

Java



Study Point :

- Java에서의 GUI Programming 기법을 익히고 제작한다.
- AWT의 기본 개념 및 구조를 알아본다.
- Java에서 제공하는 Component 및 Layout Manager에 대해 알아본다.

GUI Programming :

● GUI Programming?

- ▶ GUI는 과거에 사용하였던 DOS(CUI방식)와 같은 방식의 Text 기반 OS가 아닌 Graphic을 이용하여 User와 Program 간의 상호작용을 할 수 있도록 해주는 Interface를 의미한다. Java에서 이러한 Graphic Programming을 지원하기 위해 나온 것이 바로 AWT이다.

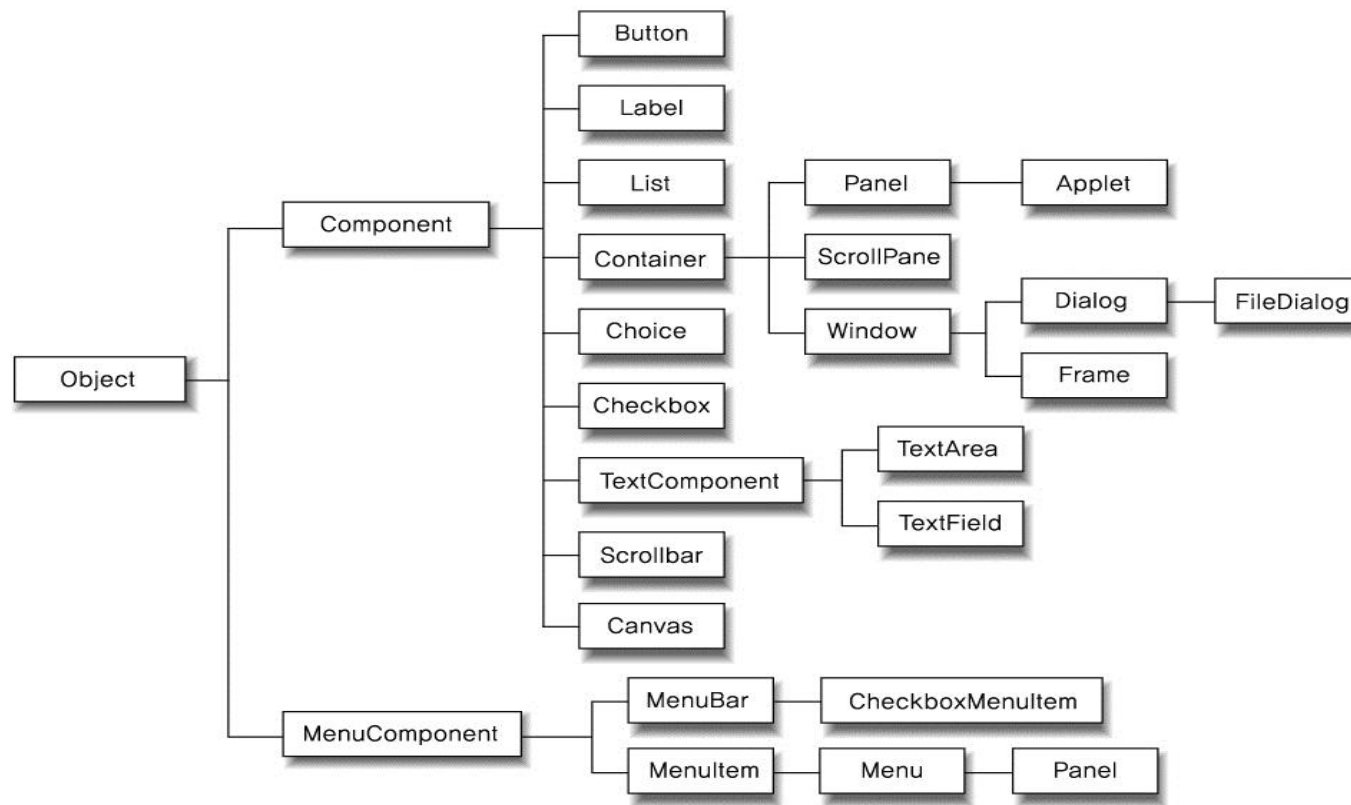
GUI Programming :

● AWT의 기본 개념

- ➡ AWT(Abstract Window Toolkit)는 GUI Programming을 제작하기 위해 Java에서 제공하는 library를 모아놓은 것이다.
- ➡ AWT는 모든 GUI Program에 사용되는 Component 및 Toolkit을 제공하고 있으며 향후에는 JFC와 같은 Swing 및 Java2D의 모태가 되는 개념.
- ➡ AWT는 OS에 구해받지 않고 쓸 수 있도록 OS의 것을 그대로 사용하지 않고 공통적이고 기본적인 Component들을 추상화시켜 제공한다.
- ➡ 실행되는 OS에 따라 다르게 보이거나 동작 방식에 차이가 있을 수 있다.
- ➡ 이러한 단점을 극복하기 위해 개발된 것이 JFC(Java Foundation Classes)이다.

GUI Programming :

• java.awt package



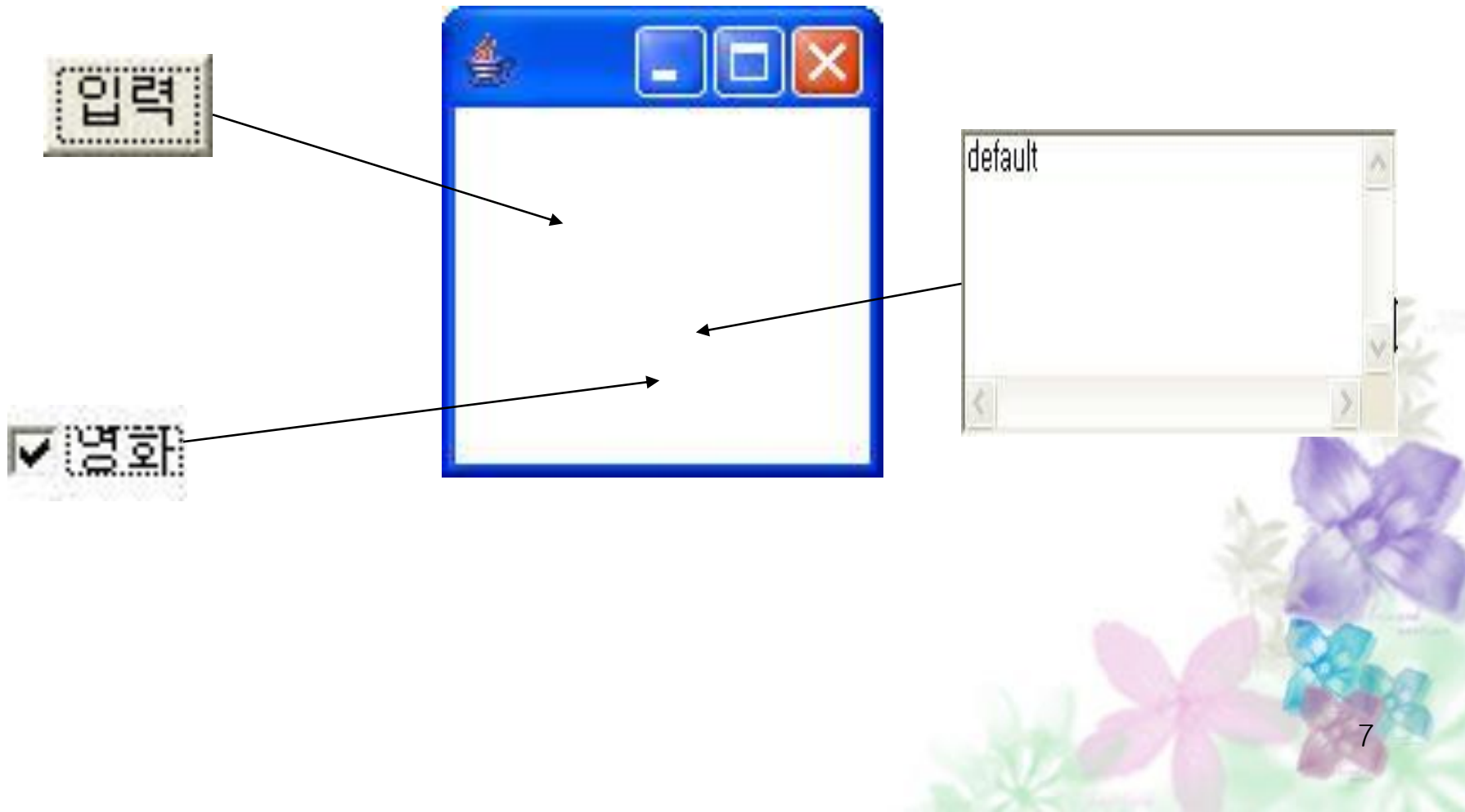
Container

● Container

- 자신의 scope에 Component를 포함시키고 관리하는 역할을 하며 Container가 다른 Container를 포함할 수도 있다.
- Component도 또한 Container에 부착시키지 않으면 독자적으로 화면에 출력될 수가 없고 반드시 Container에 부착을 시켜야만 화면에 출력이 될 수 있다.
- Container의 종류에는 Frame, Window, Panel, Applet, Dialog, FileDialog, ScrollPane이 있다.
- Container에 Component를 부착시키기 위해 add() method를 사용한다.

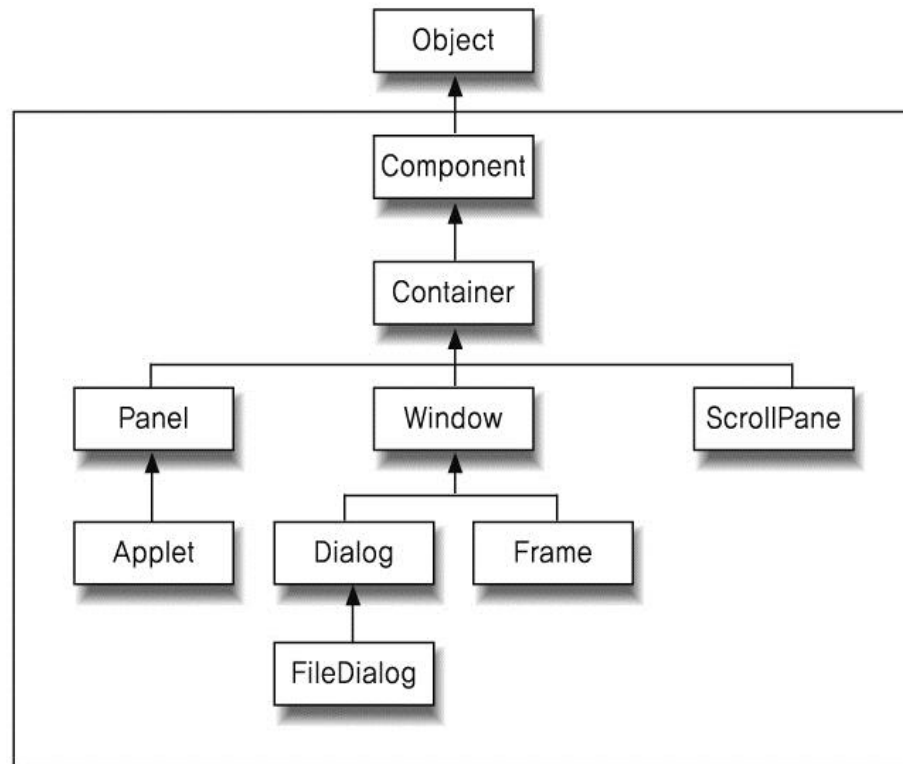
Container

- Container와 Component 관계



Container

● Container 종류 및 상속관계



Container

● Frame

- ▶ Window class의 하위 class로 일반적인 Application에서 window를 생성하기 위해 사용되는 class이다.
- ▶ Frame class의 상위 class인 Window class는 title, menu 등이 지원되지 않기 때문에 일반적으로 사용하지 않고 Frame class를 사용한다.
- ▶ Frame class는 기본적으로 Border, title, menu, System Box(최소화, 최대화, 종료 버튼) 등의 기능을 제공한다.
- ▶ Frame은 다른 window에 속해 있지 않은 window로 최상위 레벨 window라 한다.
- ▶ setSize(), setBounds() method 등을 이용해서 Window의 크기를 설정한 후 setVisible(), show() method를 통해서 화면에 출력시킬 수 있다.

Container

● frame class의 Constructor

※ Frame 클래스의 주요 생성자

생성자	설명
Frame()	가장 일반적인 생성자로 타이틀이 빈 상태로 생성
Frame (GraphicsConfiguration gc)	화면 장치의 GraphicsConfiguration을 이용하여 프레임을 생성
Frame(String title)	Title(윈도우의 타이틀 바에 나타낼 문자열)을 지정하여 프레임을 생성
Frame (String title, GraphicsConfiguration gc)	Title(윈도우의 타이틀 바에 나타낼 문자열)과 GraphicsConfiguration을 이용하여 프레임을 생성

Container

● frame class의 주요 method

반환형	메서드명	설명
int	getExtendedState()	프레임의 상태를 얻어온다.
static Frame[]	getFrames()	애플리케이션에서 생성한 모든 프레임을 리턴한다.
MenuBar	getMenuBar()	프레임의 메뉴바를 얻어온다.
int	getState()	프레임의 상태를 얻어온다.
String	getTitle()	프레임의 타이틀을 얻어온다.
void	remove (MenuComponent m)	프레임에서 지정한 메뉴바를 제거한다.
	setIconImage (Image image)	프레임이 최소화될 때 출력되는 이미지를 지정한다.
	setMenuBar (MenuBar mb)	프레임의 메뉴바를 지정한다.
	setResizable (boolean resizable)	프레임의 크기를 사용자가 변경할 수 있게 할 것인지를 지정한다.
	setState(int state)	프레임의 상태를 지정한다.
	setTitle(String title)	프레임의 타이틀을 지정한다.

Container

● Panel class

- ➡ Component들을 Group별로 묶어서 처리할 때 주로 사용한다.
- ➡ Frame에 Component를 직접 붙이지 않고 Panel에 그룹별로 붙이고, 다시 Panel을 Frame에 붙이는 경우가 많다.
- ➡ 다른 Panel을 생성하여 자신에게 붙일 수도 있어 window Program을 만들 때는 여러 개의 Panel을 사용하는 경우가 많다.

❖ Panel class의 Constructor

※ Panel 클래스의 주요 생성자

생성자	설명
Panel()	디폴트의 레이아웃 매니저를 사용해 새로운 패널을 작성
Panel(LayoutManager layout)	지정된 레이아웃 매니저를 가지는 새로운 패널을 작성

Container

● Panel class의 주요method

반환형	메서드	설명
void	addNotify()	패널의 피어를 작성한다.
AccessibleContext	getAccessibleContext()	Panel에 관련한 AccessibleContext를 얻어온다.

Container

❁ Dialog class

- ▶ 메인 window 외에 Message를 출력하거나, User로부터 Data를 입력 받을 때 주로 사용하는 Container이다,
- ▶ 보통은 Dialog class로부터 상속을 받아 새로운 기능을 가진 대화상자를 만드는데 사용된다,

❖ Dialog class의 Constructor

※ Dialog 클래스의 주요 생성자

생성자	설명
Dialog(Dialog owner)	생성되는 Dialog 객체를 소유하는 객체가 owner인 Frame을 생성
Dialog(Dialog owner, String title)	생성되는 Dialog 객체의 소유자를 owner라는 객체로 설정하고 타이틀을 설정
Dialog(Dialog owner, String title, boolean modal)	소유자로 owner 객체를 설정하고 타이틀을 가지며, 모달인지 모달이 아닌지를 설정하여 Dialog 객체를 생성
Dialog(Frame owner)	생성되는 Dialog 객체를 소유하는 객체가 owner라는 Frame 객체를 생성

Container

● Dialog class의 주요method

반환형	메서드	설명
void	addNotify()	패널의 피어를 작성한다.
AccessibleContext	getAccessibleContext()	Panel에 관련한 AccessibleContext를 얻어온다.

❖ Component

- 모든 Component들의 super class로서 GUI Program을 구성하는 구성단위로 각 Component들에서 공통으로 사용되어지는 method들을 가지고 있다.

Component

● Component class의 주요method

※ Component 클래스의 크기 및 위치와 관련있는 주요 메서드

반환형	메서드	설명
int	getX()	컴포넌트의 현재의 X 좌표를 얻어온다.
	getY()	컴포넌트의 현재의 Y 좌표를 얻어온다.
	getWidth()	컴포넌트의 현재의 폭을 얻어온다.
	getHeight()	컴포넌트의 현재의 높이를 얻어온다.
Dimension	getSize()	컴포넌트의 크기를 크기 객체(Dimensioned Object)로 얻어온다.
	getMaximumSize()	컴포넌트의 최대 크기를 크기 객체로 얻어온다.
	getMinimumSize()	컴포넌트의 최소 크기를 크기 객체로 얻어온다.
Rectangle	getBounds()	컴포넌트의 경계를 직사각형 객체(Rectangle Object)로 얻어온다.
void	setSize(int width, int height)	컴포넌트의 폭, 높이를 지정한다.
	setLocation(int x, int y)	컴포넌트의 새로운 위치를 지정하여 이동시킨다.
	setBounds(int x, int y, int width, int height)	컴포넌트의 위치와 크기를 지정한다.
	setBounds(Rectangle r)	새로운 경계 Rectangle r에 적합하도록 컴포넌트의 위치와 크기를 지정한다.

