

## Anexo 14 - Instalando e configurando o ObjectAid e UML Doclet ( ex yDocs )

Gerando o modelo de Classes via engenharia reversa.

Para este trabalho estaremos utilizando duas ferramentas distintas:

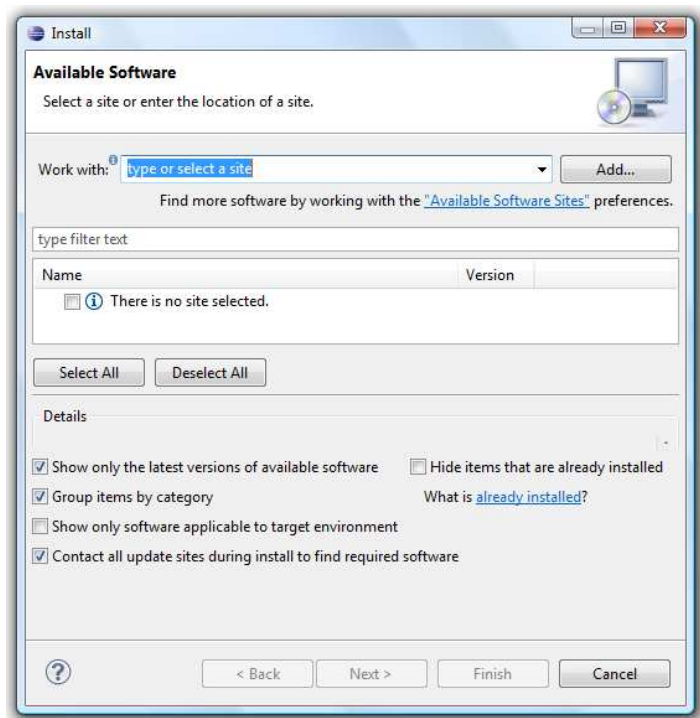
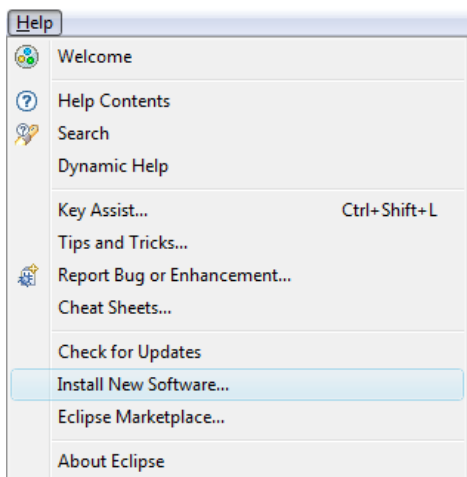
- Para gerar o modelo global do projeto: utilizaremos um plugin do Eclipse, o ObjectAid (<http://www.objectaid.com/class-diagram> )
- Para documentar as classes: utilizaremos um plugin do JavaDoc, o UML Doclet (ex yDoc) da yWorks ([http://www.yworks.com/en/products\\_ydoc.html](http://www.yworks.com/en/products_ydoc.html)).

### 14.1 - Instalando os produtos

#### 14.1.1 - ObjectAid

A instalação se dá pelo eclipse.

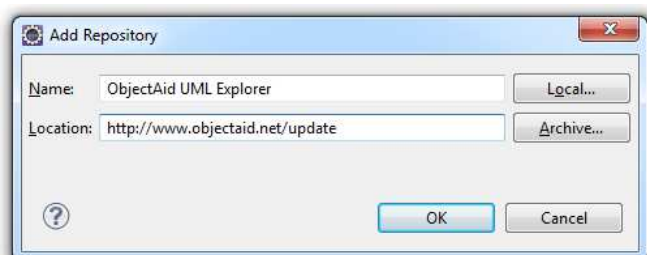
**Help > Install New Software ...** Na página "Available Software" pressione **Add ...**



