 붙이도록 지정한다.
	setTopComponent(Component comp)	위 패널에 컴포넌트를 붙이도록 지정한다.
	setBottomComponent(Component comp)	아래 패널에 컴포넌트를 붙이도록 지정한다.

스윙 컨테이너

- JDesktopPane
 - JInternalFrame을 담기 위해 만들어진 가상 데스크 탑을 생성하는 컨테이너로 JInternalFrame 객체를 생성하여 JDesktopPane에 추가한다.
 - 다중 레이어를 지원하며 내부에는 DesktopManager를 가지고 있어 JInternalFrame을 관리할 수 있다.
 - 내부 프레임을 관리하는 주요 기능인 최대화, 원래크기로, 아이콘화, 이동, 크기조절, 닫기와 같은 기능을 가지고 있다.

스윙 컨테이너

[표 12-15] JDesktopPane 클래스의 주요 필드

자료형	필드명	설명
static int	LIVE_DRAG_MODE	프레임을 드래그할 때 내용을 매번 그리도록 하는 모드다.
	OUTLINE_DRAG_MODE	프레임을 드래그할 때 내용을 매번 보여주지 않고 경계선만 그리도록 하는 모드다.

[표 12-16] JDesktopPane 클래스의 주요 메서드

반환형	메서드	설명
JInternalFrame[]	getAllFrames()	데스크탑에 포함된 모든 프레임을 배열 형태로 얻어온다.
JInternalFrame	getSelectedFrame()	현재 선택된 프레임을 얻어온다. 만약, 선택된 프레임이 없으면 null을 얻어온다.
void	setDragMode(int dragMode)	지정한 모드로 드래그 모드를 지정한다.
	setSelectedFrame(JInternalFrame f)	지정한 프레임을 선택한 상태로 지정한다.

스윙 컨테이너

[표 12-17] JDesktopPane 클래스의 주요 메서드

반환형	메서드	설명
void	activateFrame(JInternalFrame f)	지정한 프레임을 활성화시키고 맨 앞으로 이동시킨다.
	closeFrame(JInternalFrame f)	지정한 프레임을 닫고 데스크탑에서 제거 한다.
	deiconifyFrame(JInternalFrame f)	지정한 프레임을 아이콘화 상태에서 원래 상태로 복구시킨다.
	iconifyFrame(JInternalFrame f)	지정한 프레임을 아이콘화 상태로 만든다.
	maximizeFrame(JInternalFrame f)	지정한 프레임을 최대화 상태로 만든다.
	minimizeFrame(JInternalFrame f)	지정한 프레임을 원래 상태로 되돌린다.
	resizeFrame(JComponent f, int newX, int newY, int newWidth, int newHeight)	지정한 프레임의 크기를 지정한다.
	setBoundsForFrame(JComponent f, int newX, int newY, int newWidth, int newHeight)	지정한 프레임의 크기를 지정한다.

스윙 컨테이너

- DefaultDesktopManager 클래스의 주요메서드

메서드	
반환형	메서드
void	activateFrame(JInternalFrame f) 지정한 프레임을 활성화시키고 맨 앞으로 이동시킨다.
void	closeFrame(JInternalFrame f) 지정한 프레임을 닫고 데스크탑에서 제거한다.
void	deiconifyFrame(JInternalFrame f) 지정한 프레임을 아이콘화 상태에서 원래 상태로 복구시킨다.
void	iconifyFrame(JInternalFrame f) 지정한 프레임을 아이콘화 상태로 만든다.
void	maximizeFrame(JInternalFrame f) 지정한 프레임을 최대화 상태로 만든다.
void	minimizeFrame(JInternalFrame f) 지정한 프레임을 원래 상태로 되돌린다.
void	resizeFrame(JComponent f, int newX, int newY, int newWidth, int newHeight) 지정한 프레임의 크기를 지정한다.
void	setBoundsForFrame(JComponent f, int newX, int newY, int newWidth, int newHeight) 지정한 프레임의 크기를 지정한다.

스윙 컨테이너

- JInternalFrame
 - 윈도우의 MDI(Multiple Document Interface) 기능을 제공하는 프레임이다.
 - 일반 프레임과 마찬가지로 타이틀바와 윈도우 제어 버튼(최대화, 원래 크기로, 닫기)을 가지고 있으며 크기 변경, 이동, 도구모음 등의 기능을 지원하고 있다.
 - 특별한 경우를 제외하고는 JDesktopPane과 함께 사용하며 외부에서 보기에는 JFrame과 유사한데 룩앤파일에 의해서 그 모양이 바뀔 수 있다.
 - JInternalFrame 클래스는 WindowEvent 대신 InternalFrameEvent를 발생하며 종료와 같은 이벤트는 InternalFrameEvent를 통해 처리하면 된다.

스윙 컨테이너

[표 12-18] JInternalFrame 클래스의 주요 생성자

생성자	설명
JInternalFrame()	타이틀이 없고 크기조절, 닫기, 최대화, 아이콘화가 되지 않는 JInternalFrame 객체를 생성한다.
JInternalFrame(String title)	지정된 타이틀을 통해 크기조절, 닫기, 최대화, 아이콘화가 되지 않는 JInternalFrame 객체를 생성한다.
JInternalFrame(String title, boolean resizable)	지정된 타이틀을 통해 크기조절 속성이 설정된 JInternalFrame 객체를 생성한다.
JInternalFrame(String title, boolean resizable, boolean closable)	지정된 타이틀을 통해 크기조절, 닫기 속성이 설정된 JInternalFrame 객체를 생성한다.
JInternalFrame(String title, boolean resizable, boolean closable, boolean maximizable)	지정된 타이틀을 통해 크기조절, 닫기, 최대화 속성이 설정된 JInternalFrame 객체를 생성한다.
JInternalFrame(String title, boolean resizable, boolean closable, boolean maximizable, boolean iconifiable)	지정된 타이틀을 통해 크기조절, 닫기, 최대화 아이콘화 속성이 설정된 JInternalFrame 객체를 생성한다.

스윙 컨테이너

[표 12-19] JInternalFrame 클래스의 주요 메서드

반환형	메서드	설명
void	addInternalFrameListener(InternalFrameListener l)	JInternalFrame에서 원도우에서 발생하는 이벤트를 받기 위해 지정된 InternalFrameListener를 추가한다.
	setDesktopIcon(JInternalFrame, JDesktopIcon d)	JDesktop에서 JInternalFrame에 사용할 아이콘을 지정한다.
	setFrameIcon(Icon icon)	지정한 아이콘을 프레임의 타이틀바에 보여줄 아이콘으로 지정한다.
	setTitle(String title)	지정한 타이틀을 프레임의 타이틀로 지정한다.
	setDefaultCloseOperation(int operation)	프레임의 닫기 버튼을 클릭했을 때의 기본 동작을 지정한다.
	dispose()	프레임을 닫고 자원을 돌려준다.
boolean	isClosed()	프레임이 닫힌 상태인지를 얻어온다.
	isIcon()	프레임이 아이콘화된 상태인지를 얻어온다.
	isMaximum()	프레임이 최대화 상태인지를 얻어온다.
	isSelected()	프레임이 현재 선택 상태인지를 얻어온다.
void	setClosed(boolean b)	프레임을 닫거나 복구시키는 것을 지정한다.
	setIcon(boolean b)	프레임을 아이콘화 시키거나 복구시키는 것을 지정 한다.
void	setSelected(boolean selected)	프레임을 선택하거나 선택되지 않은 상태로 지정한다.
	reshape(int x, int y, int width, int height)	프레임의 크기와 위치를 지정한다.
	moveToBack()	프레임이 속한 JDesktopPane의 다른 프레임보다 뒤로 이동시킨다.
	moveToFront()	프레임이 속한 JDesktopPane의 다른 프레임보다 앞으로 이동시킨다.