Component

● Component class의 주요method

※ Component 클래스의 색상, 폰트와 관련있는 주요 메서드

반환형	메서드	설명
Color	getBackground()	컴포넌트의 배경색을 색상 객체(Color Object)로 얻어온다.
	getForeground()	컴포넌트의 전경색을 색상 객체로 얻어온다.
void	setBackground(Color c)	컴포넌트의 배경색을 Color c로 지정한다.
	setForeground(Color c)	컴포넌트의 전경색을 Color c로 지정한다.
Font	getFont()	컴포넌트의 글꼴을 글꼴 객체(Font Object)로 얻어온다.
void	setFont(Font f)	컴포넌트의 글꼴을 Font f로 지정한다.

Component

● Component class의 주요method

※ Component 클래스의 설정과 관련있는 주요메서드

반환형	메서드	설명
void	setEnabled(boolean b)	파라미터 b값에 의해 컴포넌트의 활성화와 비활성화를 지정한다.
	setVisible(Booleen b)	파라미터 b값에 의해 컴포넌트를 출력하거나 숨기는 것을 지정한다.
String	getName()	컴포넌트의 이름을 얻어온다.
Container	getParent()	컴포넌트를 소유하고 있는 컨테이너를 얻어온다.
void	requestFocus()	현 컴포넌트에 포커스를 요청한다.

Component

● Basic Component

※ 기본 컴포넌트의 종류와 기능

종류	프로그래밍 언어
Button	버튼을 만들 때 사용한다.
Canvas	비어 있는 공간으로 그래픽을 처리할 때 사용한다.
Checkbox	체크 박스나 라디오 버튼을 만들 때 사용한다.
Choice	드롭-다운 리스트를 만들 때 사용한다.
Label	고정 문자열을 표시할 때 사용한다.
List	리스트를 만들 때 사용한다.
Scrollbar	스크롤바를 만들 때 사용한다.

Component

● Button

- ➡ 버튼을 사용자가 눌렀을 때 특정한 액션을 실행할 수 있도록 만든 Component이다,

❖ Button class의 Constructor

※ Button 클래스의 주요 생성자

생성자	설명
Button()	비어 있는 버튼 객체를 생성한다.
Button(String label)	label을 지정하여 버튼 객체를 생성한다.

Component

● Button class의 주요 method

반환형	메서드	설명
void	<code>addActionListener (ActionListener l)</code>	버튼으로부터 액션 이벤트를 받기 위해 지정된 액션 리스너를 추가한다.
String	<code>getActionCommand()</code>	버튼에서 발생하는 액션 이벤트의 커맨드명을 얻어온다.
	<code>getLabel()</code>	버튼의 레이블을 얻어온다.
void	<code>setLabel(String label)</code>	버튼의 레이블을 지정한 label로 설정한다.

Component

■ Checkbox

- ▶ User가 여러 종류의 옵션을 선택할 것인지의 여부를 지정할 때 사용.
- ▶ 여러 개의 Checkbox를 묶어 하나의 group으로 만들어 group내에서는 하나만이 값을 유지할 수 있는 Radiobutton 형태로도 사용할 수 있는 Component.
- ▶ group으로 묶을 때는 CheckboxGroup class를 사용.

❖ Checkbox class의 Constructor

※ Checkbox 클래스의 주요 생성자

생성자	설명
Checkbox()	label이 없는 체크박스 객체를 생성한다.
Checkbox(String label)	지정된 label을 가지는 체크박스 객체를 생성한다.
Checkbox(String label, Boolean state)	지정된 label과 지정된 state를 넣어서 체크박스 객체를 생성한다.
Checkbox(String label, Boolean state, CheckboxGroup group)	지정된 label, 지정된 state를 넣어, 지정된 group에 속하는 체크박스 객체를 생성한다.

Component

❁ Checkbox class의 주요method

반환형	메서드	설명
void	addItemListener (ItemListener l)	체크박스로부터 아이템 이벤트를 받기 위해 지정된 아이템 리스너를 추가한다.
	setLabel(String label)	체크박스의 레이블을 지정한다.
String	getLabel()	체크박스의 레이블을 얻어온다.
void	setState(boolean state)	체크박스 상태를 지정된 상태로 설정한다.
boolean	getState()	체크박스가 'On' 또는 'Off' 상태인지를 얻어온다.
void	setCheckboxGroup (CheckboxGroup g)	체크박스 그룹을 지정한다.

Component

Choice

- ▶ List class와 거의 유사한 기능을 가지고 있는 Component로 User가 drop-down button을 사용하여 여러item중에 하나를 선택할 수있는 기능을 제공,
- ▶ Component를 생성한 후에 drop-down list에 항목에 추가시켜 사용한다,

❖ Choice class의 Constructor

※ Choice 클래스의 주요 생성자

생성자	설명
Choice()	새로운 선택 메뉴 객체를 생성한다.

Component

● Choice class의 주요method

반환형	메서드	설명
void	add(String item)	Choice 메뉴에 항목을 추가한다.
	addItemListener (ItemListener I)	Choice 메뉴로부터 아이템 이벤트를 받기 위해 지정된 아이템 리스너를 추가한다.
	insert(String item, int index)	Choice에 지정된 위치에 항목을 삽입한다.
	remove(int position)	Choice 메뉴에 지정된 위치에 있는 항목을 제거한다.
	remove(String item)	Choice 메뉴로부터 item이 첫 번째로 발견된 항목을 제거한다.
	removeAll()	Choice 메뉴로부터 모든 item을 제거한다.
String	getItem(int index)	Choice 메뉴에서 지정한 위치의 항목의 문자열을 얻어온다.
int	getItemCount()	Choice 메뉴에서 항목의 개수를 얻어온다.
	getSelectedIndex()	현재 선택된 항목의 위치를 얻어온다.
String	getSelectedItem()	현재 선택된 항목의 문자열을 얻어온다.
void	select(int index)	지정한 위치의 항목을 선택한다.
	select(String str)	지정한 이름의 항목을 선택한다.

Component

● Label

- ▶ 사각형의 scope에 문자열을 표시할 때 사용하는 Component이다.
- ▶ Label은 경계선이 없고 특별한 상태를 가지지도 않는다, 그러므로 Label을 Container에 포함시키게 되면 Label의 문자만 화면에 표시가 된다,
- ▶ Label의 문자열은 좌, 우, 중앙으로 정렬시킬 수 있다.

❖ Label class의 주요 Member Field

※ Label 클래스의 주요 멤버 필드

자료형	필드명	설명
static int	CENTER	레이블의 문자를 중앙에 정렬시킨다.
	LEFT	레이블의 문자를 왼쪽에 정렬시킨다.
	RIGHT	레이블의 문자를 오른쪽에 정렬시킨다.

Component

● Label class의 Constructor

※ Label 클래스의 주요 생성자

생성자	설명
Label()	빈 레이블을 생성한다.
Label(String text)	레이블에 지정한 text를 가지고 왼쪽 정렬이 된 상태로 생성한다.
Label(String text, int alignment)	레이블에 지정한 text를 가지고, 지정한 정렬이 된 상태로 생성한다.

❖ Label class의 주요method

반환형	메서드	설명
String	getText()	레이블의 텍스트를 얻어온다.
void	setText(String text)	레이블에 지정한 text로 설정한다.
	setAlignment(int align)	레이블의 텍스트를 지정한 정렬로 정렬시킨다.

Component

List

- ▶ Choice와 유사한 기능이지만 여러 개의 항목을 보여주고 사용자가 하나 또는 여러 개의 항목을 선택할 수 있도록 지원하는 Component이다.
- ▶ 기본적으로는 하나의 항목만을 선택할 수 있지만 MultipleMode를 설정하면 한번에 여러 개의 항목을 선택할 수 있다.

❖ List class의 Constructor

※ List 클래스의 주요 생성자

생성자	설명
List()	새로운 리스트 객체를 생성한다.
List(int rows)	지정한 숫자만큼의 항목을 보여주는 새로운 리스트 객체를 생성한다.
List(int rows, Boolean multipleMode)	지정한 숫자만큼의 항목을 보여주는 새로운 리스트 객체를 생성하며, 단일 선택 모드나 다중 선택 모드를 지정할 수 있다.

Component

List class의 주요method

반환형	메서드	설명
void	add(String item)	지정한 항목을 List의 끝에 추가한다.
	add(String item, int index)	List의 지정된 위치에 항목을 삽입한다.
	addItemListener (ItemListener I)	List로부터 아이템 이벤트를 받기 위해 지정된 아이템 리스너를 추가한다.
	remove(int position)	List에 지정한 위치에 있는 항목을 제거한다.
	remove(String item)	List로부터 item이 첫번째로 발견된 항목을 제거한다.
	removeAll()	List에 있는 모든 item을 제거한다.
String	getItem(int index)	List에서 지정한 위치의 항목의 문자열을 얻어온다.
String[]	getItems()	List의 항목들을 문자열 배열로 얻어온다.
int	getItemCount()	List에서 항목의 개수를 얻어온다.
	getSelectedIndex()	현재 선택된 항목의 위치를 얻어온다.
	getSelectedIndexes()	다중 선택 모드일 때, 현재 선택된 항목의 위치 값들을 배열로 얻어온다.
String	getSelectedItem()	현재 선택된 항목의 문자열을 얻어온다.
String[]	getSelectedItems()	다중 선택 모드일 때, 현재 선택된 항목들의 문자열을 배열로 얻어온다.
void	select(int index)	지정한 위치의 항목을 선택한다.
	replaceItem(String newValue, int index)	지정한 위치의 항목의 newValue값을 바꾼다.

Component

● Canvas

- ▶ 특정한 모양을 갖고 있지 않고 사각형의 scope만을 갖고 있는 Component, 그림을 그릴 수 있는 도화지의 역할을 하는 Component로 Container에 포함되어 Graphic 처리를 할 수 있다.

❖ Canvas class의 Constructor

※ Canvas 클래스의 주요 생성자

생성자	설명
Canvas()	새로운 캔버스 객체를 생성
Canvas (GraphicsConfiguration gc)	화면 장치의 GraphicsConfiguration을 이용하여 캔버스 객체를 생성

❖ Canvas class의 주요method

반환형	메서드	설명
void	paint(Graphics g)	캔버스를 업데이트할 때 사용된다.
	update(Graphics g)	캔버스에 그림을 그릴 때 사용된다.

Component

● TextComponent

- ➡ Text를 다루는 class의 super class로 Text를 처리하는 method를 제공,
- ➡ 독립적으로는 생성되지는 못한다.

❖ TextComponent class의 주요method

※ 텍스트 편집과 관련있는 메서드

반환형	메서드	설명
int	getCaretPosition()	텍스트 컴포넌트의 텍스트가 삽입될 캐럿의 현재의 위치를 얻어온다.
void	setCaretPosition (int Position)	텍스트 컴포넌트의 텍스트가 삽입될 캐럿의 위치를 지정한다.
String	getText()	텍스트 컴포넌트가 가지고 있는 텍스트를 얻어온다.
void	setText(String t)	텍스트 컴포넌트에 표시될 텍스트를 설정한다.
	setEditable(boolean b)	텍스트 컴포넌트의 편집 가능 여부를 결정한다.

Component

● TextComponent class의 주요 method

※ 텍스트 선택과 관련있는 메서드

반환형	메서드	설명
String	getSelectedText()	텍스트 컴포넌트에서 선택되어진 텍스트를 얻어온다.
int	getSelectionEnd()	텍스트 컴포넌트에서 선택되어진 영역의 끝 위치를 얻어온다.
	getSelectionStart()	텍스트 컴포넌트에서 선택되어진 영역의 시작 위치를 얻어온다.
void	setSelectionEnd (int selectionEnd)	텍스트 컴포넌트에서 선택할 영역의 끝 위치를 지정한다.
	setSelectionStart (int selectionStart)	텍스트 컴포넌트에서 선택할 영역의 시작 위치를 지정한다.
	select(int selectionStart, int selectionEnd)	지정된 시작과 끝 위치의 텍스트를 선택 상태로 만들어준다.

※ TextComponent 클래스의 하위 클래스

종류	기능
TextField	문자 한 줄만 입력받을 때 사용한다.
TextArea	여러 줄의 문자를 입력받을 때 사용한다.

Component

● TextField

- ➡ 한 줄 내의 Text를 입력 받거나 편집할 수 있는 Component.
- ➡ 한 줄에 표시할 수 있는 Column수를 지정할 수 있고 Echo Character를 지정하면 입력되는 문자대신 Echo Character로 지정한 문자로 출력된다.

❖ TextField class의 Constructor

※ TextField 클래스의 주요 생성자

생성자	설명
TextField()	비어있는 텍스트 필드 객체를 생성한다.
TextField(int columns)	지정한 컬럼수만큼 문자를 보여줄 수 있는 크기로 텍스트 필드 객체를 생성한다.
TextField(String text)	지정한 텍스트로 초기화하여 텍스트 필드 객체를 생성한다.
TextField(String text, int columns)	지정한 텍스트로 초기화하여 출력하고, 지정한 컬럼 수만큼 문자를 보여줄 수 있는 크기로 텍스트 필드 객체를 생성한다.

Component

● TextField class의 주요method

반환형	메서드	설명
int	getColumns()	텍스트 필드의 컬럼 수를 얻어온다.
void	setColumns(int columns)	텍스트 필드의 컬럼 수를 지정한다.
	getEchoChar()	현재 설정되어 있는 반향 문자를 얻어온다.
	setEchoChar(char c)	텍스트 필드의 반향 문자를 지정한다.

Component

■ TextArea

- ▶ 여러 줄의 Text를 사용자로 부터 입력받거나 편집할 수 있는 Component.
- ▶ 화면에 출력되는 scope이 벗어나면 Scrollbar 표시 방식에 따라 자동으로 Scrollbar가 생성된다.
- ▶ 사용자가 필요에 따라 일부 Scrollbar만 나타나게 할 수도 있다.

❖ TextArea class의 주요 Member Field

※ TextArea 클래스의 주요 멤버 필드

자료형	필드명	설명
static int	SCROLLBARS_BOTH	수평/수직 스크롤바를 모두 표시한다.
	SCROLLBARS_HORIZONTAL_ONLY	수평 스크롤바만 표시한다.
	SCROLLBARS_VERTICAL_ONLY	수직 스크롤바만 표시한다.
	SCROLLBARS_NONE	스크롤바를 표시하지 않는다.

Component

● TextArea class의 Constructor

※ TextArea 클래스의 주요 생성자

생성자	설명
TextArea()	비어있는 텍스트 영역 객체를 생성한다.
TextArea(int rows, int columns)	지정된 행수와 컬럼수만큼 표현할 수 있는 텍스트 영역 객체를 생성한다.
TextArea(String text)	지정된 문자를 가지고 텍스트 영역 객체를 생성한다.
TextArea(String text, int rows, int columns)	지정된 문자를 가지고 초기화하여 지정된 행수와 컬럼수만큼 표현할 수 있는 텍스트 영역 객체를 생성한다.
TextArea(String text, int rows, int columns, int scrollbars)	지정된 문자를 가지고 초기화하여 지정된 행수와 컬럼수만큼 표현할 수 있는 텍스트 영역 객체를 생성한다. 그리고 스크롤바의 모습이 어떻게 나타낼 것인지를 지정한다.

Component

● TextArea class의 주요method

반환형	메서드	설명
void	append(String str)	지정된 문자열을 기존 내용의 끝에 추가한다.
	insert(String str, int pos)	지정된 문자열을 지정된 위치에 삽입한다.
	replaceRange(String str, int start, int end)	지정된 시작과 끝 위치의 문자열을 지정된 문자열로 바꾼다.
int	getColumns()	텍스트 영역의 컬럼수를 얻어온다.
	getRows()	텍스트 영역의 행수를 얻어온다.

Component

● Menu Component

- ▶ Menu는 보통 최상위 Level의 window Titlebar 아래에 존재하는 것으로 User가 Program의 기능을 선택할 수 있도록 해주는 기능을 가지고 있는 Component.
- ▶ Menu의 구성은 MenuBar, Menu, MenuItem으로 구성된다.

❖ MenuComponent class의 Sub class

※ MenuComponent 클래스의 하위 클래스

종류	기능
MenuBar	메뉴를 올려 놓을 수 있는 메뉴바를 만들 때 사용한다.
Menu	메뉴 바에 올려 놓을 수 있는 메뉴를 만들 때 사용한다.
MenuItem	메뉴의 하위 메뉴를 만들 때 사용한다.
CheckboxMenuItem	체크박스가 들어 있는 메뉴아이템을 만들 때 사용한다.
PopupMenu	동적으로 표현할 수 있는 메뉴를 만들 때 사용한다.