**Na página "Add Repository" entre com as informações abaixo e pressione OK**

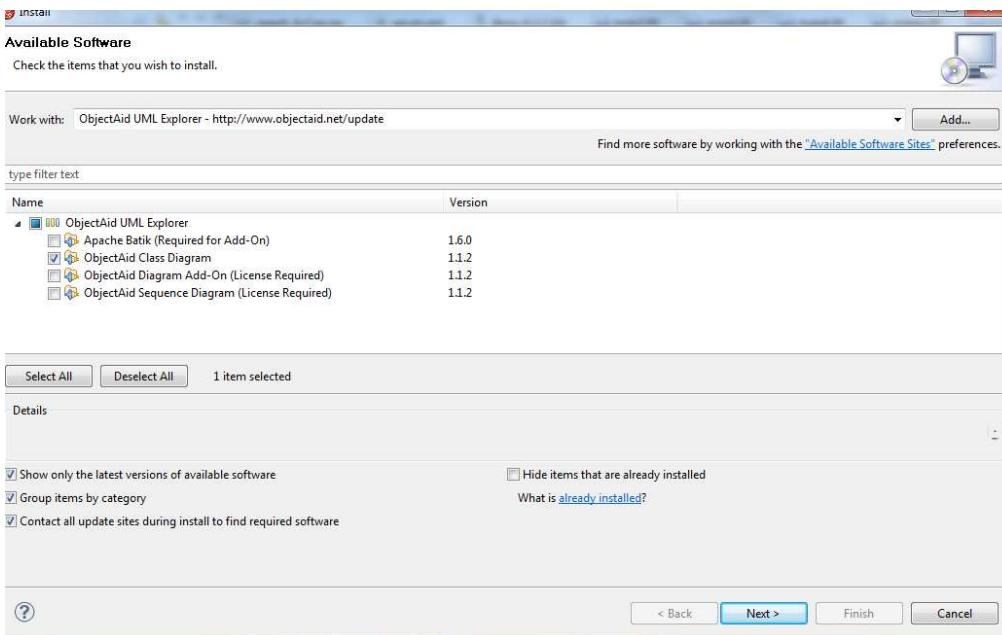
Name: ObjectAid UML Explorer

URL: <http://www.objectaid.net/update>



Serão mostrados todos plug-ins do objectAid.

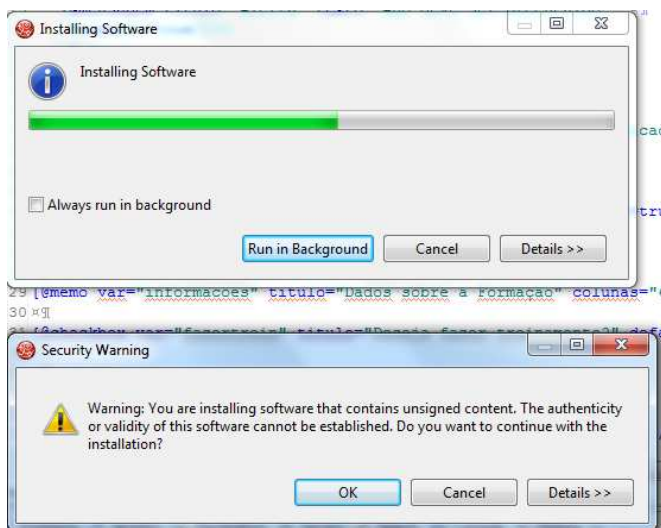
Selecione ObjectAid Class Diagram e pressione Next.



Aceite o License Agreement.

O processo de instalação dará início.

Um aviso ( ... contains unsigned content ... ) aparecerá, simplesmente pressione OK. O Plug-in será instalado e solicitará que o IDE seja restartado.



### 14.1.2 - UML Doclet (ex yDoc)

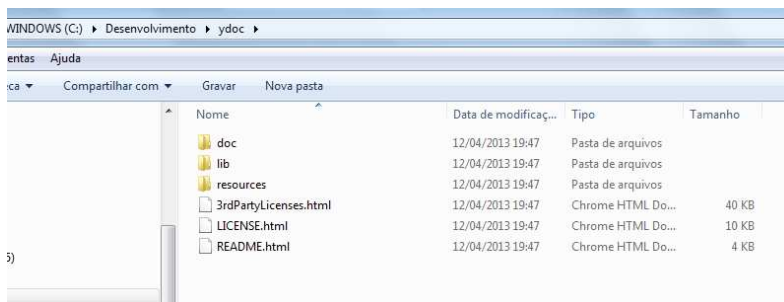
O UML Doclet é um plug-in para o javadoc da empresa yWorks. Existe uma versão paga e uma free (edição comunidade) que será utilizada.

