



μζ^{3/4}Æ'ëÇÐ±³ ÄÄÇ»Áí°øÇÐ°ú
Àì¹î±Ô

1997, 8. 16.

QUICK HELP

NOTATION MENU

▲ HOME

?

HELP

PLASTIC 1.0

OBJECT MODELING TOOL

INTRODUCTION

DESIGN PATTERNS

OBJECT MODEL NOTATION

JAVA EDITOR

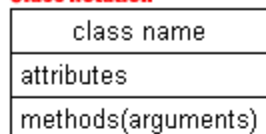
PLASTIC USAGE

EXAMPLES

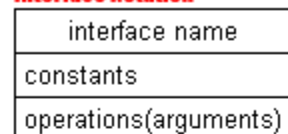
MODELING WITH PLASTIC

QUICK HELP - NOTATION

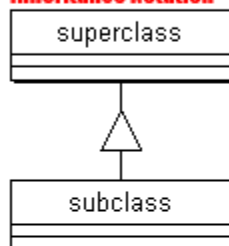
Class Notation



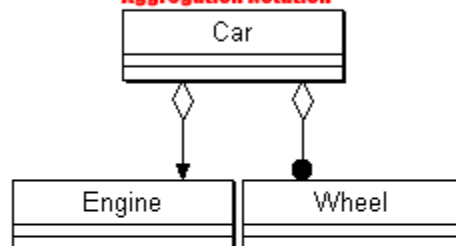
Interface Notation



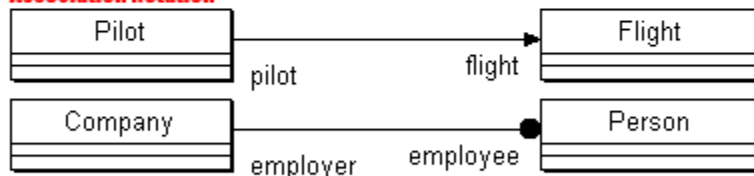
Inheritance Notation



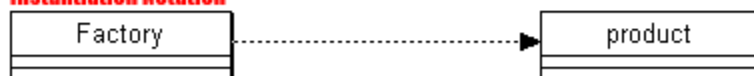
Aggregation Notation



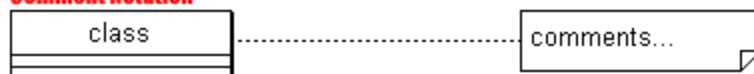
Association Notation

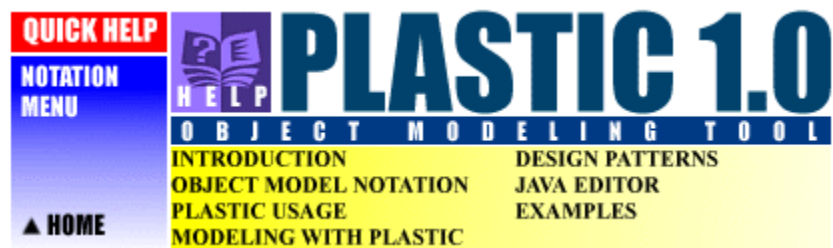


Instantiation Notation

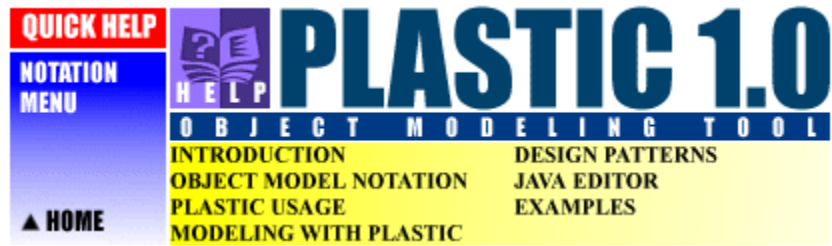


Comment Notation



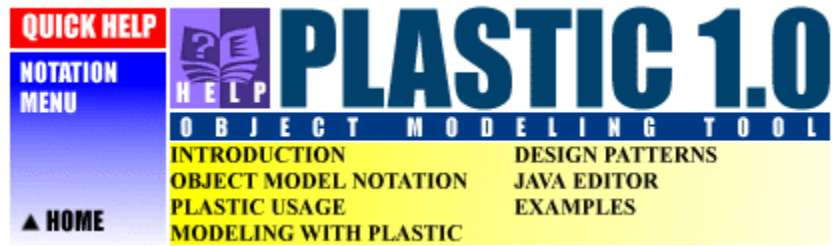


Menu Help



Plastic 1.0 for Java Help Introduction

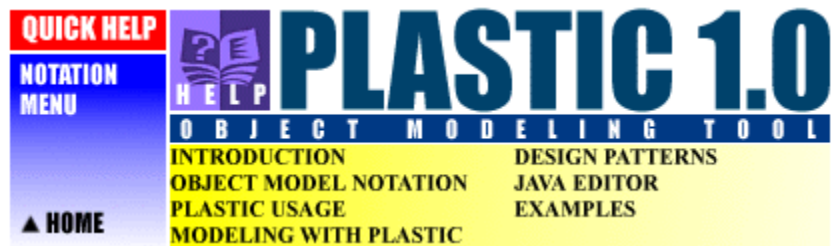
- [PlasticÀìîõ?](#)
- [PlasticÀÇ ±â´Éµé](#)
 - [PlasticÀÇ ±â´É](#)
 - [Object Model Notation](#)
 - [Design Patterns](#)
 - [Supporting java](#)
 - [Language Translators](#)
 - [±x¹UÀÇ ±â´É](#)



Plastic 1.0 for Java Help

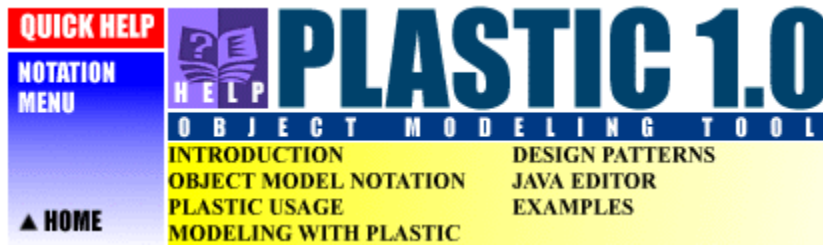
Object Model Notation

- [Class Notation](#)
- [Interface Notation](#)
- [Inheritance Notation](#)
- [Association Notation](#)
- [Aggregation Notation](#)
- [Instantiation Notation](#)
- [Comments Notation](#)
- [Modeling Example](#)



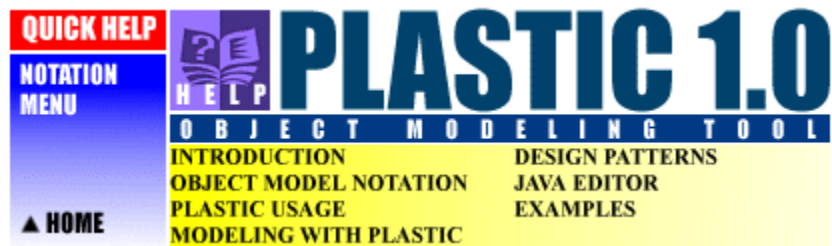
Plastic 1.0 for Java Help Plastic Usage

- [PlasticÀÇ ÀÛÃ¼È,é ¼³.í](#)
- [File Menu](#)
- [Edit Menu](#)
- [Symbols Menu](#)
- [Patterns Menu](#)
- **View Menu**
 - [Browser...](#)
 - [Package...](#)
 - [Design Information...](#)
- **Tool Menu**
 - **Options...**
 - [Options...](#)
 - [Environment Page](#)
 - [Diagram Page](#)
 - [Editor Page](#)
 - [Java Editor...](#)
 - [Java Code Generator...](#)
 - [Translators...](#)
- [Help Menu](#)



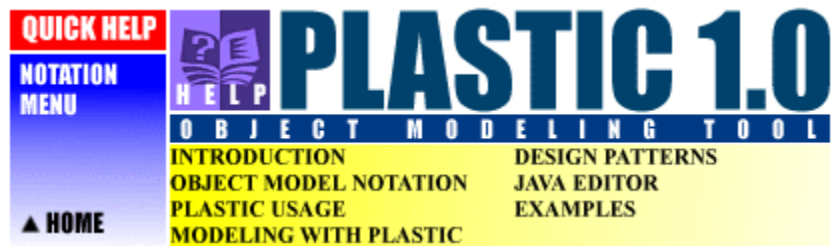
Plastic 1.0 for Java Help Modeling with Plastic

- **Class, Interface**
 - [Class Inspector](#)
 - [From Ancestors Dialog](#)
 - [Interface Inspector](#)
- **Field, Method**
 - [Attribute Inspector](#)
 - [Access Modifier](#)
 - [Constant Inspector](#)
 - [Method Inspector](#)
 - [Operation Inspector](#)
 - [Constructor Inspector](#)
 - [Argument Inspector](#)
 - [Type Dialog](#)
- **Inheritance**
 - [Class Extends](#)
 - [Interface Extends](#)
 - [Implements](#)
- **Association, Aggregation**
 - [Association Inspector](#)
- **Comment**
 - [Comment Inspector](#)
 - [Text Inspector](#)
- **Package**
 - [Package Window](#)
 - [Import Window](#)
- **Design Information**
 - [Design Information Window](#)



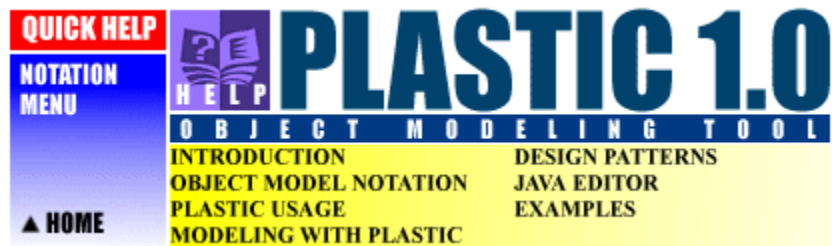
Plastic 1.0 for Java Help Design Patterns

- Design Patterns
 - [What Is a Design Pattern](#)
 - [Plastic and the Design Pattern](#)
- [Design Pattern](#)
 - [Factory Method \(Class Creational\)](#)
 - [Composite \(Object Structural\)](#)



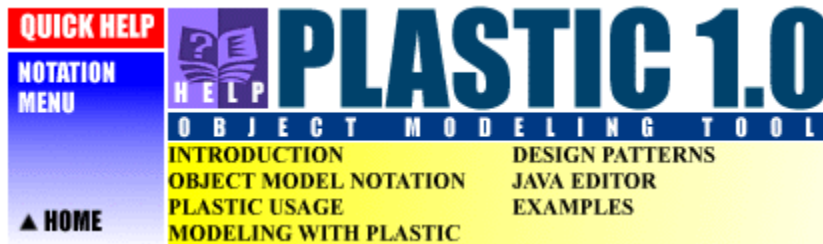
Plastic 1.0 for Java Help **Java Editor**

- Java EditorÀÇ¼Ò³
- Java EditorÀÇ¼Ò³



Plastic 1.0 for Java Help Examples

- **Java Applet Example**
 - [Applet Template](#)
 - [Applet Demo](#)
 - [Applet Demo Source](#)
 - [Applet Demo Result](#)
- **Java Application Example**
 - [Application Demo](#)
 - [Application Demo Source](#)
 - [Application Demo Result](#)

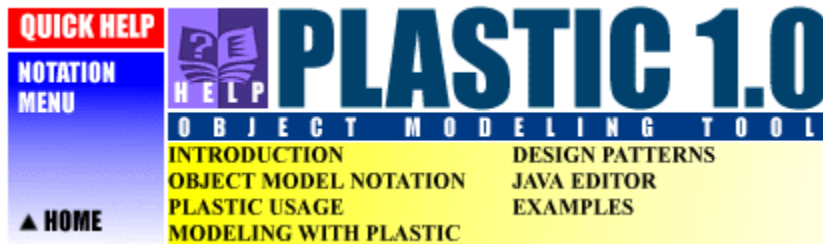


Introduction Plastic Àèŋð?

PlasticÀ° °Ã¼ÄöÇâ ,ðμ¨,μ Åð·î½á °Ã¼ÄöÇâ ¼ÖÇÁÆ®çþ¾î °³¹ΒÀ» ÀŞÇØ¼ ÇÊçäçÑ ÀýÂ÷Àî ,ðμ¨,μÀÇ ÀÛ¾÷À» ÀŞÇÑ μμ±, ¹× È¯æÀ» Áí°øÇĬç, çĭ¼°μÈ ,ðμ¨À° Java¾ð¾îÀÇ prototypeÀ, Ĩ ÀÚμç»ý¼°ÇØ ÁÖ´Â ±â´ÉÀ» °ĭÁð°í ÀÖ´Ù.

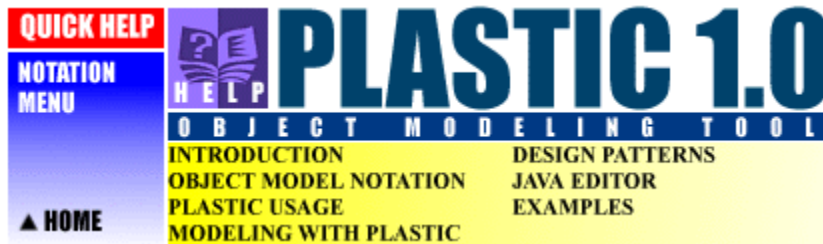
PlasticÀ° μĭ°ĭÁð Åð,éçĭ¼° °ÁĭÀ» °ĭÁð°í ÀÖ´Âμ¥, Ã¹Â°·ĭ °Ã¼ÄöÇâ ,ðμ¨,μÀÇ Åð,éçĭ¼´Â »ð·Ĭçĭ ±â´ÉÀĭ Design PatternsÀÇ ÁðçØÀ» μέ ¼ð ÀÖ´Ù. Àĭ°ĭÁ° ¼ÖÇÁÆ®çþ¾î ,ðμ¨,μçĭ¼ ÀÚÁÖ ¹Β°ΒμÇ´Â ¹®ÁĭμέÀ» ±â¼úÇĭ°í, ŋÇÇÑ ±×·ÇÑ ¹®ÁĭÀÇ ÇØ°á ¹æ¹ýçĭ ´èçÑ ÇÙ½ÉÀ» ±â¼úÇĭç© ³ðÀ° °ĭÀ, ĭ½á, ¹®Áĭ ÇØ°áçĭ ´èçÑ μçÀĭÇÑ ³è·ÂÀÇ ¹Ý¹¾ðÀĭ ¹®Áĭ ÇØ°áÀÇ Èç°ú,ĭ ¾ðÀ» ¼ð ÀÖ´Â °ĭÁĭ´Ù. Áĭ, ,ðμ¨,μ ÀÛ¾÷çĭ ÀÖ¾ĭ¼ ,Áçĭ Å« μμçðÀĭ μÇ¾ĭ¼, ¼ÖÇÁÆ®çþ¾î Àç»ççè¼° ¹× °ü,¥ ¾ÖÇÃ,®ÄÉÀ¼Ç °³¹ΒÀĭ çèÀĭÇĭ´Ù.

μÑÂ°·ĭ Â Java¾ð¾îçĭÀÇ ÅèÇÖμÈ È¯æÀĭ´Ù. ¾ð¾îçĭ μŋ,³Âûĭ ¼ÖÇÁÆ®çþ¾î ,ðμ¨,μÀÇ ÀýÂ÷°ĭ ³ĭ³ª,é Æ¯Áð ¾ð¾î,ĭ ÀĭèçÑ ±,ÇðÀ» ÇØ¾ΒÇÑ´Ù. ÇĭÁð,, °á±¹,ðμ¨,μ È¯æÀ» ŋ°³¹¼ ÀÛ¾÷À» ÇØ¾Β Çĭ¹Ç·ĭ ,Áçĭ °ÒÆĭÇÑ ÀÛ¾÷À» Çĭ°ÔμÈ´Ù. Àĭ°ĭμέÀ° ÅèÇÖ½ÃÑ ÁØ´Ù,é ,ðμ¨,μ, ±,Çð, ¼ðÁð ¹× °,çĭ ÀÛ¾÷À» μçÀĭÇÑ È¯æçĭ¼ ÇÒ ¼ð ÀÖ´Ô μÇ¹Ç·ĭ Æĭ,®Çĭ°ĭμμ °ü,£° ÀÛ¾÷À» ÇÒ ¼ð ÀÖ´Ù. Plasticçĭ¼´Â Java Editor ¹× Java source code generation ±â´ÉÀ» °®ÃΒ°í ÀÖ¾ĭ¼ ,ðμ¨,μ ÀÛ¾÷ ÀĭÈÄçĭ ±,ÇðÀÛ¾÷çĭ¼μμ ,¹À° °ĭðÀ» ÀÚμçÈÇĭç°í, ŋÇÇÑ ÆĭÁý È¯æÀ» ÅèÇÖÇĭç© Àĭ·ÇÑ ÀÖÁĭÀ» ¾ðÀ» ¼ð ÀÖ´Ô μÇ¾ĭÀÖ´Ù.



Introduction PlasticÀÇ ±â´É

PlasticÀ° OMT(Object Modeling Technique)ÀÇ OMN(Object Model Notation)À» Java¼ð¼îçj ,Â°Ô È®ÀâÇîç´°í, ¶ÇÇÑ ºÀ ÇÊçäÇÑ °î°Ð,,À» Æ÷¼·Çîç© Design PatternsÀ» Ç¥ÇöÇÒ ¼ð ÀÖ´Â ´Ü¼ðÇÑ ÁýÇÖÀ,·î ±¼°Çîç´´Ü. (ÀîÇî Modified OMNÀî¶ó Ä°ÇÔ) µû¶ó¼ Ç¥±â¹ýÀÇ ±³À° ¹× ÀìÇØ°j ½±°Ô µÇ¼ú´Ü. ±×,®°í °´Ä¼ÄöÇâ ¼ðÇÁÆ®çþ¼î ,ðµ´,µçj ÀÖ¼î¼ ÀÚÁÖ ¹³°ßµÇ´Â ÆÐÄîµé Äí, Design PatternsÀ» ÀÜ¼°Çî°í, »ðÀÖÇÒ ¼ð ÀÖ´Â ±â´ÉÀ» °jÁüÀ,·î »ý»ê¼° Çâ»ó, Àç»ççè¼°, °ü,¥ ¼ðÇÃ,®ÄÉÀ¼ÇÀÇ °³¹³çj Â« µµçðÀ» ÁÜ ¼ð°j ÀÖ´Ü. ¶ÇÇÑ ÄÖ±Ü °jÀâ °û±±¹°í ÀÖ´Â °´Ä¼ÄöÇâ ¼ð¼îÀî Javaçj ´èÇÑ prototypeÀ» ÀÜµçÀ,·î »ý¼°ÇØÁÖ,é¼µµ ÄÄÆÄÄ¹× ÆíÁý±â´ÉÀ» Áí°ðÇî´Â ÂèÇÖµÈ È´°æÀ» Áí°ðÇîç© ´ðçí Æí,®Çî°Ô ¼ðÇÁÆ®çþ¼î °³¹³À» ÇÒ ¼ð°j ÀÖ´Ü.



Introduction Design Patterns

patternÀì¶õ ¼ÒÇÁÆ®¿þ¾îÀÇ ,ðµ¨,µ¿¿¼ ÀÚÁÖ ¹ß³µÇ´Â ¹®Á!µéÀ» ±â¼úÇĩ°í, ¶ÇÇÑ ±×·ÇÑ ¹®Á!ÀÇ solution¿¿ ´ëÇÑ ÇÙ½ÉÀ» ±â¼úÇØ ³ðÀ° °íÀ, ·î½á, ¹®Á! ÇØ°á¿¿ ´ëÇÑ µ¿ÀĩÇÑ ³ë·ÀÀÇ ¹Ý¹¾¼À ¹®Á! ÇØ°áÀÇ È¿°ú, ¾ðÀ» ¼ð ÀÖ´Â ,Á¿ì °·ÂÇÑ ÀâÁ¿À» °¿Áø °íÀì´Ù.

Plastic¿¿¼´Â Àì·ÇÑ patternµéÀ» »¿¿ëÀÚ°¿ Á÷Á¢ ±â¼úÇĩ¿À ÀúÀâÇÒ ¼ð ÀÖÀ,¿, ¶ÇÇÑ ½±°Ö ÀÚ½ÀÀÇ ,ðµ¨·î »ðÀÔÇÒ ¼ð°¿ ÀÖ´Ù. ±×,®°í Gamma, Helm, Johnson, Vlissides°¿ Á!¾ÈÇÑ ¿©·°¿ÁöÀÇ patternsÀ» Æ÷ÇÔÇĩ°í ÀÖ¾î¼¼ ¼¼³°èÀÙ¾÷¿¿ ,Á¿ì µµ¿ðÀ» ÁÙ ¼ð°¿ ÀÖ´Ù. »ðÀÔµÈ patternµµ Java source,¿ »ý¼°Çĩ·Âµ¥ ¿µÇâÀ» ÁÖ±â ¶§¹®¿¿, patternÀ» °ß°íÇĩ°Ö ,µé¾î µí,é ÀÇ»¿¿ëÀì¶õ ãø,é¿¿¼µµ ÁÁÀ° °á°ú, ¾ðÀ» ¼ð ÀÖ´Ù.