Component

● Menu 사용법

1. MenuBar Object를 생성.

```
MenuBar mb = new MenuBar();
```

2. MenuBar에 삽입할 Menu를 생성한 후 Menu를 MenuBar에 붙인다.

```
Menu menu_file = new Menu( "파일" );  
mb.add(menu_file);
```

3. Menu에 붙일 MenuItem을 생성한 후 해당 Menu에 붙인다.

```
MenuItem menu_file_new = new MenuItem( "새문서" );  
menu_file.add(menu_file_new);
```

4. MenuBar를 window에 붙인다.

```
setMenuBar(mb);
```

LayoutManager

● LayoutManager

- ➡ Container는 자기 자신에 Component를 붙일 때 어디에, 어떤 방식으로 배치하여 붙일 것인가를 이미 결정하고 있다.
- ➡ 미리 정해진 Layout에 따라 Component들을 자동으로 배치하는 기능을 가지고 있는 Object를 Container들은 가지고 있는데 이것을 Layout Manager라 한다.
- ➡ Java 에서 사 용 하 는 Layout Manager 는 FlowLayout, BorderLayout, GridLayout, GridBagLayout, CardLayout의 5가지가 있다.
- ➡ Layout Manager는 각자 다른 방식으로 배치기능을 가지고 있으며 Container는 기본적으로 하나의 Layout Manager를 가지고 있다.
- ➡ 사용자가 임의로 Layout Manager는 다시 설정할 수 있으며 Layout Manager를 제거하고 수동으로 좌표를 이용해서 배치할 수도 있다.

※ 컨테이너의 기본 배치관리자

컨테이너	기본 배치관리자	컨테이너	기본 배치관리자
Frame	BorderLayout	Dialog	BorderLayout
Panel	FlowLayout	Applet	FlowLayout

LayoutManager

● FlowLayout

- ▶ Component들을 수평으로 순서대로 늘어놓는 배치 기능을 가지고 있다.
- ▶ 처음에 배치를 하게되면 상단, 중앙부터 배치가 되는데 배치를 하다가 더 이상 배치할 공간이 없으면 자동으로 다음 줄로 이동하여 배치하게 된다.
- ▶ Component를 배치할 때 Component의 간격을 gap이라고 하는데 Component들 사이의 수평, 수직간 간격을 설정할 수 있다.



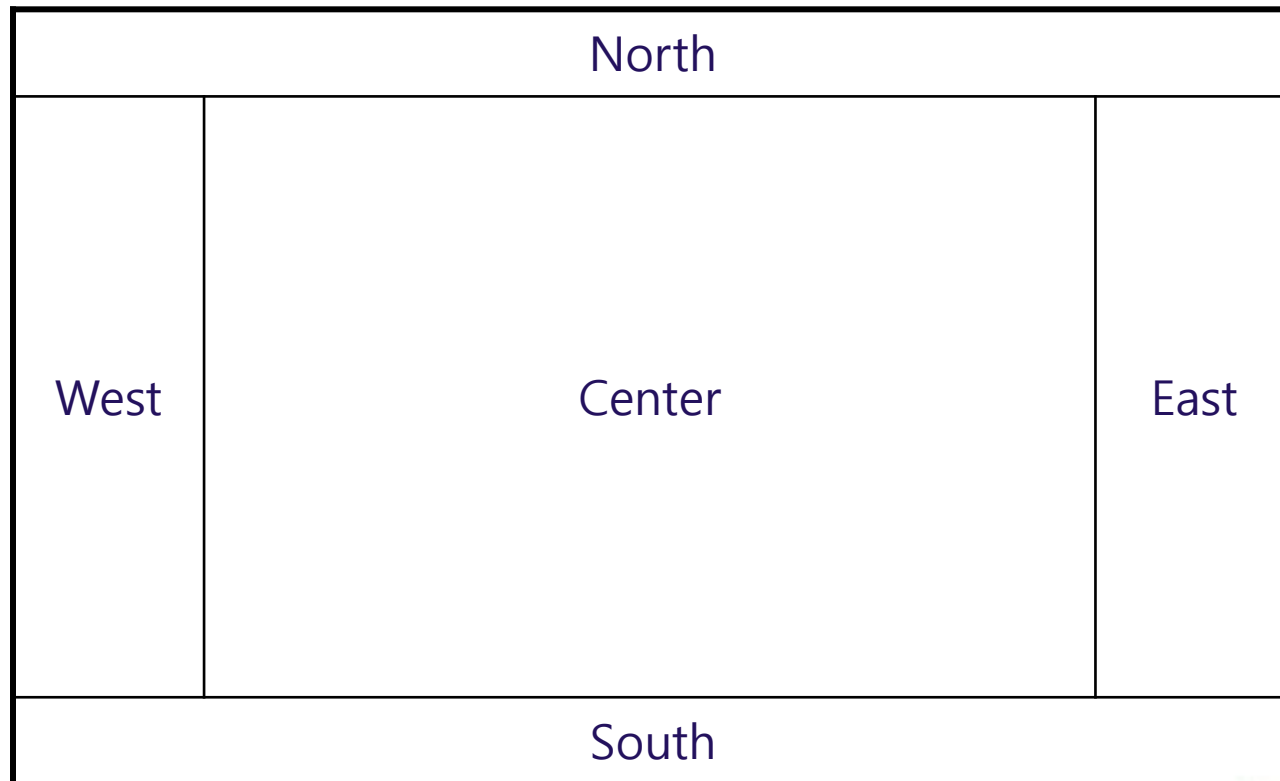
LayoutManager

● BorderLayout

- ▶ Container의 scope을 5개의 scope으로 분할하여 Component를 배치하는 관리자이다,
- ▶ 기본적으로 Component를 BorderLayout에 붙일 때 아무런 scope을 지정하지 않은경우는 기본적으로 CENTERscope에 붙이게 된다,
- ▶ CENTERscope은 다른scope에 아무것도 존재하지않으면 그 scope까지 포함해서 scope이 잡히게 된다,
- ▶ SOUTH, NORTHscope은 Component의 높이는 제대로 나타나지만 폭의 길이는 인정되지 않는다,
- ▶ WEST, EASTscope은 Component의 폭의 길이는 제대로 나타나지만 높이는 제대로 인정되지 않고 항상 그 scope의 길이만큼 잡히게 된다,

LayoutManager

● BorderLayout



※ BorderLayout 경계

LayoutManager

● GridLayout

- ➡ 격자모양(모눈종이와 같은 모양)과 같이 가로와 세로가 같은 크기의 비율로 나누어 각 공간(cell)에 Component를 배치할 수 있는 관리자이다,
- ➡ GridLayout 배치 관리자를 만들 때 행과 열의 수를 지정하는데, 값은 0 이상의 값으로 지정하며 만약 0으로 지정하게 되면 무한대로 Component를 추가하여 붙일 수 있다,
- ➡ 행과 열의 수하고 붙이는 Component의 수가 더 많은 경우는 행의 수를 우선으로 맞춘다,

LayoutManager

● GridBagLayout

- ▶ GridLayout과 유사한 기능을 제공하는 배치 관리자로 가장 복잡한 구조를 가지고 있다.
- ▶ GridLayout은 하나의 cell에는 하나의 Component를 가질 수 있는데 GridBagLayout은 여러 cell에 걸쳐서 서로 다른 크기와 간격으로 하나의 Component가 배치될 수 있다.
- ▶ GridBagLayout을 사용하는 경우는 GridBagConstraints class를 더 사용하여 배치를 시킨다.
- ▶ GridBagConstraints class는 GridLayout으로 지정된 Container에 Component가 얼마만큼의 scope을 차지하여 배치할 것인가에 대한 자세한 scope 구조에 대해 지정을 한다.

컴포넌트 1		컴포넌트 2
컴포넌트 3		
컴포넌트 4	컴포넌트 5	
	컴포넌트 6	

※ GridBagLayout 배치 예

LayoutManager

● CardLayout

- ▶ 여러 개의 카드를 쌓아둔 것 처럼 Component를 하나만 보여주는 Layout Manager이다.
- ▶ 맨 위의 Component만 보여주므로 한번에 하나의 Component만 볼 수 있다.
- ▶ CardLayout에는 맨 위에 위치할 Component를 지정할 수 있는 method가 지원되며, 또한 그 다음에 나올 Component를 이동시킬 수 있는 method를 지원한다.

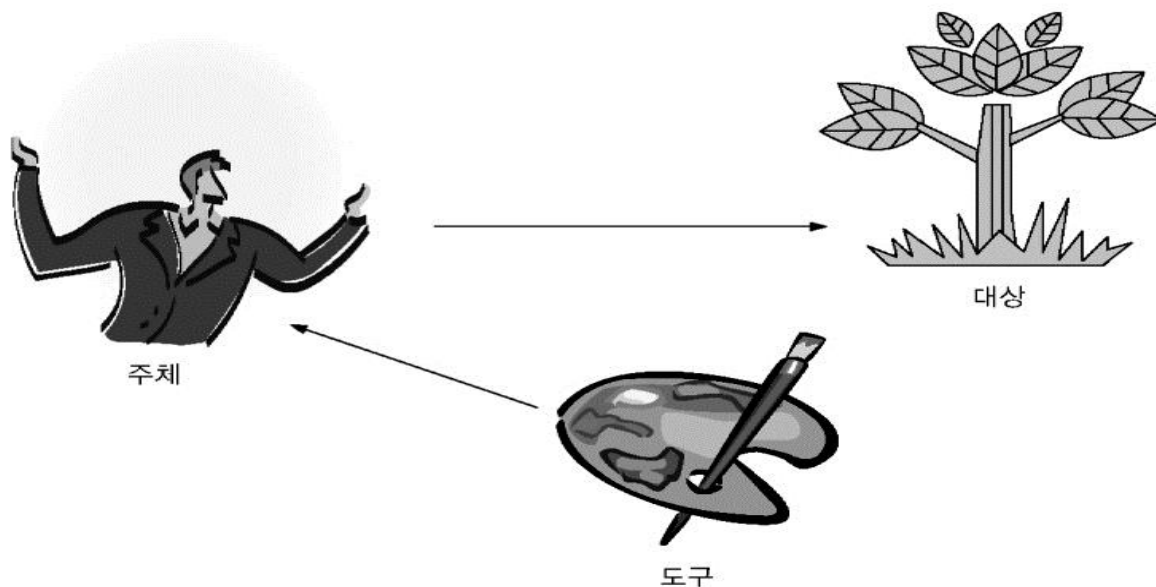
GUI 프로그래밍

GUI 환경에서 작동하는 프로그램에 대하여
윈도우 프로그램의 작성 방법
그래픽과 이미지 디스플레이 방법
오디오 파일 재생 방법



Graphic Structure

- 일반적으로 Graphic을 처리하기 위해서는 그림을 그릴 주체가 있고 그 주체가 그림을 그릴 때 사용할 수 있는 펜이나 붓 등의 도구가 있어야 한다. 또한 그 도구를 가지고 그림을 그릴 수 있는 장소, 즉 도화지 등과 같은 대상이 있어야 한다



Graphic Structure

● Graphic 주체

- ▶ Java에서의 Graphic 주체는 그림을 직접 그리도록 명령을 내리는 사람, 즉 Programmer 그 자체가 주체가 될 수 있으며, Java 자체적으로 자동으로 그림을 그릴 수 있도록 만들어놓은 Java가상머신이 될 수 있다.
- ▶ 따라서 실제로 그림을 그리는 작업이 일어나도록 하는 주체를 의미한다.

● Graphic 도구

- ▶ Java에서의 Graphic 도구는 주체가 그림을 그릴 수 있도록 펜, 붓, 폰트, 팔레트 등과 같은 것을 의미한다.
- ▶ 이러한 도구들을 만든 후 여러 번 사용할 수 있도록 정보를 저장하는데, 이를 Graphic Context라 한다.
- ▶ 가장 중요하게 보게 될 내용으로 Java에서는 Graphic Context를 추상화시킨 class가 `java.awt.Graphics` class이다.

Graphic Structure

● Graphic 대상

- ▶ Java에서의 Graphic 대상은 그림을 그릴 수 있는 곳을 의미하는데, 우리가 앞에서 배운 AWT의 모든 Component나 Image, Printer 등이 될 수 있다.
- ▶ Java에서 제공하는 모든 Component는 paint() method가 포함되어 있어 자신이 그려져야 할 때 자동으로 호출되어 지정된 곳에 그릴 수 있다.
- ▶ 그러기 때문에 Programmer가 일일이 paint() method를 만들어 주지 않아도 화면에 출력시킬 수가 있다.

Graphic Context

● Graphic Context

- ➡ 그림을 그리기 위해 사용하는 도구들을 추상화시킨 것으로 두 가지 기능을 가지고 있다,
- ➡ 첫 번째로 실제로 그림을 그리는 작업을 할 수 있도록 `Graphic method`를 가지고 있다,
- ➡ 두 번째는 `Graphic method`를 이용해서 그림을 그릴 때 사용하는 각종 도구들, 즉, `Graphic` 정보를 설정할 수 있는 기능이다,
- ➡ 예를 들어 원, 사각형을 그릴 수 있도록 하는 `method`와 그러한 그림을 그릴 때 사용하는 색상, 굵기 등을 설정할 수가 있는데 이러한 기능들을 담당하는 것이 `Graphic Context`이다,

● Graphics class

- ➡ `Graphics class`는 `Graphic` 작업을 할 수 있도록 기능들을 추상화시킨 `class`로 그림을 그릴 수 있는 각종 `method`를 지원,
- ➡ API `method`중에 `fill`이 붙어 있는 것들은 채우기 기능을 가지고 있는 `method`들이다,

Graphic Context

※ Graphics 클래스 메서드

반환형	메서드	설명
abstract void	drawString(String str, int x, int y)	그래픽 컨텍스트의 글꼴과 색상을 이용하여 지정한 문자를 지정한 위치에 그려준다.
	drawLine(int x1, int y1, int x2, int y2)	그래픽 컨텍스트 좌표 내에서 x1, y1과 x2, y2 사이에 현재 색상을 이용하여 선을 그려준다.
	drawRect(int x, int y, int width, int height)	지정한 위치에 폭과 높이가 있는 사각형을 그려준다.
	drawRoundRect(int x, int y, int width, int height, int arcWidth, int arcHeight)	지정한 위치에 폭과 높이, 각도가 있는 모서리가 둥근 사각형을 그려준다.
	drawOval(int x, int y, int width, int height)	지정한 위치에 폭과 높이가 있는 타원을 그려준다.
abstract void	drawArc(int x, int y, int width, int height, int startAngle, int arcAngle)	지정한 위치에 폭과 높이, 각도가 있는 원호를 그려준다.
	drawPolygon(int[] xPoints, int[] yPoints, int nPoints)	지정한 x 좌표와 y 좌표의 배열을 가지는 닫힌 다각형을 그려준다.

Graphic Context

※ Graphics 클래스 메서드(계속)

반환형	메서드	설명
abstract boolean	<code>drawImage(Image img, int x, int y, ImageObserver observer)</code>	현재 이용할 수 있는 이미지를, x와 y를 시작 좌표로 기준하여 그려준다.
	<code>drawImage(Image img, int x, int y, int width, int height, ImageObserver observer)</code>	현재 이용할 수 있는 이미지를, x와 y를 시작 좌표로 기준하여 폭(width)과 높이(height)만큼 그려준다.
	<code>drawImage(Image img, int dx1, int dy1, int dx2, int dy2, int sx1, int sy1, int sx2, int sy2, ImageObserver observer)</code>	로딩된 이미지를 sx1, sy1을 시작 좌표로 하고 폭(sx2), 높이(sy2)만큼을 잘라서 출력시킬 화면의 dx1, dy1을 시작 좌표로 하여 폭(dx2)과 높이(dy2)만큼의 공간에 그려준다.
void	<code>setColor(Color c)</code>	그래픽 컨텍스트의 현재 색상을 지정한 색상으로 설정한다.
	<code>setFont(Font font)</code>	그래픽 컨텍스트의 현재 글꼴을 지정한 글꼴로 설정한다.
FontMetrics	<code>getFontMetrics()</code>	그래픽 컨텍스트의 현재 글꼴에 대한 FontMetrics 객체를 얻어온다.
	<code>getFontMetrics(Font f)</code>	지정한 글꼴에 대한 FontMetrics 객체를 얻어온다.

Graphic Context

● Color class

- ▶ Graphic Context에 색상을 설정하기 위해 사용하는 class이다.
- ▶ Color class Object를 생성할 때 사용할 색상을 지정하여 Graphic Context에 설정하면 그 이후의 모든 색상에 적용하여 사용할 수 있다.

※ Color 클래스의 주요 생성자

생성자	설명
Color(float r, float g, float b)	범위(0.0~1.0)의 지정된 red, green, blue값을 사용하여 색상 객체를 생성한다.
Color(int r, int g, int b)	범위(0~255)의 지정된 red, green, blue값을 사용하여 색상 객체를 생성한다.