Realizar o download do plug-in em [http://www.yworks.com/en/products\\_ydoc.html](http://www.yworks.com/en/products_ydoc.html)

Download	Buy
Download the yWorks UML Doclet Community Edition: yWorks UML Doclet Community Edition (JDK 1.5.0 / 1.6.0 / 1.7.0) yWorks UML Doclet Community Edition (J2SDK 1.4.x)	Buy the yWorks UML Doclet Professional Edition (licenses are sold by our sales partner share-it!): yWorks UML Doclet Professional Edition

ydoc e unzipar o download para

lá.

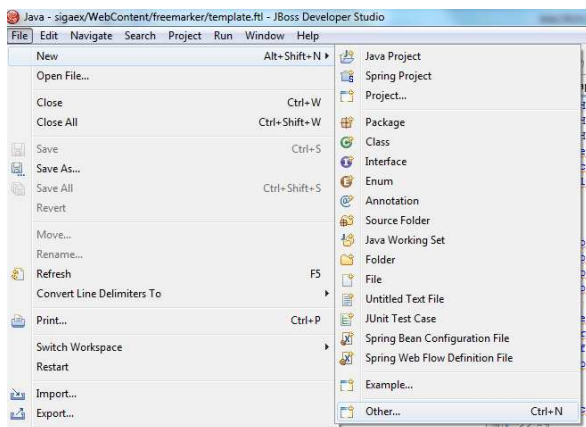


O UML Doclet será rodado pelo javadoc na interface da IDE como veremos a seguir.