Introduction

Supporting Java

JavaÀÇ ,ðµç ±â´Éµé, ç¹,µ é final, static, abstract class, interface, synchronized µî, À» ,ðµî Ç¥ÇöÇï°í
ÆíÁýÇÒ ¼ö ÀÖÀ,,ç method, constructorç; ´ëÇÑ Á÷Á¢ÀûÂî ±,Çöµµ »ðÀÔÇÒ ¼ö°í ÀÖ´Ù. ¶ÇÇÑ
package,¼ ÇÁúÁûÂî ,ðµµ,µÀÇ ´ÙÀ§·î °íÁ¢Çï±â ¶§¹®ç; , °ü,®ÀÇ Æø,éç¼µµ ´ð Æí,®Çï´Ù. (Àì·ÇÑ
´ÙÀ§°í ÁÖ¾îÁöÁö ¾É´Á´Ù,é »ççèÀÚ°í Â÷ÈÄç; ÇÁ·î±×·¥ ±,¼°ç; ¼ÖÀ» °Á¾ß Çï´Â ÀìÀì »ý±â°Ô µÈ´Ù.)
Àì·ÇÑ ,ðµµ,µ ´ÙÀ§ç; Â,ÀìÆ², ÀÚ¼°ÁÚ ¹× ÁÖ¼®À» °ÜÀì ¼ö°í ÀÖ°í, packageç; ÇÊçäÇÑ Áº,µéµµ
ÆíÁýÇÒ ¼ö°í ÀÖ´Ù. µû¶ó¼ çìÀüÇÑ Java programmingÀ» Çï±âÀ§ÇÑ µµ±,°í ,ðµî Áì°øµÇ,ç Java Editor
; Æ÷ÇÔµÇ¾î ÀÖ°í compile and error reportingÀ» ÇÑ¹øç; ÇÒ ¼ö ÀÖ´Â Æí,®ÇÔÀ» °®Ãß°í ÀÖ´Ù.





Introduction

±× ¹ÛÀÇ ±â´É

Java Editor Â ModelingÀÇ °á°ú·Î °ÍÁÍ »ý¼°µÈ °á°ú,| °ð¹Û·Î ¹þ¾Æ¹¼ ÆÍÁÝÇÒ ¼ö ÀÖµµ·Í ÇØÁÖ,ç, ´ÙÁß
¹®¼ ÆÍÁÝ°ú JDK¿Í ¿¬èÇĬ© compile ¹× error report,| ÇÑ ¹ø¿ ÇÒ ¼ö ÀÖ´Â ±â´ÉÀ» °®Ãß°í ÀÖ´Ù.
Browser´Â ÆÍÁÝÁßÂĬ ,ðµ·ÀÇ classes, interfaces, methods, attributes µĬÀ» IconÀ,·Î ÀĬ,ñ¿ă¿¬ÇĬ°Ô
°,¿©ÁÖ,ç, ÇØ´ç ¾ÆÀĬÛ¿Ĭ ěÇÑ Java¾ð¾ĬÀÇ Ç×ÇöÀ» ÁĬ½Ã °,¿©ÁÖ´Â ±â´ÉÀ» °ĬÁö°í ÀÖ´Ù.



class_i 'éÇÑ Ç¥±â±ýÀ° Á±»ç°¢ÇüÀÇ ,ð¼çç¿ çÀ,¥ÂÊ ÇĩÜÀ,·ĩ±×,²ÀÚ,|µâ,®çĩ °ÍÀĩÜ. ±×,²ÀÚ,|³ÖÀ° ÀìÀ¬À¹Ü·ĩ interface¿ÀÇ ±,°Ð ¶§¹®Àîµ¥, class°; interface°,Ü´ð concreteÇĩÜ´À °ÍÀ°³ªÀ,³¹² °ÍÀĩÜ. Á±»ç°¢ÇüÀ° 3°ĩ°ÐÀ,·ĩ³ªµ³⁄î ÀÖ´Àµ¥, °¿Àâ ÂÂÊ Äç¿´À classÀÇ Àì,§À» ±âÀÔÇĩ°í, °¿çîµ¥ °ĩ°Ðç¿´À data field,|°¿Àâ ¾Æ·§°ĩ°Ðç¿´À method,|ç°ÂÇÑÜ.

[illegible]



Object Model Notation Interface Notation

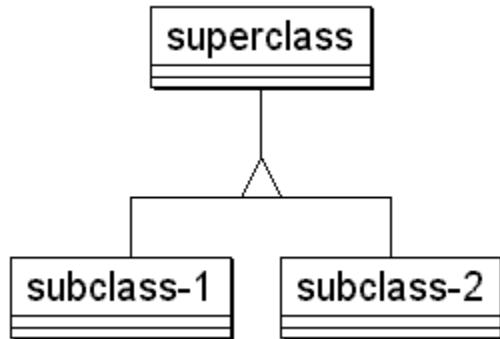
interface-name
constantType constant = initValue;
returnType operation(argType arg)

interface **name** {
 constantType **constant** = **initValue**;
 returnType **operation**(**argType** **arg**)
}

The notation for an interface is as follows: **interface** **name** { **constantType** **constant** = **initValue** ; **returnType** **operation**(**argType** **arg**) }. The **name** section, constants section, operation section and the closing brace **}** are mandatory. The **name** section is the name of the interface. The **constantType** is the type of the constant. The **constant** is the name of the constant. The **initValue** is the initial value of the constant. The **returnType** is the type of the return value. The **operation** is the name of the operation. The **argType** is the type of the argument. The **arg** is the name of the argument.



Object Model Notation Inheritance Notation

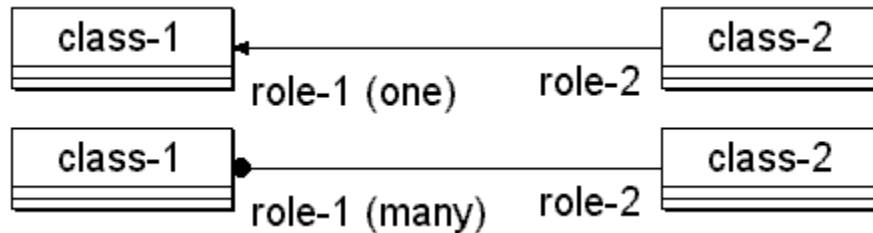


»ó¼ÓÀ° ÀÛÀ° »ï°ÇÇüÀÇ ,ð¾çÀ» superclass¿ subclasses»çÀ¿ µí°í »ÇÁÓÇÑ °ïÐÀ¿ superclass·Í, ±×·, Áö¾ÊÀ° °ïÐÀ¿ subclasses·Í °µµ·Í ¿¬áÇ¿, é µË´Ù. ¿©±â¼ ÁÖ, ñÇÒ Á¿À° Ç¿³aÀÇ superclass¿ ¿©¬³ÀÇ subclass°¿ »ï°ÇÇü ½É¹¼À» Ç¿³a µí°í ¼·Í °°À¿ ¿¬áÇÒ ¼ö ÀÖ´Ù´Â °ÍÀ¿Ù. À¿, °Ô Ç¿, é ³ÀâÇÔÀ» ÇÇÇÒ ¼ö ÀÖ°í, °, ±âµµ ½±´Ù.

À¿¬ÇÑ »ó¼Ó°ü°èÀÇ Ç×ÇöÀ° interfaces»çÀ¿ µµ ±×´è·Í Àû¿ëµË´Ù. ´Ù classes¿ interfaces°¿ ¼·Í ¿¬°áµÉ ¶§¿´Â class´Â interfaceÀÇ super°¿ µÉ ¼ö ¾øÁö,, interface´Â classÀÇ super°¿ µÉ ¼ö ÀÖ´Ù´Â Á¿¿ ÁÖÀÇÇ¿±â ¹Ù¶¶´Ù.

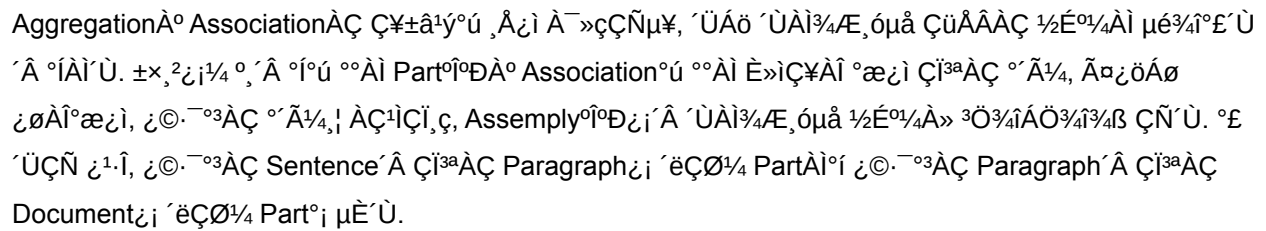


Object Model Notation Association Notation



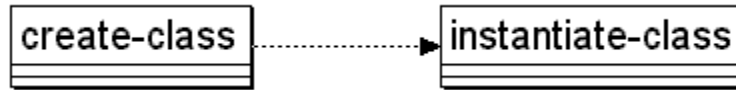
Association is a relationship between two or more classes. It is represented by a line connecting the two classes. The line is labeled with the role name and the multiplicity of each class. The multiplicity is the number of instances of the class that can be associated with the other class. The role name is the name of the role that the class plays in the association. The multiplicity is the number of instances of the class that can be associated with the other class.

For example, in the diagram above, class-1 is associated with class-2. The role name for class-1 is 'role-1' and the multiplicity is '(one)'. The role name for class-2 is 'role-2' and the multiplicity is '(many)'. This means that one instance of class-1 can be associated with many instances of class-2.





Object Model Notation Instantiation Notation

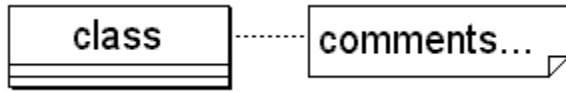


Á;¼±À,·Î μÈ È»ìÇ¥ ÇüÂÂÀÇ Ç¥±â¹ýÀ° instanceÀÇ »ý¼°À» ÀÇ¹ÌÇÑ´Ù. Áï, create-class´Â instantiate-classÀÇ instance,¡ »ý¼°ÇÒ ¼ð°¡ ÀÖ´Ù´Â ¶æÀÌ´Ù. DollFactory´Â DollÀÇ instance,¡ »ý¼°ÇÒ ¼ð ÀÖ´Ù.



Object Model Notation

Comments Notation

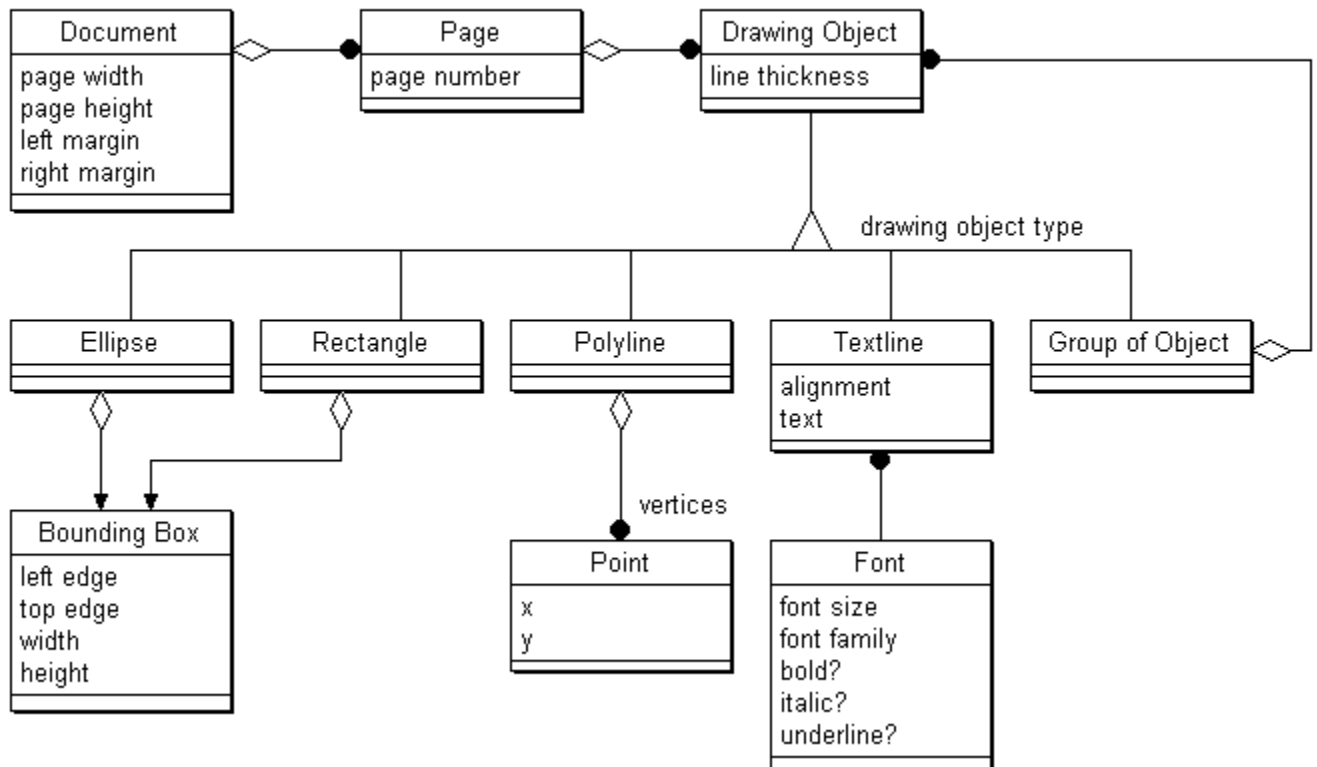


¾¶¶² Å¬·½º¿ ÁÖ¼®À» ´ªíÀÚ ÇÒ ¶§¿í´Â ¿À,¥ÂÊ Çĭ'Üŵ°¿ ¾à°£ Á¢Çð ¿Ã¶ó°£ »ç°¢ÇüÀÇ Á¾ÀìÃ³.³
Ç¥ÇöÀ» Çĭ°í ±× ³»°ĭ¿ ¿øÇĭ´Âµ¥·Î ÁÖ¼®À» ´ª,é µÈ´Ù. ±×,®°í ±×°ÍÀ° ÇØ´ç Å¬·½º¿ Í Á¿¼±À,·Î ¿¬°áÀ»
ÇÑ´Ù. Àì°ÍÀ° ¾¶¶² ÀÇ¹¿,¿ °¿Áö´Â °ÍÀì ¾¼£ĭ¶ŵ´ÜÁö ÀìÇø¿¿ µµ¿ðÀ» ÁÖ°Å³ª,¿¿ Áß¿äÇÑ »çÇ×µéÀ»
±â·ĭÇĭ´Â ¿ëµµ·Î »ç¿ëÇÑ´Ù.

Object Model Notation Modeling Example

OBJECT-ORIENTED MODELING AND DESIGN - Page 393
Rumbaugh, Blaha, Premerlani, Eddy, Lorensen

- Desktop Publishing System -

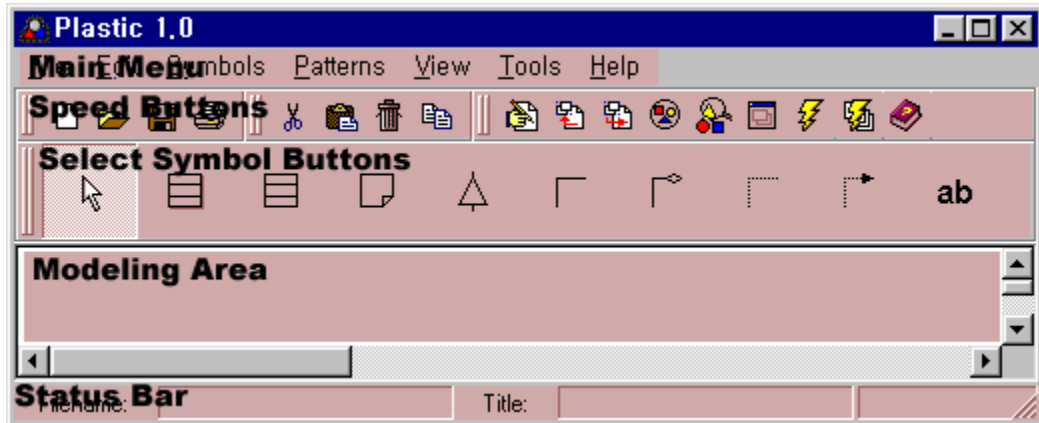


ÀŠ Desktop Publishing System¿i ¿ëÇÑ Object DiagramÀì'Ù. Àì ¿¹Áì' Â Object-Oriented Modeling and Design(Prentice Hall, ISBN:0-13-629841-9)Ã¥ÀÇ Page 393¿i ÀÖ' Â ¿¹Áì,ì Àì¿ëÇÑ °ÍÀì,ç PlasticÀ» Àì¿ëÇì© ÆíÁýÇÑ °ÍÀì'Ù.



Plastic Usage

Plastic



Plastic is a UML modeling tool. It provides a Main Menu, a Speed Buttons toolbar, a Select Symbol Buttons toolbar, a Modeling Area, and a Status Bar. The Main Menu includes File, Edit, View, Tools, and Help. The Speed Buttons toolbar includes icons for file operations (New, Open, Save, Save As, Print, Close, Close All, Exit) and editing (Undo, Redo, Cut, Copy, Paste, Delete, Find, Find Next, Find Previous, Find All, Replace, Replace All, Go to Line, Go to Page, Go to Symbol, Go to Class, Go to Interface, Go to Comment, Go to Inheritance, Go to Association, Go to Aggregation, Go to Comment Link, Instantiate, Text). The Select Symbol Buttons toolbar includes icons for selection (Select, Select All, Select None, Select by Name, Select by Class, Select by Interface, Select by Comment, Select by Inheritance, Select by Association, Select by Aggregation, Select by Comment Link, Select by Instantiate, Select by Text). The Modeling Area is a large workspace for creating and editing UML diagrams. The Status Bar displays the current filename and title.



Plastic Usage File Menu

ÆÄÄÏÇ; °ü·ÃµÈ ,p´°; ,ðç©ÄÖ´Âµ¥, **New**´Â ÇöÁ! ÆíÁýÁßÂÎ ÆÄÄÏÄ»´Ý°í, ,ðµç °ÍÄ» Äöçì°í ÆÊ±âÈ ÇÑ´Ù. **Open... [Ctrl+O]** À° Àìì Á,ÀçÇí´Â ¼³°èµÈ ,ðµ´À» °Ò·¬Ç´Â´Ù. °Ò·¬Ç´Â ¼ö ÄÖ´Â ÆÄÄÏÄÇ Çü½ÄÄ° .plaÄÎµ¥, PlasticÇ;¼ ÄÜ¾÷ÇÑ ,ðµ´ÀÖÄ»³a, ¾½´Ù. **Save [Ctrl+S]**´Â ÇöÁ! ÄÜ¾÷ÁßÂÎ ,ðµ´À» ÄúÄâÇÑ´Ù. Àì°ÍÄ° ÇöÁ! ,ðµ´Ç; ÆÄÄÏ,íÄì °Íç©µÇ¾î ÄÖÄ» ¶§Ç; , ÄúÄâÄ» Çí°í, ±×·, Ä;¾Ê´Â °æçìç;´Â **Save As...**,p´°Ä» ½ÇÇàÇÑ´Ù. **Save As...**´Â ÆÄÄÏ Äì,§Äì °Íç©µÇ¾î ÄÖÄö¾Ê°Ä³a, ÈÄ° »ð·Îçî ÆÄÄÏÄì,§Ä,·Î ÄúÄâÇí°í ½ÍÄ»¶§Ç; »ççèÇÑ´Ù. **Print...**´Â Äö±Ý ÄÜ¾÷ÁßÂÎ ,ðµ´À» ÇÁ,°Á·Î Äâ·Ä½ÄÄÑÁØ´Ù. **Print Setup...**Ä° ÇÁ,°Áç; °üÇÑ ¼³ÄÄ» Çí°í ½ÍÄ»¶§¾ÄÄÇÇ;é µÈ´Ù. **Exit**´Â PlasticÄ» Á¾·á½ÄÄ²´Ù.





Plastic Usage Symbols Menu

Select´Â ÇöÁ|, ðµ´¿;¼ ½É¹¼µéÀ» ¼±ÃÃÇÒ ¼ö ÀÖ´Â, ðµâ·Î ÀüÈ´ÇÑ´Ù. ½É¹¼ ÀS¿;¼ Â¬,´À» ÇĬ,é, ðµç ¼±ÃÃÀĬ Ãë¼ÖµÈ ÈÄ¿; Â¬,´µÈ ½É¹¼ÀĬ ¼±ÃÃµÈ´Ù. ÀĬ ¶S ¼±ÃÃµÈ ½É¹¼À° °ÓÀ° »öÀ,·Î ³aÃ,³´Ù. ±×,®´Ĭ Æ´Á¤ ¿µ¿³¼ÈÄÇ, ðµç ½É¹¼À» ¼±ÃÃÇĬ´ĬÁÚ ÇÑ´Ù,é µâ·Ĭ±× ¾Ø µâ·ÓÀ,·Î ¿µ¿ªÀ» ÁöÁ¤ÇĬ¿© ÁÖ,é µÈ´Ù. ±×,®´Ĭ [Shift]Â°,Ĭ´©,£,é¼ ¼±ÃÃÇĬ,é, ÀĬÀü¿;¼ ¼±ÃÃµÇ¾Ĭ ÀÖ´Â °ĬµéÀ» À´ÁöÇĬ,é¼ ¼±ÃÃÀ» ÃB°;ÇÒ ¼ö°; ÀÖ´Ù.

