NATIONAL UNIVERSITY OF COMPUTER AND EMERGING SCIENCES

Software Design and Analysis (CS 3004)

Instructor: Nida Munawar

nida.munawar@nu.edu.pk

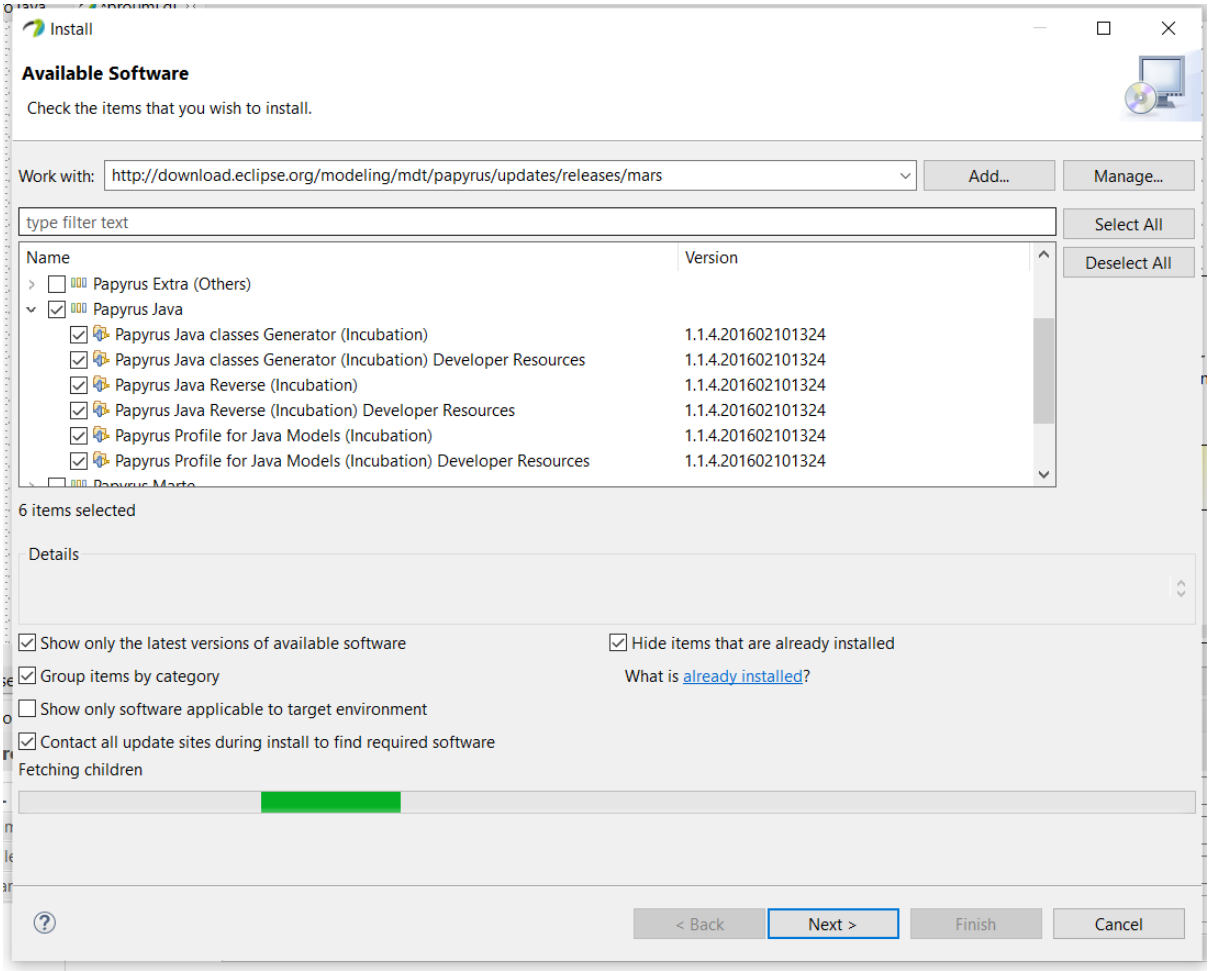
Installation instructions

Generate java code using diagram

You need to **install Papyrus Java Classes Generator**.