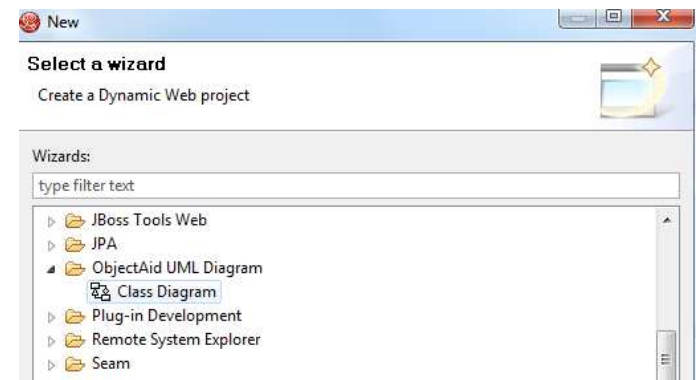
## 14.2 - Utilizando os produtos

### 14.2.1 - ObjectAid

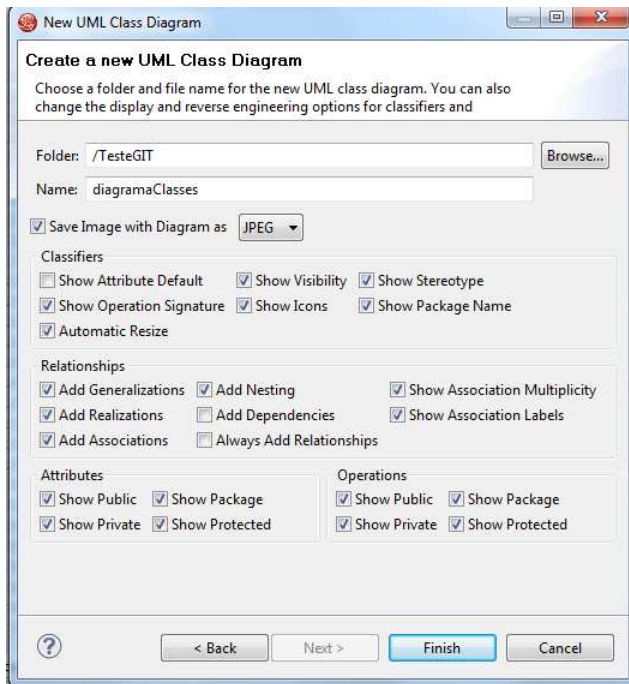
**File > New > Other**



**Selecionar ObjectAid / Class Diagram**

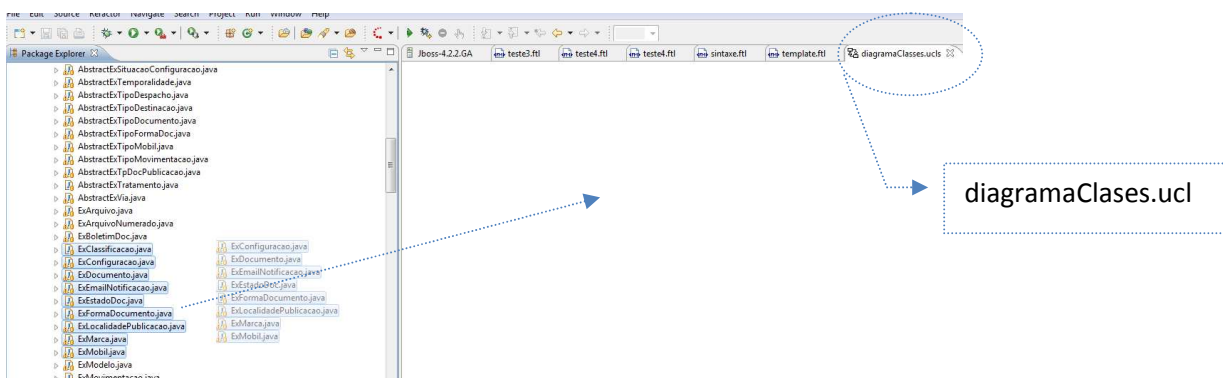


Selecionar o folder (caminho) onde residirá o diagrama a ser gerado. Fornecer um nome para o diagrama. Selecionar o formato da imagem e marcar as opções desejadas.



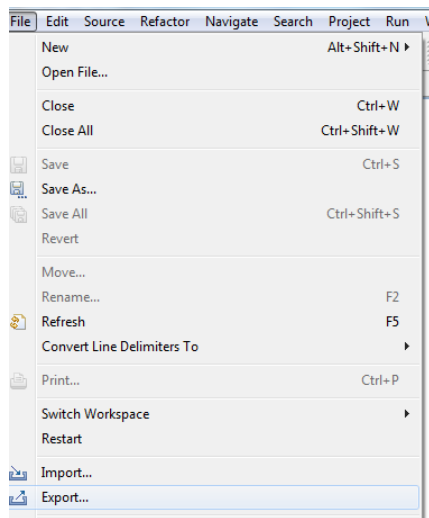
Será criada uma aba na IDE com o nome\_do\_diagrama\_fornecido.ucl (no caso, diagramaClasses.ucl).

Agora, navegue nos projetos / pacotes, selecione as classes e arraste-as para a aba recém criada. O diagrama será gerado.