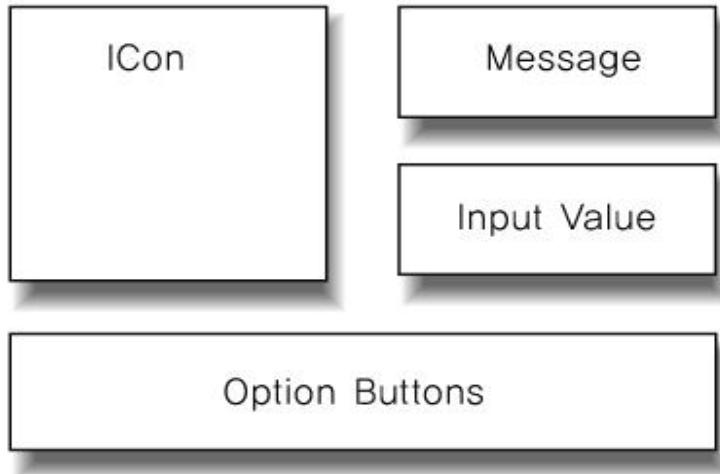
스윙 컨테이너

- JOptionPane
 - 프로그램을 실행하는 도중에 사용자로부터 데이터를 입력 받거나 특정한 메시지를 출력시켜 확인 시키는 작업들을 할 수 있도록 지원하는 컴포넌트이다.
 - 자체적인 기능을 가지고 있는 것은 아니기 때문에 객체를 생성한 후 반드시 showXXXDialog() 메서드를 이용해서 어떤 다이얼로그 박스를 출력시킬 것인가를 정해야 된다.

[표 12-20] JOptionPane 클래스의 다이얼로그 박스 종류

종류	기능	호출 함수
MessageDialog	사용자에게 메시지를 보여주는 다이얼로그 박스	showMessageDialog()
ConfirmDialog	Yes, No, Cancel과 같은 버튼으로 확인하는 다이얼로그 박스	showConfirmDialog()
InputDialog	사용자로부터 자료를 입력받기 위한 다이얼로그 박스	showInputDialog()
OptionDialog	위 세 가지를 포함하여 맞춘 다이얼로그 박스	showOptionDialog()

스윙 컨테이너



[그림 12-19] 다이얼로그 박스의 기본적인 외관

- Component parentComponent : 다이얼로그 박스를 소유하고 있는 부모가 컴포넌트이다.
- Object message : 다이얼로그 박스에서 보여줄 메시지이다.
- String title : 다이얼로그 박스에서 타이틀 부분에 보여줄 타이틀이다.
- int optionType : 다이얼로그 박스 하단에 표시되는 옵션 버튼이다.
- Icon icon : 다이얼로그 박스 좌측에 보여주는 아이콘이다.

스윙 컨테이너

○ JOptionPane 클래스의 메시지 관련 필드

자료형	필드명	설명
static int	ERROR_MESSAGE	오류 메시지를 이용한다.
	INFORMATION_MESSAGE	정보 메시지를 이용한다.
	PLAIN_MESSAGE	아이콘이 없는 메시지를 이용한다.
	QUESTION_MESSAGE	질문 메시지를 이용한다.
	WARNING_MESSAGE	경고 메시지를 이용한다.

• JOptionPane 클래스의 옵션 관련 필드

자료형	필드명	설명
static int	DEFAULT_OPTION	오류 메시지를 이용한다.
	OK_CANCEL_OPTION	정보 메시지를 이용한다.
	YES_NO_CANCEL_OPTION	아이콘이 없는 메시지를 이용한다.
	YES_NO_OPTION	질문 메시지를 이용한다.

스윙 컨테이너

○ JOptionPane 클래스의 반환값과 관련 필드

자료형	필드명	설명
static int	YES_OPTION	경고 메시지를 이용한다.
	NO_OPTION	경고 메시지를 이용한다.
	CANCEL_OPTION	경고 메시지를 이용한다.

[표 12-24] JOptionPane 클래스의 메시지ダイ얼로그 박스와 관련있는 메서드

반환형	메서드	설명
static void	showMessageDialog(Component parentComponent, Object message)	지정한 메시지를ダイ얼로그 박스에 보여주는 메시지 대화창을 만든다.
	showMessageDialog(Component parentComponent, Object message, String title, int messageType)	지정한 메시지, 타이틀, 메시지 타입을ダイ얼로그 박스에 보여주는 메시지 대화창을 만든다.
	showMessageDialog(Component parentComponent, Object message, String title, int messageType, Icon icon)	지정한 메시지, 타이틀, 메시지 타입, 아이콘을ダイ얼로그 박스에 보여주는 메시지 대화창을 만든다.

스윙 컨테이너

[표 12-25] JOptionPane 클래스의 확인 다이얼로그 박스와 관련있는 메서드

반환형	메서드	설명
static int	showConfirmDialog(Component parentComponent, Object messge)	지정한 메시지를 다이얼로그 박스에 보여주는 확인 대화창을 만든다.
	showConfirmDialog(Component parentComponent, Object messge, String title, int optionType)	지정한 메시지, 타이틀, 옵션 타입을 다이얼로그 박스에 보여주는 확인대화창을 만든다.
	showConfirmDialog(Component parentComponent, Object messge, String title, int optionType, int messageType)	지정한 메시지, 타이틀, 옵션 타입, 메시지 타입을 다이얼로그 박스에 보여주는 확인 대화창을 만든다.

[표 12-26] JOptionPane 클래스의 입력 다이얼로그 박스와 관련있는 메서드

반환형	메서드	설명
static String	showInputDialog(Component parentComponent, Object messge)	지정한 메시지를 다이얼로그 박스에 보여주는 입력 대화창을 만든다.
	showInputDialog(Component parentComponent, Object messge, String title, int messageType)	지정한 메시지, 타이틀, 메시지 타입을 다이얼로그 박스에 보여주는 입력 대화창을 만든다.

스윙 컨테이너

- JFileChooser
 - 프로그램을 실행하는 도중에 데이터를 파일로부터 불러오거나 파일에 저장할 수 있도록 파일 선택ダイ얼로그 상자를 사용할 수 있도록 만든 컴포넌트이다.
 - 파일 선택창은 FileSystemView, FileView, FileFilter 등과 같은 컨트롤러가 조합되어 만들어진다.
 - FileSystemView는 파일 시스템과 디렉토리 정보를 제공하고, FileView는 디렉토리 내부에 있는 파일들에 대한 정보를 제공하며 FileFilter는 파일을 원하는 종류만 보여줄 수 있도록 걸러주는 역할을 한다.

[표 12-27] JFileChooser 클래스의 주요 필드

자료형	필드명	설명
static int	APPROVE_OPTION	응답 버튼(yes, ok)이 선택되었을 때 반환되는 값이다.
	CANCEL_OPTION	Cancel이 선택되었을 때 반환되는 값이다.

스윙 컨테이너

[표 12-28] JFileChooser 클래스의 생성자

생성자	설명
JFileChooser()	사용자의 기본 디렉토리를 가리키는 JFileChooser 객체를 생성 한다.
JFileChooser(File currentDirectory)	지정된 디렉토리를 가리키는 JFileChooser 객체를 생성한다.
JFileChooser(FileSystemView fsv)	지정된 FileSystemView를 사용하여 JFileChooser 객체를 생성 한다.
JFileChooser(String currentDirectoryPath)	지정한 경로를 가리키는 JFileChooser 객체를 생성한다.