1. Help --> install new software
2. work with
   : http://download.eclipse.org/modeling/mdt/papyrus/updates/releases/mars
3. Check *group items by category*
4. Check *show only the latest version of available software*
5. Check Papyrus Java
6. Next --> accept terms and conditions
7. finish

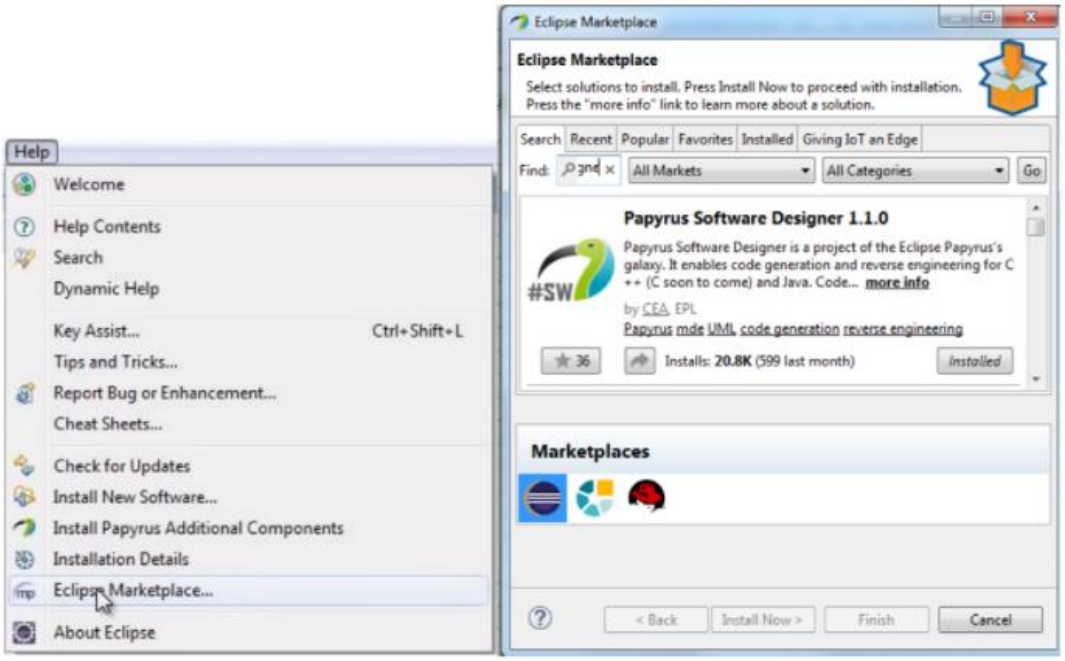
Now, in your model.di : right click on your class --> Java --> Generate java code. And you're done !



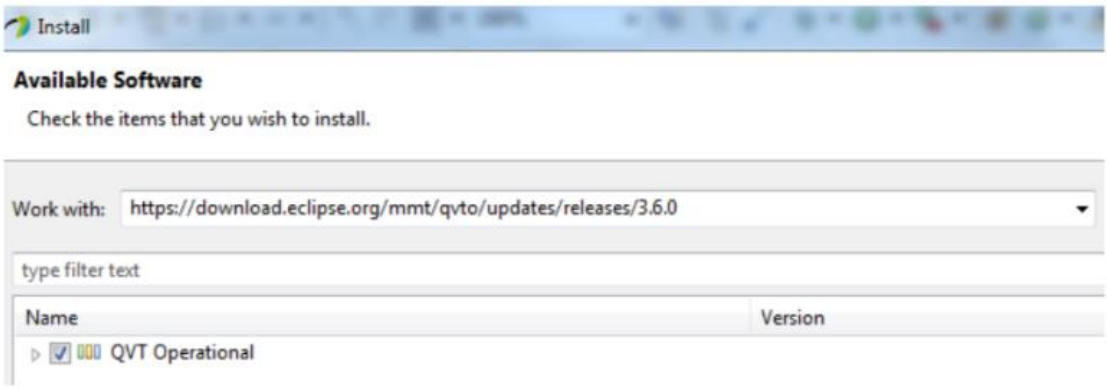
how to install Marketplace client on Eclipse

Simply select All sites and search for "marketplace" on the *Install new software...* wizard.