Graphic Context

※ Color 클래스의 주요 멤버필드

자료형	필드명	설명
static Color	black, BLACK	검정색
	Blue, BLUE	파란색
	cyan, CYAN	하늘색
	darkGray, DARK_GRAY	어두운 회색
	Gray, GRAY	회색
	green, GREEN	녹색
	lightGray, LIGHT_GRAY	밝은 회색
	magenta, MAGENTA	진홍색
	orange, ORANGE	주황색
	pink, PINK	분홍색
	red, RED	빨간색
	white, WHITE	흰색
	yellow, YELLOW	노란색

Graphic Context

Font class

- ▶ Graphic Context에 글꼴을 설정하기 위해 사용하는 class.
- ▶ Font class Object를 생성할 때 사용할 글꼴의 속성을 지정하여 Graphic Context에 설정하면 그 이후의 모든 글자에 적용하여 사용할 수 있다.

※ Font 클래스의 주요 멤버필드

자료형	필드명	설명
static int	BOLD	굵은 스타일 상수
	ITALIC	이탤릭 스타일 상수
	PLAIN	일반 스타일 상수

※ Font 클래스의 생성자

생성자	설명
Font(Map<? extends AttributedCharacterIterator.Attribute,?> attributes)	지정된 속성으로 새로운 폰트 객체를 생성한다.
Font(String name, int style, int size)	지정한 폰트 이름, 스타일, 크기를 통해 새로운 폰트 객체를 생성한다.

GUI 프로그래밍

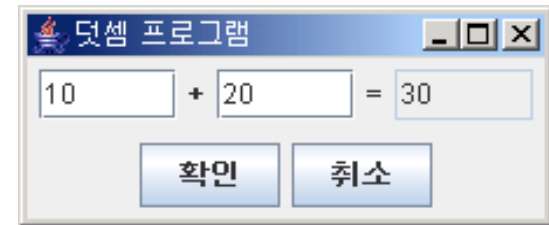
01. GUI 환경에서 작동하는 프로그램

윈도우 프로그램에 대하여

- 텍스트 모드 프로그램과 윈도우 프로그램



a) 텍스트 모드의 프로그램



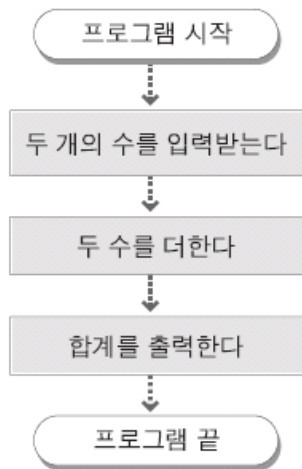
b) 윈도우 프로그램

GUI 프로그래밍

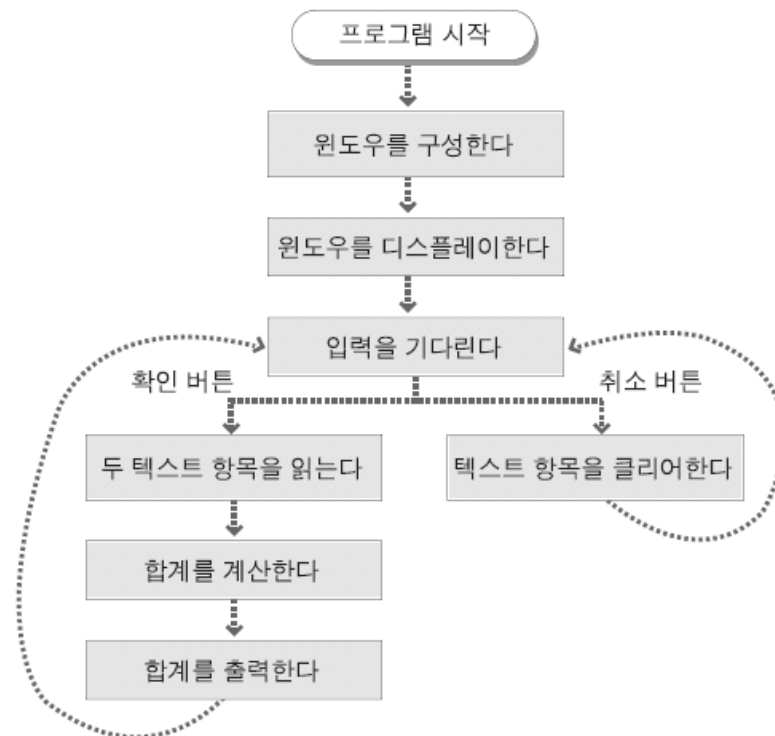
01. GUI 환경에서 작동하는 프로그램 윈도우 프로그램에 대하여

• 텍스트 모드 프로그램과 윈도우 프로그램의 구조적 차이

a) 텍스트 모드 프로그램의 구조



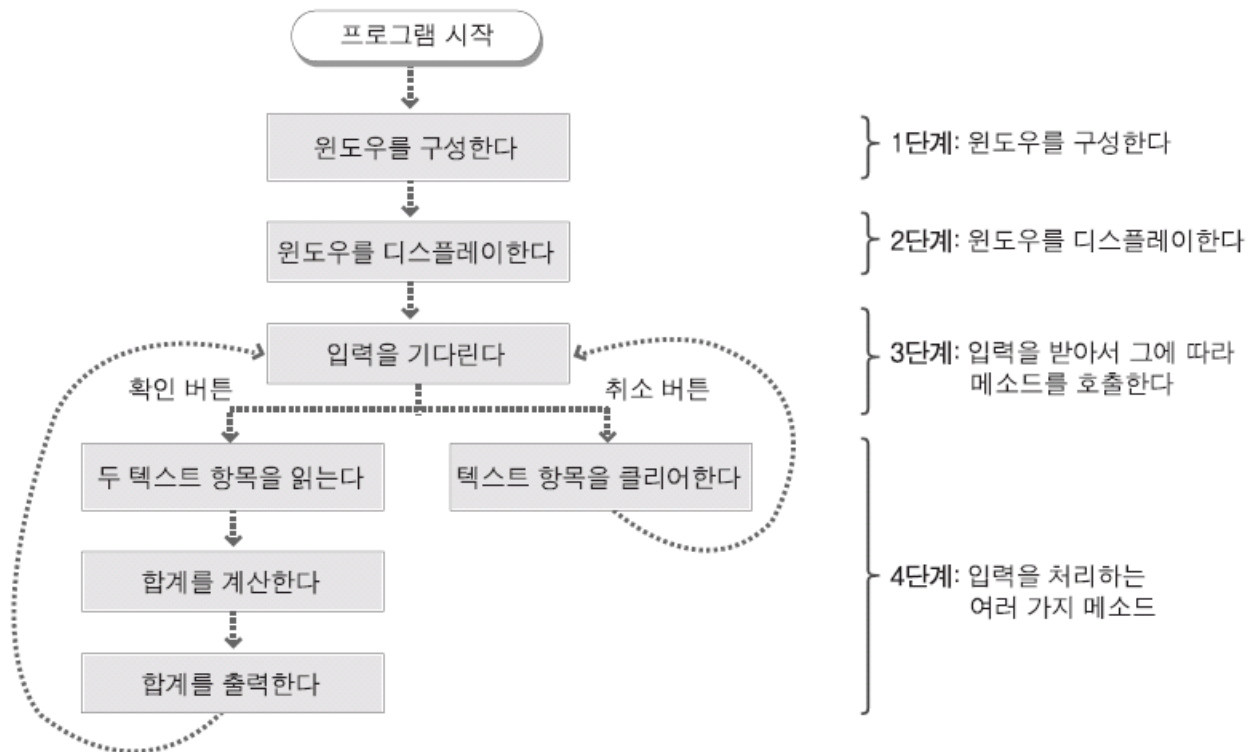
b) 윈도우 프로그램의 구조



02. 윈도우 프로그램의 작성 방법

윈도우 프로그램의 구조

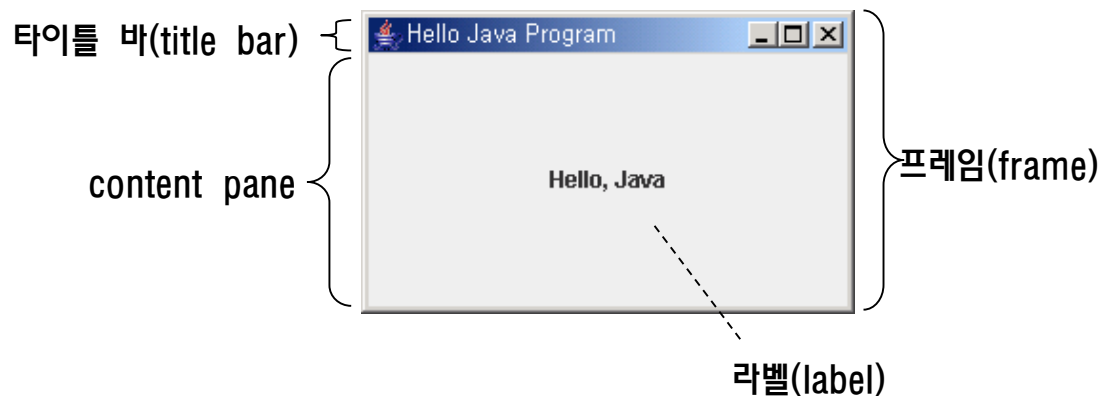
• 네 단계로 실행되는 윈도우 프로그램



02. 윈도우 프로그램의 작성 방법

윈도우의 구성요소

• 윈도우의 기본적인 구성요소



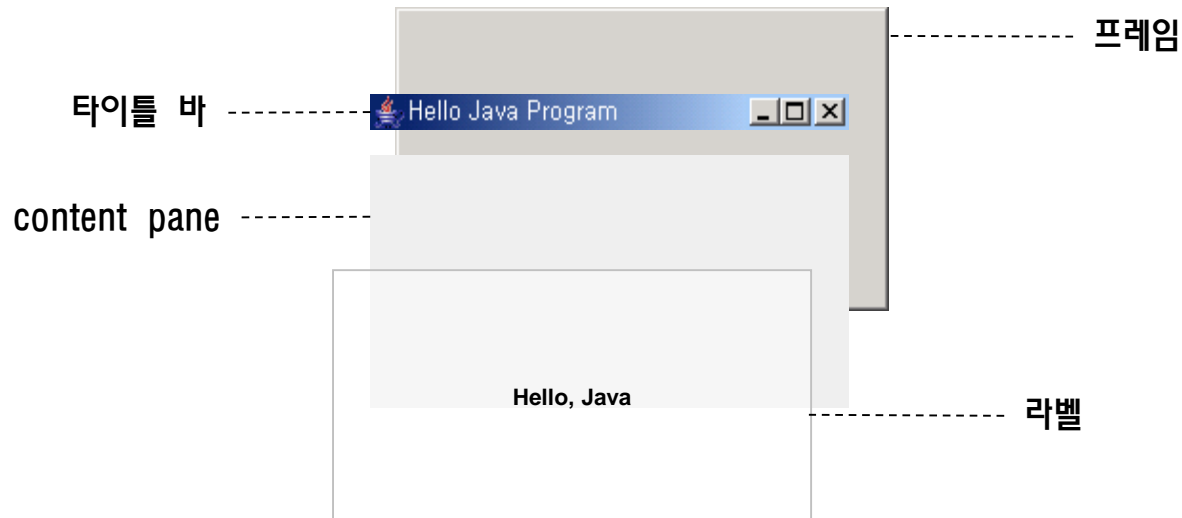
컴포넌트(component)

– 프레임, 타이틀 바, content pane, 라벨 등의 구성요소를 통틀어 이르는 용어

02. 윈도우 프로그램의 작성 방법

윈도우의 구성요소

• 컴포넌트들이 겹쳐진 층으로 구성되는 윈도우



02. 윈도우 프로그램의 작성 방법

윈도우를 구성하는 방법

- 프레임 만드는 방법

```
JFrame frame = new JFrame();
```



프레임을 생성하는 식

02. 윈도우 프로그램의 작성 방법

윈도우를 구성하는 방법

- 타이틀이 표시된 프레임 만드는 방법

```
JFrame frame = new JFrame("Hello Java Program");
```

↑
타이틀 바에 표시되는 문자열

02. 윈도우 프로그램의 작성 방법

윈도우를 구성하는 방법

- 라벨 만드는 방법

```
JLabel label = new JLabel("Hello, Java");
```

↑
라벨을 생성하는 식

● 02. 윈도우 프로그램의 작성 방법

● 윈도우를 구성하는 방법

- 프레임에 종속된 `content pane`을 가져오는 방법

```
Container contentPane = frame.getContentPane();
```

↑
content pane을 리턴하는 메소드

02. 윈도우 프로그램의 작성 방법

윈도우를 구성하는 방법

- content pane 위에 라벨을 올려놓는 방법

```
contentPane.add(label);
```

이 컴포넌트 위에 이 컴포넌트를 올려놓습니다

02. 윈도우 프로그램의 작성 방법

윈도우를 디스플레이하는 방법

- 프레임을 적절한 크기로 조정하는 방법

`frame.pack();`



프레임을 적절한 크기로 만드는 메소드

02. 윈도우 프로그램의 작성 방법

윈도우를 디스플레이하는 방법

- 프레임을 디스플레이하는 방법

```
frame.setVisible(true);
```



이 프레임



파라미터로 true 값을 넘겨주면

GUI 프로그래밍

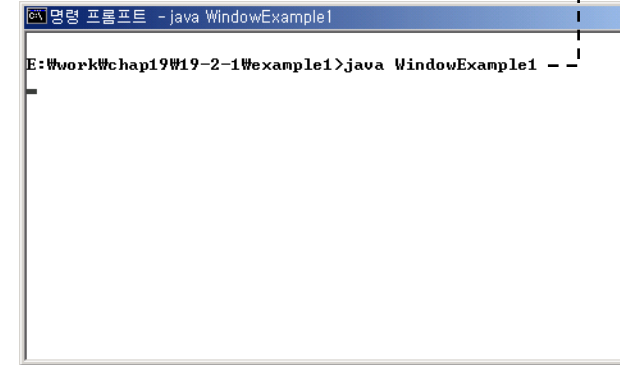
02. 윈도우 프로그램의

윈도우로 작동하는 Hello, Java

• [예제 19-1] 윈도우로 작동하는 Hello, Java

```
1  import java.awt.*;
2  import javax.swing.*;
3  class WindowExample1 {
4      public static void main(String[] args) {
5          JFrame frame = new JFrame("Hello Java Program");
6          Container contentPane = frame.getContentPane();
7          JLabel label = new JLabel("Hello, Java");
8          contentPane.add(label);
9          frame.pack();
10         frame.setVisible(true);
11     }
12 }
```

} 2 단계 : 윈도우를
디스플레이한다



02. 윈도우 프로그램의 작성 방법

Hello, Java 프로그램 완성하기

- 윈도우의 위치를 지정하는 방법

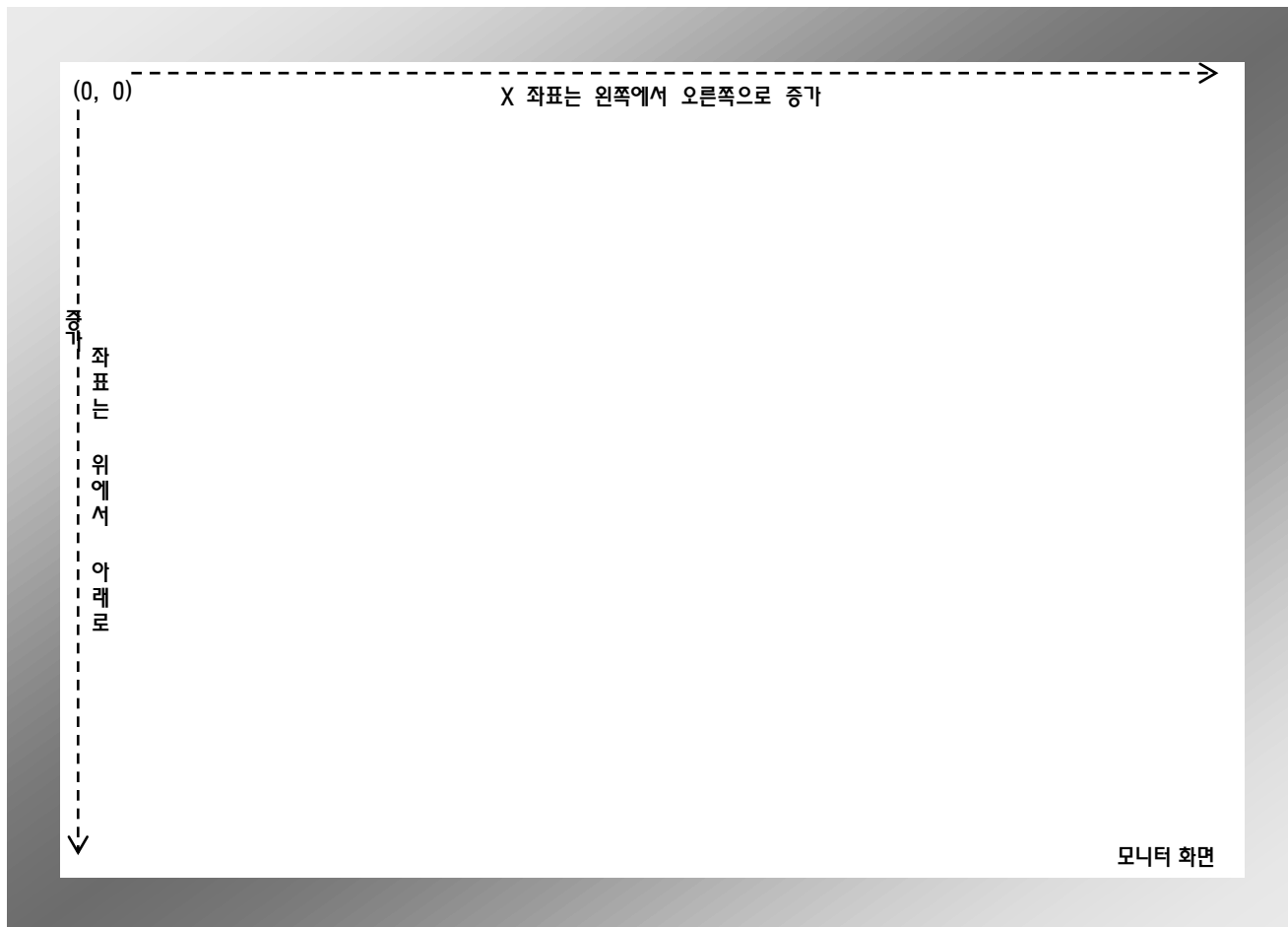
```
frame.setLocation(500, 400);
```

↑ ↑
X 좌표 Y 좌표

GUI 프로그래밍

02. 윈도우 프로그램의 작성 방법

모니터상의 좌표 체계

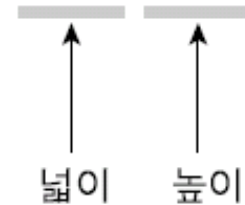


02. 윈도우 프로그램의 작성 방법

Hello, Java 프로그램 완성하기

- 프레임의 크기를 바꾸는 방법

```
frame.setPreferredSize(new Dimension(300, 200));
```



02. 윈도우 프로그램의 작성 방법

Hello, Java 프로그램 완성하기

- 라벨의 위치를 지정하는 방법

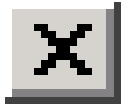
```
JLabel label = new JLabel("Hello, Java", SwingConstants.CENTER);
```