스윙 컨테이너

[표 12-29] JFileChooser 클래스의 주요 메서드

반환형	메서드	설명
void	addActionListener (ActionListener l)	JFileChooser에서 파일이 선택되었을 때 발생하는 이벤트를 받기 위해 지정된 ActionListener를 추가한다.
	addChoosableFileFilter (FileFilter filter)	사용자가 선택 가능한 파일 필터의 리스트에 필터를 추가한다.
	setAccessory(Jcomponent newAccessory)	액세서리 컴포넌트를 지정한다.
	setFileFilter(FileFilter filter)	파일 필터를 지정한다.
	setCurrentDirectory(File dir)	현재 디렉토리를 지정한다.
	setDialogTitle(String dialogTitle)	다이얼로그의 타이틀을 지정한다.
int	setOpenDialog(Component parent)	열기용 JFileChooser 다이얼로그 박스를 만든다.
	setSaveDialog(Component parent)	저장용 JFileChooser 다이얼로그 박스를 만든다.
void	setFileSelectionMode(int mode)	사용자가 선택할 수 있는 것을 파일만, 디렉토리만, 또는 파일과 디렉토리의 양쪽 모두를 선택할 수 있게 할 수 있도록 JFileChooser를 설정한다.
	setMultiSelectionEnabled(boolean b)	다중 파일 선택 모드를 지정한다.
	setSelectedFile(File file)	지정한 파일을 선택 상태로 만든다.
	setSelectedFiles(File[] selectedFiles)	지정한 파일들을 선택 상태로 만든다.
File	getSelectedFile()	선택한 파일을 얻어온다.
File[]	getSelectedFiles()	다중 선택 모드일 때 선택한 파일들을 얻어온다.

스윙 컨테이너

- JColorChooser
 - 사용자가 색상을 선택할 수 있도록 기능을 제공하는 컴포넌트이다.
 - 일반적으로는 `showDialog()`메서드를 이용해서 화면에 출력을 시키는데 이런 방법외에 `createDialog()`메서드를 이용할 수도 있고, 다른 윈도우나 컨테이너에 포함시켜 사용할 수도 있다.

[표 12-30] JColorChooser 클래스의 생성자

생성자	설명
<code>JColorChooser()</code>	초기 색을 하얀색으로 가지는 JColorChooser 객체를 생성한다.
<code>JColorChooser(Color initialColor)</code>	초기 색을 지정한 색으로 가지는 JColorChooser 객체를 생성한다.
<code>JColorChooser (ColorSelectionModel model)</code>	지정된 ColorSelectionModel로 JColorChooser 객체를 생성한다.

스윙 컨테이너

[표 12-31] JColorChooser 클래스의 주요 메서드

반환형	메서드	설명
static JPanel	createDialog (Component c, String title, boolean modal, JColorChooser chooserPane, ActionListener okListener, ActionListener cancelListener)	“OK”, “Cancel” 및 “Reset” 버튼과 함께 지정된 JColorChooser 객체를 생성하여 그 객체를 얻어 온다.
static Color	showDialog(Component component, String title, Color initialColor)	모달 모드의 JColorChooser 객체를 생성하여 보여주고 다이얼로그 박스가 숨겨질 때까지 블록 상태가 된다.
Color	getColor()	현재 선택된 색을 얻어온다.
void	setColor(Color color)	지정한 색을 선택 상태로 만든다.

스윙 컴포넌트

- AbstractButton 클래스

- JButton, JToggleButton, JRadioButton 등의 모든 버튼의 종류를 추상화한 클래스로 대부분의 버튼이 이 클래스로부터 상속을 받는다.
- 버튼들의 모양이나 기능은 다르지만 기본적인 기능들은 이 클래스 구현되어 있다.

[표 12-32] AbstractButton 클래스의 주요 메서드

반환형	메서드	설명
void	setModal(ButtonModel newModel)	버튼의 모델을 정한다.
	setMarginInsets(m)	버튼의 경계선과 텍스트와의 경계를 정한다.
	setRolloverEnabled(boolean b)	롤오버 효과의 활성화 상태를 정한다.
	setIcon(Icon defaultIcon)	버튼의 기본 아이콘을 정한다.
	setSelectedIcon(Icon selectedIcon)	버튼이 선택된 상태의 아이콘을 정한다.
	setDisableIcon(Icon disabledIcon)	버튼이 비활성화된 상태의 아이콘을 정한다.
	setDisableSelectedIcon(Icon disabledSelectedIcon)	버튼이 비활성화되고 선택된 상태의 아이콘을 정한다.
	setPressedIcon(Icon pressedIcon)	버튼이 눌려진 상태의 아이콘을 정한다.
	setRolloverIcon(Icon rolloverIcon)	마우스가 버튼 위에 놓여진 상태의 아이콘을 정한다.
void	setHorizontalTextPosition(int textPostion)	버튼의 수평 텍스트의 위치를 지정한다.
	setVerticalTextPosition(int textPostion)	버튼의 수직 텍스트의 위치를 지정한다.

스윙 컴포넌트

- JButton 클래스
 - JButton 클래스는 AWT의 버튼과 거의 유사하며 버튼에 텍스트와 이미지를 모두 넣을 수 있고 다양한 상태에서 보여줄 버튼의 이미지를 각각 지정하여 롤어버(Roll-Over) 버튼을 만들 수도 있다.

[표 12-33] JButton 클래스의 생성자

생성자	설명
JButton(Icon icon)	지정한 아이콘을 보여주는 버튼 객체를 생성한다.
JButton(String text, Icon icon)	지정한 텍스트와 아이콘을 보여주는 버튼 객체를 생성한다.

스윙 컴포넌트

- `JToggleButton` 클래스
 - 버튼의 상태를 기본과 선택된 상태 두 가지를 가지는 버튼으로 선택된 상태를 계속 유지하고 있다.
 - 기본 상태에서 한 번 클랙하면 선택상태로 계속 유지하고 있다가 다시 선택하면 기본 상태를 계속 유지할 수 있는 버튼이다.
 - 일반적으로 기본 상태와 선택된 상태를 구분하기 위해서 서로 다른 아이콘을 지정하여 사용을 한다.

[표 12-34] `JToggleButton` 클래스의 생성자

생성자	설명
<code>JToggleButton(Icon icon)</code>	지정한 아이콘을 보여주는 토글 버튼 객체를 생성한다.
<code>JToggleButton(Icon icon, boolean selected)</code>	지정한 아이콘을 보여주며 버튼의 선택 여부를 지정하여 토글 버튼 객체를 생성한다.
<code>JToggleButton(String text)</code>	지정한 텍스트를 보여주는 토글 버튼 객체를 생성한다.
<code>JToggleButton(String text, boolean selected)</code>	지정한 텍스트를 보여주며 버튼의 선택 여부를 지정하여 토글 버튼 객체를 생성한다.

스윙 컴포넌트

- `JRadioButton` 클래스
 - AWT의 `Checkbox` 클래스를 이용한 라디오형 체크박스와 유사한 형태의 컴포넌트이다.
 - 여러 항목 중에서 하나의 항목만 선택할 수 있도록 만든 컴포넌트로 여러 개의 항목을 그룹으로 묶어주기 위하여 `ButtonGroup` 클래스를 이용해서 그룹으로 묶어준다.

[표 12-35] `JRadioButton` 클래스의 생성자

생성자	설명
<code>JRadioButton(Icon icon)</code>	지정한 아이콘을 보여주는 라디오 버튼 객체를 생성한다.
<code>JRadioButton(Icon icon, boolean selected)</code>	지정한 아이콘을 보여주며 버튼의 선택 여부를 지정하여 라디오 버튼 객체를 생성한다.
<code>JRadioButton(String text)</code>	지정한 텍스트를 보여주는 라디오 버튼 객체를 생성한다.
<code>JRadioButton(String text, boolean selected)</code>	지정한 텍스트를 보여주며 버튼의 선택 여부를 지정하여 라디오 버튼 객체를 생성한다.

스윙 컴포넌트

- JCheckBox 클래스
 - AWT의 Checkbox 클래스의 기능과 유사하고, 컴포넌트에 텍스트와 이미지를 모두 넣을 수 있다.

[표 12-36] JCheckBox 클래스의 생성자

생성자	설명
JCheckBox(Icon icon)	지정한 아이콘을 보여주는 체크박스 객체를 생성한다.
JCheckBox(Icon icon, boolean selected)	지정한 아이콘을 보여주며 버튼의 선택 여부를 지정하여 체크박스 객체를 생성한다.
JCheckBox(String text)	지정한 텍스트를 보여주는 체크박스 객체를 생성한다.
JCheckBox(String text, boolean selected)	지정한 텍스트를 보여주며 버튼의 선택 여부를 지정하여 체크박스 객체를 생성한다.