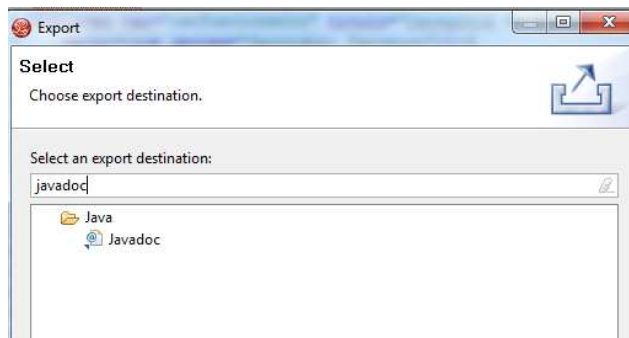


### 14.2.2 - UML Doclet

#### File > Export



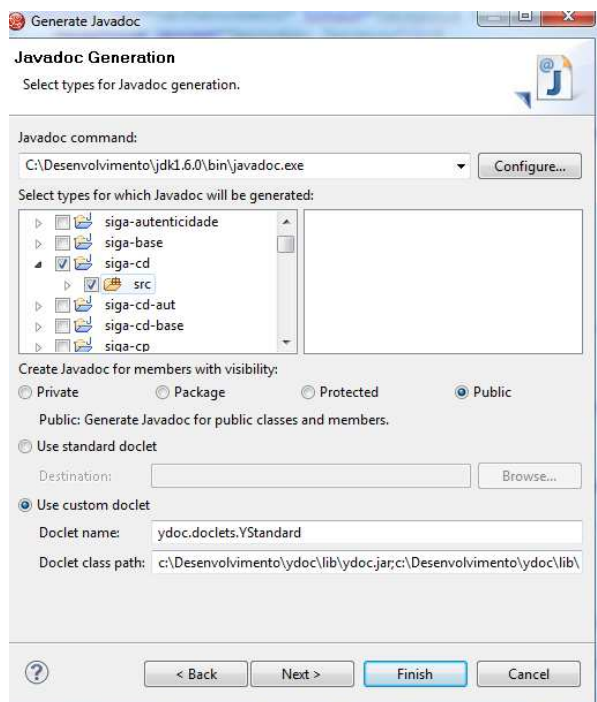
#### Selecionar Javadoc



Selecionar o(s) projeto(s) e/ou pacote(s). Marcar "Use custom doclet" e informar também:

- **Doclet name:** ydoc.doclets.YStandard
- **Doclet class path:** c:\Desenvolvimento\ydoc\lib\ydoc.jar;  
c:\Desenvolvimento\ydoc\lib\styleed.jar;  
c:\Desenvolvimento\ydoc\resources;

#### Clicar em Next

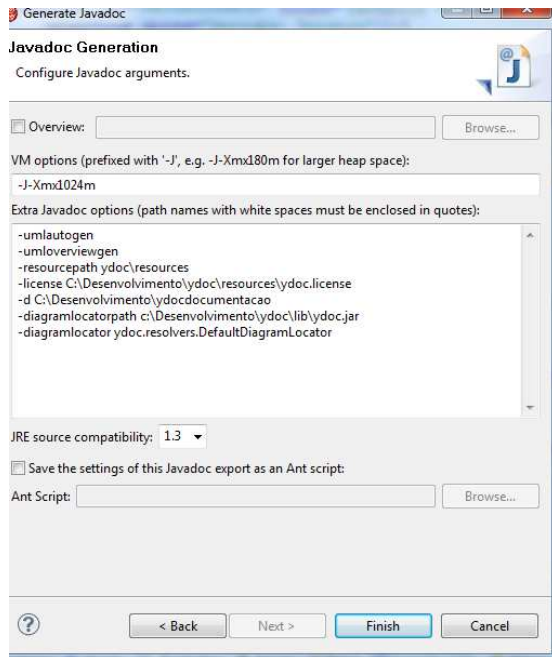


Na tela seguinte, fornecer as seguintes informações:

**VM Options:** -J-Xmx1024m

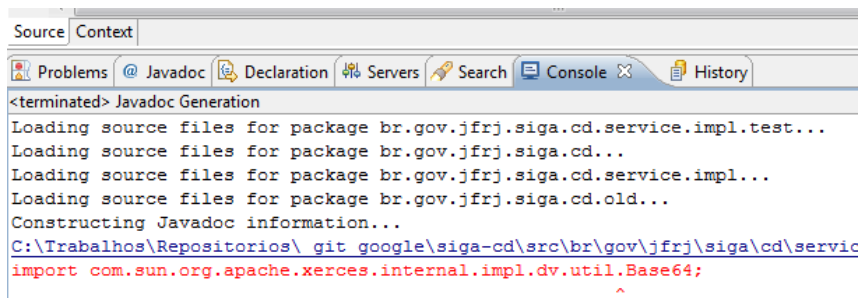
### Extra Javadoc options:

```
-umlautogen
-umloverviewgen
-resourcepath ydoc\resources
-license C:\Desenvolvimento\ydoc\resources\ydoc.license
-d C:\Desenvolvimento\ydocdocumentacao
-diagramlocatorpath c:\Desenvolvimento\ydoc\lib\ydoc.jar
-diagramlocator ydoc.resolvers.DefaultDiagramLocator
```



### Clicar em Finish.

Na console será exibido o progresso da documentação.