Class´Â Â¬·Ĭ½° ½É¹¼À» »ý¼°ÇĬ´Â, ðµâ·Î, **Interface**´Â ÀĬÁĬÆÐÀĬ½° ½É¹¼À» »ý¼°ÇĬ´Â, ðµâ·Î ÀüÈ´ÇĬ,ç, Modeling Area¿;¼ ¿ÇĬ´Â ÀSÄ¿; Â¬,´À» ÇĬ,é »ð·Î¿ ½É¹¼ÀĬ »ý±â´Ĭ, Select,ðµâ·Î´Ù½Ã ÀüÈ´ÀĬ µÈ´Ù. **Extends** . **Implements** ´Â »ó¼Ó°ü°è,Ĭ ³aÃ,¾¼ ¼ö ÀÖ´Â, ðµâ·Î ¹Ù²Ĭ Áµ¥, ¹Ýµâ½Ã ¼- °êÂ¬·Ĭ½°(ÀĬÁĬÆÐÀĬ½°)¿;¼ ¼öÆÛÂ¬·Ĭ½°(ÀĬÁĬÆÐÀĬ½°)·Î µâ·Ĭ±× ¾Ø µâ·ÓÀ» ÇĬ¿©¾B ÇÑ´Ù. **Association, Aggregation**À° Association Symbol, Aggregation SymbolÀ» »ý¼°ÇĬ´Â, ðµâ·Î ÀüÈ´ÇĬ,ç, **Instantiation**À° »ý¼°°°ü°è,Ĭ ³aÃ,¾¼ ¼ö ÀÖ´Â, ðµâ·Î ÀüÈ´ÇÑ´Ù. **Comment**´Â ÁÖ¼@½É¹¼À» »ý¼°ÇÒ ¼ö ÀÖÀ,ç, **Comment Link**´Â ÁÖ¼@½É¹¼À» Â¬·Ĭ½° ¶Ç´Â ÀĬÁĬÆÐÀĬ½°·Î ¿¬°áÇÒ ¼ö ÀÖ´Â, ðµâ·Î ¹Ù²Ĭ´Ù. **Text**´Â ÇÑÁÚÀ¥,® ¼³,ĬÀ» ´B ¼ö ÀÖµµ·Ĭ ÇØ ÁÖ´Ù.



Plastic Usage Patterns Menu

Save As Design Pattern...

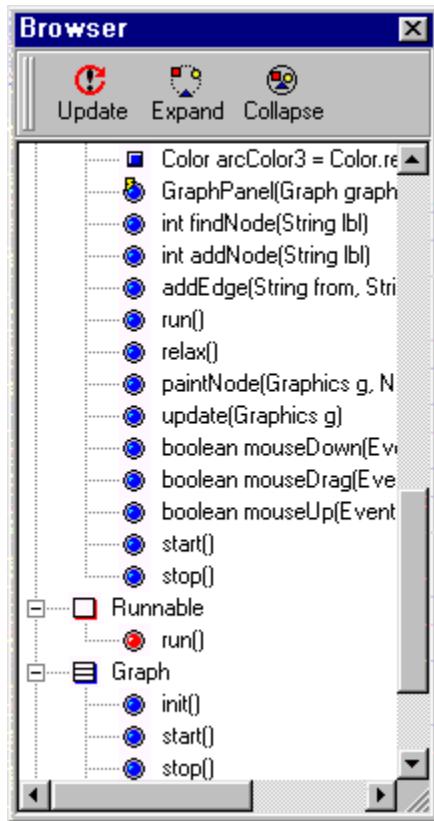
Äö±Ý ÆíÁýÇĩ°í ÀÖ´Â ,ðµ´Àì PatternÀì¶ó,é, Àì ,p´°,ı ¼±ÄÄÇĩç© ±×°íÄ» Äç»ççèÀì °ı´ÉÇÑ ÇüÄÄ·İ
ÀúÄâÇĩç© ÁÙ ¼ö ÀÖ´Ù. È®Äâ,íÄ° .patÄ,·İ ÀúÄâÀì µÇ´Âµ¥, ÄúÄâÀì µÉ ¶§çı´Â Packageçı °ü·ÄµÈ
Áº°,³ª, Design Informations´Â ,ðµİ ÄúÄâçı¼ ÄıçÜµÇ°ı ¼ø¼öÇĩ°Ö ,ðµ´,Äì ÄúÄâµÈ´Ù.

Insert Design Pattern...

Plasticçı¼ Äı°øÇĩ´Â Design Patterns³ª »ççèÄÚ°ı Àìıı Á±ÄÇÇĩç© ÀúÄâÇø µÐ Patterns,ı ÇöÁı ÆíÁýÁßÄİ
,ðµ´çı »ðÄÔÇÒ ¼ö°ı ÀÖ´Ù. »ðÄÔÄ» Çİ,é´Ù,¥ Áº°,çı´Â ÄüÇô çµÇâÄ» ıÄıÄö ¾ÊÄ,,ç, ´Üö Patternçı
Á,ÄçÇĩ´Â ½É°¼µé,,Äì »ð·Ó°Ö Äß°ıµÉ »ÓÄİ´Ù.



Plastic Usage Browser...



Browser ʼĀ ÇöÄ! ,ðμ´ζ; ÄÖ´Ā Ā¬·;½ºζÍ ĀĪĀÍÆÐÄĪ½ºζ; °üÇÑ ,®½ºÆ®ζÍ ±×°ÍμéÄĪ °;Āö´í ÄÖ´Ā ÇÊμĀ ¹×
,p¼ÖμĀ, »ý¼ºÄÜμéÄ» Æ®,®ÇüĀĀ·Ī°,ζ©ĀÖ´í ÄÖ´Ü. ¶ÇÇÑ °,±â ½±ºÖ ¾ÆĀĪÄÜ ÇüĀĀ·Ī³ª,³»´í
ÄÖĀ,ç, ¾Æ·;ζ;ĪĀ ¼±ĀĀμĒ Ā¬·;½º, ĀĪĀÍÆÐÄĪ½º, ÇÊμĀ, ,p¼ÖμĀ ¹× »ý¼ºÄÜζ; Java¾ð¾ĪĀÇ Ç¥ÇöÄ»
¾ÆĀÖ °ü,£º°°,ζ©ĀÖ´í ÄÖ´Ü. ±×,®´í ĀϕĀÇ Ā©±â ¹× °æ°è,Ī ÄÜĀ¬·Óº Ā¶ĀýÇÖ ¼ö ÄÖ¾Ī¼ »çζèÄÜ°;
»çζèÇĪ±âζ; ÆíÇĪμμ·Ī ÇĪ´í ÄÖ´Ü. ÄÖ»ó´Üζ; ÄÖ´Ā 3ºĀÇ ¹öÆºÄº Ā÷·Ê´Ī Refresh, Collapse,
ExpandÄĪμ¥, Refresh´Ā °è¶óζ;ĪÄú°; ζ,º ÄĪÈÄζ; ÆíÁýμÈ ½Éº¼μéÄ» ,ðμĪ ´Ü½Ā ¹ÝζμÇĪ´í ½ĪĀ»¶S
»çζèÇĪ,ç, Collapse´Ā ÇÊμĀ, ,p¼ÖμĀ, »ý¼ºÄÜζ; ´èÇÑ ´ĪĀ» ,ðμĪ ¼üºÜ¼ Ā¬·;½ºζÍ ĀĪĀÍÆÐÄĪ½º,Ä»
°,ζ©ĀÖ,ç, Expand´Ā ¹Ý´è·Ī ,ðμĪ ´Ü°,ζ©ĀØ´Ü.



Plastic Usage Package...

ÆÐÅ°Äöç|´ëÇÑ Á²°,!¼³Á²ÇÒ¼öÀÖμμ·Ï Package Window,!ç¬´Ù.



Plastic Usage Design Information...

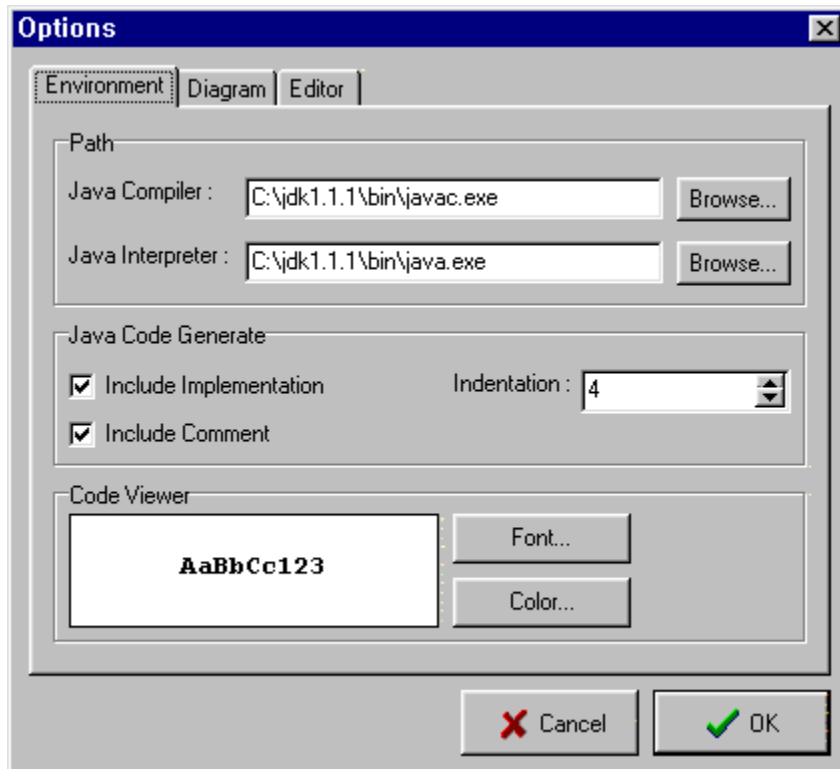
¼³°è¿; °üÇÑ Á²°,! ±âÀÔÇÒ ¼ö ÀÖμμ·İ Design Information Window,! ¿¾îÁØ´Ù.



Plastic Usage Options...

¿É¼ÇÀ° Plastic¿i ´ëÇÑ È¯æ(Environment Page)¿i ´ëÇÑ ¼³Á¤, ¿ðµ´´ÙÀì¾î±×·¥(Diagram Page)°üÇÑ
¼³Á¤ 1× JavaEditor(Editor Page)¿i °üÇÑ ¼³Á¤À, ·î ÃÑ 3ÆäÀìÁö·î ³a´µ¾î ÀÖ´Ù. ¿©±â¼ ¼³Á¤µÈ °aÀ»
´ÙÀ½¿i PlasticÀ» ½ÇÇàÇÒ ¶§¿i ¿ðµî°i ±×´ë·î À´ÁöµÈ´Ù.

Plastic Usage Environment Page



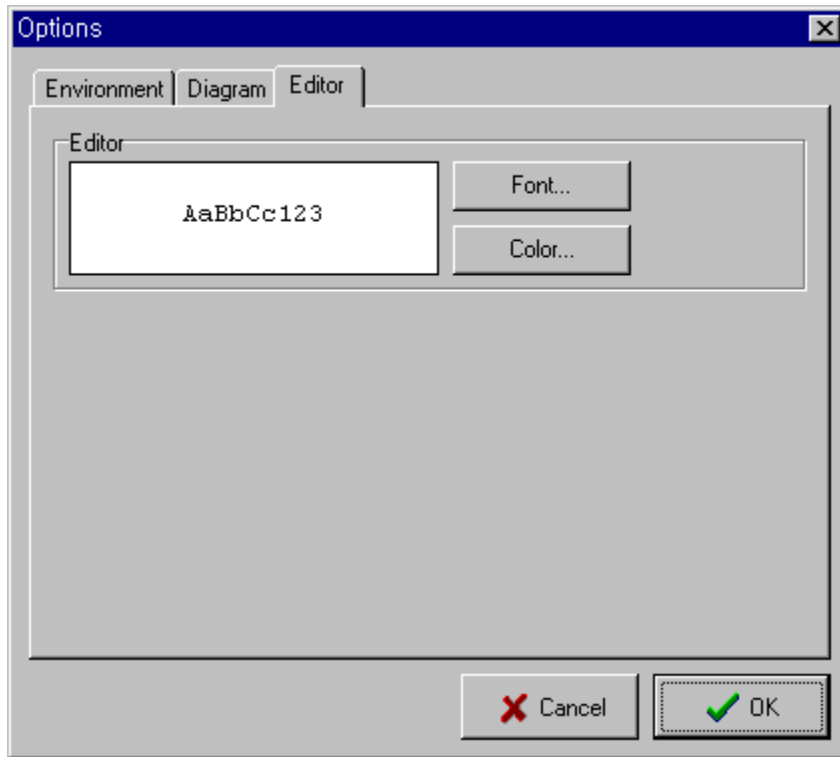
Environment Page, é Java Compiler, Java Code Generate, Browser. 3°iÁöÀÇ ±x·iÀ,·Î ±,°ĐÀİ'Ü. Java Compilerç;ı'Â Java Compiler°; ÀÖ'Â °æ·Î,íÁ» ¼³Á±ÇØÁÖ¾¼¾¾¾ Çı'Âµ¥, Àì'ÍÁ° Java Editorç;ı¼ Compile,ı'°À» ¼±ÄÄÇİç'À»¶İ, »ççëµÉ °ÍÀİ'Ü. ç©±âç;ı¼'Â JDK1.1.1 Àİ»ó ıöÁ'ÀÇ javac.exe,ı' »ççëÇÖ °ÍÀ» ±ÇÀâÇÑ'Ü. ±x,°°ı Java Code Generator±x·ıç;ı¼ Include Implementation Ä¼Ä© ıÜ½°'Â »ý¼°ÇÖ ÀÚı½¾¾¾ıÄÇ ÄÚµâç;ı ±,Çö°ı°ĐÀ» Æ÷ÇÖÇÖ °ÍÀİÁöç;ı 'ëÇÑ ç©°ı,ı, Include Comment'Â Comment ½É°ı¼ç;ı ÄÇÇÑ ÁÖ¼¼@ı@À» Æ÷ÇÖÇÖ °ÍÀİ°ıç;ı 'ëÇÑ ç©°ı,ı Ç¥½ÄÇÑ'Ü. ±x,°°ı IndentationÀ» »ý¼°ÇÖ ÀÚı½¾¾¾ı ÄÚµâÄÇ µéç©¾²±âÄÇ °øıéÄÇ °ıı¼ö,ı ¼³Á±ÇÑ'Ü. Browser±x·ıç;ı'Â Browserç;ı¼ ÀÚı½¾¾¾ıÄÇ Ç¥ÇöÀ» °,ç©ÁÖ'Â °ı°ĐÀÇ ÆùÆ® ıx »ö»óÀ» °áÁ±ÇÖ ¼ö ÀÖµµ·ı Çİç'Ü.



Diagram Page ღı'Â È,éღı Ç¥½ÃµÇ'Â °¢Á¾¼ ½É°¼ÀÇ ÆùÆ® ,ð¾Çღı 'èÇÑ °İÀ» ¼¾³Ã°ÇÒ ¼ø ÀÖµµ.İ Çİ
 'Â Font±×.ı°ú Modeling AreaÀÇ ღµĲªÀÇ Â©±â,ı ¼¾³Ã°ÇÒ ¼ø ÀÖµµ.İ µÇ¾İ ÀÖ'Â Size±×.İÀİ ÀÖ'Ü.
 Modeling AreaÀÇ Â©±â,ı °°æÇÒ ¶Şღı'Â ÄÖÀÇ,ı ĲäÇİ'Âµ¥, Â©±â,ı Â©°Ö ÇÒ ¶Şღı'Â °°İ®Ä'ı ¾øÄ,³ª,
 ÄÛ°Ö ÇÒ ¶Şღı'Â ღµĲªÀ» ıþ¾İ³ª Ä¢±ÜÇÒ ¼ø ¾øÄ ½É°¼Äİ »ý±æ ¼ø ÀÖÄ,¹Ç.İ ÄÖÀÇÇİ±â ıÜ¶đ'Ü.
 ±×,°ı°İ ÆùÆ®ÀÇ ,ð¾ÇÀ° ÇÁ,°ÆÄÇÒ ¶Şღıµµ ღµÇÄâ» ıİİı¹Ç.İ ĲıµĲı µİı°İ ÄÖÄ,é ÄÁÄ» °İÄİ'Ü.



Plastic Usage Editor Page



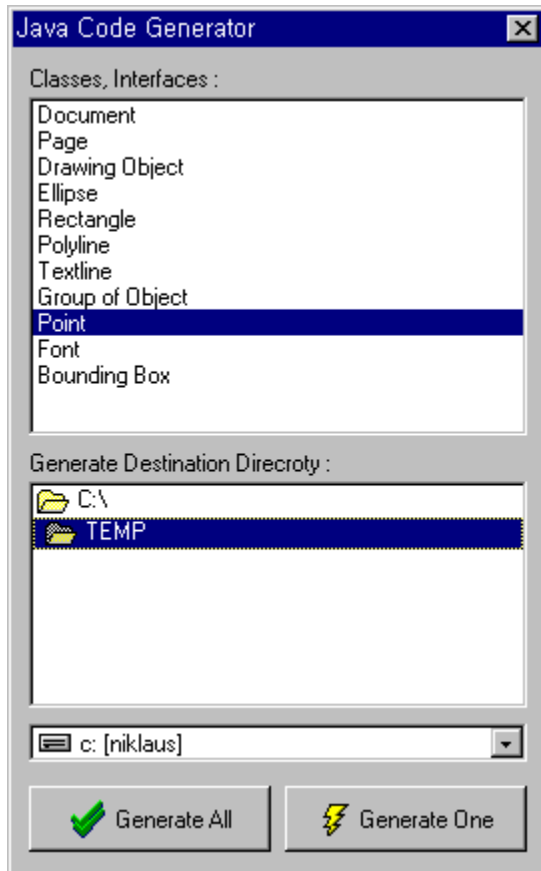
Editor Page¿í Â Java Editor¿i¼ ÆíÁýÇíÂ °í°ÐÀÇ ÆùÆ®ÀÇ ,ð¾ç ¹× »ö»óÀ» ÁµÀÇÇÒ ¼ö ÀÖ´Ù.



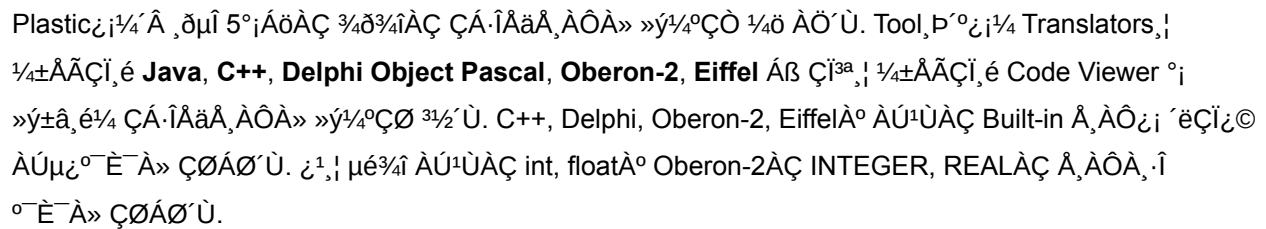
Plastic Usage Java Editor...

Java Editor... [F3] ^ Á Java¾ð¾¼îÀÇ ¼¼Ö½²¹× html, text ÆÄÄÏÄ» ÆíÁýÇÒ ¼¼ö ÄÖµµ·Ï Java Editor,¡ ¿¾îÁØ
Û.

Plastic Usage Java Code Generator...



Java Code Generator' Â ÇöÁ! ,ðµ"¿i¼ Package MemberÂÎ Â¬·i½° 1× ÂÎÁÍÆÐÀì½°¿i ´ëÇÑ Java¾¼ð¾¼ÎÀÇ
¼Ò½°ÄÜµâ,! ÀÜµ¿À» »ý¼°½ÄÄÑÁØ´Ù. ,,¾¼à Çï³aÀÇ Â¬·i½° ¶Ç´Â ÂÎÁÍÆÐÀì½°ÀÇ ¼¼Ò½°ÄÜµâ »ý¼°Àì
ÇÊ¿äÇï´Ù,é ÇØ´ç ¾¼ÆÀÎÄÜÀ» ¼¼±ÄÄÇÑ ÈÄ, Generate One¹öÆ°À» ´©,£,é Â¬·i½° ¶Ç´Â ÂÎÁÍÆÐÀì½°ÀÇ
Àì,§°ú µ¿ÄïÇÑ ÀÜ¹Ü¾¼ð¾¼ÎÀÇ ¼¼Ò½°°i Generate Destination Directory¿i¼¼ ¼¼³ÁµÈ µð·°Ää,®¿i
»ý¼°ÀìµÈ´Ù. Generate All¹öÆ°À» ´©,£,é ,ðµç Â¬·i½°¿Í ÂÎÁÍÆÐÀì½°¿i ´ëÇï¿© ÄÜµâ,! »ý¼°Çïí
Package Name°ú µ¿ÄïÇÑ Àì,§ÀÇ ÀÜ¹Ü¾¼ð¾¼Î ¼¼Ò½°°i »ý¼°Àì µÈ´Ù. Generate Destination Directory´Â
ÄÊ±â°aÄ,·Î Package Path·Î ¼¼³ÁµÀì µÇÁö,, ´Ù,¥ µð·°Ää,®¿i »ý¼°À» ¿ØÇÑ´Ù,é ¹Ü²Ü¾¼îµµ ¹«¹æÇï´Ù.