Go to help and Eclipse Client and install software designer



Go to https://download.eclipse.org/mmt/qvto/updates/releases/3.6.0 this for QVT

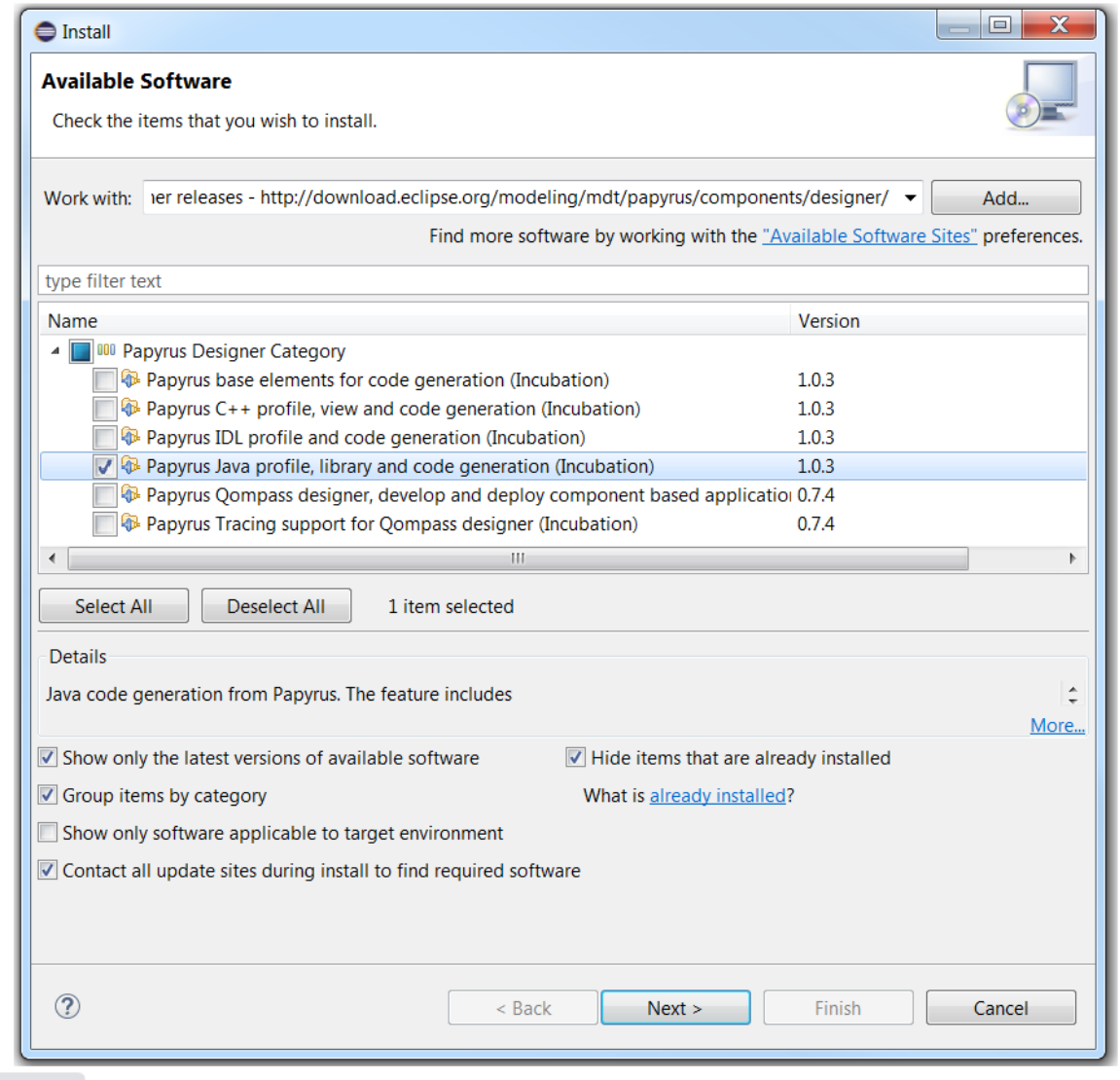


Restart the software after each step now your system is ready for Code Generation