스윙 컴포넌트

- JLabel 클래스
 - AWT의 Label 클래스의 기능과 유사하고, 컴포넌트에 텍스트와 이미지를 모두 넣을 수 있다.
 - 텍스트와 이미지를 붙일 때 정렬하는 방식이 좀 복잡하다. 여기서 말하는 정렬이라 함은 자체 정렬 방식과 텍스트와 이미지 사이의 정렬 상태를 의미한다.

[표 12-37] JLabel 클래스의 생성자

생성자	설명
JLabel(Icon image)	지정한 아이콘을 보여주는 레이블 객체를 생성한다.
JLabel(Icon image, int horizontalAlignment)	지정한 아이콘을 지정한 수평정렬 방식에 따라 보여주는 레이블 객체를 생성한다.
JLabel(String text, int horizontalAlignment)	지정한 문자열을 지정한 수평정렬 방식에 따라 보여주는 레이블 객체를 생성한다.
JLabel(String text, Icon icon, int horizontalAlignment)	지정한 텍스트와 지정한 아이콘을 지정한 수평정렬 방식에 따라 보여주는 레이블 객체를 생성한다.

스윙 컴포넌트

[표 12-38] JLabel 클래스의 주요 메서드

반환형	메서드	설명
void	setHorizontalAlignment(int alignment)	수평 정렬 상태를 정한다.
	setVerticalAlignment(int alignment)	수직 정렬 상태를 정한다.
	setHorizontalTextPosition(int textPosition)	아이콘에 대한 텍스트의 수평 위치를 정한다.
	setVerticalTextPosition(int textPosition)	아이콘에 대한 텍스트의 수직 위치를 정한다.
	setIconTextGap(int iconTextGap)	아이콘과 텍스트 사이의 간격을 지정한다.

스윙 컴포넌트

- `JList` 클래스

- AWT의 `List` 클래스의 기능과 유사하지만 스크롤바가 자동으로 생성되지 않는다는 차이점이 있다.
- 그래서 `JList`를 `JScrollPane`에 포함시켜 자동으로 스크롤할 수 있도록 지원한다.
- 컨트롤러를 자체적으로 가지고 있는 컴포넌트로 아이템을 직접 컴포넌트에 추가할 수 있는 모델 접근 메서드를 지원하지 않는다.
- 아이템을 추가하기 위해서는 아이템을 저장할 수 있는 모델이 있어야 하는데 주로 배열이나 `Vector`를 이용한다.
- 모델이나 컨트롤러를 지정하지 않으면 `Default`가 붙는 기본 모델과 컨트롤러가 만들어진다.

[표 12-39] `JList` 클래스의 생성자

생성자	설명
<code>JList(ListModel dataModel)</code>	지정한 리스트 모델을 사용하는 리스트 객체를 생성한다.
<code>JList(Object[] listData)</code>	지정한 배열의 자료를 보여주는 리스트 객체를 생성한다.
<code>JList(Vector listData)</code>	지정한 벡터의 자료를 보여주는 리스트 객체를 생성한다.

스윙 컴포넌트

[표 12-40] JList 클래스의 주요 메서드

반환형	메서드	설명
void	addListSelectionListener(ListSelectionListener listener)	리스트에서 선택하는 항목이 바뀌었을 때 발생하는 이벤트를 받기 위해 지정된 ListSelectionListener를 추가한다.
	ensureIndexIsVisible(int index)	지정한 인덱스의 아이템이 보여줄 수 있도록 리스트를 스크롤시킨다.
	setSelectionBackground(Color selectionBackground)	선택된 아이템의 배경색을 지정한다.
	setSelectionForeground(Color selectionForeground)	선택된 아이템의 전경색을 지정한다.
	setSelectionModel(ListSelectionModel selectionModel)	지정된 ListSelectionModel로 지정한다.
	setSelectionMode(int selectionMode)	아이템의 선택모드를 selectionMode로 지정한다. • SINGLE_SELECTION: 한번에 하나의 아이템만 선택 • SINGLE_INTERVAL_SELECTION: 연속된 아이템 선택 • MULTIPLE_INTERVAL_SELECTION: 불연속된 아이템 선택
boolean	isSelectionEmpty()	선택된 아이템이 있는지 검사하여 있으면 true, 없으면 false으로 얻어온다.
int	getSelectedIndex()	선택된 첫 번째 아이템의 인덱스를 얻어온다.
Int[]	getSelectedIndices()	선택된 모든 아이템의 인덱스를 배열로 얻어온다.
Object	getSelectedValue()	선택된 아이템을 얻어온다.
Object[]	getSelectedValues()	선택된 모든 아이템을 배열로 얻어온다.
void	setSelectedIndex(int index)	지정된 인덱스의 아이템을 선택된 상태로 만든다.
void	setSelectedValue(Object anObject, boolean shouldScroll)	지정된 객체를 가진 아이템을 선택된 상태로 만들고 shouldScroll이 true면 그 아이템이 보이도록 스크롤 한다.
	setModel(ListModel model)	지정된 모델로 리스트의 모델을 지정한다.
	setListData(Object[] listData)	지정된 배열 자료를 사용하는 모델을 만들고 지정한다.
	setListData(Vector listData)	지정된 벡터 자료를 사용하는 모델을 만들고 지정한다.

스윙 컴포넌트

- JComboBox 클래스
 - AWT의 Choice 클래스의 기능과 유사하고, 컴포넌트에 텍스트와 이미지를 모두 넣을 수 있다.
 - 주의할 점은 사용자가 아이템을 선택했을 때 ItemEvent가 발생되고, 사용자가 선택한 아이템을 바꾸면 ActionEvent가 발생된다.
 - 즉, AWT에서는 ItemEvent가 발생하였지만 스윙에서는 발생되지 않는다.

[표 12-41] JComboBox 클래스의 생성자

생성자	설명
JComboBox(ComboBoxModel aModel)	지정한 콤보박스 모델을 사용하는 콤보박스 객체를 생성한다.
JComboBox(Object[] items)	지정한 배열의 자료를 보여주는 콤보박스 객체를 생성한다.
JComboBox(Vector items)	지정한 벡터의 자료를 보여주는 콤보박스 객체를 생성한다.

스윙 컴포넌트

[표 12-42] JComboBox 클래스의 주요 메서드

반환형	메서드	설명
void	addActionListener(ActionListener l)	콤보박스에서 아이템을 선택하였을 때 발생하는 이벤트를 받기 위해 지정된 ActionListener를 추가한다.
	addItemListener(ItemListener aListener)	콤보박스에서 아이템의 선택이 바꿨을 때 발생하는 이벤트를 받기 위해 지정된 ItemListener를 추가한다.
	showPopup()	팝업 윈도우를 보이게 한다.
void	hidePopup()	팝업 윈도우를 숨긴다.
Object	getItemAt(int index)	지정한 인덱스의 아이템을 얻어온다.
void	addItem(Object anObject)	지정한 아이템을 목록에 추가한다.
	insertItemAt(Object anObject, int index)	지정한 아이템을 지정한 인덱스 위치에 추가한다.
	removeAllItems()	모든 아이템을 제거한다.
	removeItem(Object anObject)	지정된 아이템을 목록에서 제거한다.
	removeItemAt(int anIndex)	지정된 인덱스의 아이템을 목록에서 제거한다.
	setEditable(Boolean aFlag)	콤보박스의 편집 가능 상태를 지정한다.

스윙 컴포넌트

- `JTextComponent` 클래스
 - `JTextComponent` 클래스는 모든 스윙 텍스트 컴포넌트의 Super 클래스로 텍스트 편집에 대한 기능(Copy, Cut, Paste 등)을 제공한다.