Plastic Usage Help Menu

Contents... 'Â µµζòÀÌ ÇÉζäÇÒ ¶§ ¼±ÅÃÇÑ'Ù. ±×.¯,é µµζò,»ÀÇ ³»ζëÀÌ ³aÅ,³'Ù. **About...** 'Â Plasticζi
°üÇÑ Àİ¹ÝÀûÀÎ Á²°,°; ³aÅ,³'Ù.

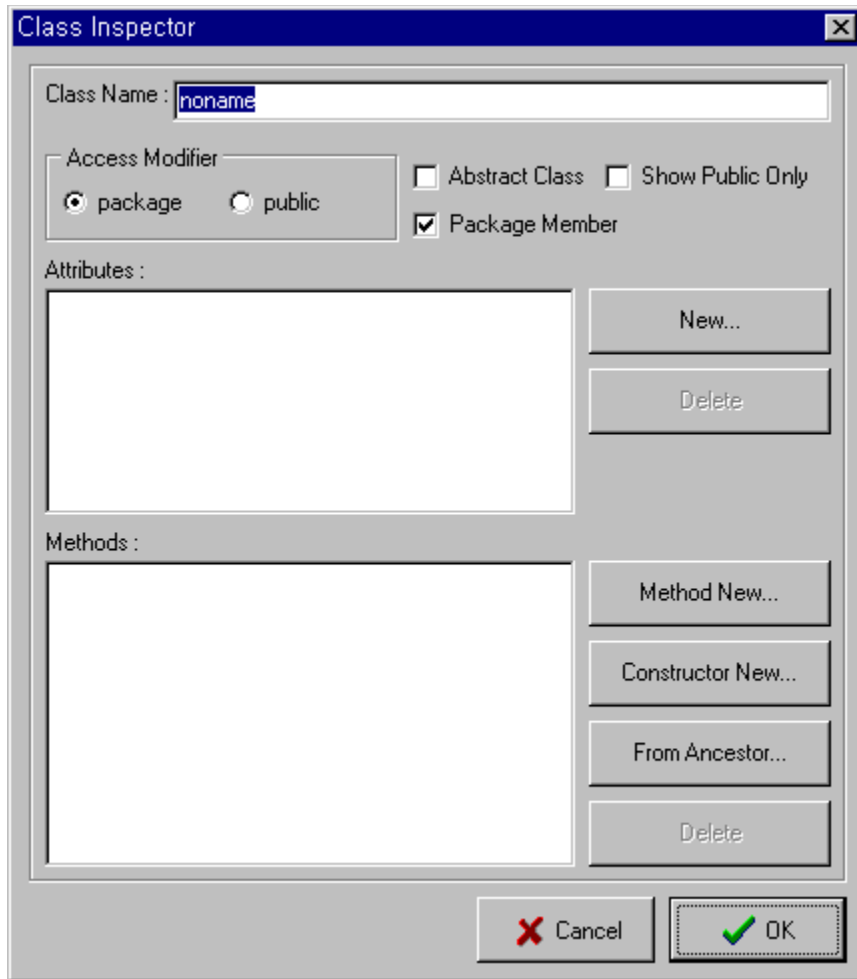


Modeling with Plastic Class Inspector

Java 3/4 d 3/4 i j' A A- j 1/2 o, i ± â 1/4 ú Ç Ĩ Â µ ¥ , ĩ ° i Á ö Â ÷ À i Á j À » Ç ¥ Ç ö Ç ĩ ± â À § Ç Ø abstract, public ° ° À ° ° í µ é À ĩ ° Û À »
1/4 ö À Ö ´ Û. abstract Å ° ç ö µ â ° i ° Û À , é Ã ß » ó Å - j 1/2 ° , i 3 a Å , 3 » , ç , public À ĩ ° Û À , é package ç Û ö ĩ ç 1/4 µ µ Å - j 1/2 ° , i
Á ¢ ± Û Ç Ò 1/4 ö À Ö ° Ö µ È ´ Û. public À ĩ ° Û À ö 3/4 Ê À , é package ° i µ È ´ Û. À ĩ ° í À ° Á ÷ Á ¢ package ¶ ó Á ç 13/4 à 3/4 ĩ , i
° Û À ĩ Á ö 3/4 Ê ´ Â µ ¥ , Å - j 1/2 ° i package 3 » ° ĩ ç j 1/4 , Á ¢ ± Û À ĩ ° í ´ É Ç Ĩ Û ´ Á ° í À » ¶ æ Ç Ñ ´ Û.

Java ç j 1/4 ´ Â Å - j 1/2 ° ° i Ç ĩ 3 a À ĩ » ó À Ç Ã ß » ó , p 1/4 Ò µ â (Abstract Method) , i ° i Á ø ´ Û , é ± × Å - j 1/2 ° ´ Â 1 Ý µ â 1/2 Ã
Ã ß » ó Å - j 1/2 ° ° i µ Ç 3/4 ĩ 3/4 ß Ç Ñ ´ Û. À ĩ Á j À » À ¯ À Ç Ç ĩ ç © 3/4 ß Ç ĩ , ç À ĩ ° í À » Á ö Å ° Á ö 3/4 Ê À , é Ä Ä Æ Ä Ä ĩ Å , À Ö ç i ç i ¯ , i
33/4 ° í À ĩ ´ Û. ´ Û À 1/2 ç j , ĩ ° i Á ö Å - j 1/2 ° ç j ´ è Ç Ñ Ç ¥ Ç ö À ĩ À Ö ´ Û.

```
class MyClass { ... }  
abstract MyClass { ... } // Ã ß » ó Å - j 1/2 °  
public MyClass { ... } // package ç Û ö ĩ ç 1/4 Á ¢ ± Û ° í ´ É.  
public abstract MyClass { ... }
```



Class Inspector ʼĀ Ā-·j½° ½É¼Ā» ʼðíĀ-·Çĭ, é ³ĀĀ, ³Ű. °Ā±âĸiĀ Ā ĸ·-°iĀö Ā°, ĭ°, ĸ·ĀÖ°í, ¶Ç ĄĭÁýÇÖ ¼ö ĀÖµµ-ĭ ÇÖĀØŰ. ĸi¼± ĀÖ»óŰĸi Ā-·j½°ĀÇ Āĭ, ŠĀ» ±âĀÖÇĭ°í, ĀÇ±ŰĀĭÇŅĀ° public¶ÇĀ package-ĭ µĭĀµŸ, publicĀ, ĭ µĭ°ÖµÇ, é »ý¼°µĒ Java¼ĀÖ½°ĸi¼ Ā-·j½°°i publicĀĭ¶óĀ ĸi¼â³¼ĭ, ĭ°ŰĀĭ°Ö µĒŰ. ±×, Āi ¼ĒĒ, é ¼ĒĒ«°ĭµµ °ŰĀö ¼ĒĒĀŰ. Abstract Class¶óĀ Ā¼ĀĀ°Ű½°ĸi ÇŸ¼Āĭ, Çĭ, é abstract¶óĀ Ā ĸi¼â³¼ĭ, ĭ°ŰĀĭ°Ö µĒŰ. ÇŅ°iĀö ĀÖ, ĩÇÖ °ĭĀĭ ĀÖĀµŸ ±×°ĭĀ° Package Memberĸi Ā¼ĀĀ°ÇŸ¼Āĭ, ¼öĸi ¼ö, ®, é JavaĀŰµĀ°i »ý¼°µÇĀö ¼ĒĒ°í, ¶Ç ½É¼Āĭ Āi¼±Ā, ĭ ÇŸ¼ĀĭµÇ°Ö µĒŰĀ ĀiĀĭŰ. Āĭ°ĭĀ°, ðµ, µĸi Ā÷ĀÇĀŰĀ, ĭ »óŰĀĭ ¼öĀ, ³Ā, »ó¼ĀÖŰ°è ¶ÇĀ ĸi-Ű°Ű°è, ĭ ÇŸ¼ĀĭÇĭ±â ĀŠÇØ¼, ÇĒĸiĀÇŅ °æĸiĸi »çĸèÇĭ°Ö µĒŰ. ³Ā, ÖĀö Show Public Onlyĸi Ā¼ĀĀ°, ĭ Çĭ, é, Fields, Methods ĀĒ publicĀĭ°, Ē, éĸi ÇŸ¼ĀĭÇĭĸi ĀØŰ.

Attributes(or Fields)°ĭ°Đĸi¼ Ā ¼ĀÖ¼°µéĀĭ , ®½°Ē®Ű½°ĸi ³ĀĸiĀĭ µÇ, ç, ʼðíĀ-·Ā» Çĭ°Ö µÇ, é Attribute InspectorĀ°µµĸi°i ĸi, °Ű. ±×, ®°í ĸĀ, ŸĀĒĸi µĭ³ĀÇ ¼öĒ°Āĭ ĀÖĀµŸ, NewĀ »ð·ĭĸi ¼ĀÖ¼°Ā» »ý¼°Çĭ°ĭĀŰ ÇÖ ¶Š, DeleteĀ , ®½°Ē®Ű½°ĸi¼ ¼±ĀĭµĒ ¼ĀÖ¼°Ā» »èĀĭÇĭ°ĭĀŰ ÇÖ ¶Š »çĸèÇĭĀ °ĭĀĭŰ.

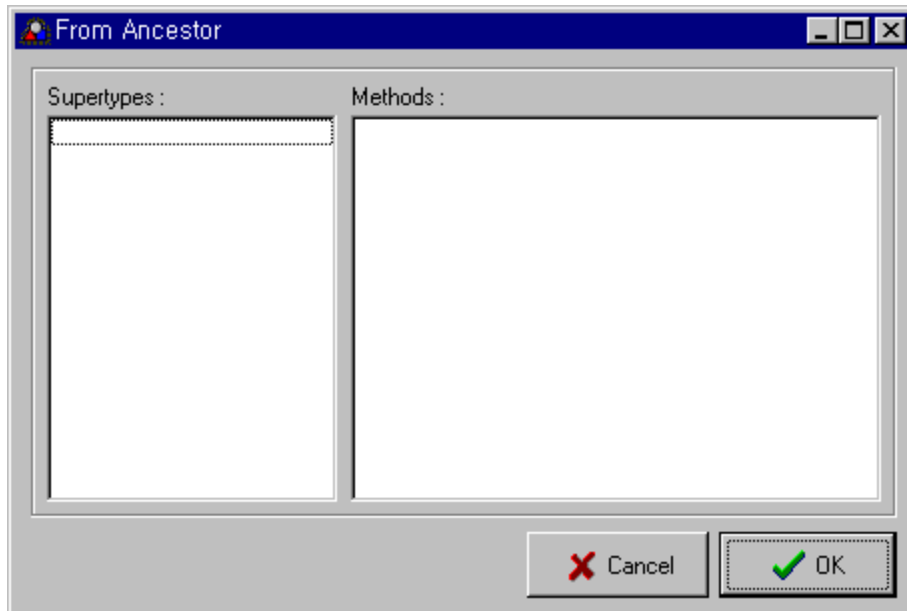
Methods°ĭ°Đĸi¼µµ , ¼¼ÖµĀµéĀĭ , ®½°Ē®Ű½°ĸi ³ĀĸiĀĭ µĒŰ. ĸĀ, ŸĀĒĸi ĀÖĀ 4°³ĀÇ ¼öĒ°Ā» ¼³, ĭÇĭ, é, ĸi¼± Method NewĀ »ð·ĭĸi , ¼¼ÖµĀ, ĭ »ý¼°ÇÖ ¶Š °Ç, Ē, é Method Inspector°i ĸi, ®, é¼ »ð·ĭĸi Method, ĭ ĀŰ¼°ÇÖ ¼ö°i ĀÖ°Ö µÇ°í, Constructor NewĀ »ð·ĭĸi »ý¼°ĀŰ, ĭ ĀĒ°iÇĭ°ĭĀŰ ÇÖ ¶Š »çĸèÇŅŰ. From AncestorĀ Āĭ Ā-·j½°ĀÇ ¼öĒ°ĀŰĀ-·j½° ¶ÇĀ ¼öĒ°ĀŰĀĭĀĭĀĀĭ½°i ĀÖĀ»°æĸi ±×°íµéĀÇ , ¼¼ÖµĀĀĒ Çĭ³Ā, ĭ ±×°Ē·ĭ°»çÇØ ĸĀ ¼ö ĀÖĀ ±âĒĒĀĭŰ. , ¼¼ÖµĀ, ĭ overrideÇĭĀ °æĸi , ¼¼ÖµĀĀÇ ÇĀ·ĭĀĀ, ĀÖĀĭ

ÄÏÄ¡ÇØ¾ß Ç¡¹Ç·Î Àì ±â´ÉÀ° ,Å¿ì ÆÉì,®Ç¡´Ù. Delete´Â Àì¹ì Á,ÀçÇ¡´Â ,p¼ÒµåÀÇ »èÁ!Àì´Ù.

ConstructorÀÇ °æ¿ì, »ý¼°ÀÚÀÇ Àì,§À° Å¬·¡½°ÀÇ Àì,§°ú µ¿ÀÏÇÑµ¥, Àì°ÍÀ° Inspector¿¡¼´Â ¹Ý¿µÀì µÇÁö ¾É°Í, OK¹öÆ°À» ´©,¥ ÈÄ°ÍÁÍ ¹Ý¿µÀì µÇ¹Ç·Î ÁÖÀÇÇØ¾ß ÇÑ´Ù. CancelÀ» ¼±ÄÃÇÒ °æ¿ì Attributes¿Í Methods¿¡ ´ëÇÑ Ã³,®¿Ü¿¡´Â ,ðµí ¹«½ÃµÈ´Ù.



Modeling with Plastic From Ancestor Dialog



supertypes ,@½°/Æ® ¹Ú½°¿;¼´Â Å¬·;½°°; »ó¼Ô¹ÐÀ° ,ðµç Å¬·;½° ¶Ç´Â ÀîÁí/ÆÐÀ½°°; ³ªÅ,³´Ù. ¹°·Ð
°í,ð(Parent) »Ó,,¾/Æ´í¶ó ¼±Á¶(Ancestor)±îÁö ,ðµí°; ´Ù ³ª¿À´Âµ¥, ¾/ÆÀìÀÙÀ» ¼±ÃÃÇï°Ô µÇ,é
¿À,¥ÂÊÀÇ Methods°í°Ð¿; ÇØ´ç ¾/ÆÀìÀÙÀÇ ,Ð¼Òµå°; ,ðµî ³ªÅ,³»°Ô µÇ´Âµ¥ ÀìÁß¿; ÇÊ¿äÇÑ °íÀ»
¼±ÃÃÇï¿© OK¹ö/Æ°À» ´©,£,é µÊ´Ù. Methods°í°Ð¿;´Â ³ªÅ,³ª´Â °íµéÀ° Constructor,¡ Á¿¿ÜÇÑ
Operations¿í MethodsÀì´Ù.



Modeling with Plastic Interface Inspector

ÀÎÁÍÆÐÀÌ½° Â ±× ¾Î¶ÇÑ ±,Çöμμ °;ÁöÁö ¾Ê° Â´Ù. ÀÎÁÍÆÐÀÌ½°ÀÇ Fields´Â ,ðμÎ staticÀÌ°í finalÀÌ´Ù. ÇĬÁö, ±×.ÇÑ ç¹¾à¾Ĭ Â °ÙÀÌö ¾Ê° Â´Ù. μŭ¶ó¾ ÀÌ°íμéÀ »ó¾ö(Constant)°; μÊ´Ù. ±×,®°í ,ðμç Method ´Â Ãß»ó,Þ¾ÖμâÀÌ,ç, ç¹¾Ã abstract¶ó´Â ç¹¾à¾Ĭ Â °ÙÀÌö ¾Ê° Â´Ù. ÀÌ°ÍÀ° ÀüÇö ±,ÇöÀ» °;ÁöÁö ¾Ê¾Æ¾ß ÇÑ´Ù. ç©±â¾´Â ÀÌ°ÍÀ» çĬ,®´Â OperationÀÌ¶ó ¶Ĭ,£°Ú´Ù. Å¬½°çĬ ,¶Àù°;Áö·Ĭ publicÀÌ μÇ,é ÆÐÂ°Áö çÜ¶çĬ¾ÀÇ ÀÎÁÍÆÐÀÌ½°ÀÇ Áç±ÙÀ» ÇãçëÇÑ´Ù. ´ÙÀ½çĬ ,¶°;Áö ÀÎÁÍÆÐÀÌ½°ÀÇ Ç¥ÇöÀÌö ´Ù.

```
interface MyInterface { ... }  
public interface MyInterface { ... } // ÆÐÂ°Áö çÜ¶çĬ¾ÀÇ Áç±Ù°;É.
```

ÀÎÁÍÆÐÀÌ½° ÀÌ½°ÆÑÁçĬ¾À ÖÖ»ó´ÙçĬ´Â ÀÎÁÍÆÐÀÌ½°ÀÇ ÀÌ,ßÀ», Access ModifierçĬ¾À´Â Áç±ÙÇãçëçĬ °üÇÑ °ÍÀ» ¾¾ÁçĬ,ç, Package MemberçĬ¾À´Â ,ðμ,μçĬ °ü½ÉÀÌ ¾ø´Â °ÍÀ,é Ã¾Áç,Ĭ ¾ø¾Ö°í, Å¬½°ÀÇ °æçĬçĬ ,¶Àù°;Áö·Ĭ ÁÜμâ»ý¾°çĬ¾À ÁççÜμÇ,é¾À, ÀÎÁÍÆÐÀÌ½° ½É°¾ÀÌ Áç¾À,·Ĭ Ç¥½ÃμÇ°Ö μÊ´Ù.

Constant °¶ÐÀÇ ,®½°Æ@½°½° Â »ó¾öçĬ ´ëÇÑ °ÍÀ» ç°ÁçĬ,ç, ´ð°íÅ¬,ÀÎ °æçĬçĬ Constant Inspector,Ĭ ç- °í, NewçĬ Delete´Â »ó¾öçĬ ´ëÇÑ Ãß°;çĬ »èÁçö ¶ßçĬ »ççëÇÑ´Ù. ÀÌ°ÍÀ° Operation°¶ÐçĬ ´ëçø¾μμ ±×

ἔ·ἴ· Ἄ·ὠ·ζ·ἐ·μ·Ἐ·Ὠ·.



Modeling with Plastic Attribute Inspector

Attribute Inspector: Attribute Type: Initial Value:
☐ Static ☐ Final
Access Modifier
☒ package ☐ private ☐ protected ☐ public
Cancel OK

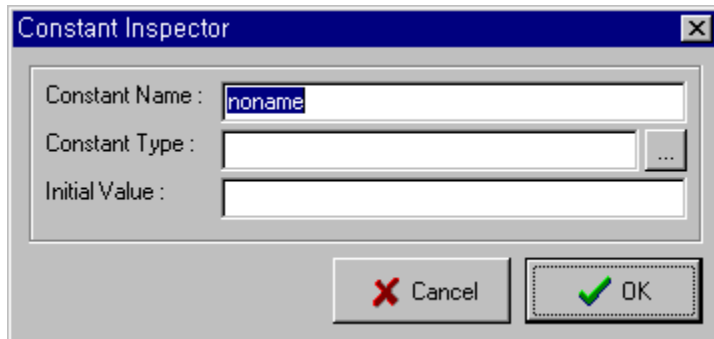


Modeling with Plastic Access Modifier

¼Ó¼°, , Þ¼Òμå μîÀÇ Á±Ù ¯æÀÚçí´Â package, private, protected, pulbicÀÇ 4°íÁö°í ÀÖ´Ù. package´Â
¾Æ¹«·± ç¹¼à¾î,í °ÙÀíÁö ¾Ê¾ÒÀ» ¶§ ÇØ´ç ¼Ó¼°Àì³ª, Þ¼Òμå°í ÆÐÅ°ÁöÀÇ ³»îçí¼, Á±ÙÀì °í´ÉÇí
´Ù´Â °íÀ» ÀÇ¹ÇÑ´Ù. ±×,®°í private´Â Å¬·í½° ³»îçí¼, Á±ÙÀì °í´ÉÇí,ç, protected´Â Å¬·í½°Àì ³»î ¹×
ÀÚ¼Õ Å¬·í½°çí ÀÇÇØ¼μμ Á±ÙÀì °í´ÉÇí´Ù. publicÀ° Å¬·í½°ÀÇ çÜ°îçí¼μμ Á±ÙÀì °í´ÉÇí´Ù´Â °íÀ»
ÀÇ¹ÇÑ´Ù.