↑
라벨의 문자열을 중앙에 표시하도록 만드는 파라미터

02. 윈도우 프로그램의 작성 방법

Hello, Java 프로그램 완성하기

- 버튼을 눌렀을 때 윈도우가 닫히도록 만드는 방법



```
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

✕ 버튼을 눌렀을 때 해야 할 일을 지시하는 메소드

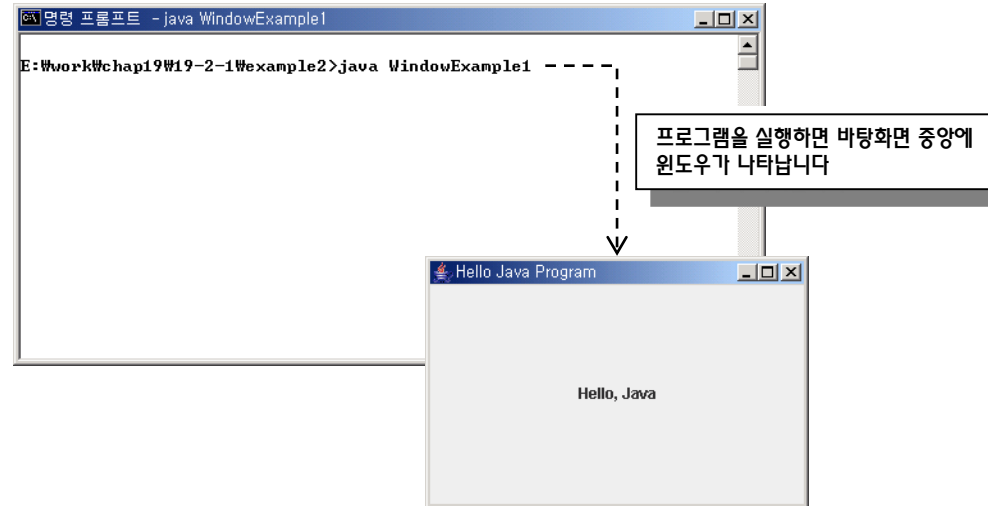
GUI 프로그래밍

02. 윈도우 프로그램의 작성 방법

Hello, Java 프로그램 완성하기

- [예제 19-2] 윈도우로 작동하

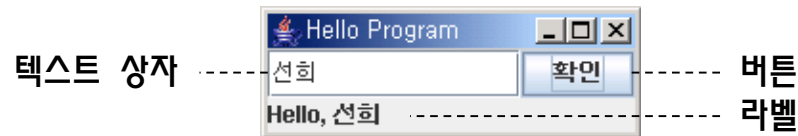
```
1  import java.awt.*;
2  import javax.swing.*;
3  class WindowExample1 {
4      public static void main(String[] args) {
5          JFrame frame = new JFrame("Hello, Java");
6          frame.setLocation(500, 400);
7          frame.setPreferredSize(new Dimension(300, 200));
8          Container contentPane = frame.getContentPane();
9          JLabel label = new JLabel("Hello, Java",
10 SwingConstants.CENTER);
11          contentPane.add(label);
12          frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
13          frame.pack();
14          frame.setVisible(true);
15      }
16  }
```



02. 윈도우 프로그램의 작성 방법

텍스트 상자와 버튼

•• 지금부터 작성할 예제의 윈도우



02. 윈도우 프로그램의 작성 방법

텍스트 상자와 버튼

- 텍스트 상자를 만드는 방법

```
JTextField text = new JTextField();
```

↑
텍스트 상자를 생성하는 식

02. 윈도우 프로그램의 작성 방법

텍스트 상자와 버튼

• 버튼을 만드는 방법

```
JButton button = new JButton("확인");
```

↑
버튼을 생성하는 식

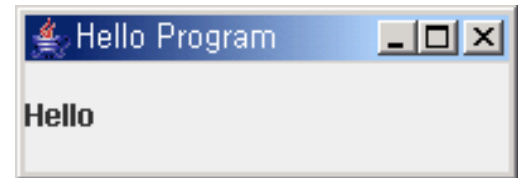
02. 윈도우 프로그램의 작성 방법

텍스트 상자와 버튼

- 여러 개의 컴포넌트를 `content pane` 위에 올려놓는 방법 – 잘못된 예

```
contentPane.add(text);  
contentPane.add(button);  
contentPane.add(label);
```

이렇게 하면 마지막에
추가된 라벨만 보이게 됩니다



02. 윈도우 프로그램의 작성 방법

텍스트 상자와 버튼

- content pane의 기본 레이아웃 – border layout



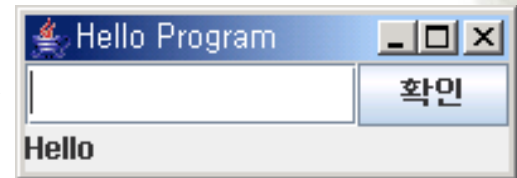
02. 윈도우 프로그램의 작성 방법

텍스트 상자와 버튼

- 여러 개의 컴포넌트를 *content pane* 위에 올려놓는 방법 – 올바른 예

```
contentPane.add(text, BorderLayout.CENTER);  
contentPane.add(button, BorderLayout.EAST);  
contentPane.add(label, BorderLayout.SOUTH);
```

이렇게 배치됩니다



GUI 프로그래밍

02. 윈도우 프로그램의 작성 방법

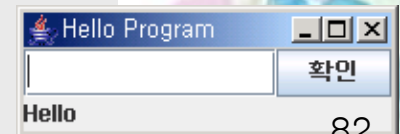
텍스트 상자와 버튼

- [예제 19-3] 텍스트 상자와 버튼이 있는 Hello 프로그램 (미완성)

```
1  import java.awt.*;
2  import javax.swing.*;
3  class WindowExample2 {
4      public static void main(String[] args) {
5          JFrame frame = new JFrame("Hello Program");
6          frame.setPreferredSize(new Dimension(200, 70));
7          frame.setLocation(500, 400);
8          Container contentPane = frame.getContentPane();
9          JTextField text = new JTextField();
10         JButton button = new JButton("확인");
11         JLabel label = new JLabel("Hello");
12         contentPane.add(text);
13         contentPane.add(button, BorderLayout.EAST);
14         contentPane.add(label, BorderLayout.SOUTH);
15
16         frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
17         frame.pack();
18     }
```

1단계: 윈도우를
구성한다

2단계: 윈도우를 디스플레이한다



02. 윈도우 프로그램의 작성 방법

버튼 이벤트의 처리 방법

- 텍스트 상자의 텍스트를 가져오는 방법

```
String str = text.getText();
```



텍스트 상자의 텍스트를 가져오는 메소드

● 02. 윈도우 프로그램의 작성 방법

● 버튼 이벤트의 처리 방법

- 라벨에 텍스트를 표시하는 방법


```
label.setText(str);
```

라벨에 쓰여진 텍스트를 바꾸는 메소드

02. 윈도우 프로그램의 작성 방법

버튼 이벤트의 처리 방법

- 버튼을 눌렀을 때 해야할 일을 기술하는 방법

```
class ConfirmButtonActionListener implements ActionListener {  
    public void actionPerformed(ActionEvent e) {  
          
    }  
}
```

버튼을 눌렀을 때 해야할 일을 기술하는 부분

02. 윈도우 프로그램의 작성 방법

버튼 이벤트의 처리 방법

- 버튼을 눌렀을 때 `actionPerformed` 메소드가 호출되도록 만드는 방법

```
ActionListener listener = new ConfirmButtonActionListener();  
  
button.addActionListener(listener);
```

action listener 객체를 만들어서 버튼에 등록합니다

GUI 프로그래밍

02. 윈도우 프로그램의 작성 방법

버튼 이벤트의 처리 방법

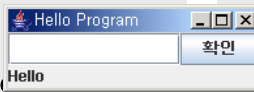
- [예제 19-4] 텍스트 상자와 버튼이 있는 Hello 프로그램 (완성)

main 메소드를 포함하는 클래스

```
1 import java.awt.*;
2 import javax.swing.*;
3 import java.awt.event.*;
4 class WindowExample2 {
5     public static void main(String[] args) {
6         JFrame frame = new JFrame("Hello Program");
7         frame.setPreferredSize(new Dimension(200, 70));
8         frame.setLocation(500, 400);
9         Container contentPane = frame.getContentPane();
10        JTextField text = new JTextField();
11        JButton button = new JButton("확인");
12        JLabel label = new JLabel("Hello");
13        contentPane.add(text, BorderLayout.CENTER);
14        contentPane.add(button, BorderLayout.EAST);
15        contentPane.add(label, BorderLayout.SOUTH);
16        ActionListener listener =
17            new ConfirmButtonActionListener(text, label);
18        button.addActionListener(listener);
19        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
20        frame.pack();
21        frame.setVisible(true);
22    }
23 }
```

확인 버튼을 처리하는 리스너 클래스

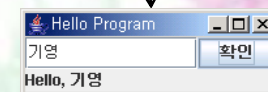
```
1 import javax.swing.*;
2 import java.awt.event.*;
3 class ConfirmButtonActionListener implements ActionListener {
4     JTextField text;
5     JLabel label;
6     ConfirmButtonActionListener(JTextField text, JLabel label) {
7         this.text = text;
8         this.label = label;
9     }
10    public void actionPerformed(ActionEvent e) {
11        String name = text.getText();
12        label.setText("Hello, " + name);
13    }
14 }
```



텍스트 상자에 “선희” 라고 입력하고
확인 버튼을 눌렀을 때



텍스트 상자에 “기영” 이라고 입력하고
확인 버튼을 눌렀을 때



02. 윈도우 프로그램의 작성 방법

레이아웃과 패널

- 자주 사용되는 레이아웃들

a) grid layout



b) flow layout



c) box layout



02. 윈도우 프로그램의 작성 방법

레이아웃과 패널

• content pane의 레이아웃을 설정하는 방법

```
FlowLayout layout = new FlowLayout();
```

content pane의 레이아웃을
flow layout으로 설정합니다

```
contentPane.setLayout(layout);
```

```
GridLayout layout = new GridLayout(2, 3);
```

content pane의 레이아웃을
2행 3열의 grid layout으로 설정합니다

```
contentPane.setLayout(layout);
```

02. 윈도우 프로그램의 작성 방법

레이아웃과 패널

- content pane의 레이아웃을 설정하는 방법 (계속)

```
BoxLayout layout = new BoxLayout(contentPane, BoxLayout.X_AXIS);
```

content pane의 레이아웃을
가로 방향의 box layout으로 설정합니다

```
contentPane.setLayout(layout);
```

GUI 프로그래밍

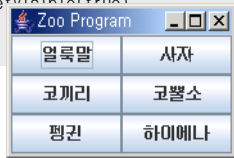
02. 윈도우 프로그램의 작성 방법

레이아웃과 패널

• [예제 19-5] 여러 가지 레이아웃을 사용하는 프로그램들

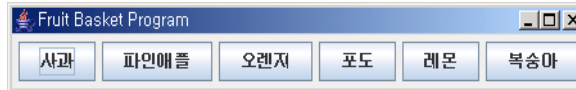
grid layout을 사용하는 프로그램

```
1 import java.awt.*;
2 import javax.swing.*;
3 class WindowExample3 {
4     public static void main(String[] args) {
5         JFrame frame = new JFrame("Zoo Program");
6         frame.setLocation(500, 400);
7         Container contentPane = frame.getContentPane();
8         GridLayout layout = new GridLayout(3, 2);
9         contentPane.setLayout(layout);
10        contentPane.add(new JButton("얼룩말"));
11        contentPane.add(new JButton("사자"));
12        contentPane.add(new JButton("코끼리"));
13        contentPane.add(new JButton("코뿔소"));
14        contentPane.add(new JButton("펭귄"));
15        contentPane.add(new JButton("하이에나"));
16        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
17        frame.pack();
18        frame.setVisible(true);
19    }
20 }
```



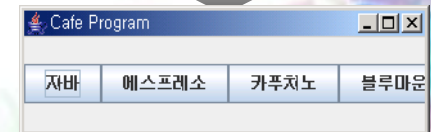
flow layout을 사용하는 프로그램

```
1 import java.awt.*;
2 import javax.swing.*;
3 class WindowExample4 {
4     public static void main(String[] args) {
5         JFrame frame = new JFrame("Fruit Basket Program");
6         frame.setLocation(500, 400);
7         Container contentPane = frame.getContentPane();
8         FlowLayout layout = new FlowLayout();
9         contentPane.setLayout(layout);
10        contentPane.add(new JButton("사과"));
11        contentPane.add(new JButton("파인애플"));
12        contentPane.add(new JButton("오렌지"));
13        contentPane.add(new JButton("포도"));
14        contentPane.add(new JButton("레몬"));
15        contentPane.add(new JButton("복숭아"));
16        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
17        frame.pack();
18        frame.setVisible(true);
19    }
20 }
```



box layout을 사용하는 프로그램

```
1 import java.awt.*;
2 import javax.swing.*;
3 class WindowExample5 {
4     public static void main(String[] args) {
5         JFrame frame = new JFrame("Cafe Program");
6         frame.setLocation(500, 400);
7         Container contentPane = frame.getContentPane();
8         BoxLayout layout = new BoxLayout(contentPane,
9                                         BoxLayout.X_AXIS);
10        contentPane.setLayout(layout);
11        contentPane.add(new JButton("자바"));
12        contentPane.add(new JButton("에스프레소"));
13        contentPane.add(new JButton("카푸치노"));
14        contentPane.add(new JButton("블루마운틴"));
15        contentPane.add(new JButton("콜럼비아"));
16        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
17        frame.pack();
18        frame.setVisible(true);
19    }
20 }
```



02. 윈도우 프로그램의 작성 방법

레이아웃과 패널

• 다음과 같은 윈도우는 어떤 레이아웃을 이용해서 만들 수 있을까요?

패널(panel)을 만들어서
content pane 위에 올려놓고
그 위에 컴포넌트를 올려놓으면 됩니다.

02. 윈도우 프로그램의 작성 방법

레이아웃과 패널

- 패널을 이용하여 복합적인 레이아웃을 설정하는 방법
 - 1) 패널을 생성해서 content pane 위에 추가합니다.

```
JPanel panel = new JPanel();
```

패널을 생성해서
content pane에 추가합니다

```
contentPane.add(panel);
```

02. 윈도우 프로그램의 작성 방법

레이아웃과 패널

- 패널을 이용하여 복합적인 레이아웃을 설정하는 방법
 - 2) 패널의 레이아웃을 설정합니다.

```
GridLayout layout = new GridLayout(3, 2);
```

패널의 레이아웃을 3행 2열의
grid layout으로 설정합니다

```
panel.setLayout(layout);
```

02. 윈도우 프로그램의 작성 방법

레이아웃과 패널

- 패널을 이용하여 복잡한 레이아웃을 설정하는 방법
 - 3) `add` 메소드를 이용하여 패널 위에 컴포넌트를 추가합니다.