[표 12-43] `JTextComponent` 클래스의 주요 메서드

반환형	메서드	설명
String	<code>getText(int offs, int len)</code>	지정한 offs의 위치부터 지정한 len 영역까지의 텍스트를 얻어온다.
void	<code>copy()</code>	선택된 텍스트를 시스템 클립보드에 복사한다.
	<code>cut()</code>	선택된 텍스트를 시스템 클립보드에 이동시킨다.
	<code>paste()</code>	시스템 클립보드에의 내용을 현재 캐럿의 위치에 붙여 넣는다.
	<code>replaceSelection(String content)</code>	선택된 텍스트를 지정한 content로 바꾼다.
	<code>setSelectedTextColor(Color c)</code>	선택된 텍스트의 색상을 지정한다.
	<code>setSelectionColor(Color c)</code>	선택된 영역의 색상을 지정한다.

스윙 컴포넌트

- JTextField 클래스와 JPasswordField 클래스
 - JTextField 클래스는 AWT의 TextField 클래스와 거의 기능이 유사하며 그 중 반향 문자를 설정해서 암호를 입력했던 것을 스윙에서는 JPasswordField 클래스가 암호를 입력할 때 다른 문자로 바꾸어 보여 주는 기능을 가지는 컴포넌트이다.

[표 12-44] JPasswordField 클래스의 주요 메서드

반환형	메서드	설명
char[]	getPassword()	현재 입력된 암호를 얻어온다.
char	getEchoChar()	사용자가 입력한 문자를 반향 문자로 화면에 나타낸다.
void	setEchoChar(char c)	지정한 문자를 반향 문자로 지정한다.

스윙 컴포넌트

- JTextArea 클래스

- AWT의 TextArea 클래스와 거의 기능이 유사하며 자체적으로 스크롤을 처리하지 않으므로 JScrollPane에 포함시켜 스크롤을 지원한다.
- 한 줄의 폭이 텍스트 영역의 폭보다 클 때 자동으로 줄바꿈을 해주지 않으므로 옵션을 통하여 설정을 해야한다.
- 줄바꿈을 할때는 보통 문자 단위로 하지만 setWrapStyleWord() 메서드를 이용하면 단어 단위로 줄바꿈할 수도 있다.

[표 12-45] JTextArea 클래스의 주요 메서드

반환형	메서드	설명
void	setLineWrap(boolean wrap)	줄바꿈 정책을 사용할지를 지정한다.
	setWrapStyleWord(Boolean word)	단어 단위로 줄바꿈 정책을 사용할지를 지정한다.
	setTabSize(int size)	탭의 크기를 지정한다.

스윙 컴포넌트

- JSlider 클래스
 - 사용자가 최대 값과 최소값으로 정해진 범위 내에서 값을 지정하여 사용하는 컴포넌트이다.
 - 수직, 수평으로 모양을 지정할 수 있으며, 필요시 눈금과 값을 표시할 수 있다.
 - 눈금(Tick)은 주눈금과 보조눈금으로 나누어 사용할 수 있다.

[표 12-46] JSlider 클래스의 생성자

생성자	설명
JSlider()	범위가 0~100, 초기값이 50으로 설정된 수평 슬라이드 객체를 생성한다.
JSlider(int orientation)	지정한 방향을 가진 슬라이드를 범위가 0~100, 초기값이 50으로 설정된 객체를 생성한다.
JSlider(int min, int max)	지정한 최소값과 최대값의 범위를 가지고 초기값이 50으로 설정된 슬라이드 객체를 생성한다.
JSlider(int min, int max, int value)	지정한 최소값과 최대값의 범위를 가지고 지정한 초기값으로 설정된 슬라이드 객체를 생성한다.
JSlider(int orientation, int min, int max, int value)	지정한 방향과 지정한 최소값과 최대값의 범위를 가지고 지정한 초기값으로 설정된 슬라이드 객체를 생성한다.

스윙 컴포넌트

[표 12-47] JSlider 클래스의 주요 메서드

반환형	메서드	설명
void	addChangeListener (ChangeListener l)	슬라이드에서 값이 변할 때 발생하는 이벤트를 받기위해 지정된 ChangeListener를 추가한다.
Int	getValue()	슬라이드 값을 얻어온다.
	getMaximum()	슬라이드에서 제공되는 최대값을 얻어온다.
	getMinimum()	슬라이드에서 제공되는 최소값을 얻어온다.
void	setMaximum(int maximum)	모델의 최대 속성을 지정한다.
	setMinimum(int minimum)	모델의 최소 속성을 지정한다.
	setOrientation(int orientation)	슬라이드의 수평, 수직 방향을 지정한다.
	setMajorTickSpacing(int n)	슬라이드의 주 눈금 간격을 지정한다.
	setMinorTickSpacing(int n)	슬라이드의 보조 눈금 간격을 지정한다.

스윙 컴포넌트

- JProgressBar 클래스
 - 우리가 보통 프로그램을 설치할 때나 인터넷 익스플로러에서 페이지가 열릴 때 오랜 시간이 걸리 때 작업의 진행 상태를 나타낼 수 있는 컴포넌트이다.
 - 슬라이드와는 달리 사용자가 직접 값을 변경시킬 수는 없고 보여주는 기능만 가지고 있다.

[표 12-48] JProgressBar 클래스의 생성자

생성자	설명
JProgressBar()	보더를 보여주지만 진행 문자열이 없는 수평 진행바 객체를 생성한다.
JProgressBar(int orient)	지정된 방향을 가지는 진행바 객체를 생성한다.
JProgressBar(int min, int max)	지정된 최소값과 최대값을 가지는 수평 진행바 객체를 생성한다.
JProgressBar(int orient, int min, int max)	지정한 방향과 지정한 최소값과 최대값이 있는 진행바 객체를 생성한다.

스윙 컴포넌트

[표 12-49] JProgressBar 클래스의 주요 메서드

반환형	메서드	설명
void	addChangeListener(ChangeListener l)	진행바에서 값이 변할 때 발생하는 이벤트를 받기 위해 지정된 ChangeListener를 추가한다.
double	getPercentComplete()	현재 진행바로부터 백분율값으로 얻어온다.
Int	getValue()	진행바의 값을 얻어온다.
	getMaximum()	진행바의 최대값을 얻어온다.
	getMinimum()	진행바의 최소값을 얻어온다.
void	setMaximum(int n)	진행바의 최대값을 지정한다.
	setMinimum(int n)	진행바의 최소값을 지정한다.
	setOrientation(int newOrientation)	진행바의 수평, 수직 방향을 지정한다.
	setString(String s)	진행바와 같이 보여줄 문자열을 지정한다.
	setValue(int n)	진행바의 현재 값을 지정한다.

스윙 컴포넌트

- JToolBar 클래스
 - 자주 사용하는 메뉴를 버튼 형태의 모양으로 보여주는 컴포넌트이다.
 - 부모 컴포넌트로 분리되거나 도킹될 수 있는데 이 기능을 사용하기 위해서는 부모 컴포넌트는 반드시 BorderLayout이어야 한다.

[표 12-50] JToolBar 클래스의 생성자

생성자	설명
JToolBar()	새로운 툴바 객체를 생성한다. 수평 방향으로 만들어지는 것이 기본이다.
JToolBar(String name)	지정된 이름을 가지는 새로운 툴바 객체를 생성한다.
JToolBar(int orientation)	지정된 방향을 가지고 새로운 툴바 객체를 생성한다.
JToolBar(String name, int orientation)	지정된 이름과 방향을 가지고 새로운 툴바 객체를 생성한다.

스윙 컴포넌트

[표 12-51] JToolBar 클래스의 주요 메서드

반환형	메서드	설명
void	addSeparator()	구분자를 기본 사이즈로 툴바 끝에 추가한다.
	addSeparator(Dimension size)	지정한 크기로 구분자를 툴바 끝에 추가한다.
	setFloatable(boolean b)	툴바를 이동시킬 수 있게 만들 것인지를 지정한다.
	setMarginInsets(n)	툴바와 붙이는 컴포넌트 사이의 경계선의 간격을 지정한다.
	setOrientation(int o)	툴바의 수평, 수직 방향을 지정한다.