Modeling with Plastic Constant Inspector



The image shows a 'Constant Inspector' dialog box. It has a title bar with a close button. Inside, there are three input fields: 'Constant Name' with the text 'noname', 'Constant Type' with a dropdown arrow, and 'Initial Value'. At the bottom, there are 'Cancel' and 'OK' buttons.

»ó¼ö´Â ¾¼½ÄÄÛÄ,Î staticÄìí finalÄì´Ù. µû¶ó¼ °aÄ» °ü²Ü ¼ö ¾øÄ,,ç Initial Value¶ðçí °aÄ» ²Ä
³Ö¾µî¾¾¾¾¾ ÇÑ´Ù. ç¹,í µé,é float Pi = 3.141592 çÍ °°Äì ÄÜ¼°ÇÒ ¼ö ÄÖ´Ù. »ó¼öÄÇ Ä,ÄÖµµ Types
Dialogçí¼ ¼±ÄÄÇÒ ¼ö ÄÖ´Ù. ÇÑ°;Äö ÄÖÄÇÇÒ Ä;Ä° °aÄì int, float °°Ä° °æçí´Ä 255, 3.141592Ã³.³ °aÄ»
±×´ë´Î ³ÖÄ,,é µÇÄö,, StringÇüÄî °æçìçí´Ä ¹Ýµâ½Ã “Lee”, “Kim” Ã³.³ ÄìÁß µûçÈÇ¥,í °Ùç©¾ß ÇÑ´Ù.



, ¼Òµâ Á Ò Õ, ³³ä; òñÇØ ¾ä°É Á» ¹ÄâÇĭ Ò. Method Name, Return Type ĭ Á °¢°¢, ¼ÒµâÄÇ Àĭ, §°ú
 , ®Äĭ °âÀĭ ÄÖ° Á °æĭ, ®Äĭ °âÄÇ Ä, ÄÖÄ» ±âÄÖÇÑ Ò. , ¾ä °âÀĭ ¾Ø Ò, é voidÄĭ Ò. ±×, ®°ĭ throws° Á
 Exception Handling ĭ ¼ ³ª; Ä° Á °³³äÄµ, Äĭ, ¼Òµâ° ĭ ¹ß»ý½Ä³ ¼ö ÄÖ° Á ExceptionÄÇ Ä, ÄÖÄ» ³ª; ÇÑ
 Ò. , ¾ä Çĭ³ª Äĭ°óÄÇ ĭ ĭ ÜÄ, ÄÖÄĭ ¹ß»ýÇÖ ¼ö ÄÖ° Ò, é, (comma)·ĭ ¼·ÎÄÇ Ä, ÄÖÄ» ±, °ÐÇØ ÄØ Ò.

Static Method' Â Ä¬·½°ÄÇ ,â°öÀÖÀ» ÄÇ¹¹ÇÑ'Û. °,ÄëÄÇ °æ¿ì Ä¬·½°ÄÇ Ä¹½°Ä¹½°ÄÇ ,â,ÓÀìÄö,,
staticÄÇ °æ¿ì¿ì·Ä Ä¬·½°ÄÇ ,â°öÀì'Û. Abstract Method' Â Äß»ö, p¹¼Öµâ·î¿á ±,ÇÖÀ» °,ÄöÄö ¾Ê'Ä

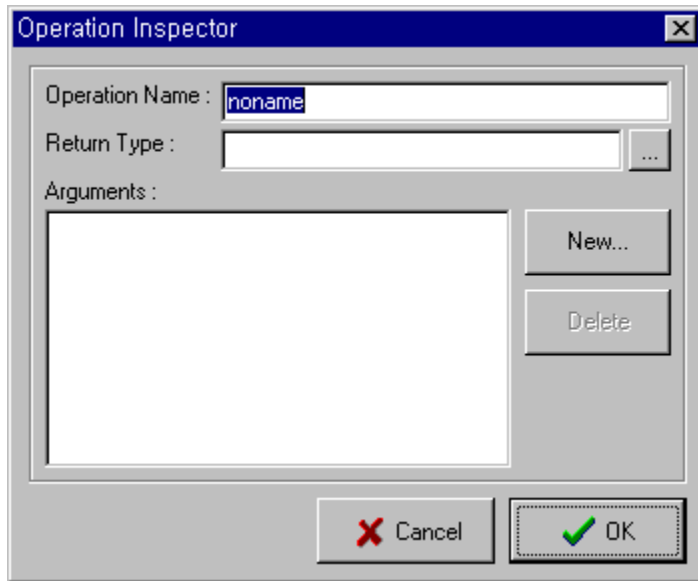
,P¼ÖµåÀÓÀ» ÀÇ¹Çĭ,ç Àĭ ,P¼Öµå,ĭ »ó¼Ó¹PÀ° Å¬·ĭ½° Áß ¹Ýµå½Ã Çĭ³ª´ Å ±,ÇöÀĭ µÇ¾ĭ¾ß ÇÑ´Ù.
±×·,Äĭ ¾ÊÄ,é ÄÄÆÄÄĭ Å,ÀÓĭ ĭ ĭ·ĭ,ĭ ¹ß»ý½ÃÅ³ °ĭÀĭ´Ù. finalÀ° Àĭ ,P¼Öµå´ Å ´ð Àĭ»ó overrideÀĭ µÉ ¼ö
¾ØÄ½À» ³ªÅ,³»,ç, synchronized´ Å multi thread,ĭ ÀĭĭëÇĭ´ĭÀÚ ÇÒ ¶ß, ,P¼Öµå° ĭ µĭ½Ã¼°À» °ĭÁø´Ù´ Å
°ĭÀ» ³ªÅ,¾½´Ù.

,P¼Öµå° ĭ Çĭ³ª Àĭ»óÀÇ ÄĭÀÚ(argument),ĭ °ĭÁú °æĭ ĭ Argument °ĭ°ÐÀÇ New¹öÆ°À» ´©,ßÄ,·ĭ½á »ð·ĭĭ
ÄĭÀÚ,ĭ Äß° ĭÇÒ ¼ö ÀÖ°ĭ, Delete¹öÆ°À,·ĭ Àĭĭ Á,ÀÇÇĭ´ Å ÄĭÀÚ,ĭ »èÄĭÇÒ ¼ö ÀÖ´Ù. ±×,®°ĭ
Implementation°ĭ°Ðĭ´ Å ,P¼ÖµåÀÇ ±,ÇöÀ» »ðÀÔÇÒ ¼ö° ĭ ÀÖ´Ù. Àĭ °ĭ°ÐÀ° ÀÚ¹Ù ÄÚµå »ý¼°ĭ ĭ¼ ±×´è·ĭ
°¹»çµÇ¾ĭ µé¾ĭ° ĭ¹Ç·ĭ ÁÖÀÇÇĭĭ© ÄÚµùÇØ¾ß ÇÑ´Ù. ´ÙÀ½ĭĭ´ Å Java¾ð¾ĭ,ĭ »ý¼°ÇÑ ĭ¹ÀÇ °ĭ°ÝÀ» °,ĭ©
ÁÖ°ĭ ÀÖ´Ù.

```
protected final void method_name(int value)
    throws exception1, exception2
{
    implementation
    ...
}
```



Modeling with Plastic Operation Inspector

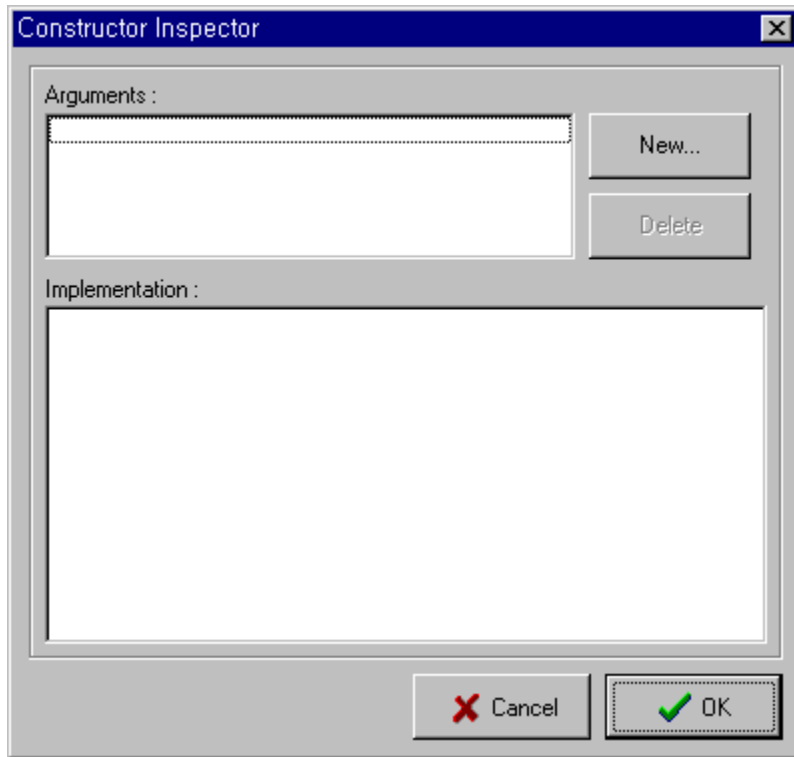


Method Inspector
Arguments
Types Dialog

```
String AsString();  
void Store(Writer writer);  
void Clone();
```



Modeling with Plastic Constructor Inspector



»ý¼°ÀÚ´Â Àì, \$μμ , @ÀĬÀ, ÀÔμμ ¾ĬŲÇÑ Access Modifierμμ ¾ø´Ù. ÇÖ³ÄÇĬ, é ±x·± °ĬμέÀ° ÄÇ¹Ĭ°; ¾ø±â
ŲŲ¹@ÀĬ´Ù. Àì, ŒÀ° Ä¬·;½°ÄÇ Àì, Œ°ú μÇÄĬÇø¾ß ÇĬ°í, , @ÀĬ À, ÀÔÀ° ¾øÄ,, Ç Access Modifier´Â ¹Ýμâ½Ä
publicÀì±â ŲŲ¹@Ç; ±»Àì ±âÀÔÇÒ ÇĬÉçä°; ÀüÇð¾ø´Ù. ÇĬÁö,, ImplementationÀ° °;Áú ¼ø°; ÀÖ´Ù.

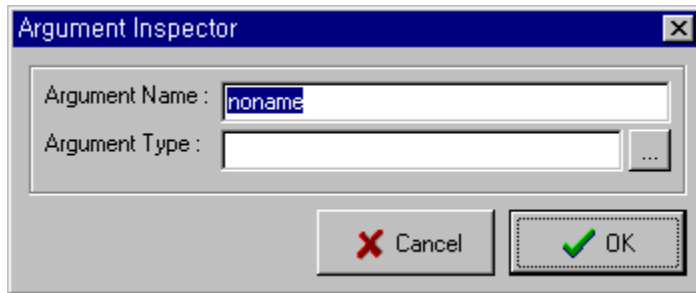
»ý¼°ÀÚ´Â Ç©·¬³ÄÇ ÄĬÀÚμέÀ» °;Áú ¼ø°; ÀÖ´Âμϕ, ÄĬÀÚμέÀÇ Ä, ÀÔÀì³ª, °¹¼øÇ; μŲŲó¼ ¼·Ĭ´ÙÀ°
»ý¼°ÀÚ°; μĒ´Ù. Áĭ, ÇĬ³ªÄÇ Ä¬·;½°´Â Ç©·¬³ÄÇ »ý¼°ÀÚ,Ĭ °;Áú ¼ø ÀÖ´Ù. ±x, @°í »ý¼°ÀÚμέÀ°
»ó¼ÓμÇÁö ¾ĬÉ´Â´Ù. »ý¼°ÀÚ°; ¾ø´Â °æçĬ ÀÚ¹ÙÇ;¼´Â ÄĬÀÚ°; ¾ø´Â μðÆúÆ® »ý¼°ÀÚ,Ĭ ÀÚμçÄ,Ĭ Äß°
;½ÄÑÁø´Ù. ÇĬÁö,, »ý¼°ÀÚ°; ÀÖ´Â °æçĬçĬ´Â ±x·¬ÇÑ ÄĬÀ» ÇĬÁö ¾ĬÉ´Â´Ù. ¾Æ·çĬ »ý¼°ÀÚçĬ ´èÇÑ ç¹°
; ÀÖ´Ù.

```
class MyClass {  
    int MyValue;  
    MyClass() {  
        MyValue = 500;  
    }  
}
```

```
        ...  
    }  
    ...  
}  
...  
MyClass myObject = new MyClass();
```



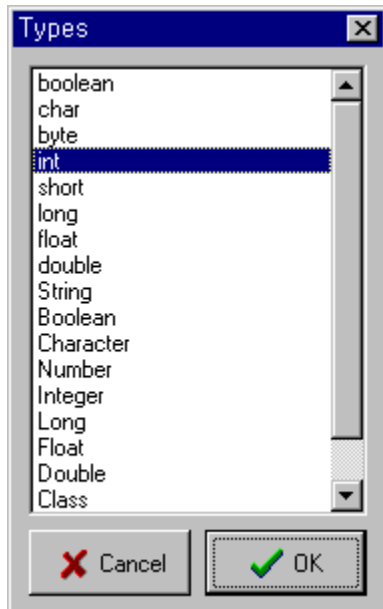
Modeling with Plastic Argument Inspector



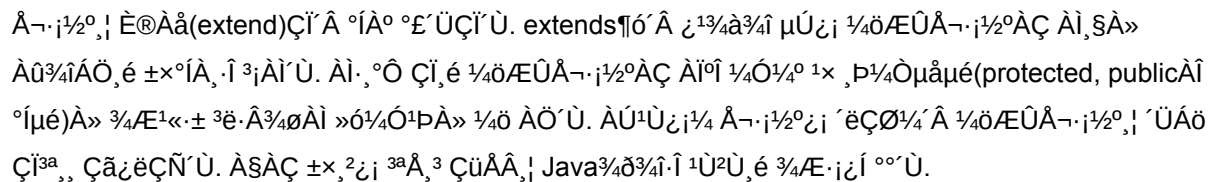
ÀÎÄÚ(Argument) Ä ÄÎÄÚÄÇ ÄÏ, s 1x ÄÎÄÚÄÇ Ä, ÄÖ,, ±â¼úÇÖ ¼ö ÄÖÄ,,é µË´Ù. Ä,ÄÖÄ» ±â¼úÇÏ±â
ÄSÇØ Types Dialog, ÄÏÇëÇÖ ¼ö°; ÄÖ´Ù. ÄÚ¹ÙÇ;¼´Ä ,p¼ÖµåÇ; ´ëÇÑ ,ðµÇ ÄÎÄÚµéÄ° Call By ValueÄÇ
ÇøÄÇÇ; µû,¥´Ù.



Modeling with Plastic Types Dialog



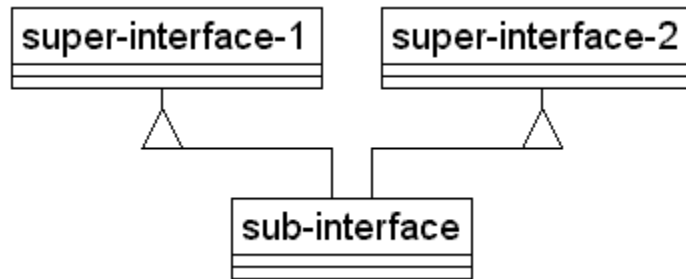
Types Dialog: A dialog box for selecting a data type. The list includes: boolean, char, byte, int, short, long, float, double, String, Boolean, Character, Number, Integer, Long, Float, Double, and Class. The 'int' type is selected. The 'OK' button is highlighted.



Plasticç;¼´ Â Inheritance ½Éº¼À» ¼±ÄÃÇÑ ÈÄç; ¼ºeÄ¬·½ºç;¼ ºÄ ¼ºÆÜÄ¬·½º·Î mã·j±× ¾Ø mã·Ó(Drag & Drop)À» Çİ,é »ð·Îç´ »iº¢Çü ½Éº¼ÄÎ »ý±ä,é¼ çºàÄÎ mã·Ü. ,¾Ä Çİ³aÀÇ ¼ºÆÜÄ¬·½ºç; ç©. ¯³ÀÇ ¼ºeÄ¬·½º,Î çºàÇİºİÀÜ ÇÑ´Ü,é, ÄÖÄÊÀÇ ¼ºeÄ¬·½ºº·Ä ÀSçİ ººº 1æ¹ýÄ,·Î Çİºİ ³a,ÓÄº ¼ºeÄ¬·½º,Î çºàÇÒ ¶Sç;¼ Â mã·j±× ¾Ø mã·Óç;¼ mã·Ó(Drop)ÇàÀS,Î »iº¢Çü ½Éº¼ ÀSçİ´Üº Çİ,é(¶ç;½º 1ºÆ·À» »iº¢Çü ½Éº¼ç;¼ ¾³Ä,é) º,±Ä ÁÁºº »iº¢Çü ½Éº¼À» ºØÄÇİ,é¼¼¼ çºàÄ» ÇÒ ¼ºº; ÄÖ´Ü.



Modeling with Plastic Interface Extends

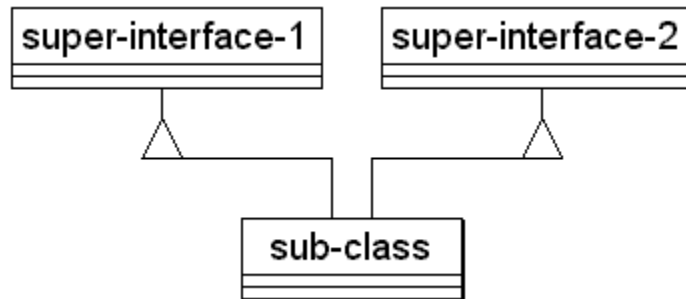


ÀÎÁÍÆÐÀÌ½°ÀÇ »ó¼ÓÀÇ Ç¥±â¹ýÀ° Å¬·½°¿Í µ¿ÀİÇİÁö,, Å¬·½°ÀÇ °í´´Å ¾à°£ ´Ù,£´Ù. ÀÎÁÍÆÐÀÌ½°´Å ¼öÆÛÀÎÁÍÆÐÀÌ½°,İ Çİ³ª Àİ»óÀ» Çă¿ëÇÑ´Ù. Àİ·,°Ô ÇØ¼¼ ,,µé¾îÁø ¼°êÀÎÁÍÆÐÀÌ½°´Å ,ðµç ¼öÆÛÀÎÁÍÆÐÀÌ½°ÀÇ Æ¯ÅİÀ» ¹°·Á¹ÐÀ,,ç, ¼öÆÛÀÎÁÍÆÐÀÌ½°,İ ¿ă±,Çİ´Å ¹®,Æ¿İ ¼°êÀÎÁÍÆÐÀÌ½°°İ ´ë½Å ¾²Àİ¼ö ÀÖ´Ù. (Substitutability)

```
interface subinterface extends superinterfacel, superinterface2 {  
    // new properties...  
}
```



Modeling with Plastic Implements



Ä¬·j½° Â Çï³ª Àì»óÀÇ ÀîÄíÆÐÀì½°·îíÄí »ó¼ÓÀ» ¹ᐅÀ» ¼ö°j ÀÖ´Ù. (¿ªÀ° °Ò°j´É!) ´ÜÁö,
¼öÆÜÂîÄíÆÐÀì½°ÀÇ ¿ÄÆÜ·¹À¼ÇÀ» ,ðµí ±,ÇöÇØ¾ß,, ÇÑ´Ù. ,îjÁö ±,ÇöµÇÁö ¾ÊÀ° °íÀì ÀÖ´Ù,é ±×
Ä¬·j½° Â Ãß»óÄ¬·j½°(abstract class)Àì¾î¾ß Çï°í ±× ÀÚ¼ÖÄ¬·j½°¿j´Â ¿îÄüË÷ ±,ÇöµÇ¾î¾ß,, ÇÑ´Ù.
¼°êÄ¬·j½°ÀÇ Àî½°Äî½°(instance)´Â ¼öÆÜÂîÄíÆÐÀì½° ÀîÄíÆÐÀì½°¿ ¿ä±,Çï´Â ÀÚ,®¿j ´ë½Ä »¿¿ëµÉ
¼ö ÀÖ´Ù.

```
class subclass implements superinterfacel, superinterface2 {  
    ...  
    // implements operations of superinterfaces.  
}
```

[illegible]

Person°ú Company¿; ´ëÇÑ ¿¬°üÀ» 3aA,3»,é From Class°; PersonÀìµÇ°í To Class°; Company°; µÇ°Âµ¥,
 PersonÀ° Company¿; ´ëÇØ¼¼ ÇÇ°í¿ëÄÎ(employee) ÀìÇ.Î Role NameÀ° employee, Company° Â Person¿;
 ´ëÇØ¼¼ °í¿ëÄÎ(employer)ÀìÇ.Î Role NameÀ° employer°; µÈ°Ù. ¶ÇÇÑ PersonÀ° CompanyÀÇ Part°; ¾/Æ