```
panel.add(label1);  
panel.add(text1);  
panel.add(label2);  
panel.add(text2);  
panel.add(label3);  
panel.add(text3);
```

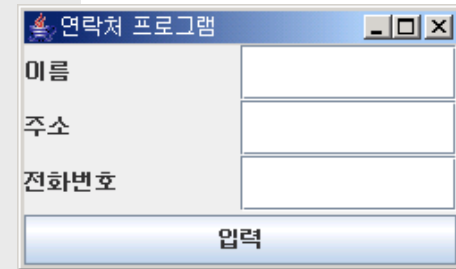
} 패널에 6개의 컴포넌트를 추가합니다

02. 윈도우 프로그램의 작성 방법

레이아웃과 패널

- [예제 19-6] JPanel의 사용 예를 보여주는 프로그램

```
1 import java.awt.*;
2 import javax.swing.*;
3 class WindowExample6 {
4     public static void main(String[] args) {
5         JFrame frame = new JFrame("연락처 프로그램");
6         frame.setPreferredSize(new Dimension(250, 150));
7         frame.setLocation(500, 400);
8         Container contentPane = frame.getContentPane();
9         JPanel panel = new JPanel(); ----- 패널을 생성합니다
10        panel.setLayout(new GridLayout(3, 2)); ----- 패널의 레이아웃을 설정합니다
11        JTextField text1 = new JTextField();
12        JTextField text2 = new JTextField();
13        JTextField text3 = new JTextField();
14        panel.add(new JLabel("이름"));
15        panel.add(text1);
16        panel.add(new JLabel("주소"));
17        panel.add(text2);
18        panel.add(new JLabel("전화번호"));
19        panel.add(text3);
20        contentPane.add(panel, BorderLayout.CENTER); ----- 패널을 content pane에 추가합니다
21        contentPane.add(new JButton("입력"), BorderLayout.SOUTH);
22        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
23        frame.pack();
24        frame.setVisible(true);
25    }
26 }
```



02. 윈도우 프로그램의 작성 방법

테이블

•• 지금부터 작성할 예제의 윈도우



02. 윈도우 프로그램의 작성 방법

테이블

• 테이블을 만드는 방법

- 1) 테이블을 생성합니다.

```
String colNames[] = { "이름", "나이", "성별" };  
Object data[][] = { { "김철수", 24, '남' },  
                    { "이순희", 21, '여' },  
                    { "박지영", 26, '여' } };  
...  
JTable table = new JTable(data, colNames);
```

데이터 항목이 있는 2차원 배열

컬럼 이름이 있는 1차원 배열

02. 윈도우 프로그램의 작성 방법

테이블

• 테이블을 만드는 방법

- 2) 스크롤 판을 만들어서 테이블을 올려놓고, 그 스크롤 판을 content pane 위에 올려놓습니다.

```
JScrollPane scrollPane = new JScrollPane(table);
```

테이블이 올려진
스크롤 판을 생성

스크롤 판을 content pane에 추가합니다

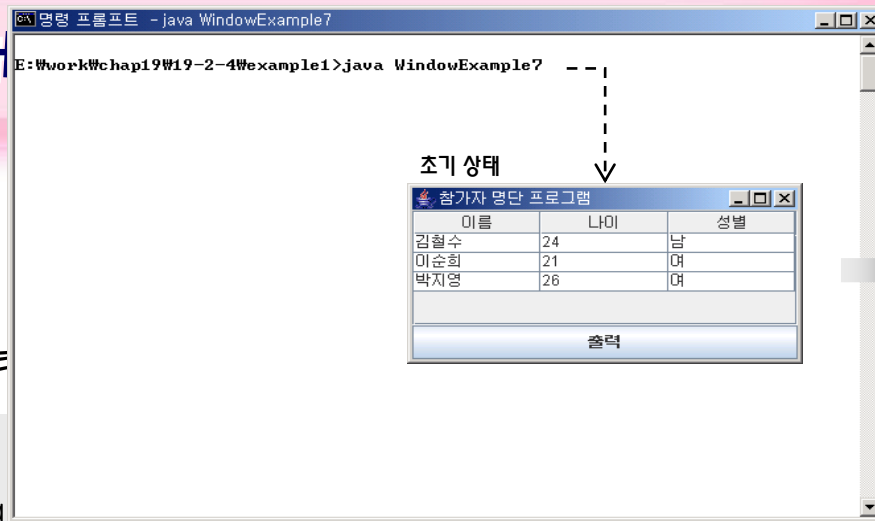
```
contentPane.add(scrollPane);
```

GUI 프로그래밍

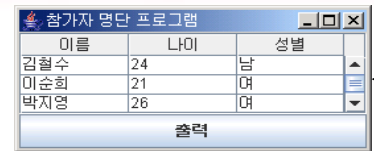
02. 윈도우 테이블

• [예제 19-7] 테이블

```
1 import java.awt.*;
2 import javax.swing.*;
3 class WindowExample7 {
4     public static void main(String[] args) {
5         JFrame frame = new JFrame("참가자 명단 프로그램");
6         frame.setPreferredSize(new Dimension(300, 150));
7         frame.setLocation(500, 400);
8         Container contentPane = frame.getContentPane();
9         String colNames[] = { "이름", "나이", "성별" };
10        Object data[][] = { { "김철수", 24, '남' },
11                             { "이순희", 21, '여' },
12                             { "박지영", 26, '여' } };
13        JTable table = new JTable(data, colNames);
14        JScrollPane scrollPane = new JScrollPane(table);
15        contentPane.add(scrollPane, BorderLayout.CENTER);
16        JButton button = new JButton("출력");
17        contentPane.add(button, BorderLayout.SOUTH);
18        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```



변



크기를 줄이면
스크롤 바가 나타납니다

02. 윈도우 프로그램의 작성 방법

테이블에 있는 정보 가져오기

- 테이블의 데이터는 `JTable` 객체가 아닌 별도의 객체에 저장됨

↑
모델(model)

02. 윈도우 프로그램의 작성 방법

테이블에 있는 정보 가져오기

- 테이블의 모델을 가져오는 방법

```
TableModel model = table.getModel();
```

테이블의 모델을 가져오는 메소드

02. 윈도우 프로그램의 작성 방법

테이블에 있는 정보 가져오기

•• 모델의 데이터를 가져오거나 바꾸는 방법

```
Object obj = model.getValueAt(2, 3);    model.setValueAt("김수영", 1, 0);
```

(2, 3) 위치에 있는 데이터를 가져옵니다

(1, 0) 위치에 있는 데이터를
"김수영"으로 바꿉니다

02. 윈도우 프로그램의 작성 방법

테이블에 있는 정보 가져오기

•• 모델로부터 테이블 정보를 가져오는 방법

```
int rowNum = model.getRowCount();   int colNum = model.getColumnCount();
```

↑
행의 수를 가져오는 메소드

↑
열의 수를 가져오는 메소드

```
String str = model.getColumnName(cnt);
```

↑
cnt번째 컬럼 이름을 가져옵니다

GUI 프로그래밍

02. 윈도우 프로그램의 작성 방법

테이블에 있는 정보 가져오기

- [예제 19-8] 테이블이 있는 윈도우 프로그램 (완성)

main 메소드를 포함하는 클래스

```
1 import java.awt.*;
2 import javax.swing.*;
3 class WindowExample7 {
4     public static void main(String[] args) {
5         JFrame frame = new JFrame("참가자 명단 프로그램");
6         frame.setPreferredSize(new Dimension(300, 150));
7         frame.setLocation(500, 400);
8         Container contentPane = frame.getContentPane();
9         String colNames[] = { "이름", "나이", "성별" };
10        Object data[][] = { {
11            "김철수", 24, "남",
12            "이순희", 21, "여",
13            "박지영", 26, "여"
14        } };
15        JButton button = new JButton("출력");
16        contentPane.add(button);
17        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
18        frame.pack();
19        frame.setVisible(true);
20    }
21 }
```

출력 버튼을 처리하는 리스너 클래스

```
1 import java.awt.event.*;
2 import javax.swing.*;
3 import javax.swing.table.*;
4 class PrintActionListener implements ActionListener {
5     JTable table;
6     PrintActionListener(JTable table) {
7         this.table = table;
8     }
9     public void actionPerformed(ActionEvent e) {
```

```
        Model model = table.getModel(); // 모델을 가져옵니다.
        int rows = model.getRowCount(); // 모델에서 행과 열의 수를 가져옵니다.
        Count count = new Count(rows);
        um: col++; {
            setColumnName(col); // 컬럼 이름을 가져와서 출력합니다.
            + "Wt");
        }
        +) {
        col++; {
        t(row, col);
        };
        + "Wt");
    }
}
```

출력 버튼을 누르면
명령 프롬프트 창으로
테이블 내용이 출력됩니다

이름	나이	성별
김철수	24	남
이순희	21	여
박지영	26	여

출력

● 02. 윈도우 프로그램의 작성 방법

● 테이블에 행 추가/삭제하기

- 테이블을 생성할 때 자동으로 생기는 모델에는 행의 추가/삭제 기능이 없습니다.
 - -> 테이블에 행을 추가/삭제하려면 모델을 직접 만들어서 사용해야 합니다.

02. 윈도우 프로그램의 작성 방법

테이블에 행 추가/삭제하기

- 행 추가/삭제 기능이 있는 모델을 생성하는 방법

```
String colNames[] = { "이름", "나이", "성별" };  
  
DefaultTableModel model = new DefaultTableModel(colNames, 0);
```

컬럼 이름이 있는 1차원 배열

행의 수

02. 윈도우 프로그램의 작성 방법

테이블에 행 추가/삭제하기

- 직접 만든 모델에 연관된 테이블을 생성하는 방법

```
JTable table = new JTable(model);
```

↑
이 모델에 데이터를 저장하는
테이블을 생성합니다

02. 윈도우 프로그램의 작성 방법

테이블에 행 추가/삭제하기

- DefaultTableModel에 행을 추가하는 방법

```
Object data[] = { "박승규", 27, '남' };
```

```
model.addRow(data);
```

테이블의 마지막에 행을 추가합니다

```
Object data[] = { "최지영", 25, '여' };
```

```
model.insertRow(2, data);
```

테이블의 2번째 위치에 행을 추가합니다

02. 윈도우 프로그램의 작성 방법

테이블에 행 추가/삭제하기

- DefaultTableModel에 있는 행을 삭제하는 방법

```
model.removeRow(3);
```

↑
테이블의 3번째 위치에 있는
행을 삭제합니다

GUI 프로그래밍

02. 윈도우 프로그램의 작성 방법

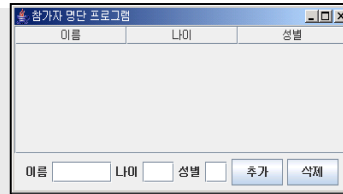
테이블에 행 추가/삭제하기

- [예제 19-9] 테이블에 행을 추가/삭제하는 윈도우 프로그램

main 메소드를 포함하는 클래스

```
1 import java.awt.*;
2 import javax.swing.*;
3 import javax.swing.table.*;
4 class WindowExample8 {
5     public static void main(String[] args) {
6         JFrame frame = new JFrame("참가자 명단 프로그램");
7         frame.setPreferredSize(new Dimension(400, 200));
8         frame.setLocation(500, 400);
9         Container contentPane = frame.getContentPane();
10        String colNames[] = { "이름", "나이", "성별" };
11        DefaultTableModel model = new DefaultTableModel(colNames, 0);
12        JTable table = new JTable(model);
13        contentPane.add(new JScrollPane(table), BorderLayout.CENTER);
14        JPanel panel = new JPanel();
15        JTextField text1 = new JTextField(6);
16        JTextField text2 = new JTextField(3);
17        JTextField text3 = new JTextField(2);
18        JButton button1 = new JButton("추가");
19        JButton button2 = new JButton("삭제");
20        panel.add(new JLabel("이름"));
21        panel.add(text1);
22        panel.add(new JLabel("나이"));
23        panel.add(text2);
24        panel.add(new JLabel("성별"));
25        panel.add(text3);
26        panel.add(button1);
27        panel.add(button2);
28        contentPane.add(panel, BorderLayout.SOUTH);
29        button1.addActionListener(
30            new AddActionListener(table, text1, text2, text3));
31        button2.addActionListener(new RemoveActionListener(table));
32        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
33        frame.pack();
34        frame.setVisible(true);
35    }
```

[화면 설계]



삭제 버튼을 처리하는 리스너 클래스

```
1 import java.awt.event.*;
2 import javax.swing.*;
3 import javax.swing.table.*;
4 class RemoveActionListener implements ActionListener {
5     JTable table;
6     RemoveActionListener(JTable table) {
7         this.table = table;
8     }
9     public void actionPerformed(ActionEvent e) {
10        int row = table.getSelectedRow();
11        if (row == -1)
12            return;
13        DefaultTableModel model = (DefaultTableModel) table.getModel();
14        model.removeRow(row);
15    }
16 }
```

추가 버튼을 처리하는 리스너 클래스

```
1 import java.awt.event.*;
2 import javax.swing.*;
3 import javax.swing.table.*;
4 class AddActionListener implements ActionListener {
5     JTable table;
6     JTextField text1, text2, text3;
7     AddActionListener(JTable table, JTextField text1,
8         JTextField text2, JTextField text3) {
9         this.table = table;
10        this.text1 = text1;
11        this.text2 = text2;
12        this.text3 = text3;
13    }
14    public void actionPerformed(ActionEvent e) {
15        String arr[] = new String[3];
16        arr[0] = text1.getText();
17        arr[1] = text2.getText();
18        arr[2] = text3.getText();
19        DefaultTableModel model = (DefaultTableModel) table.getModel();
20        model.addRow(arr);
21    }
22 }
```

02. 윈도우 프로그램의 작성 방법

테이블에 행 추가/삭제하기

- [예제 19-9] 테이블에 행을 추가/삭제하는 윈도우 프로그램 - 실행 결과

초기 상태

이름	나이	성별
----	----	----

이름 나이 성별

이름, 나이, 성별을 입력하고
추가 버튼을 눌렀을 때

이름	나이	성별
이주현	25	여

이름 이주현 나이 25 성별 여

이름	나이	성별
이주현	25	여
김보경	26	여
최경준	28	남

이름 최경준 나이 28 성별 남

마우스로 행을 클릭하고
삭제 버튼을 눌렀을 때

이름	나이	성별
이주현	25	여
최경준	28	남

이름 최경준 나이 28 성별 남

● 03. 그래픽과 이미지 디스플레이

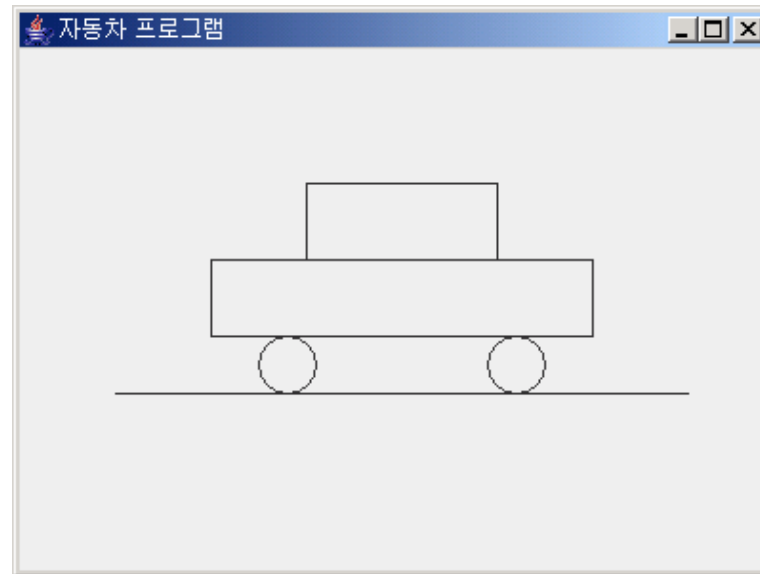
● 그래픽 프리미티브로 그림 그리기

- 윈도우에 그림을 그리기 위해 필요한 것
 - 종이의 역할 : 패널 클래스(Jpanel의 서브클래스)
 - 펴기도구의 역할 : java.awt.Graphics 클래스

03. 그래픽과 이미지 디스플레이

그래픽 프리미티브로 그림 그리기

•• 지금부터 작성할 예제의 윈도우



● 03. 그래픽과 이미지 디스플레이

● 그래픽 프리미티브로 그림 그리기

- 윈도우에 그림을 그리는 방법
 - 1) 그림이 그려진 패널을 만듭니다.
 - 2) 그 패널을 `content pane` 위에 올려놓습니다.

03. 그래픽과 이미지 디스플레이

그래픽 프리미티브로 그림 그리기

- 그림이 그려진 패널을 만드는 방법
 - 1) JPanel의 서브클래스를 선언합니다.

```
class CarDrawingPanel extends JPanel {  
    public void paint(Graphics g) {  
          
    }  
}
```

그림을 그리는 명령문을
이 부분에 써야 합니다

이렇게 선언된 paint 메소드는
패널이 디스플레이 될 때
JDK 라이브러리 모듈에 의해 자동으로 호출됩니다.