스윙 컴포넌트

- JTree 클래스
 - 데이터를 계층적인 노드 형식으로 보여주는 컴포넌트이다.
 - 윈도우에서 탐색기를 실행 시켰을 때 볼 수 있는 것처럼 하나의 부모 노드에 여러 개의 자식노드가 계층적으로 연결된 상태로 보여주는 것이다.
 - 데이터를 표현하는 하나의 객체를 노드라하고 가장 최상위에 있는 노드를 루트노드라 한다.
 - 루트노드밑에 있는 것을 자식노드라 하는데 자식노드는 가질수도 있지만 없을 수도 있다.
 - 자식노드는 또 다른 자식노드를 가질수도 있지만 없을 수도 있다. 일 반적으로 자식노드가 존재하지 않는 노드를 리프(leaf)라 부른다.
 - JTree에서 노드들은 TreeModel에 의해 다루어진다.
 - TreeModel는 랜더러와 에이터 등 각종 컨트롤러들이 관련되어 있다.
 - 이 모델은 트리에서 노드를 나타내기 위해 어떤 타입의 객체든지 사용할 수 있도록 지원하고 있다.
 - TreeCellRenderer라고 하는 특별한 객체가 있는데, 이것은 시각적으로 표현하는 객체로 변환하도록 지원해 준다.

스윙 컴포넌트

- 기본적으로 설정되어 있는 경우는 DefaultTreeCellRenderer가 되고 이는 JLabel을 확장해서 구현한 것이다.
- 노드가 선택된 상태를 기록, 제어하는데 사용되는 모델은 TreeSelectionModel이다.
- 기본적인 트리 구조를 만드는 방법은 TreeNode를 이용하는데, 이 경우 계층적인 구조를 가지는 트리 모델을 잘 표현해 낼 수가 있다.
- 사용자가 직접 트리 모델을 구현해서 트리를 생성할 수도 있다. TreeNode나 모델을 만들지 않고 좀 쉬운 방법을 이용해서 트리를 만드는 방법은 배열이나 벡터, 해쉬테이블을 이용하는 방법이다.
- 좀 더 강력한 트리구조를 만들고 싶다면 DefaultMutableTreeNode를 이용하는데, 이 클래스는 TreeNode 인터페이스를 구현한 클래스로 트리 구조를 만들기 위해 사용할 수 있는 유용한 메서드를 지원한다.

스윙 컴포넌트

[표 12-52] JTree 클래스의 생성자

생성자	설명
JTree()	샘플 모델을 가지는 새 트리 객체를 생성한다.
JTree(Hashtable<?,?> value)	해시 테이블로부터 생성된 트리 객체를 생성하고 루트노드는 보여주지 않는다.
JTree(Object[] value)	지정한 배열의 자료를 보여주는 트리 객체를 생성하고 루트노드는 보여주지 않는다.
JTree(TreeModel newModel)	지정된 트리 모델을 보여주는 트리 객체를 생성한다.
JTree(TreeNode root)	지정한 노드를 루트노드로 보여주는 트리 객체를 생성한다.
JTree(Vector<?> value)	벡터로부터 생성된 트리 객체를 생성하고 루트노드는 보여주지 않는다.

스윙 컴포넌트

[표 12-53] JTree 클래스의 모델, 컨트롤러와 관련있는 메서드

반환형	메서드	설명
void	setModel(TreeModel newModel)	트리 모델을 지정한다.
	setCellEditor(TreeCellEditor cellEditor)	노드를 편집할 때 사용할 셀 에디터를 지정 한다.
	setCellRenderer(TreeCellRenderer x)	노드를 보여줄 때 사용할 셀 랜더러를 지정 한다.
	setSelectionModel(TreeSelectionModel selectionModel)	트리의 선택 모델을 지정한다.
TreeSelectionModel	getSelectionModel()	트리로부터 선택한 모델을 얻어온다.

스윙 컴포넌트

[표 12-54] JTree 클래스의 화면 출력과 관련있는 메서드

반환형	메서드	설명
boolean	isVisible(TreePath path)	지정된 경로가 현재 보여주고 있는지를 알려준다. 경로가 노드의 루트이거나 부모가 확장된 상태이면 true를 얻어온다.
void	makeVisible(TreePath path)	지정한 경로의 노드가 표시되도록 만든다.
	scrollPathToVisible(TreePath path)	지정한 경로의 모든 컴포넌트들을 확장시키고(마지막에 경로에 있는 컴포넌트는 제외), 경로에 지정된 노드를 보여지도록 그 위치로 스크롤시킨다.
	scrollRowToVisible(int row)	지정된 행의 아이템이 보여질 때까지 스크롤한다.
	setRootToVisible(Boolean rootVisible)	루트 노드가 가시 상태인지 아닌지를 정한다.

스윙 컴포넌트

[표 12-55] JTree 클래스의 경로와 관련있는 메서드

반환형	메서드	설명
Int	getRowCount()	현재 보여주고 있는 트리의 행수를 얻어온다.
TreePath	getPathForRow(int row)	지정된 행의 경로를 얻어온다.
Int	getRowForPath(TreePath path)	지정된 경로에 의해 식별되는 노드를 보여주는 행을 얻어온다.
	getRowForLocation (int x, int y)	지정된 경로의 행을 얻어온다.
TreePath	getPathForLocation(int x, int y)	지정된 위치의 경로를 얻어온다.

스윙 컴포넌트

[표 12-56] JTree 클래스의 노드 확장, 축소와 관련있는 메서드

반환형	메서드	설명
void	collapsePath(TreePath path)	지정된 경로를 축소시킨다.
	collapseRow(int row)	지정된 행을 축소시킨다.
	expandPath(TreePath path)	지정된 경로를 확장시킨다.
	expandRow(int row)	지정된 행을 확장시킨다.
void	setToggleClickCount(int clickCount)	노드를 확장하거나 축소시킬 때 필요한 마우스 클릭수를 지정한다. 기본설정은 더블클릭이다.

스윙 컴포넌트

[표 12-57] JTree 클래스의 노드 선택과 관련있는 메서드

반환형	메서드	설명
Int	getSelectionCount()	선택된 노드의 수를 얻어온다.
TreePath	getSelectionPath()	선택된 첫 번째 노드의 경로를 얻어온다.
TreePath[]	getSelectionPaths()	선택된 모든 노드의 경로를 배열로 얻어온다.
Int[]	getSelectionRows()	guswo 선택된 행을 얻어온다.

스윙 컴포넌트

[표 12-58] JTree 클래스의 이벤트와 관련있는 메서드

반환형	메서드	설명
Int	addTreeExpansionListener (TreeExpansionListener tel)	트리가 확장되거나 축소될 때 발생하는 이벤트를 받기 위해 지정된 TreeExpansionListener를 추가한다.
	addTreeSelectionListener (TreeSelectionListener tsl)	트리에서 선택된 노드가 바뀔 때 발생하는 이벤트를 받기 위해 지정된 TreeSelectionListener를 추가한다. 단, 노드를 클릭하는 것 자체로는 이벤트가 발생하지 않는다.
	addTreeWillExpandListener (TreeWillExpandListener tel)	트리에서 확장되거나 축소되기 직전에 발생하는 이벤트를 받기 위해 지정된 TreeWillExpandListener를 추가한다.

스윙 컴포넌트

[표 12-59] DefaultMutableTreeNode 클래스의 생성자

생성자	설명
DefaultMutableTreeNode()	사용자 객체가 아닌 트리노드를 생성한다.
DefaultMutableTreeNode(Object userObject)	지정된 사용자 객체를 가지는 트리노드를 생성한다.
DefaultMutableTreeNode(Object userObject, Boolean allowsChildren)	지정된 사용자 객체를 가지는 트리노드를 생성하되, 자식 노드를 허용할 것인지를 지정한다.