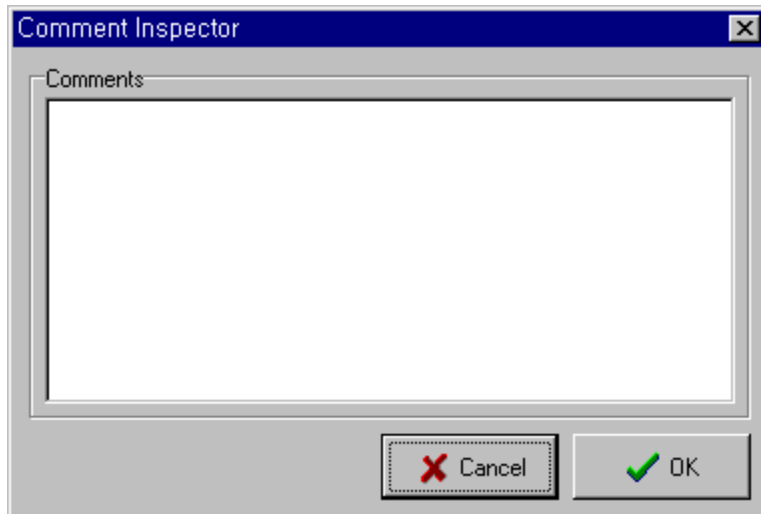
İÇ·İ Ğ¬üÀÇ ÇüÂÂ´Â AssociationÀİ´Ü.

```
class Person {  
    Company employer;  
}
```

```
class Company {  
    Person[] employee = new Person[]; // reference  
}
```



Modeling with Plastic Comment Inspector

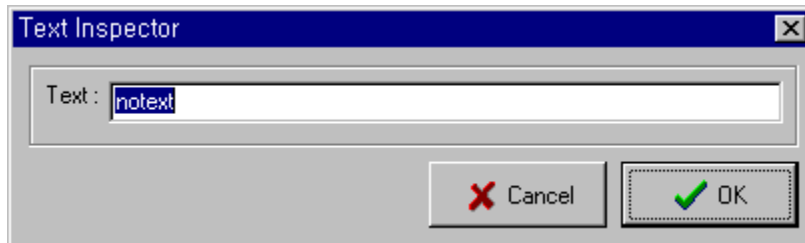


3/4î¶² Å¬·;½³ª ÄîÄíÆÐÀì½º¿; ÁÖ¼@À» °ÙÀì°íÀÚ ÇÑ´Ù,é ¿½± Comment ½É¹¼À» ¼±ÄÃÇí¿©
Comment,¡ Çí³ª »ý¼ºÇí°í, ±×°íÀ» Äí°íÅ¬, ¯Çí¿© Comment Inspector,¡ ¿¬ ´ÙÀ½¿; ¼³,íÀ» ½á³Ö´Å´Ù.
±×,®°í Comment Link ½É¹¼À» ¼±ÄÃÇí¿© Comment·î°íÄí ÁÖ¼@À» °ÙÀí Å¬·;½³ª ÄîÄíÆÐÀì½º·î µå·;±×
¾Ø µå·ÓÀ» ÇÑ´Ù. ,¾à ÁÖ¼@À» “this is a class”¶óÂ ¼³,íÀ» °Ù¿´Ù,é ¾Æ·;¿í °°Àì ÀÚ¹Ù¾ð¾î°;
»ý¼ºµÉ °íÀì´Ù.

```
// this is a class  
class Something { ... }
```



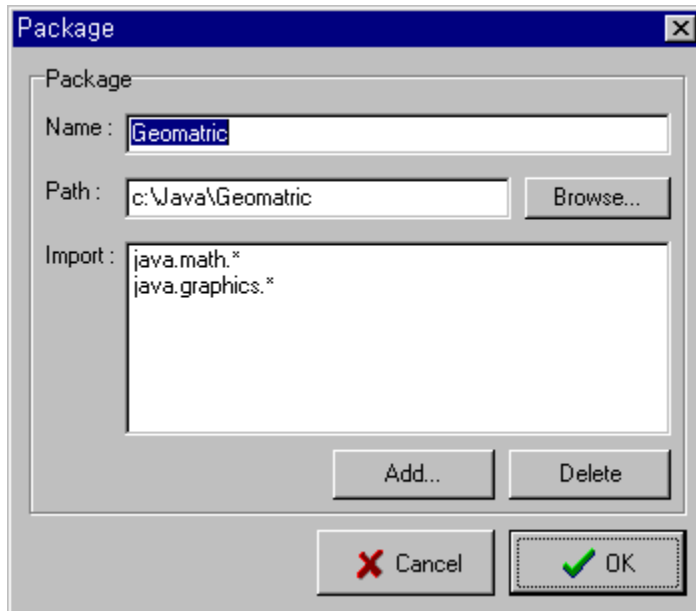
Modeling with Plastic Text Inspector



Text´Â ¾¼¶°ÇÑ ÀÇ¹µµ °;ÁöÁö ¾ÉÁö,, °;Åå ÇÊ¼öÀúÂÎ °ÍÀ¶ó ÇÍ°Ú´Ù. Àì°ÍÀ° ¼³°èµÈ ±,¼@±,¼@¿i
¼³,íÀ» ´¼ö°; ÀÖ´Ù. Àì°ÍÀ° ´ÜÁö ÇÑ ÁÜ,, °;´ÉÇÍ´Ù. µÇµµ·ÍÀì,é Àì°ÍÀ» ,¹Àì »ç¿ëÇÒ °ÍÀ» ±ÇÀåÇÑ´Ù.
¿Ö³ÄÇÍ,é ¼³,íÀ» ,¹Àì ´¼ö·Í ÀìÇØÇÍ´Âµ¥ µµ¿òÀì µÇ±â ¶§¹@Àì´Ù. Java¾öð¾¼îÄÇ »ý¼°¿;´Â ÀüÇô
¿µÇâÀ» ¹Ä;Áö ¾É´Â´Ù.



Modeling with Plastic Package Window



Plastic is a Java-based modeling language. It is used to create 3D models of objects. The Plastic Package Window is used to define the packages that are used in the model. The Name field is used to enter the name of the package. The Path field is used to enter the path to the package. The Import field is used to enter the packages that are imported into the model. The Add... button is used to add a new package to the list. The Delete button is used to delete a package from the list. The Cancel button is used to cancel the operation. The OK button is used to confirm the operation.

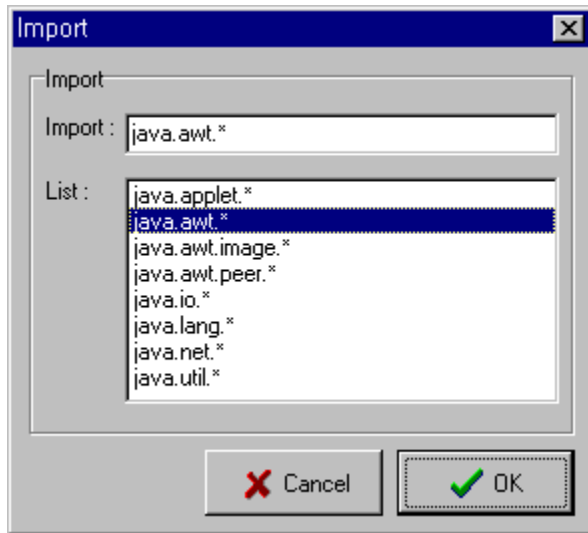
```
package Geomatic;
```

```
import java.math.*;
```

```
import java.graphics.*;
```



Modeling with Plastic Import Window



ImportÇÒ ÆÐÅ°Äö°; ÀÖ´Ù,é Import À©μμζì,¡ »çζëÇÒ ¼⁄⁄⁄; ÀÖ´Ù. Àì°ÍÀ° ÀÚ¹Ùζ;¼ ±â»»ÀûÀ,·Î Á¡°øÇÍ´Â´
´ë°ÎÐÀÇ ÆÐÅ°Äöζ; ´ëÇÑ ±â¼⁄⁄⁄À» Æ÷ÇÔÇÍ°í ÀÖ¾⁄⁄⁄, ¹À° °æζìζ; ÀÖ¾⁄⁄⁄¼ ¹Ù·Î¼±ÃÇİζ© »ðÀÔÇÒ ¼⁄⁄⁄
ÀÖ´Ù. ¾⁄⁄⁄à ¾⁄⁄⁄´Â °æζìζ;´Â Edit¹Ú¹⁄⁄⁄ζ; Á÷Á¢ ±âÀÔÇÒ ¼⁄⁄⁄ ÀÖ´Ù.

Modeling with Plastic Design Information Window

Design Information
✕

Design Info.

Title :

Author :

Comments :

More Info. :

symbols	number of symbols
classes	11
interfaces	0
total classes	11
total interfaces	0
associations	7

✕ Cancel
✓ OK

¼³ºè¿; °üÇÑ Áº, µéµµ ¼³ÁºÇÒ ¼ºº; ÀÖ´Âµ¥, Title, Author, Commentsº; ±×ºíµéÀì´Ù. Title¿;´Â , ðµ´ÀÇ Àì, SÀ», Author¿;´Â ÀúÀÚ(º³¹ÀÚ)ÀÇ Àì, S, ±×, ººí Comments¶ð¿;´Â ´ëÃæÀÇ ¼³, íÀì³ª, ±× ¿ÜÀÇ ¾î¶² ºíµµ ¿Ã ¼ºº; ÀÖ´Ù. ÀìºíÀº ÁÚ¹Ù¾ºð¾îÀÇ »ý¼º¿;¼ ÁÖ¼º¹ºÀ, ·î µé¾îº;ºÔ µÇ´Âµ¥, CommentsºíºÐÀº ÆººÇÑ ÁÖ¼ºÀÇ ÇüÀÀî/** ... */ À, ·î µé¾îº;ºÔ µÉ´Ù. Àì ÇüÀÀ´Â Æ´Áº À´Æ¿, ºÆ¼¿¿; ÀÇÇØ¼º ÁÚµ¿À, ·î ¹º¼ºº; µÉ ¼º ÀÖ´Â ÀâÀ;Àì ÀÖ´Ù. ±×, ººí , ¶Áö, ·¿;º, Àì´Â Ç¥ÇüÀÀ´Â , î;ÁöÀÇ Áº, , ¿;¿ÁÖ ´Âµ¥, Á¬·;½ºÀÇ º³¹º, ÀìÁìÆÐÀì½ºÀÇ º³¹º, ¿¬ºüÀÇ º³¹º µî ¿©º;ÁöÀÇ Áº, , ³ªÀ, ¾¾¾¾¾¾ ÁØ´Ù. ,¾à ÀS¿;ººÀì ÆíÁýµÇ¾ú´Ù, é¾Æ·¿;¿;ººº ÁÖ¼º¹ºÀì »ðÀÔµÉ ºíÀì´Ù.

/ *

* This program was generated by PLASTIC

*

* Author : Hong Gil-Dong

* Title : Geomatic Library Project

* Generated Date : 97-07-18

*/

...

/**

* Copyright(c) 1996. 4 ~ 1997. 7 All Rights Reserved.

* Dong-A University.

*/



Design Patterns

What Is a Design Pattern.

Design Pattern is a reusable solution to a common problem in software design. It is a template that can be used to solve a problem in a consistent and efficient manner. Design patterns are used to solve common problems in software design, such as how to organize a class hierarchy, how to manage a group of related objects, and how to handle data flow.

Design Patterns are divided into three main categories: Creational Patterns, Structural Patterns, and Behavioral Patterns. Creational Patterns deal with object creation, Structural Patterns deal with object composition, and Behavioral Patterns deal with object interaction. Some of the most common Design Patterns include the Singleton, Factory Method, Abstract Factory, Builder, Prototype, Singleton, Adapter, Bridge, Composite, Decorator, Facade, Flyweight, Proxy, Chain of Responsibility, Command, Interpreter, Iterator, Mediator, Memento, Observer, State, Strategy, Template Method, and Visitor.

The book "Design Patterns: Elements of Reusable Object-Oriented Software" by Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides is a classic reference for Design Patterns. It is published by Addison-Wesley.



Design Patterns

Plastic Çi¼ PatternÀ» ÀÛ¼°Çİ°í,»ðÀÔÇİ±â

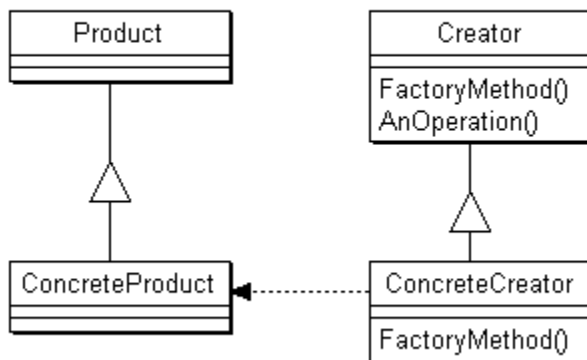
Plastic Çi¼ Â »ÇçëÀÚ ÁÀÇ PatternÀ» ÀúÀâÇİ°Â^{3a}, Àìì Á,ÀçÇİ°Â PatternÀ» ÇöÀç ,ðμ°·Î »ðÀÔÇÒ ¼ö
ÀÖ°Â ±âÉÀÌ ÀÖ°Ù. »ÇçëÀÚ ÁÀÇ PatternÀ» »ý¼°Çİ°Á,é °Ù,¥ °ÍÂ» Á!ÇÜÇÑ ¼ø¼öÇİ°Ô ÆÐÄİ Çi °ëÇÑ
°Í,,Â» ÆÍÁÝÇÑ ÈÀ Patterns, Þ°Çi¼ Save As Design PatternÀ» ¼±ÄÇİ Ç© ÀúÀâÇİ,é È®ÀâÀÚ°i .patÂİ
ÆÄÄİÂÌ »ý¼°μÇ°Ô μÈ°Ù. ±x,®°Í PatternÀ» »ðÀÔÇİ°íÀÚ ÇÑ°Ù,é Patterns, Þ°Çi¼ Insert Design
PatternÀ» ¼±ÄÇİ Ç© ÆÄÄİ,íÂ» ÁöÁÀÇØÄÖ,é ÇöÀç ,ðμ°Çi »ðÀÔÀÌ μÈ°Ù.



Design Patterns

Factory Method (Class Creational)

Design Pattern : Factory Method.



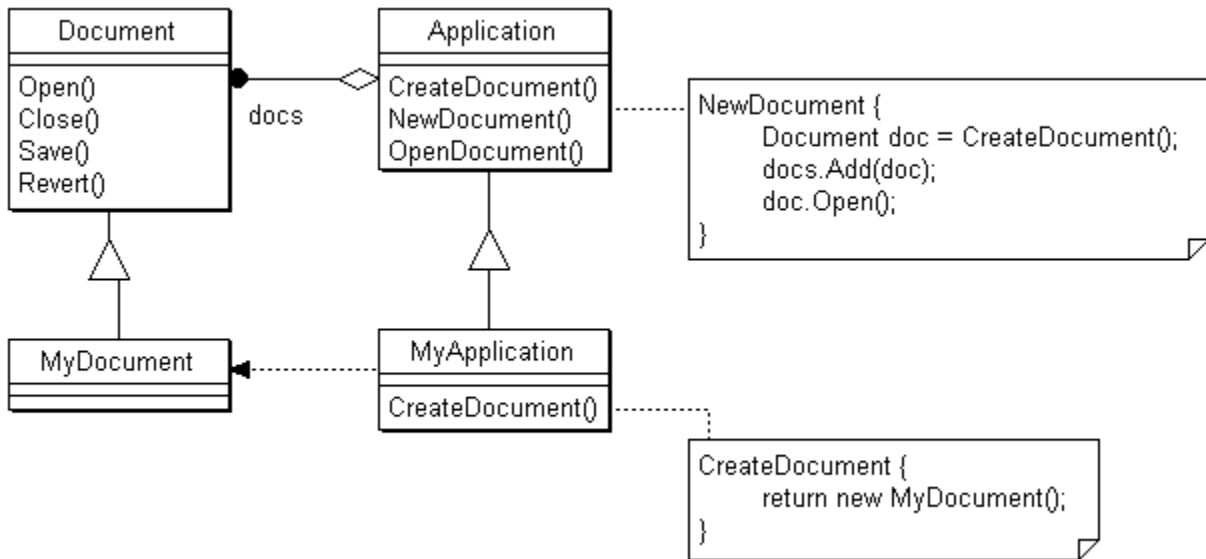
Factory Method ÆÐÄÏÄ° Virtual Constructor¶ó°íµµ ¼Ä·ÁÁ® ÀÖ´Âµ¥, °´Ä¼,¡ »ý¼°ÇÍ±â ÀŞÇÑ ÄÎÄÍÆÐÄÎ½°,¡ Ä¡°ðÇÑ´Ù. ÇİÄö,, ¼Ä¶² Ä¬·½°ÀÇ ÄÎ½°ÄÎ½°,¡ »ý¼°ÇÒ °İÄİÄö´Ä Äİ°İÄÇ ¼°êÄ¬·½°°; °áÄðÇÑ´Ù. Factory Method´Ä Ä¬·½°°; ÄÎ½°ÄÎ½°ÀÇ »ý¼°Ä» ¼°êÄ¬·½°·İ·çµµ·İ ÇÑ´Ù.

ÀŞ ±×,²ç¼ CreatorÀÇ FactoryMethod()´Ä Äß»ó, p¼Öµâ·İ½á ±×°İÄ° ¼Ä¶² °íµµ »ý¼°ÇİÄö ¼°É°İ´Ù,, ±×°İÄÇ ¼°êÄ¬·½°ÄÎ ConcreteCreatorÀÇ FactoryMethod(),¡ ÀçÁðÀÇ ÇÒÄ, ·İ½á ¼Ä¶² Ä¬·½°ÀÇ ÄÎ½°ÄÎ½°,¡ »ý¼°ÇÒ °İÄİÄö,¡ °áÄðÇÑ´Ù. ±×°İÄ° ¹Ù·İ ConcreteProduct ÀÇ ÄÎ½°ÄÎ½° Äİ´Ù. ¼ÄÆ·çİ °£ ´ÜÇÑ ÄÜ¹Ü ¼°ð¼Ä·İ´èÄæÀÇ °ñ°ÝÄ» ³aÄ,³»¼ú°İ °£´ÜÇÑ ç¹°; ÀÖ´Ù.

```
abstract class Creator {
    public abstract void FactoryMethod();
    public void AnOperation() {
        ...
    }
}
```

```
class ConcreteCreator extends Creator {
    /* It Creates Instance of ConcreteProduct Class */
    public void FactoryMethod() {
        return new ConcreteProduct();
    }
}
```

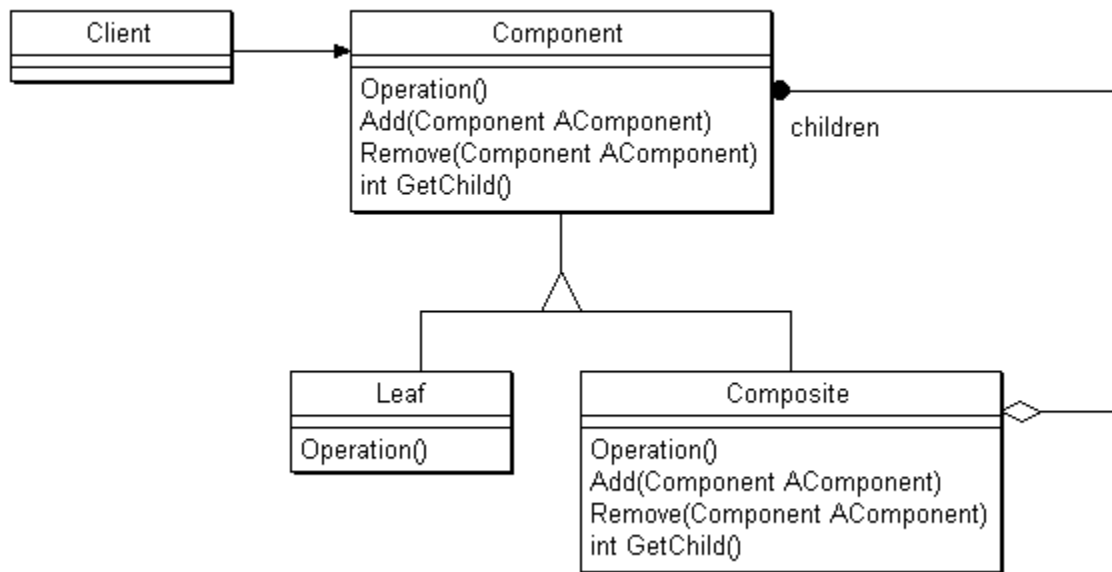
```
}  
}
```



Design Patterns

Composite (Object Structural)

Design Pattern : Composite



ο'Ã¼µéÀ» ο'Ð-1x-ÀüÃ¼ÀÇ °ÔÃþ±,Á¶,¡ ³aÃ,³»±â ÀSÇİÇ © Æ®,®±,Á¶·İ½á ±,¼°ÇÑ'Ù. Composite'Â
 Å-¶óÀİ¾ðÆ®°i °³°³ÀÇ °'Ã¼ÇÍ ±x °'Ã¼ÀÇ Á¶ÇÕ(composition)À» ±ÖÀİÇİ°Ô 'Ù·ê ¼ð ÀÖµµ·İ ÇØ ÁØ'Ù.