03. 그래픽과 이미지 디스플레이

그래픽 프리미티브로 그림 그리기

•• 그림이 그려진 패널을 만드는 방법

- 2) paint 메소드 안에 다음과 같은 메소드 호출문을 써서 그림을 그립니다.

`g.drawRect(100, 110, 200, 40);`

↑
(100, 110) 위치에 200 X 40 크기의
직사각형을 그립니다

`g.drawOval(125, 150, 30, 30);`

↑
(125, 150) 위치에 지름이 30인
원을 그립니다

`g.drawLine(50, 180, 350, 180);`

↑
(50, 180)부터 (350, 180) 위치까지
직선을 그립니다.

GUI 프로그래밍

03. 그래픽과 이미지 디스플레이

그래픽 프리미티브로 그림 그리기

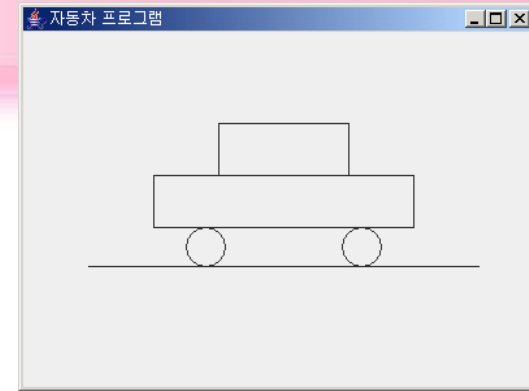
- [예제 19-10] 자동차 그림이 그려진 윈도우 프로그램

main 메소드를 포함하는 클래스

```
1 import java.awt.*;
2 import javax.swing.*;
3 class GraphicExample1 {
4     public static void main(String[] args) {
5         JFrame frame = new JFrame("자동차 프로그램");
6         frame.setLocation(500, 200);
7         frame.setPreferredSize(new Dimension(400,
8 300));
9         Container contentPane =
10 frame.getContentPane();
11         CarDrawingPanel drawingPanel = new
12 CarDrawingPanel();
13         contentPane.add(drawingPanel);
14 frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
15         frame.pack();
16         frame.setVisible(true);
17     }
18 }
```

자동차 그림이 있는 패널 클래스

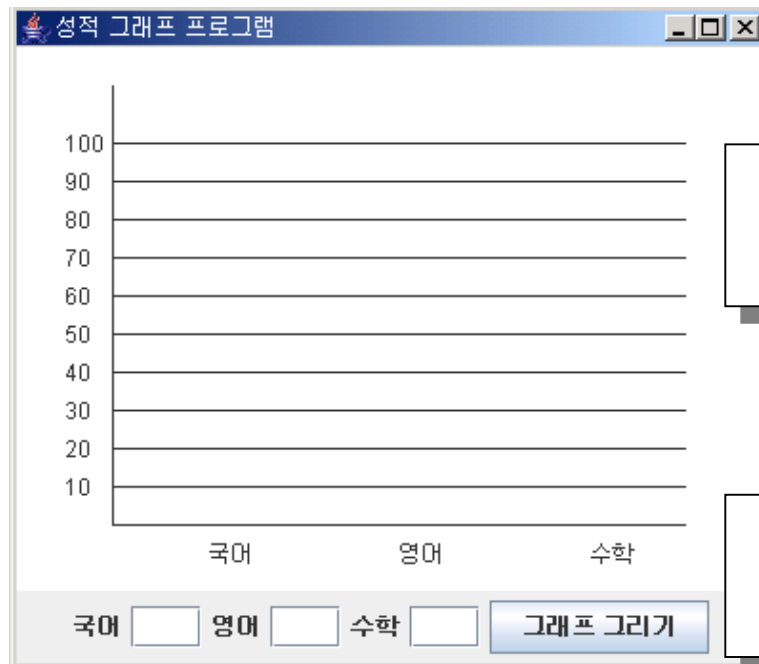
```
1 import javax.swing.*;
2 import java.awt.*;
3 class CarDrawingPanel extends
4 JPanel {
5     public void paint(Graphics
6 g) {
7         g.drawRect(100, 110, 200,
8 40);
9         g.drawRect(150, 70, 100,
10 40);
11         g.drawOval(125, 150, 30,
12 30);
13         g.drawOval(245, 150, 30,
14 30);
15         g.drawLine(50, 180, 350,
16 180);
17     }
18 }
```



02. 윈도우 프로그램의 작성 방법

데이터를 이용해서 그래프 그리기

• 지금부터 작성할 예제의 윈도우



버튼을 눌렀을 때 그래프가 그려지도록
하려면 어떻게 해야할까요?



paint 메소드 안에서
if 조건문을 사용하면 됩니다.

02. 윈도우 프로그램의 작성 방법

데이터를 이용해서 그래프 그리기

- 버튼을 눌렀을 때 막대 그래프가 그려지도록 하려면;
 - -> repaint 메소드를 호출하여 paint 메소드를 간접 호출해야 합니다.

```
drawingPanel.repaint();
```

↑
paint 메소드를 간접적으로
호출하는 메소드

GUI 프로그래밍

02. 윈도우 프로그램의 작성 방법

데이터를 이용해서 그래프 그리기

• 하나의 paint 메소드를 가지고 두 가지 그림을 그리는 방법

```
class DrawingPanel extends JPanel {  
    > int korean, english, math;  
    public void paint(Graphics g) {  
        ...  
        if (korean > 0)  
            g.fillRect(110, 250 - korean * 2, 10, korean * 2);  
        if (english > 0)  
            g.fillRect(210, 250 - english * 2, 10, english * 2);  
        if (math > 0)  
            g.fillRect(310, 250 - math * 2, 10, math * 2);  
    }  
    void setScores(int korean, int english, int math) {  
        this.korean = korean;  
        this.english = english;  
        this.math = math;  
    }  
}
```

국어, 영어, 수학
점수를
필드에 대입합니다

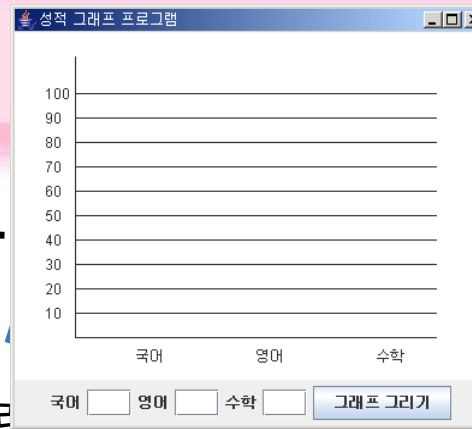
필드 값을 가져다가
막대 그래프를
그립니다

점수 필드 값을 설정하는
메소드

GUI 프로그래밍

03. 그래픽과 데이터를 이용해

• [예제 19-11] 성적 그래프



플레이

점수를 입력하고 버튼을 누르면
막대 그래프가 출력됩니다



main 메소드를 포함하는 클래스

```
1 import java.awt.*;
2 import javax.swing.*;
3 class GraphicExample2 {
4     public static void main(String[] args) {
5         JFrame frame = new JFrame("성적 그래프 프로그램");
6         frame.setLocation(500, 200);
7         frame.setPreferredSize(new Dimension(400, 350));
8         Container contentPane = frame.getContentPane();
9         DrawingPanel drawingPanel = new DrawingPanel();
10        contentPane.add(drawingPanel, BorderLayout.CENTER);
11        JPanel controlPanel = new JPanel();
12        JTextField text1 = new JTextField(3);
13        JTextField text2 = new JTextField(3);
14        JTextField text3 = new JTextField(3);
15        JButton button = new JButton("그래프 그리기");
16        controlPanel.add(new JLabel("국어"));
17        controlPanel.add(text1);
18        controlPanel.add(new JLabel("영어"));
19        controlPanel.add(text2);
20        controlPanel.add(new JLabel("수학"));
21        controlPanel.add(text3);
22        controlPanel.add(button);
23        contentPane.add(controlPanel, BorderLayout.SOUTH);
24        frame.setDefaultCloseOperation(
25            JFrame.EXIT_ON_CLOSE);
26        button.addActionListener(new DrawActionListener(
27            text1, text2, text3, drawingPanel));
28        frame.pack();
29        frame.setVisible(true);
30    }
31 }
```

그래프를 그리는 패널 클래스

```
1 import java.awt.*;
2 import javax.swing.*;
3 class DrawingPanel extends JPanel {
4     int korean, english, math;
5     public void paint(Graphics g) {
6         g.clearRect(0, 0, getWidth(), getHeight());
7         g.drawLine(50, 250, 350, 250);
8         for (int cnt = 1; cnt < 11; cnt++) {
9             g.drawString(cnt*10 + "", 25, 255 - 20*cnt);
10            g.drawLine(50, 250 - 20*cnt, 350, 250 - 20*cnt);
11        }
12        g.drawLine(50, 20, 350, 20);
13        g.drawString("국어", 100, 270);
14        g.drawString("영어", 200, 270);
15        g.drawString("수학", 300, 270);
16        g.setColor(Color.RED);
17        if (korean > 0)
18            g.fillRect(110, 250 - korean*2, 10, korean*2);
19        if (english > 0)
20            g.fillRect(210, 250 - english*2, 10, english*2);
21        if (math > 0)
22            g.fillRect(310, 250 - math*2, 10, math*2);
23        void setScores(int korean, int english, int math) {
24            this.korean = korean;
25            this.english = english;
26            this.math = math;
27        }
28    }
29 }
```

그래프 그리기 버튼을 처리하는 리스너 클래스

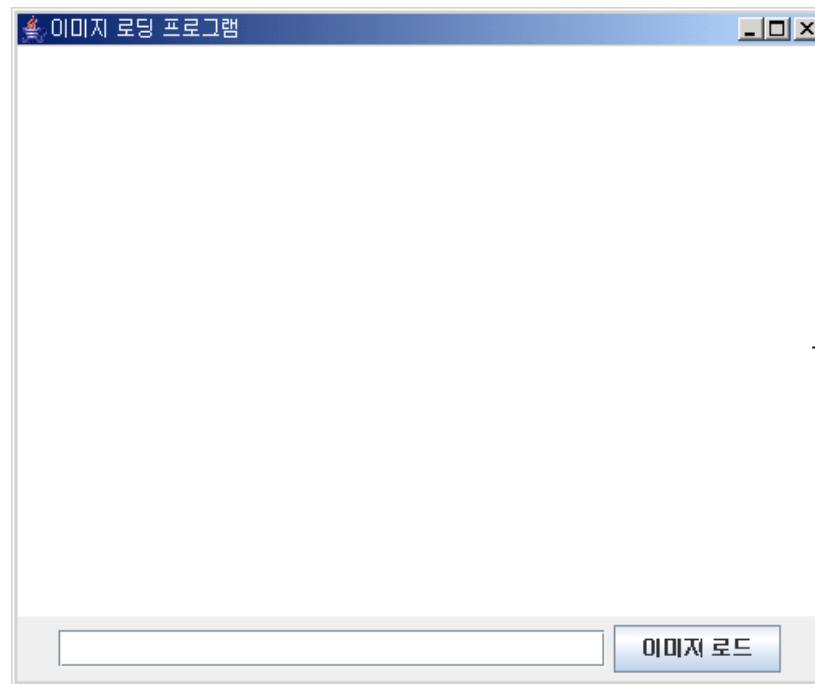
```
1 import javax.swing.*;
2 import java.awt.event.*;
3 class DrawActionListener implements ActionListener {
4     JTextField text1, text2, text3;
5     DrawingPanel drawingPanel;
6     DrawActionListener(JTextField text1, JTextField
7         text2,
8         JTextField text3, DrawingPanel drawingPanel) {
9         this.text1 = text1;
10        this.text2 = text2;
11        this.text3 = text3;
12        this.drawingPanel = drawingPanel;
13    }
14    public void actionPerformed(ActionEvent e) {
15        try {
16            int korean = Integer.parseInt(
17                text1.getText());
18            int english = Integer.parseInt(
19                text2.getText());
20            int math = Integer.parseInt(text3.getText());
21            drawingPanel.setScores(korean, english, math);
22            drawingPanel.repaint();
23        } catch (NumberFormatException nfe) {
24            JOptionPane.showMessageDialog(drawingPanel,
25                "잘못된 숫자 포맷입니다.",
26                "에러 메시지",
27                JOptionPane.ERROR_MESSAGE);
28        }
29    }
30 }
```

GUI 프로그래밍

03. 그래픽과 이미지 디스플레이

이미지 디스플레이

•• 지금부터 작성할 예제의 윈도우



이미지를 디스플레이할 패널


이미지 파일 이름을 입력하는 부분

03. 그래픽과 이미지 디스플레이

이미지 디스플레이

•• 이미지를 디스플레이하는 방법

- 1) JPanel의 서브클래스를 선언합니다.

```
class ImagePanel extends JPanel {  
    public void paint(Graphics g) {  
          
    }  
}
```

이미지를 디스플레이하는
명령문을 이 부분에 써야 합니다

03. 그래픽과 이미지 디스플레이

이미지 디스플레이

•• 이미지를 디스플레이하는 방법

- 2) Image 객체를 만드는데 필요한 Toolkit 객체를 얻습니다.

```
Toolkit toolkit = panel.getToolkit();
```

↑
Toolkit 객체를 리턴하는 메소드

03. 그래픽과 이미지 디스플레이

이미지 디스플레이

•• 이미지를 디스플레이하는 방법

- 3) Toolkit 객체를 이용해서 Image 객체를 만듭니다.

```
Image image = toolkit.getImage("Logo.jpg");
```

↑
Logo.jpg 파일을 읽어서
Image 객체로 만들어서 리턴합니다

03. 그래픽과 이미지 디스플레이

이미지 디스플레이

•• 이미지를 디스플레이하는 방법

- 4) Image 객체를 가지고 drawImage 메소드를 호출하면 이미지가 디스플레이됩니다.

```
g.drawImage(image, 0, 0, this);
```

↑
패널의 (0, 0) 위치에 image를
디스플레이합니다

03. 그래픽과 이미지 디스플레이

이미지 디스플레이

- 이미지를 그리는 로직이 포함된 paint 메소드

```
class ImagePanel extends JPanel {  
    public void paint(Graphics g) {  
        Toolkit toolkit = this.getToolkit();  
        Image image = toolkit.getImage("Logo.jpg");  
        g.drawImage(image, 0, 0, this);  
    }  
}
```

} Logo.jpg 파일을 읽어서
패널에 디스플레이합니다

프로그램의 성능에
문제를 일으키는 방법임



drawImage 메소드 호출문을 제외한
나머지를 paint 메소드 밖으로 빼는 것이 좋음

GUI 프로그래밍

03. 그래픽과 이미지 디스플레이

이미지 디스플레이

• [예제 19-12] 이미지 로딩 프로그램

main 메소드를 포함하는 클래스

```
1 import java.awt.*;
2 import javax.swing.*;
3 class ImageExample1 {
4     public static void main(String[] args) {
5         JFrame frame = new JFrame("이미지 로딩 프로그램");
6         frame.setLocation(500, 200);
7         frame.setPreferredSize(new Dimension(500, 400));
8         Container contentPane = frame.getContentPane();
9         ImagePanel imagePanel = new ImagePanel();
10        contentPane.add(imagePanel, BorderLayout.CENTER);
11        JPanel controlPanel = new JPanel();
12        JTextField text = new JTextField(30);
13        JButton button = new JButton("이미지 로드");
14        controlPanel.add(text);
15        controlPanel.add(button);
16        contentPane.add(controlPanel, BorderLayout.SOUTH);
17        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
18        button.addActionListener(
19            new LoadActionListener(text, imagePanel))
20        frame.pack();
21        frame.setVisible(true);
22    }
```

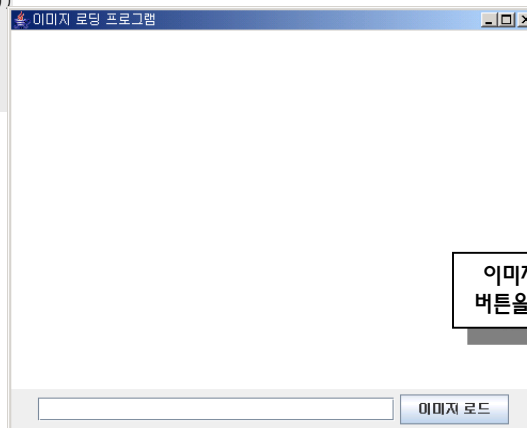
이미지를 디스플레이하는 패널 클래스

```
1 import java.awt.*;
2 import javax.swing.*;
3 class ImagePanel extends JPanel {
4     Image image;
5     Toolkit toolkit = getToolkit();
6     void setPath(String path) {
7         image = toolkit.getImage(path);
8     }
9     public void paint(Graphics g) {
10        g.clearRect(0, 0, getWidth(), getHeight());
11        if (image != null)
12            g.drawImage(image, 0, 0, this);
13    }
14 }
```

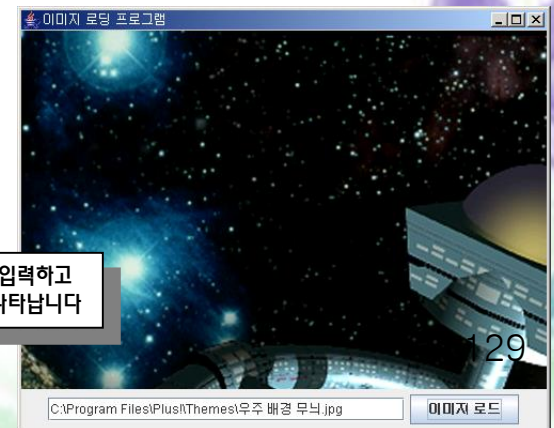
이미지 로드 버튼을 처리하는 리스너 클래스

```
1 import javax.swing.*;
2 import java.awt.event.*;
3 class LoadActionListener implements ActionListener {
4     JTextField text;
5     ImagePanel imagePanel;
6     LoadActionListener(JTextField text,
7         ImagePanel imagePanel) {
8         this.text = text;
9         this.imagePanel = imagePanel;
10    }
11    public void actionPerformed(ActionEvent e) {
12        imagePanel.setPath(text.getText());
13        imagePanel.repaint();
14    }
```

초기 상태



이미지 파일의 경로명을 입력하고
버튼을 누르면 이미지가 나타납니다



● 04. 오디오 파일 재생하기

● 오디오 파일의 재생

- 오디오 파일을 재생하는 방법
 - 1) 오디오 클립을 생성합니다.
 - 2) 오디오 클립을 플레이합니다.