### Explicação de alguns parâmetros que podem ser passados ao UMLDoclet

#### 14.2.2.1 - Tipos de Diagrama

##### -umlgen

UML diagrams will be created and embedded for all documented files with an @y.uml tag. @y.uml may be used in type, package, and overview documentation.

O @y.uml é uma anotação UML Doclet. Por exemplo, se uma classe A possui esta anotação e uma classe B não, ao pedirmos o tipo -umlgen, somente a classe A será documentada.

##### -umloverviewgen

An UML **overview diagram** will be created and embedded, even if there is no

@y.uml tag in overview.html.

#### -umlpackagegen

UML diagrams will be created and embedded for all documented **packages**, not only for those with an @y.uml tag.

#### -umltypegen

UML diagrams will be created and embedded for all documented **classes** and **interfaces**, not only for those with an @y.uml tag.

Segundo o manual, **-umlautogen** é o mesmo que utilizar **-umltypegen**, **-umlpackagegen**, e **-umloverviewgen** em combinação.

#### 14.2.2.2 - Destino da documentação

Local onde o javadoc gerará a documentação. É dado pelo parâmetro **-d**. Neste exemplo, **-d C:\Desenvolvimento\ydoc\documentacao**.

#### 14.2.2.3 - Formato da imagem

**-umlfileformat formatname**

Overrides the `uml_file_format` property in `resources/ydoc.cfg`.










Formatos suportados: GIF, JPG, PNG, SVG (Scalable Vector Graphics, a XML-based vector graphics

Format), SVGZ (Compressed SVG), SWF (Shockwave Flash, a popular binary vector graphicsformat).

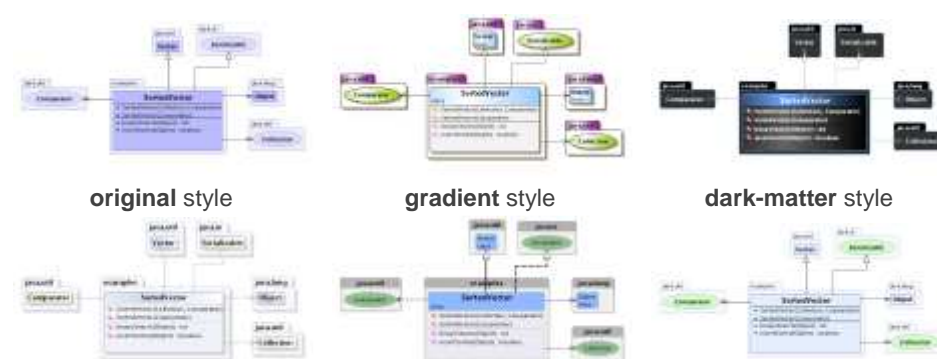
#### 14.2.2.4 - Estilos de formatação

O estilo default é o `original-style.xml`.

Os estilos reside no seguinte diretório:  
`C:\Desenvolvimento\ydoc\resources\styles`

 dark-matter-style.xml	12/04/2013 19:47	Documento XML	5 KB
 default-style.xml	12/04/2013 19:47	Documento XML	4 KB
 dotnet-style.xml	12/04/2013 19:47	Documento XML	5 KB
 gradient-style.xml	12/04/2013 19:47	Documento XML	5 KB
 muted-colors-style.xml	12/04/2013 19:47	Documento XML	6 KB
 original-style.xml	12/04/2013 19:47	Documento XML	6 KB
 theBlues-style.xml	12/04/2013 19:47	Documento XML	7 KB
 ydocstyle.xsd	12/04/2013 19:47	Arquivo XSD	11 KB
 yFiles-style.xml	12/04/2013 19:47	Documento XML	6 KB