스윙 컴포넌트

[표 12-60] DefaultMutableTreeNode 클래스의 주요 메서드

반환형	메서드	설명
void	remove(int childIndex)	지정된 위치의 자식 노드를 제거한다.
	remove(MutableTreeNode aChild)	지정된 자식 노드를 제거한다.
	removeAllChildren()	모든 자식 노드를 제거한다.
	removeFromParent()	현재 노드를 부모 노드로부터 분리시킨다.
	setParent(MutableTreeNode newParent)	현재 노드를 newParent 노드의 자식 노드로 만든다.
	getChildrenCount()	자식 노드의 개수를 얻어온다.
Enumeration	children()	자식 노드를 모두 얻어온다.
TreeNode	getChildAt(int index)	지정한 위치의 자식 노드를 얻어온다.
	getParent()	현재 노드를 가지고 있는 부모 노드를 얻어온다.
	getRoot()	현재 노드를 가지고 있는 루트 노드를 얻어온다.

스윙 컴포넌트

[표 12-61] DefaultTreeModel 클래스의 생성자

생성자	설명
DefaultTreeModel(TreeMode root)	어느 노드든 자식 노드를 가질 수 트리 모델 객체를 생성한다.
DefaultTreeModel(TreeMode root, boolean asksAllowsChildren)	어느 노드든 자식 노드를 가질 수 있게 하든지, 특정 노드만 자식 노드를 가질 수 있게 하는 것인지를 지정하여 트리 모델 객체를 생성한다.

스윙 컴포넌트

[표 12-62] DefaultTreeModel 클래스의 주요 메서드

반환형	메서드	설명
void	addTreeModelListener(TreeModelListener l)	트리의 모델이 변화되었을 때 발생하는 이벤트를 받기위해 지정된 TreeModelListener를 추가한다.
Object	getChild(Object parent, int index)	지정된 부모 노드에서 index 위치에 있는 자식 노드를 얻어온다.
Int	getChildCount(Object parent)	지정된 부모 노드에서 자식 노드의 수를 얻어온다.
	getIndexofChild(Object parent, Object child)	지정된 부모 노드에서 자식 노드의 위치를 얻어온다.
Object	getRoot()	트리의 루트 노드를 얻어온다.
boolean	isLeaf(Object node)	지정된 노드가 자식 노드가 아니라 리프 노드일 때 true 값을 얻어온다.
void	valueForPathChanged(TreePath path, Object newValue)	지정된 노드가 자식 노드가 아니라 리프 노드일 때 true값을 얻어온다.
	reload()	트리 구조나 노드에 변화가 있을 때 트리 모델을 갱신한다.
	reload(TreeNode node)	지정된 노드 이하의 트리에서 변화가 있을 때 트리 모델을 갱신한다.

스윙 컴포넌트

[표 12-63] TreeSelectionModel 인터페이스의 주요 필드

자료형	필드명	설명
static int	CONTINUOUS_TREE_SELECTION	연속적으로 되어 있는 노드들을 여러 개 선택할 수 있다.
	DISCONTINUOUS_TREE_SELECTION	선택하는 노드 수에 제한이 없고 연속적일 필요도 없다.
	SINGLE_TREE_SELECTION	한 번에 하나의 노드만 선택한다.

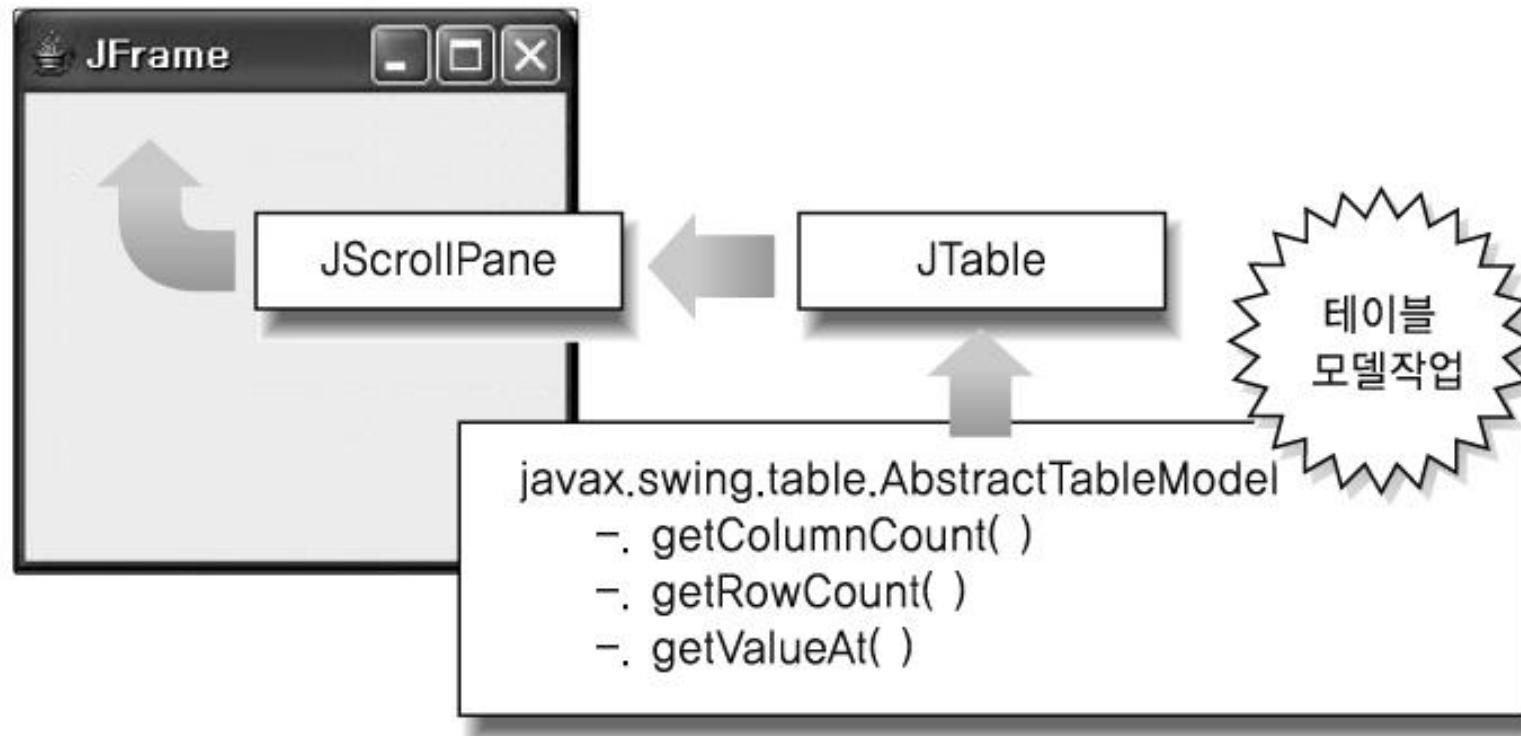
[표 12-64] TreeSelectionModel 인터페이스의 주요 메서드

반환형	메서드	설명
void	clearSelection()	현재 선택 영역을 비운다.
	setSelectionMode(int mode)	지정된 모드로 선택 모델을 설정한다.

스윙 컴포넌트

- **JTable** 클래스
 - 데이터를 행과 열로 구성되어 있는 테이블 형식으로 보여주는 컴포넌트이다.
 - **JTable** 역시 **Scrollable** 인터페이스가 구현되어 있어 **JScrollPane**에 붙여 스크롤을 할 수 있으며 모델로는 **TableModel**을 사용한다.
 - 테이블내의 각각의 행과 열이 맞나는 공간을 셀(Cell)이라고 하는데 하나의 값만이 기억된다.
 - **JTable**은 스윙의 특징에서 언급했듯이 **MVC** 모델을 가장 잘 보여주고 있는 컴포넌트이다.
 - **JTable**을 사용하기 위해서는 먼저 데이터를 저장할 모델을 만들고, 뷰인 **JTable**에 연결해 주어서 화면에 출력을 시켜야 한다.