ÀS ±x,²Çi¼ Client'Â Component,¡ Á±±ÙÇİ°i Composite'Â ComponentÀÇ ¼°êÃ-·i½°À'Ù. OperationÀ»
 ÈËÃÇÇ ¶¶S, ,¾à ±x °'Ã¼°i LeftÀÇ Àİ½°Àİ½°¶ó,é °³°°ÀüÀÎ ¾¶¶² ÀÜ¾÷À» ¼ðÇàÇİ°i, CompositeÀÇ
 Àİ½°Àİ½°¶ó,é ±x°íÀİ Æ÷ÇÔÇİ°i ÀÖ'Â Component °'Ã¼µéÇi 'êÇÑ ,ðµç ÀÜ¾÷À» ±ÖÀİÇİ°Ô ¼ðÇàÇİ°Ô
 µÉ °íÀ'Ù. 'ÙÀ½Çi ÀÜ¹Ü¾ð¾¶ÀÇ °ñ°Ý°ú Ç¹Á'°i ÀÖ'Ù.

```

class Component {
    public void Operation() {
        ...
    }
}

```

```

class Leaf extends Component {
    public void Operation() {

```

```

        ...
    }
}

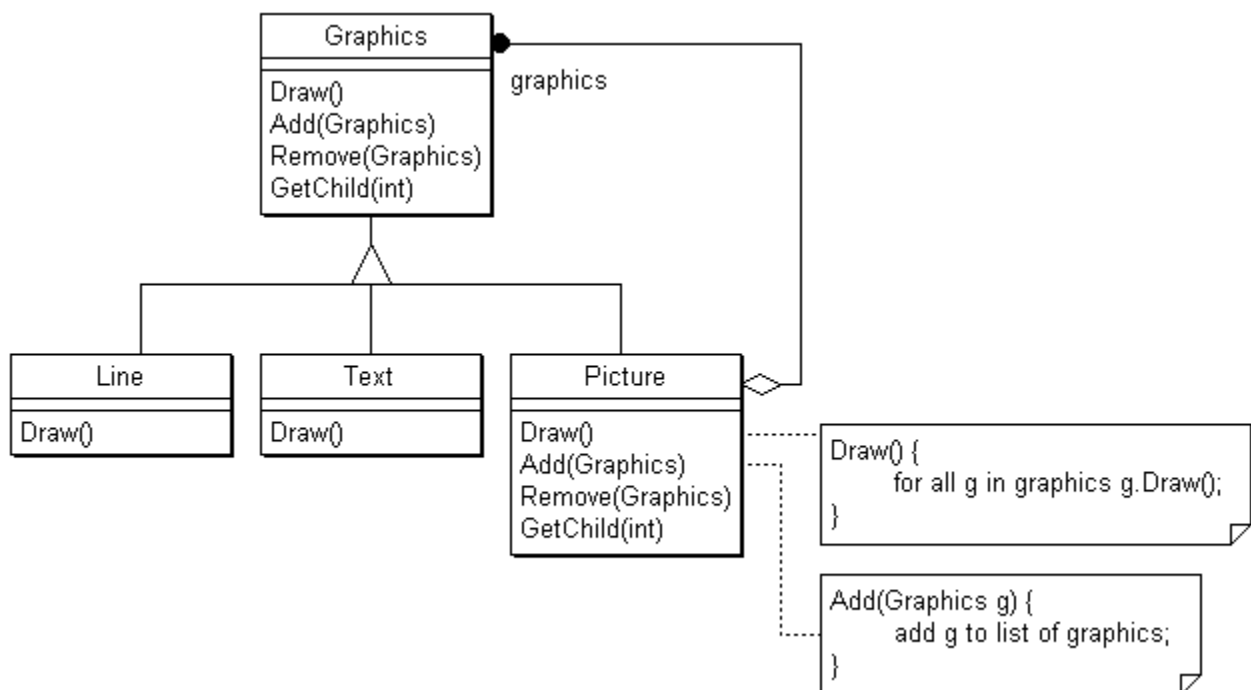
class Composite extends Component {
    List children = new List();
    public void Operation() {
        // for g all in children g.Operation();
        ...
    }

    public void Add(Component AComponent) {
        List.Add(AComponent);
    }

    public void Remove(Component AComponent) {
        List.Remove(AComponent);
    }

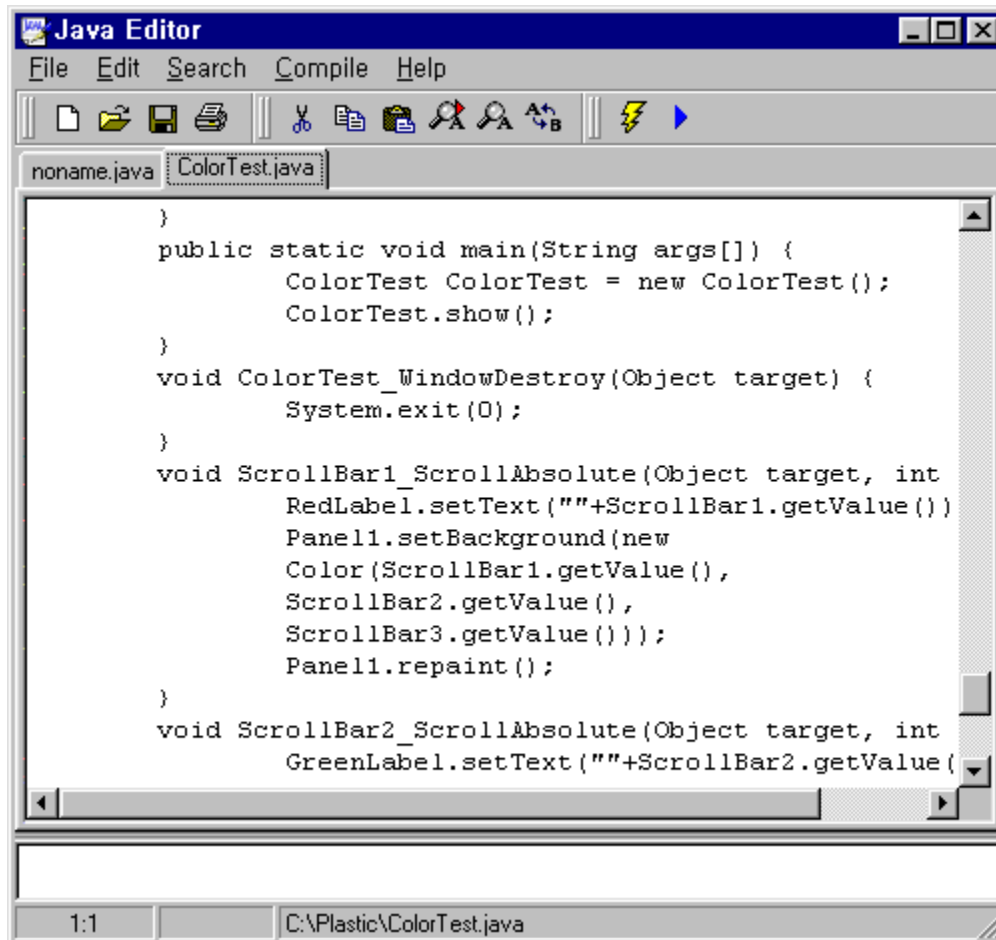
    public int GetChild() {
        return List.Count;
    }
}

```

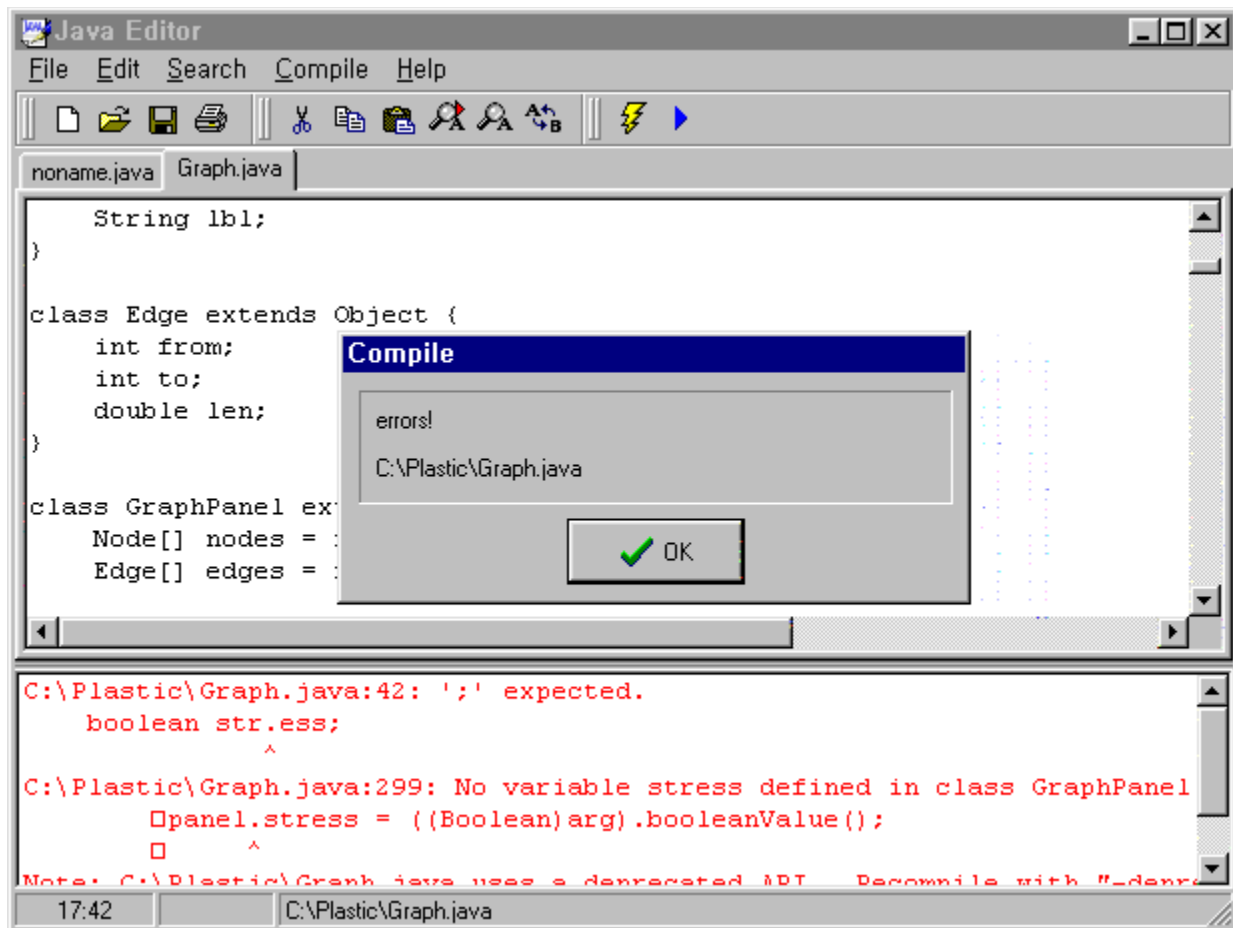


Java Editor

Java Editor



Java Editor'Ä Plastic; Ä÷ÇÔµÇ¼Ä ÄÖ'Ä ÄÜ¹Ü¼ð¼Ä, ÄŞÇÑ ;µðÄÍÄ'Ü. Äì°ÍÄ° ÄØ½°Ä® ¹@¼, ÄÍÁÝÇÒ ¼ð ÄÖµµ.Í °í¼ÈµÇ¼Ä ÄÖ'Äµ, È®ÄâÄÜ .java, .htm, .html, .txt µîÄ» °Ö.¬Ä ¼ð ÄÖ'Ü. Ä©.¬³ÄÇ ¹@¼, ÇÑ²¹ð; ÄÍÁÝÇÒ ¼ð ÄÖ'Äµ, Äì°ÍÄ° MDI(Multiple Document Interface); ±â¹ÝÇÑ ¹æ¹ýÄì ¾Ä'Ñ Page Tab» ÄìðÇÑ °ÍÄìó MDI; °ñÇØ ¹@¼ÄÜ¾÷ÄÇ ÄùÈ¬Äì °ü,Ä,Ç ÄÈ÷·Ä ´ð ÄÍ,®Ç'Ü.



Compile Error Message Window; 3a, 3»3/4i AÖ´Äµ¥ çimðÆ® À©µµçì 1Ü·Î ¾Æ·çì »ýÜ¼ µð¹ð±èµµ Æí,®ÇĩÔ ÇÒ ¼ö ÀÖµµ·µµçÍÁØÜ.

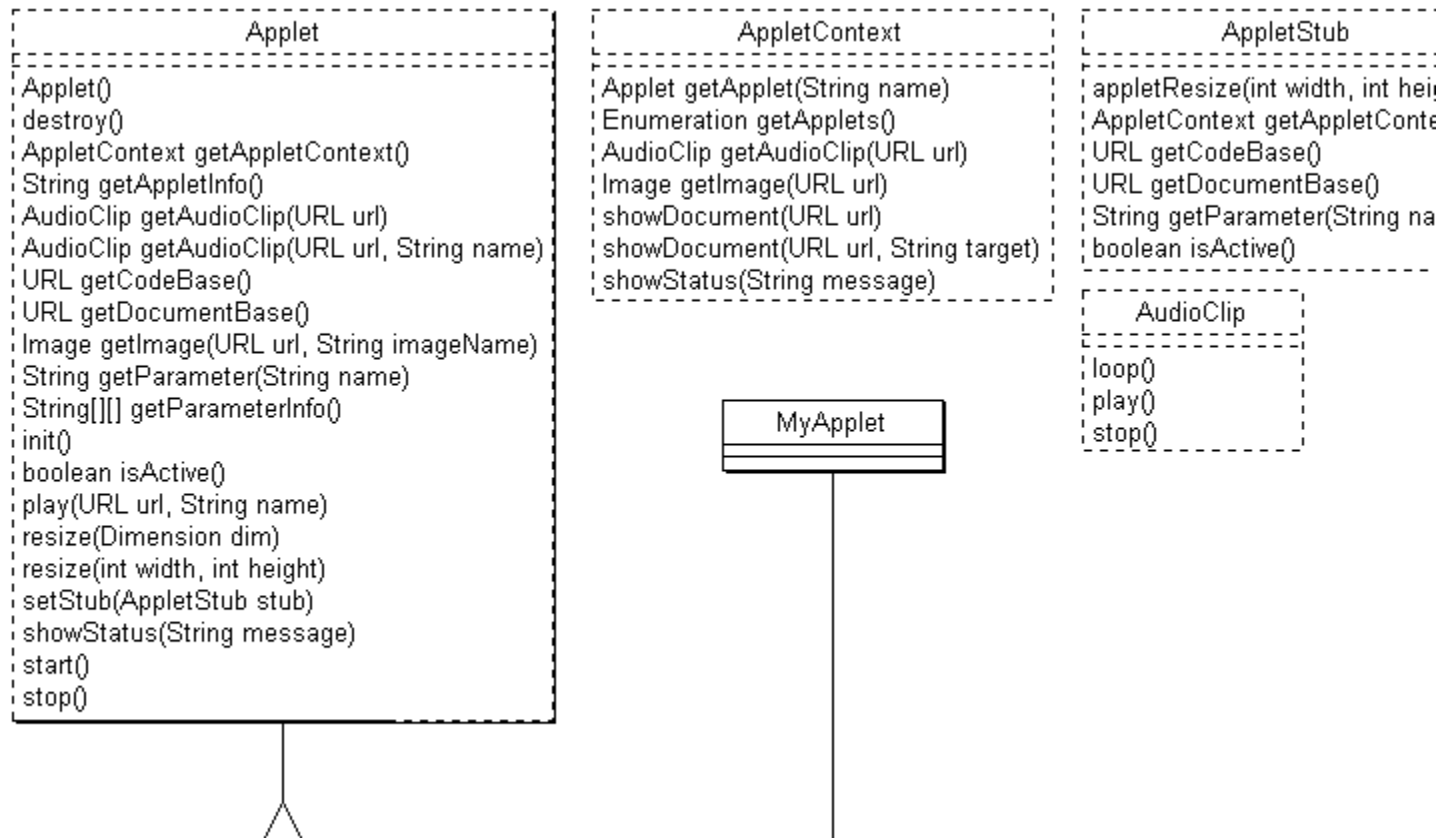
ÀÚ¹Ü çimðÁí´Â ±âÁ,ÀÇ çimðÁíçí °ñ½ÁÇÑ ,p´,í °íÁó´Äµ¥, File, Edit, Search, Compile, Help ÁµµÀÇ ,p´,í °íÁó´í AÖ´í, °ü,¥ ÄÜ¾÷À» ÀŞÇØ Speed ButtonsÀ» ÀÚÁÖ ¾²´Â ,p´°çì ÇÒ´çÇØ ¾ð¾Ö´Ü. ½°ÇÇÆ® ¹ðÆ°À° çpÂÊçì¼ Ä÷·Ê´è·Î, New, Open, Save, Print, Cut, Copy, Paste, Find, Find Next, Replace, CompileÀì´Ü. ±x,®´í ÆíÁý À©µµçìÀÇ ÁÇçì´Â ÆÄÄìÀÇ Àì,ŞÀì 3a, 3´Ü. ¾Æ·çìÀÇ »óÄÄ¹Üçì´Â ÇöÀç ÆíÁýÁßÀì ÷÷ÀÇ Column°ú Row,í 3a, 3a, ç, ±x çÀ,¥ÂÊçì´Â °°æÀÇ ç©´í, Modified¶ó´Â 1®ÀÜ,í ÆèÇØ¼ 3a, 3»ç, ±x ç·çì´Â ÆíÁýÁßÀì ÆÄÄìÀÇ çÌÄüÇÑ °æ·î; 3a, 3´Ü.



ÆíÁý ÁßÀî Æù¹ù ¾ð¾¼,í ÄÄÆÄÄî ÇïíÆÁÚ ÇÑ´Ù,é ´ÜÄö [F8]Ä» ´©,£°Ä³ª CompileÄ» ¼±ÄÄÇí,é µÊ´Ù.
±x·±µ¥ Àì, °Ô Çìâ ÄìÄüç; PlasticÄÇ Option,þ°ÄÇ Environment Pageç;¼ ÄÜ¹ùÄÄÆÄÄî´ÄÇ °æ·î,í
¹Ýµå½Ä ¼³Ä¾ÇØÄÖ¾¼¾¼ß ÇÑ´Ù. ÄÄÆÄÄîÄì ³;³µÄ» ¶§,,¾à ç;´°; ÄÖ´Ù,é ¾Æ·îÄÊç; »ð·îçî
,®½°Æ®¹Ü½°°; ³ªÄ,³ª,é¼ °ÖÄ° »öÄ,·î ç;´°; ¾Ê·ÁÄÜ°íÄì´Ù. ±x·,Ä; ¾Ê´Ù,é Compile Completely¶ó·Ä
,þ½ÄÄö°; ³ªçÄ,é¼ ÄÄÆÄÄîç; ¼°°øÇß´Ù´Ä, þ½ÄÄö°,í °,ç©ÄÜ °íÄì´Ù.

±Ù²Ã ¹× »ð»6¹Ù²Ù±âÀ» ¹Ù²Ù°í ½í¹Ù,é PlasticÀÇ OptionÀÇ Editor Page¿;¼±×°íÀ» ¼³À±ÇÒ ¼ð°; ÀÖ´Ù.
±×,®°í ÇÁ,°Æ®·î Ââ·ÃÀ» ¿øÇÑ´Ù,é File | Print...,! ¼±ÀÃÇĬ,é µĒ´Ù. ÀÚ¹Ù ¿;µðÁĬ Â µĬ,³ÂùÃĬ
¾ÖÇÃ,®ÆĒÀ!¼ÇÀ! ¾Æ´î±â ¶S¹®¿;±×°íÀÇ ÂÇ±â¿Ĭ ÀSĬ;±×,®°í ±Ù²Ã ¹× »ð»óÁĬ ,ðµĬ ´ÙÀ½²ð
½ÇÇà¶S¿;µµ À´ð°; µÇ¹Ç·Ĭ. ÀÚ½Æ¿;°Ö ÆĬ,®ÇÑ ÀSĬ;¿Ĭ ÂÇ±â,! ¼³À±ÇÇ µĬ é ÁÁ´Ù.