The Java features of Papyrus are available as part of the Papyrus Software Designer extra feature.

You can install it by using the Papyrus Software Designer update-site:

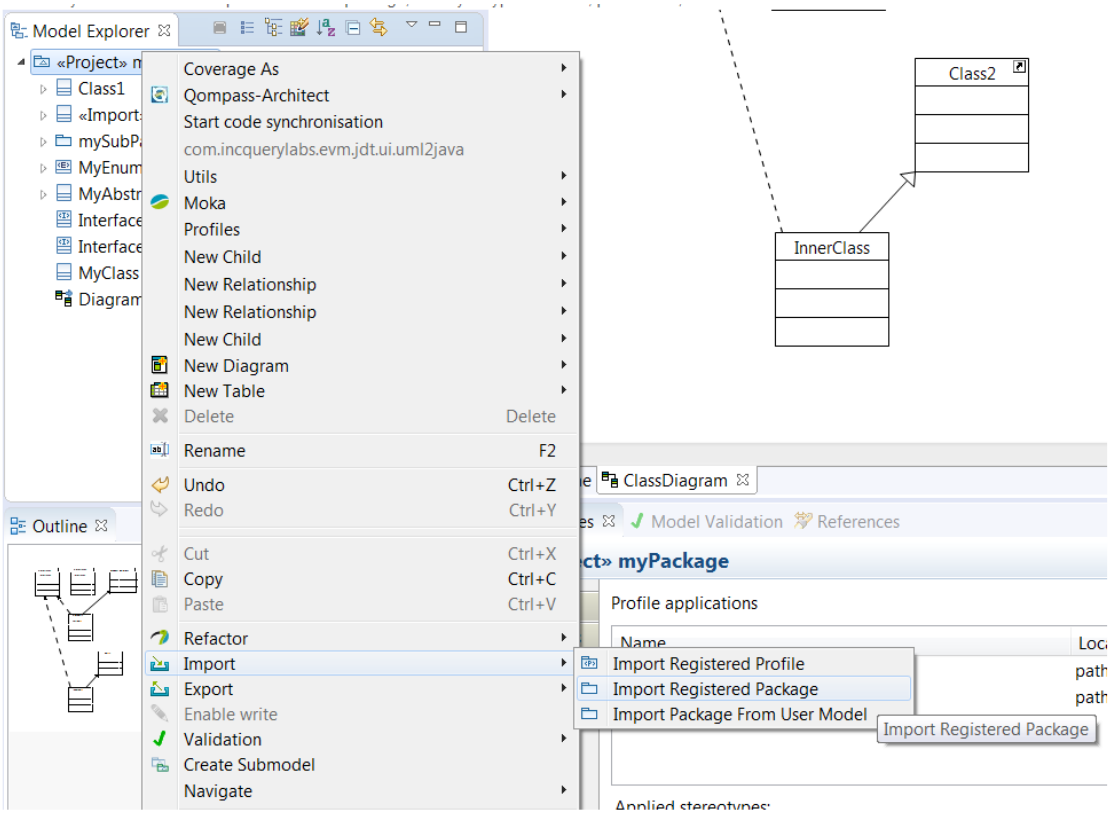
- http://download.eclipse.org/modeling/mdt/papyrus/components/designer/