04. 오디오 파일 재생하기

오디오 파일의 재생

- 오디오 클립을 생성하는 방법 (1)

```
URL url = new URL("file:\\c:\\audio\\sample.au");
```

URL 객체를 가지고 newAudioClip 메소드를 호출해서 AudioClip 객체를 생성합니다

```
AudioClip audioClip = Applet.newAudioClip(url);
```

04. 오디오 파일 재생하기

오디오 파일의 재생

.. 오디오 클립을 생성하는 방법 (2)

```
File file = new File("c:\\audio\\sample.au");  
URL url = file.toURL();
```

File 객체를 생성하고,
그 객체로부터 URL
객체를 얻습니다

URL 객체를 가지고 newAudioClip 메소드를
호출해서 AudioClip 객체를 생성합니다

```
AudioClip audioClip = Applet.newAudioClip(url);
```

04. 오디오 파일 재생하기

오디오 파일의 재생

•• 오디오 클립을 플레이하는 방법

```
audioClip.play();
```

↑
이 오디오 클립을 플레이합니다

GUI 프로그래밍

04. 오디오 파일 재생하기

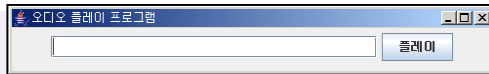
오디오 파일의 재생

• [예제 19-13] 오디오 파일을 플레이하는 프로그램

main 메소드를 포함하는 클래스

```
1 import java.awt.*;
2 import javax.swing.*;
3 class AudioExample1 {
4     public static void main(String[] args) {
5         JFrame frame = new JFrame("오디오 플레이 프로그램");
6         frame.setLocation(500, 200);
7         frame.setPreferredSize(new Dimension(500, 70));
8         Container contentPane = frame.getContentPane();
9         contentPane.setLayout(new FlowLayout());
10        JTextField text = new JTextField(30);
11        JButton button = new JButton("플레이");
12        contentPane.add(text);
13        contentPane.add(button);
14        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
15        button.addActionListener(new PlayActionListener(text));
16        frame.pack();
17        frame.setVisible(true);
18    }
19 }
```

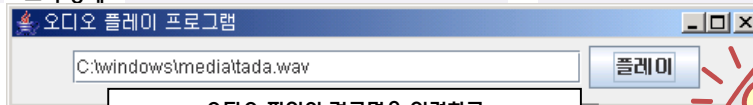
[화면 설계]



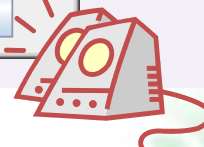
플레이 버튼을 처리하는 리스너 클래스

```
1 import javax.swing.*;
2 import java.awt.event.*;
3 import java.applet.*;
4 import java.net.*;
5 import java.io.File;
6 class PlayActionListener implements ActionListener { // 생성자
7     JTextField text;
8     PlayActionListener(JTextField text) {
9         this.text = text;
10    }
11    public void actionPerformed(ActionEvent e) {
12        try {
13            File file = new File(text.getText());
14            AudioClip audioClip = Applet.newAudioClip(file.toURL());
15            audioClip.play();
16        }
17        catch (MalformedURLException mue) {
18            JOptionPane.showMessageDialog(text,
19                "잘못된 파일명입니다.",
20                "에러 메시지", JOptionPane.ERROR_MESSAGE);
21        }
22    }
23 }
```

초기 상태



오디오 파일의 경로명을 입력하고
플레이 버튼을 누르면 스피커를 통해 소리가 납니다



Applet

- ▶ Applet은 우리가 앞에서 보았던 AWT에서 Container의 역할을 하는 것으로 Web Browser에 다운되어 실행될 수 있는 Program을 의미한다.
- ▶ Applet이 나오면서 동적으로 물체가 움직이고, Game을 하는 등 특별히 Software를 Down받아 설치 하지 않고 그냥 Web Browser상에서 Program이 실행될 수 있도록 해준 것.
- ▶ Applet은 Browser에 내장되어 있는 Java VM에 의해 동작이 된다.
- ▶ User는 별도의 Java VM을 설치하지 않아도 Applet이 실행될 수 있는데, Browser의 version이나 개발사에 따라 지원하는 version이 다를 수 있기 때문에 작성된 Applet Program이 원래 의도대로 동작하지 않을 수도 있다.
- ▶ 그래서 이러한 문제를 해결하기 위해서 Sun에서 해결책으로 나온 것이 바로 Java Plug-In이다.

Applet

● Applet의 실행

- ▶ Applet은 Application과 달리 `main()` 함수가 존재하지 않기 때문에 독자적으로는 실행이 불가능하다.
- ▶ Web Browser나 JDK가 설치되어 있는 `bin` folder에서 제공하는 `appletviewer`와 같은 Program내에서만 실행이 가능하다.
- ▶ Web Browser는 기본적으로 읽을 수 있는 문서형태가 HTML형태이므로, Applet code를 직접 Browser에 삽입하여 작성하는 것은 아니다.
- ▶ 즉, Applet code는 우리가 지금까지 작성했던 방식으로 `.java`형태로 작성해서 `compile` 시킨 후 HTML code에서 `compile`된 `CLASS` file을 불러 실행될 수 있도록 하는 방법을 사용한다.
- ▶ Browser에서 `CLASS` file을 불러오는 code는 `<applet>` tag를 사용.

Applet

● Applet관련 tag 및 method

➡ Applet관련 tag

- Web Browser에서 Applet code를 실행시키기 위해서 HTML에서 제공하는 tag가 바로 <applet>이다.
- 이 tag를 이용해서 HTML에서 Applet code를 지정해서 쉽게 Applet code를 사용할 수 있도록 해주는 기능을 한다.

❖ <applet>tag 형식

```
<applet code=" Applet class이름"  
        width=" Web Browser상의 Applet의 폭"  
        height=" Web Browser상의 Applet의 높이"  
        codebase=" Applet code가 있는 위치"  
>  
  <param name=" parameter name"  
        value=" parameter가 가지는 값"  
  >  
</applet>
```

Applet

● <applet>tag의 속성

- ➡ **code** : HTML에서 Applet으로 작성하여 compile된 CLASS file을 불러와서 실행할 수 있도록 지정하는 속성으로 CLASS file의 확장자는 생략가능하며 package자체를 지정하여도 된다.
- ➡ **width, height** : Web Browser에서 Applet이 보여줄 수 있는 영역을 지정하는 속성이다. Web Browser의 크기변경에 따라 가변적으로 변경될 수 있다.
- ➡ **codebase** : Applet file이 있는 위치를 지정하는 속성으로 일반적으로 HTMLfile과 같은 위치에 있다면 생략이 가능하고 다른 위치에 있으면 반드시 경로를 설정해줘야 된다.
- ➡ **<param>tag** : 이 tag는 Applet이 실행될 때 HTML에서 Applet으로 초기값을 넘겨줄 수 있도록 해주는 기능으로 name은 Applet에서 값을 얻어갈 수 있도록 해주는 parameter 이름이고 value는 name에 의해서 실제적으로 넘어오는 parameter값을 지정하는 속성이다.

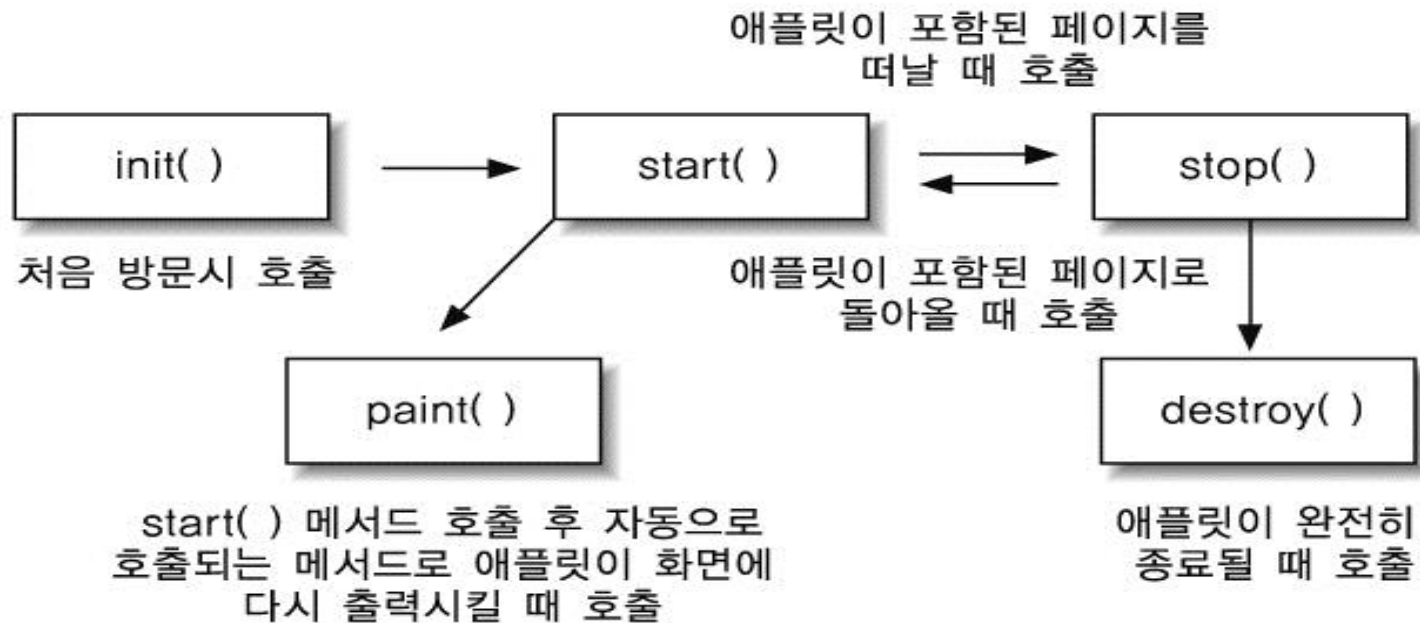
Applet

※ Applet 클래스의 주요 메서드

반환형	메서드	설명
void	init()	웹브라우저에서 처음으로 애플릿이 로드될 때 실행되는 메서드로 애플릿을 초기화한다.
	start()	init() 메서드 다음에 호출되거나 다른 페이지를 보고 있다가 다시 애플릿이 포함된 페이지로 되돌아 올 때 호출되는 메서드로 애플릿을 시작시킨다.
	stop()	애플릿이 중지되거나 애플릿이 포함되지 않은 페이지로 이동될 때 호출되는 메서드로 애플릿을 중지시킨다.
	destroy()	애플릿이 종료될 때 호출되는 메서드로, 주로 웹브라우저가 완전히 종료될 때 호출되는 메서드다.
	paint(Graphics g)	애플릿을 화면에 다시 출력시켜야 하는 경우에 호출되는 메서드다.
URL	getCodeBase()	애플릿 코드가 있는 URL을 얻어온다.
	getDocumentBase()	HTML 문서가 있는 URL을 얻어온다.
String	getParameter(String name)	HTML 문서에서 <param> 태그에 있는 name에 해당하는 값을 얻어온다.
Image	getImage(URL url)	url에 지정된 위치의 이미지를 얻어온다.
	getImage(URL url, String imgName)	url에 지정된 위치에 imgName을 가지는 이미지를 얻어온다.

Applet

● Applet Life Cycle



[그림 11-3] 애플릿이 실행되는 동작의 도식화

Applet

● Applet Security

- ➡ Java Applet은 Server로부터 Download되어 Web Browser를 통하여 실행되는 Program으로 Download된 Applet Program이 System에 접근하여 System의 정보를 읽거나 file을 가지고 갈 수도 있다면 문제가 아주 심각할 것이다.
- ➡ Applet은 이러한 행위를 하지 못하게 보안이 되어있고 이런 보안 문제가 발생하면 예외를 던지고 실행을 멈추게 한다.

❖ Applet Security Policy

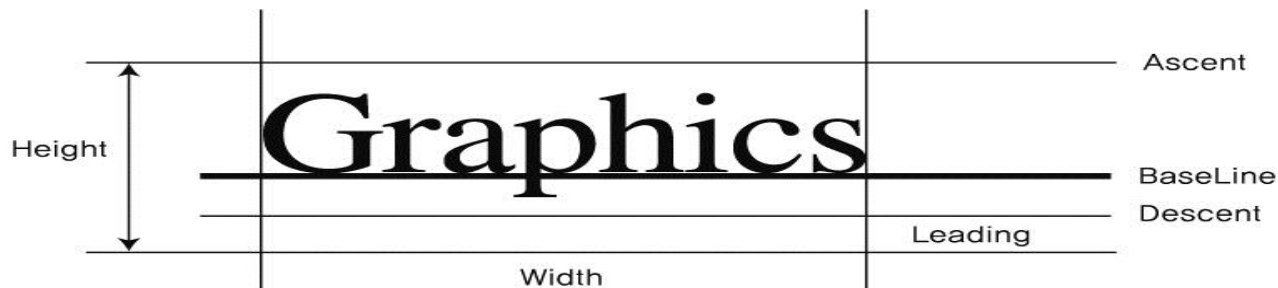
1. Applet이 Download된 System의 file을 읽거나 쓸 수 없다.
2. Applet이 Download된 System의 Program을 실행하거나 공유 library를 호출 할 수 없다.
3. Applet이 Download된 System의 정보를 알아낼 수 없다.
4. Binary code를 불러내 사용할 수 없다.
5. Applet을 제공한 Server 이외의 다른 System과 통신을 할 수 없다.

Applet

● FontMetrics class

- ➡ User가 정확한 위치에 문자를 출력시킬 수 있도록 FontMetrics class를 제공.

❖ FontMetrics의 문자열에 대한 정보



- Width : 문자열의 폭
- BaseLine : Font의 기준선(이 기준선에 의해 글자가 출력된다.)
- Descent : 기준선보다 아래에 있는 공간
- Ascent : 기준선보다 위에 있는 공간
- Leading : 현문자의 decent과 다음 문자라인의 Ascent 사이의 공간으로 아래글자와의 경계에 사용되는 빈 공간
- Height : $Ascent + Descent + Leading$

image Process

- ▶ Java에서는 Disk에 저장되어 있는 Image file을 불러와 화면에 출력시킬 수 있는 기능을 제공.
- ▶ Image를 불러 올 수 있는 method로는 getImage() method를 이용하고, 이 method는 Toolkit class와 Applet class의 method이다.
- ▶ Toolkit class의 getImage() method는 일반적인 Application에서 사용하는 method이고, Applet class의 getImage() method는 Applet Program에서 사용하는 method다.
- ▶ getImage() method는 Image를 불러오면서 Image Object를 생성해주는 method다.
- ▶ Image Object를 관리하기 위해서 Java에서는 java.awt package의 Image class를 제공.

image Process

● Double Bufferring을 이용한 Image Process

- ➡ Image를 출력시키는 경우 화면을 지웠다가 다시 그리게 되면 깜박거림을 피할 수가 없다.
- ➡ 그래서 이러한 문제점을 해결하기 위해서 많이 사용하는 방법이 Double Bufferring 기법이다.
- ➡ Double Bufferring 기법이란 화면에 Image를 직접 그리지 않고, memory상에 미리 그려 놓았다가, 필요할 때 한번만 화면에 출력시키는 방법이다.
- ➡ 매번 화면을 지웠다 그렸다하는 것이 아니라 보이지 않는 메모리상에서 지웠다 그렸다하고 화면에는 최종적으로 보여줄 때 한번만 출력을 시켜 깜박거림을 최소화시켜주는 기법인것이다.



정

이

화

시

다

^^!