스윙 컴포넌트



[그림 12-31] JTable 모델 작업의 도식화

스윙 컴포넌트

- TableModel로부터 행과 열의 개수, 데이터 타입정보, 열의 header(타이틀), Cell 데이터 편집 등을 설정하는 메서드를 지원한다.
- 일반적으로 사용자가 모델을 따로 설정하여 고유 테이블 모델을 만들어 사용하는데, 만드는 방법은 TableModel인터페이스를 구현하거나 AbstractTableModel과 DefaultTableModel을 상속하는 방법이 있다.
- DefaultTableModel은 유용한 메서드들을 지원하고 있지만 불필요한 기능이 너무 많아 일반적으로는 AbstractTableModel을 많이 사용한다.
- JTable에서 모델과 컨트롤러를 설정할때는 각 클래스마다 기본 Editor와 Renderer를 각각 설정할 수 있다.
- Cell의 데이터 타입에 따라 Editor를 두어 색상, 폰트 등과 같은 다양한 형태로 보여 주는 것이 가능하다. 물론 기본적으로는 DefaultTableCellRenderer와 DefaultCellEditor가 설정되어 사용된다.
- J2SE 5.0에는 API에 print() 메소드가 추가되어 테이블 출력, 결과물에 머리글/바닥글 삽입, 출력 페이지에 테이블크기 맞추기 등의 설정이 가능해졌다.

스윙 컴포넌트

[표 12-65] JTable 클래스의 주요 필드

자료형	필드명	설명
static int	AUTO_RESIZE_ALL_COLUMNS	모든 열의 사이즈를 조절할 때에 모든 열의 크기를 균등하게 조절한다.
	AUTO_RESIZE_LAST_COLUMN	모든 열의 사이즈를 조절할 때에 맨 마지막 열의 크기만 조절한다.
	AUTO_RESIZE_NEXT_COLUMN	모든 열의 사이즈를 조절할 때에 조절하는 열의 다음 열의 크기만 조절한다.
static int	AUTO_RESIZE_OFF	열의 크기 자동 조절은 사용하지 못하고 스크롤 바를 이용한다.
	AUTO_RESIZE_SUBSEQUENT_COLUMNS	UI를 조절할 때에 모든 열의 크기를 균등하게 조절한다.

스윙 컴포넌트

[표 12-66] JTable 클래스의 생성자

생성자	설명
JTable()	기본 자료 모델, 기본 컬럼 모델, 기본 선택 모델로 초기화된 기본 테이블 객체를 생성한다.
JTable(int numRows, int numColumns)	DefaultTableModel을 이용해 빈 셀을 numRows의 행과 numColumns의 열의 수만큼 테이블 객체를 생성한다.
JTable(Object[] rowData, Object columnNames)	2차원 배열에 값을 보여주는 테이블 객체를 생성한다. 주어진 테이블 데이터와 컬럼 이름을 가진다.
JTable(TableModel dm)	주어진 dm을 가지는 자료 모델, 기본 컬럼 모델, 기본 선택 모델로 초기화된 테이블 객체를 생성한다.
JTable(TableModel dm, TableColumnModel cm)	주어진 dm을 가지는 자료 모델, cm을 가지는 컬럼 모델, 기본 선택 모델로 초기화된 테이블 객체를 생성한다.
JTable(TableModel dm, TableColumnModel cm, ListSelectionModel sm)	주어진 dm을 가지는 자료 모델, cm을 가지는 컬럼 모델, sm을 가지는 선택 모델로 초기화된 테이블 객체를 생성한다.
JTable(Vector rowData, Vector columnNames)	주어진 Vectors의 Vector의 값을 보여주는 테이블 객체를 생성한다. 주어진 테이블 자료와 컬럼 이름을 가진다.

스윙 컴포넌트

[표 12-67] JTable 클래스의 모델과 컨트롤러와 관련있는 메서드

반환형	메서드	설명
void	setModel(TableModel dataModel)	주어진 자료 모델을 테이블의 자료 모델로 지정한다.
	setColumnModel (TableColumnModel columnModel)	주어진 컬럼 모델을 테이블의 컬럼 모델로 지정한다.
	setCellEditor(TableCellEditor anEditor)	주어진 셀 자료를 테이블의 셀 에디터로 지정한다.

[표 12-68] JTable 클래스의 자료와 관련있는 메서드

반환형	메서드	설명
void	setValueAt(Object aValue, int row, int column)	데이블 모델의 셀에 주어진 row, column 위치에 자료를 설정한다.
Object	getValueAt(int row, int column)	주어진 row, column 위치에 있는 셀값을 얻어온다.

스윙 컴포넌트

[표 12-69] JTable 클래스의 색, 모양과 관련있는 메서드

반환형	메서드	설명
void	setAutoResizeMode(int mode)	컬럼의 자동 크기 조절 모드를 설정한다.
	setIntercellSpacing (Dimension intercellSpacing)	셀 사이의 간격을 설정한다.
	setSelectionBackground (Color selectionBackground)	선택된 셀의 배경색을 설정한다.
	setSelectionForeground (Color selectionForeground)	선택된 셀의 전경색을 설정한다.
	setShowHorizontalLines (boolean showHorizontalLines)	셀 사이의 수평 구분선을 그릴지를 설정한다.
	setShowVerticalLines(boolean showVerticalLines)	셀 사이의 수직 구분선을 그릴지를 설정한다.

스윙 컴포넌트

[표 12-70] JTable 클래스의 선택과 관련있는 메서드

반환형	메서드	설명
void	clearSelction()	선택된 행과 열, 셀을 지운다.
	selectAll()	전체 테이블의 셀, 행과 열을 선택한다.
	setCellSelectionEnabled (boolean cellSelectionEnabled)	행과 열을 동시에 선택할 수 있도록 하는 선택모드를 사용할지를 설정한다.
	setColumnSelectionAllowed (boolean columnSelectionAllowed)	테이블의 열을 동시에 선택할 수 있도록 하는 선택 모드를 사용할지를 설정한다.
void	setRowSelectionAllowed (boolean rowSelectionAllowed)	테이블의 행을 동시에 선택할 수 있도록 하는 선택 모드를 사용할지를 설정한다.
	setSelectionMode(int selectionMode)	테이블의 선택 모드를 단일 선택하거나, 연속적 단일선택, 복수 선택 모드로 설정한다.
int	getSelectedColumn()	선택된 열의 첫번째 위치를 얻어온다. 선택된 열이 없으면 -1을 얻어온다.
Int[]	getSelectedColumns()	선택된 열들을 배열 형태로 얻어온다.
int	getSelectedRow()	선택된 행의 첫번째 위치를 얻어온다. 선택된 열이 없으면 -1을 얻어온다.
Int[]	getSelectedRows()	선택된 행들을 배열 형태로 얻어온다.
boolean	isCellSelected(int row, int column)	주어진 위치의 셀이 선택 상태인지를 얻어온다.

스윙 컴포넌트

[표 12-71] TableModel 클래스의 주요 메서드

반환형	메서드	설명
int	getColumnCount()	열 수를 얻어온다.
	getRowCount()	행 수를 얻어온다.
Object	getValueAt(int rowIndex, int columnIndex)	주어진 위치의 셀값을 얻어온다.
void	setValueAt(Object aValue, int rowIndex, int columnIndex)	주어진 위치의 셀값을 설정한다.

[표 12-72] AbstractTableModel 클래스의 주요 메서드

반환형	메서드	설명
void	addTableModelListener (TableModelListener l)	테이블의 모델이 변화되었을 때 발생하는 이벤트를 받기위해 지정된 TableModelListener를 추가한다.
int	findColumn(String columnName)	주어진 열의 이름을 가지고 테이블에서 찾아 위치를 얻어온다.
Class<?>	getColumnClass(int columnIndex)	주어진 열의 클래스를 얻어온다.
String	getColumnName(int columnIndex)	지정된 열 위치에 해당하는 이름을 얻어온다.

스윙 컴포넌트

[표 12-73] DefaultCellEditor 클래스의 주요 생성자

생성자	설명
DefaultCellEditor(JCheckBox checkBox)	주어진 체크박스를 편집기로 사용하는 셀 편집기 객체를 생성한다.
DefaultCellEditor(JComboBox comboBox)	주어진 콤보박스를 편집기로 사용하는 셀 편집기 객체를 생성한다.
DefaultCellEditor(JTextField textField)	주어진 텍스트필드를 편집기로 사용하는 셀 편집기 객체를 생성한다.