Examples Applet Template

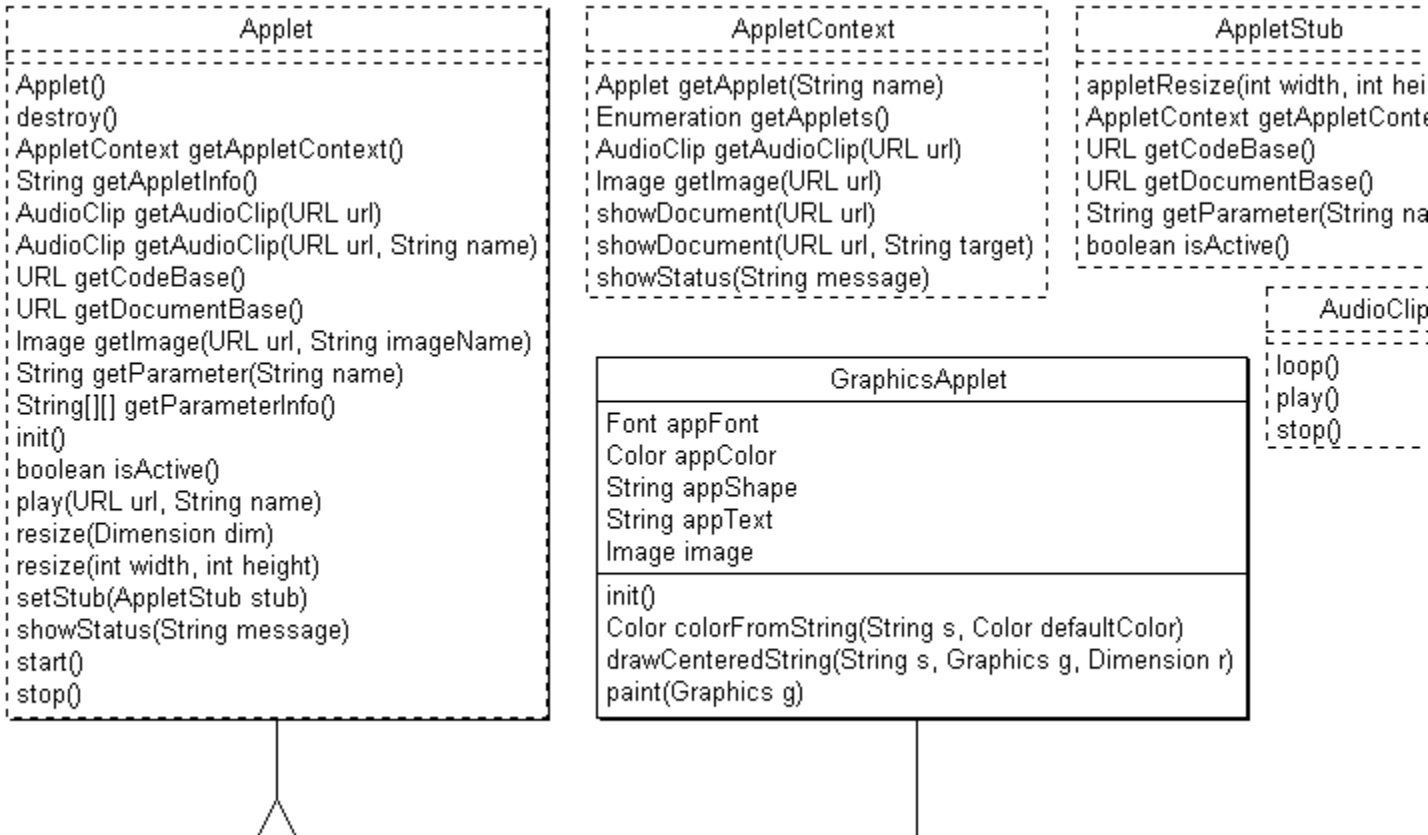


PlasticÀ° ÀÚ¹ÙÀÇ Applet³¹βζι μμζδÀì μÉ ¼ö ÀÖ´À ¾ÖÇÃ,´ ÅÜÇÃ,´(AppletTemplate.pla),! Áì°øÇì°í ÀÖ´Ù. Àì°íÀ° ¾ÖÇÃ,´ζι °ü·ÃμÈ Å¬·½º¹× ÀÎÁÍÆÐÀì½ºζí ±×°íÀì °íÄö°í ÀÖ´À ,Þ¼ÖμâμéÀ» °,ζ©ÄÖ°í ÀÖ¾î¼ ¾ÖÇÃ,´³¹βζι ,¹À° μμζδÀ» ÁØ´Ù. Applet, AppletContext, AppletStub, AudioClipÀ° java.appletÆÐÄ°Áöζι Æ÷ÇÔμÇ¾î ÀÖ´À °íμéÀì´Ù. ±×·½¼ Package Window,! ζ¬ ÈÄ, Import°í°ÐÄ» °,é java.applet.*° í »ðÀÔμÇ¾î ÀÖ´À °íÀ» °¼ ¼ö ÀÖÀ» °íÀì´Ù. ÀÚ¹ÙÀÇ ¾ÖÇÃ,´À° AppletÅ¬·½º·î°íÁÍ »ó¼ÓÀ» ¹Þ¼Æ¾β ÇÑ´Ù. MyApplet°í°Ðζι ÀÚ½ÄÀì ζøÇì´À ¾ÖÇÃ,´À,î Ãβ°í, È©ÄâÇì±â,, Çì,é μÉ´Ù.



Examples

Applet Demo



Applet Demo
MyAppletApplet GraphicsApplet
Java Code Generator
Java Source
GraphicsApplet.class
Applet Demo



Examples

Applet Demo Source

```
/*
 * This program was generated by PLASTIC
 *
 * Author : Min-Kyu Lee
 * Title : GraphicsApplet
 * Generated Date : 97-07-22
 */

package GraphicsApplet;

import java.awt.*;
import java.applet.*;

/**
 * This example is from :
 *
 * HOOKED ON JAVA
 * Arthur van Hoff, Sami Shaio, Orca Starbuck.
 * Addison-Wesley Publishing Company.
 *
 * An applet that draws different shapes depending
 * on the configuration of the HTML tag. It
 * accepts the following attributes in addition
 * to the standard attributes:
 *
 * SHAPE - one of "Line", "Rectangle", "String" or "Image"
 * COLOR - which foreground color to use.
 * TEXT - If SHAPE is "String" this is the text to draw.
 * IMAGE - If SHAPE is "Image" this is the image to draw.
 */
```



```

public class GraphicsApplet extends Applet {
    Font appFont;
    Color appColor;
    String appShape;
    String appText;
    Image image;
    public void init() {
        appFont = new Font("Helvetica", Font.BOLD, 14);
        String arg = getParameter("COLOR");
        if (arg != null) {
            appColor = colorFromString(arg, Color.red);
        }
        appShape = getParameter("SHAPE");
        if (appShape == null) {
            appShape = "Line";
        }
        appText = getParameter("TEXT");
        if (appText == null) {
            appText = "Graphics";
        }
        arg = getParameter("IMAGE");
        if (arg != null) {
            image = getImage(getDocumentBase(), arg);
        }
    }
    public Color colorFromString(String s, Color defaultColor) {
        Integer i;
        try {
            i = Integer.valueOf(s, 16);
            return new Color(i.intValue());
        } catch (NumberFormatException e) {
            return defaultColor;
        }
    }
    public void drawCenteredString(String s, Graphics g, Dimension r) {
        FontMetrics fm = g.getFontMetrics(appFont);

```

```

        . drawString(s,
                    (r.width - fm.stringWidth(s)) / 2,
                    (r.height - fm.getHeight()) / 2);
    }
    public void paint(Graphics g) {
        Dimension r = size();
        g.setColor(appColor);
        if (appShape.equalsIgnoreCase("line")) {
            g.drawLine(0, 0, r.width, r.height);
        } else if (appShape.equalsIgnoreCase("rectangle")) {
            g.drawRect(0, 0, r.width - 1, r.height - 1);
        } else if (appShape.equalsIgnoreCase("image")) {
            g.drawImage(image, 0, 0, r.width, r.height, this);
        } else if (appShape.equalsIgnoreCase("string")) {
            g.setFont(appFont);
            drawCenteredString(appText, g, r);
        }
    }
}

```

Examples

Applet Demo Result

이 페이지는 Netscape Navigator, Internet Explorer 등의 Java Appletviewer로 실행됩니다. 이 페이지는 Plastic 1.0 for Java로 생성된 GraphicsApplet.html입니다. 이 페이지는 Netscape 4.0 이상의 HTML 4.0으로 생성되었습니다.



```
<HTML>
<HEAD>
  <TITLE>GraphicsApplet Demo</TITLE>
</HEAD>
<BODY>
```

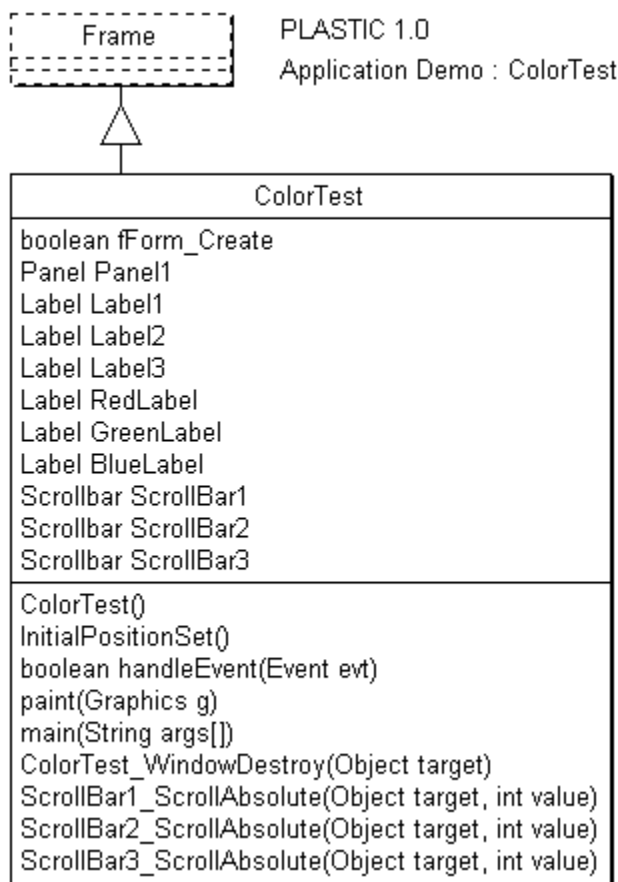
```
<H1>GraphicsApplet</H1>
This Applet was generated by Plastic 1.0 for Java.<br>
Author : Min-Kyu Lee<br>
Title : GraphicsApplet<br>
Generated Date : 97-07-21<br>
<HR>
<APPLET code="GraphicsApplet.class" width=350 height=50>
  <PARAM NAME="SHAPE" VALUE="Rectangle">
  <PARAM NAME="COLOR" VALUE="ffffff">
</APPLET>
<BR>
<APPLET code="GraphicsApplet.class" width=350 height=50>
  <PARAM NAME="SHAPE" VALUE="String">
  <PARAM NAME="COLOR" VALUE="ff0000">
  <PARAM NAME="TEXT" VALUE="Object-Oriented Modeling Tool - Plastic 1.0">
</APPLET>
<BR>
<APPLET code="GraphicsApplet.class" width=350 height=50>
  <PARAM NAME="SHAPE" VALUE="String">
  <PARAM NAME="COLOR" VALUE="0000ff">
  <PARAM NAME="TEXT" VALUE="Java AppletDemo (using AppletTemplate.pla)">
</APPLET>
</BODY>
</HTML>
```



Examples

Application Demo

ÀÚ¹Ù¿¼ ¼ÖÇÃ,®ÄÉÀ¼ÇÀ» ÀŞÇŃ ¿¹Áµµ Á¹°øµÉ´Ù. ApplicationDemo.pla´Â Red, Green, BlueÀÇ
Á¶ÁýÀ» ÄëÇ¿¿© ±×°íµéÀÇ Á¶ÇÖÀ¿ „µé¼î³»´Â »ð»óÀ» °¿¿©ÁÖ´Â ÇÁ·Î±×·¥À¿Ù. À¿ÍÀ» ¼ÖÇÃ,´À¿ ¼⁄Æ
´¿¹Ç·Î À¥°é¶¿ó¿ìÀú³ª, appletviewer¿¼¿¹Â ½ÇÇàÇØ ¼⁄¼ö ¼⁄ø°¿, JDKÀÇ java µ¿ÀÇ ÇÁ·Î±×·¥À»
À¿¿ëÇØ¼⁄ß ÇŃ´Ù.



À¿ ¼ÖÇÃ,®ÄÉÀ¼ÇÀ» À©µµ¿ì,¿ ¿¼î³⁄ß Ç¿¹Ç·Î FrameÀ,·Î¶ÍÁ »ó¼⁄ÓÀ» ¹b¼⁄Ö´Ù. ±×,®°¿ mainÀ¿
½ÇÇàµÇ°Ô µÉ´Ù. handleEvent´Â »ç¿ëÄÚ°¿ ÇÒ ¼⁄ö ÀÖ´Â Çàµ¿¿¿¿ëÇŃ À¿°¥Æ® Ä³,®,¿ Ç¿¿¿ ÀÖ°¿, paint
´Â È,é¿¿ ±×·ÁÁú ¶§ °Ö,®´Â ,b¼⁄ÒµàÀ¿Ù. ¿©±â¼¿Â ¿,¿¹¿ö ÄÄÆ®·Ń(Control)µéÀ» À¿ëÇ¿¿¿ ÀÖ´Âµ¥,
Panel, Label, Scrollbar µ¿À¿ ±×°íµéÀ¿Ù.



Examples

Application Demo Source

```
/*
 * This program was generated by PLASTIC
 *
 * Author : Min-Kyu Lee
 * Title : RGB Color Test
 * Generated Date : 97-07-22
 */

import java.awt.*;

/**
 * This examples is "BLUETTE 0.62" Exmaple.
 * By Hyoung-Gook Kim.
 */

public class ColorTest extends Frame {
    boolean fForm_Create;
    Panel Panel1;
    Label Label1;
    Label Label2;
    Label Label3;
    Label RedLabel;
    Label GreenLabel;
    Label BlueLabel;
    Scrollbar ScrollBar1;
    Scrollbar ScrollBar2;
    Scrollbar ScrollBar3;
    ColorTest() {
        // Frame Initialization
        setForeground(Color.black);
        setBackground(Color.lightGray);
    }
}
```

```

setFont(new Font("Dialog",Font.BOLD,12));
setTitle("ColorTest");
setLayout(null);

// Component Initialization
Panel1 = new Panel();
Panel1.setLayout(null);
Panel1.setForeground(Color.black);
Panel1.setBackground(Color.black);
Panel1.setFont(new Font("Dialog",Font.BOLD,12));
Label1 = new Label("Red :",Label.LEFT);
Label1.setFont(new Font("Dialog",Font.BOLD,12));
Label2 = new Label("Green :",Label.LEFT);
Label2.setFont(new Font("Dialog",Font.BOLD,12));
Label3 = new Label("Blue :",Label.LEFT);
Label3.setFont(new Font("Dialog",Font.BOLD,12));
RedLabel = new Label("0",Label.LEFT);
RedLabel.setFont(new Font("Dialog",Font.BOLD,12));
GreenLabel = new Label("0",Label.LEFT);
GreenLabel.setFont(new Font("Dialog",Font.BOLD,12));
BlueLabel = new Label("0",Label.LEFT);
BlueLabel.setFont(new Font("Dialog",Font.BOLD,12));
ScrollBar1 = new Scrollbar(Scrollbar.HORIZONTAL);
ScrollBar1.setValues(0,22,0,255);
ScrollBar1.setBackground(Color.lightGray);
ScrollBar2 = new Scrollbar(Scrollbar.HORIZONTAL);
ScrollBar2.setValues(0,22,0,255);
ScrollBar2.setBackground(Color.lightGray);
ScrollBar3 = new Scrollbar(Scrollbar.HORIZONTAL);
ScrollBar3.setValues(0,22,0,255);
ScrollBar3.setBackground(Color.lightGray);

// Add()s
add(ScrollBar3);
add(ScrollBar2);
add(ScrollBar1);

```

```

        add(BlueLabel);
        add(GreenLabel);
        add(RedLabel);
        add(Label3);
        add(Label2);
        add(Label1);
        add(Pane11);

        fForm_Create = true;
    }

    void InitialPositionSet() {
        reshape(191,107,283,229);
        Pane11.reshape(11,29,262,95);
        Label1.reshape(26,143,48,16);
        Label2.reshape(13,167,64,16);
        Label3.reshape(22,192,54,16);
        RedLabel.reshape(76,143,55,16);
        GreenLabel.reshape(77,168,50,16);
        BlueLabel.reshape(77,194,54,16);
        ScrollBar1.reshape(142,141,121,16);
        ScrollBar2.reshape(142,168,121,16);
        ScrollBar3.reshape(142,195,121,16);
        fForm_Create = false;
    }

    public boolean handleEvent(Event evt) {
        // handleEvent()
        if (evt.id == Event.WINDOW_DESTROY && evt.target == this)
            ColorTest_WindowDestroy(evt.target);
        else if (evt.id == Event.SCROLL_ABSOLUTE && evt.target == ScrollBar1)
            ScrollBar1_ScrollAbsolute(evt.target,
                                     ((Scrollbar) (evt.target)).getValue());
        else if (evt.id == Event.SCROLL_LINE_DOWN && evt.target == ScrollBar1)
            ScrollBar1_ScrollAbsolute(evt.target,
                                     ((Scrollbar) (evt.target)).getValue());
        else if (evt.id == Event.SCROLL_LINE_UP && evt.target == ScrollBar1)
            ScrollBar1_ScrollAbsolute(evt.target,

```



```

        ((Scrollbar) (evt.target)).getValue());
else if (evt.id == Event.SCROLL_PAGE_DOWN && evt.target == ScrollBar1)
    ScrollBar1_ScrollAbsolute(evt.target,
        ((Scrollbar) (evt.target)).getValue());
else if (evt.id == Event.SCROLL_PAGE_UP && evt.target == ScrollBar1)
    ScrollBar1_ScrollAbsolute(evt.target,
        ((Scrollbar) (evt.target)).getValue());
else if (evt.id == Event.SCROLL_ABSOLUTE && evt.target == ScrollBar2)
    ScrollBar2_ScrollAbsolute(evt.target,
        ((Scrollbar) (evt.target)).getValue());
else if (evt.id == Event.SCROLL_LINE_DOWN && evt.target == ScrollBar2)
    ScrollBar2_ScrollAbsolute(evt.target,
        ((Scrollbar) (evt.target)).getValue());
else if (evt.id == Event.SCROLL_LINE_UP && evt.target == ScrollBar2)
    ScrollBar2_ScrollAbsolute(evt.target,
        ((Scrollbar) (evt.target)).getValue());
else if (evt.id == Event.SCROLL_PAGE_DOWN && evt.target == ScrollBar2)
    ScrollBar2_ScrollAbsolute(evt.target,
        ((Scrollbar) (evt.target)).getValue());
else if (evt.id == Event.SCROLL_PAGE_UP && evt.target == ScrollBar2)
    ScrollBar2_ScrollAbsolute(evt.target,
        ((Scrollbar) (evt.target)).getValue());
else if (evt.id == Event.SCROLL_ABSOLUTE && evt.target == ScrollBar3)
    ScrollBar3_ScrollAbsolute(evt.target,
        ((Scrollbar) (evt.target)).getValue());
else if (evt.id == Event.SCROLL_LINE_DOWN && evt.target == ScrollBar3)
    ScrollBar3_ScrollAbsolute(evt.target,
        ((Scrollbar) (evt.target)).getValue());
else if (evt.id == Event.SCROLL_LINE_UP && evt.target == ScrollBar3)
    ScrollBar3_ScrollAbsolute(evt.target,
        ((Scrollbar) (evt.target)).getValue());
else if (evt.id == Event.SCROLL_PAGE_DOWN && evt.target == ScrollBar3)
    ScrollBar3_ScrollAbsolute(evt.target,
        ((Scrollbar) (evt.target)).getValue());
else if (evt.id == Event.SCROLL_PAGE_UP && evt.target == ScrollBar3)
    ScrollBar3_ScrollAbsolute(evt.target,

```

```

        ((Scrollbar) (evt.target)).getValue());
        return super.handleEvent(evt);
    }

    public void paint(Graphics g) {
        if (fForm_Create) InitialPositionSet();
    }

    public static void main(String args[]) {
        ColorTest ColorTest = new ColorTest();
        ColorTest.show();
    }

    void ColorTest_WindowDestroy(Object target) {
        System.exit(0);
    }

    void ScrollBar1_ScrollAbsolute(Object target, int value) {
        RedLabel.setText(""+ScrollBar1.getValue());
        Panell1.setBackground(new
            Color(ScrollBar1.getValue(),
                ScrollBar2.getValue(),
                ScrollBar3.getValue()));
        Panell1.repaint();
    }

    void ScrollBar2_ScrollAbsolute(Object target, int value) {
        GreenLabel.setText(""+ScrollBar2.getValue());
        Panell1.setBackground(new
            Color(ScrollBar1.getValue(),
                ScrollBar2.getValue(),
                ScrollBar3.getValue()));
        Panell1.repaint();
    }

    void ScrollBar3_ScrollAbsolute(Object target, int value) {
        BlueLabel.setText(""+ScrollBar3.getValue());
        Panell1.setBackground(new
            Color(ScrollBar1.getValue(),
                ScrollBar2.getValue(),
                ScrollBar3.getValue()));
        Panell1.repaint();
    }

```




Examples

Application Demo Result

»ý¼°µÈ ÀÚ¹Ù ¼Ö½°ÄÚµâ,¡ Java EditorÀÇ Compile ±â´ÉÄ,·Î ÄÄ/ÆÄÀiÇĬ,é ColorTest.class°i »ý¼°Ài µÇ
´Âµ¥, Ài°ÍÄ» JDKÀÇ java·Î ½ÇÇàÇØ °¼¼¼ö ÀÖ´Ù. (¿¹, c:\jdk1.1.1\bin\java ColorTest) ±x°ÍÄ» ½ÇÇàÇĬ,é
´ÙÄ½ ±x,²°ú °°ÀĬ »ð·Î¿Ĭ À©µµ¿Ĭ°i ¿,®,é¼¼ ¾ÖÇÃ,®ÄÉÀi¼ÇÀi ½ÇÇàµÈ´Ù.