Exemplos de estilos (skins):



dotnet style

muted-colors style

yFiles style

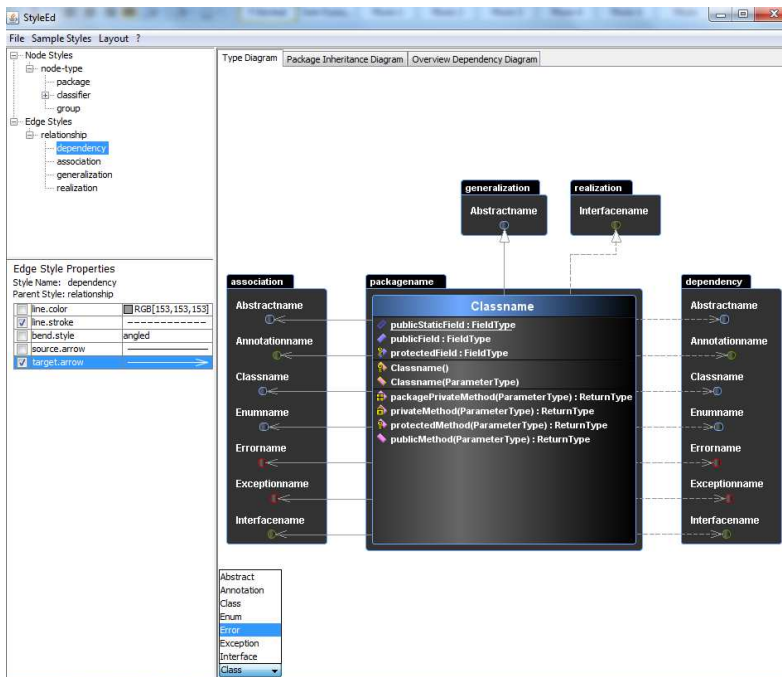
#### 14.2.2.5 - Alterando o estilo

Acessar e alterar o arquivo de configuração do UMLDoclet (ver abaixo), informando o estilo css desejado.

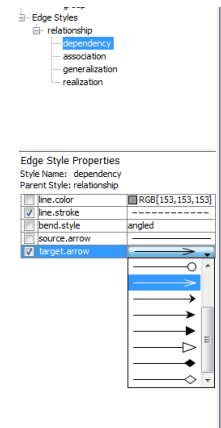
#### 14.2.2.6 - Customizando um estilo

Além dos estilos que já vem com o produto, pode-se customizar um estilo. Existem dezenas de opções, como por exemplo: tipo de seta para os relacionamentos, cor, tamanho e etc.

Executar a aplicação: C:\Desenvolvimento\ydoc\lib\styleed.jar



Alterando a seta para um relacionamento de dependência.





#### 14.2.2.7 - Arquivo onde reside a configuração do UMLDoclet:

Os parâmetros passados em runtime ao UMLDoclet fazem um override nos parâmetros no arquivo de configuração.

Arquivo de configuração: C:\Desenvolvimento\ydoc\resources\ydoc.cfg

```
<?xml version="1.0" encoding="UTF-8"?>
- <configuration xmlns="http://www.yworks.com/xml/ydoc">
  - <group name="diagrams">
    - <group name="overview">
      - <group name="diagram">
        <property name="style" value="./styles/original-style.xml"/>
        <property name="type" value="dependency"/>
        <property name="id" value="0"/>
        + <group name="layout">
        + <group name="insets">
        + <group name="include">
      </group>
    - <group name="diagram">
      <property name="style" value="./styles/original-style.xml"/>
      <property name="type" value="inheritance"/>
      <property name="id" value="1"/>
      + <group name="layout">
      + <group name="insets">
      + <group name="include">
    </group>
  </group>
  - <group name="package">
    - <group name="diagram">
      <property name="style" value="./styles/original-style.xml"/>
      + <group name="layout">
      + <group name="insets">
      + <group name="include">
    </group>
  </group>
  - <group name="type">
    - <group name="diagram">
      <property name="style" value="./styles/original-style.xml"/>
      + <group name="include">
        <!-- Sample exclude group. All classes in java.* packages are excluded
        <!-- <group name="exclude"> <group name="pattern"> <property name="pattern" value="java.*"/> </group> </group>
        <property name="realizations" value="javax.*"/> </group> </group>
      + <group name="insets">
      - <group name="order">
        <property name="fields" value="mod-lex"/>
        <property name="constructors" value="lex-ic"/>
        <property name="methods" value="lex-ic"/>
      </group>
      + <group name="layout">
    </group>
  </group>
  - <group name="formats">
    <property name="fileformat" value="PNG"/>
    + <group name="vectorgraphics">
    + <group name="image">
  </group>
  + <group name="misc">
</configuration>
```

Estilos CSS

Formato Imagem