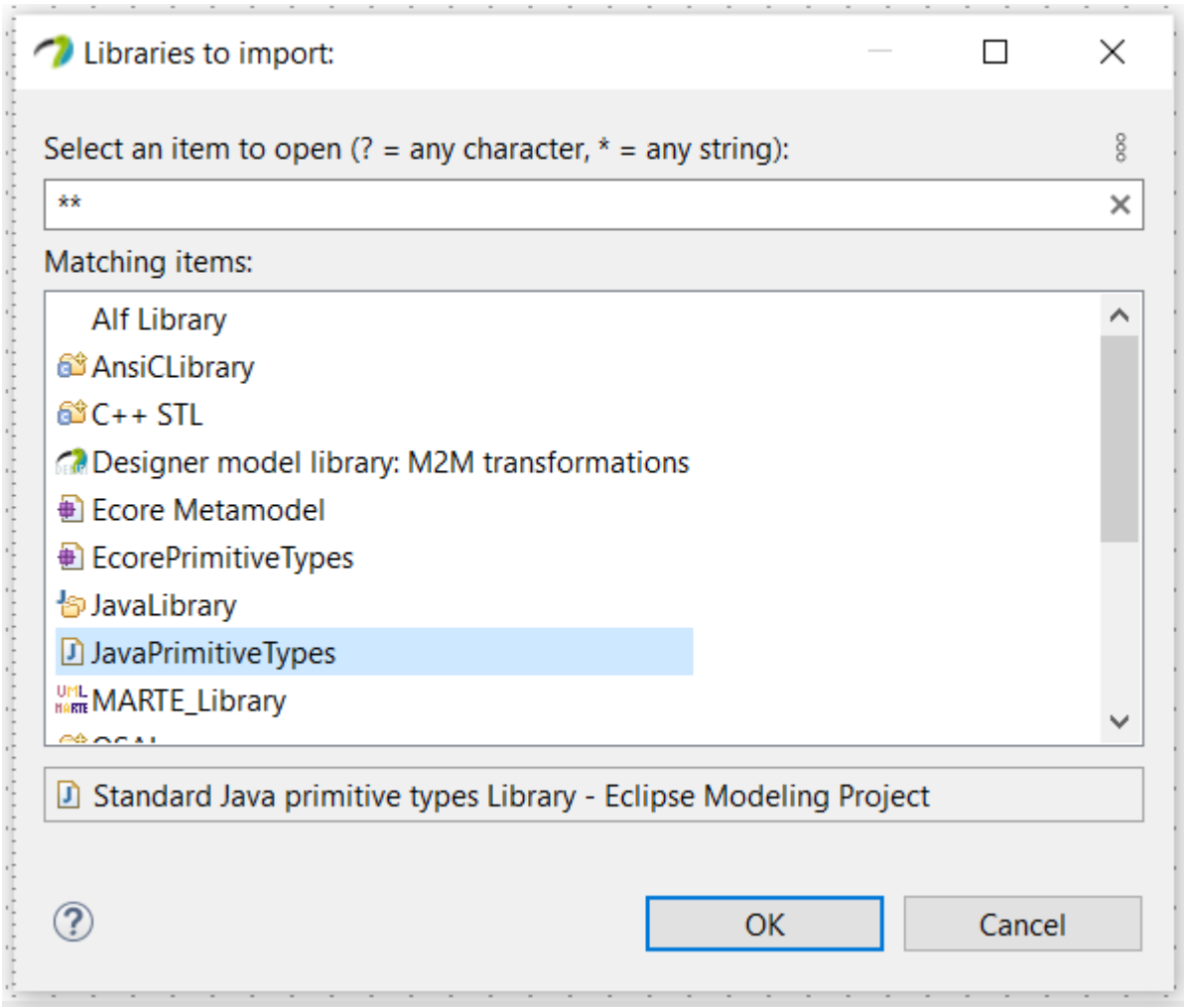


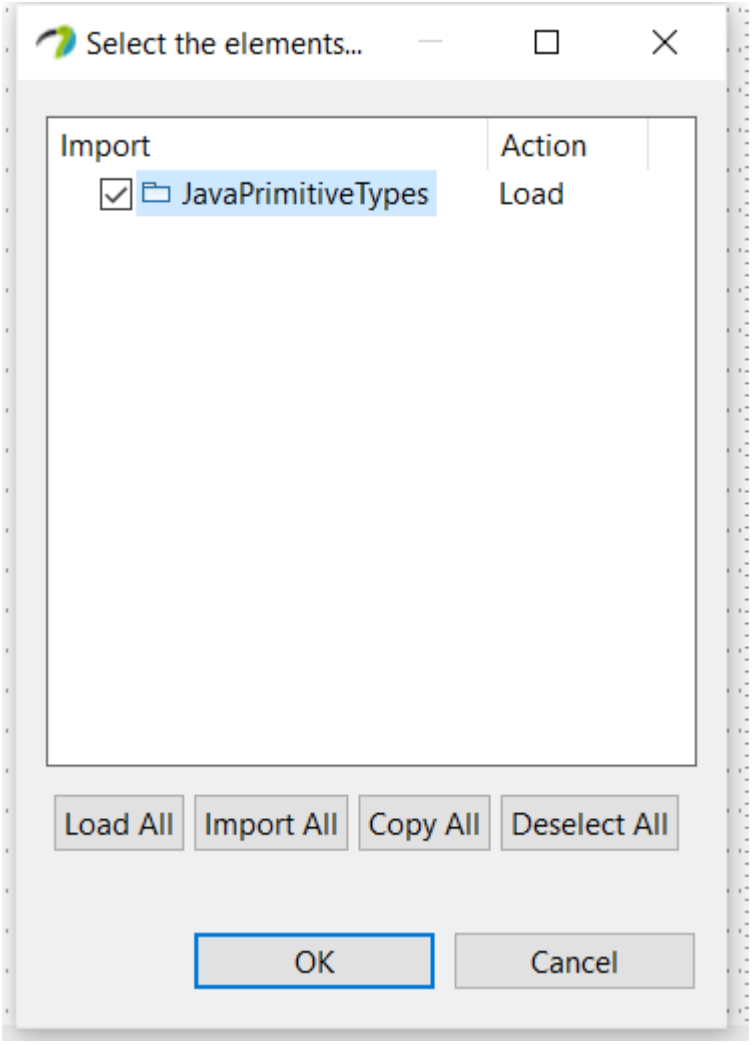
Java library

The Java library provides Java primitives and their wrapper classes. To import, this library:

- Right click on your model (in the Model Explorer view)
- Import > Import Registered Package
- Choose the JavaPrimitiveTypes
- Then you can use the Java primitives in this package, when you type attributes, parameters, etc...


Model Explorer
Class1
MyEnum
MyClass
Coverage As
Qompass-Architect
Start code synchronisation
com.incquerylabs.evm.jdt.ui.uml2java
Utils
Moka
Profiles
New Child
New Relationship
New Relationship
New Child
New Diagram
New Table
Delete Delete
Rename F2
Undo Ctrl+Z
Redo Ctrl+Y
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Refactor
Import
Export
Enable write
Validation
Create Submodel
Navigate
Outline
Class2
InnerClass
ClassDiagram
Model Validation
References
myPackage
Profile applications
Import Registered Profile
Import Registered Package
Import Package From User Model
Import Registered Package


Libraries to import:
Select an item to open (? = any character, * = any string):
**
Matching items:
Alf Library
AnsiCLibrary
C++ STL
Designer model library: M2M transformations
Ecore Metamodel
EcorePrimitiveTypes
JavaLibrary
JavaPrimitiveTypes
MARTE_Library
Standard Java primitive types Library - Eclipse Modeling Project
OK
Cancel

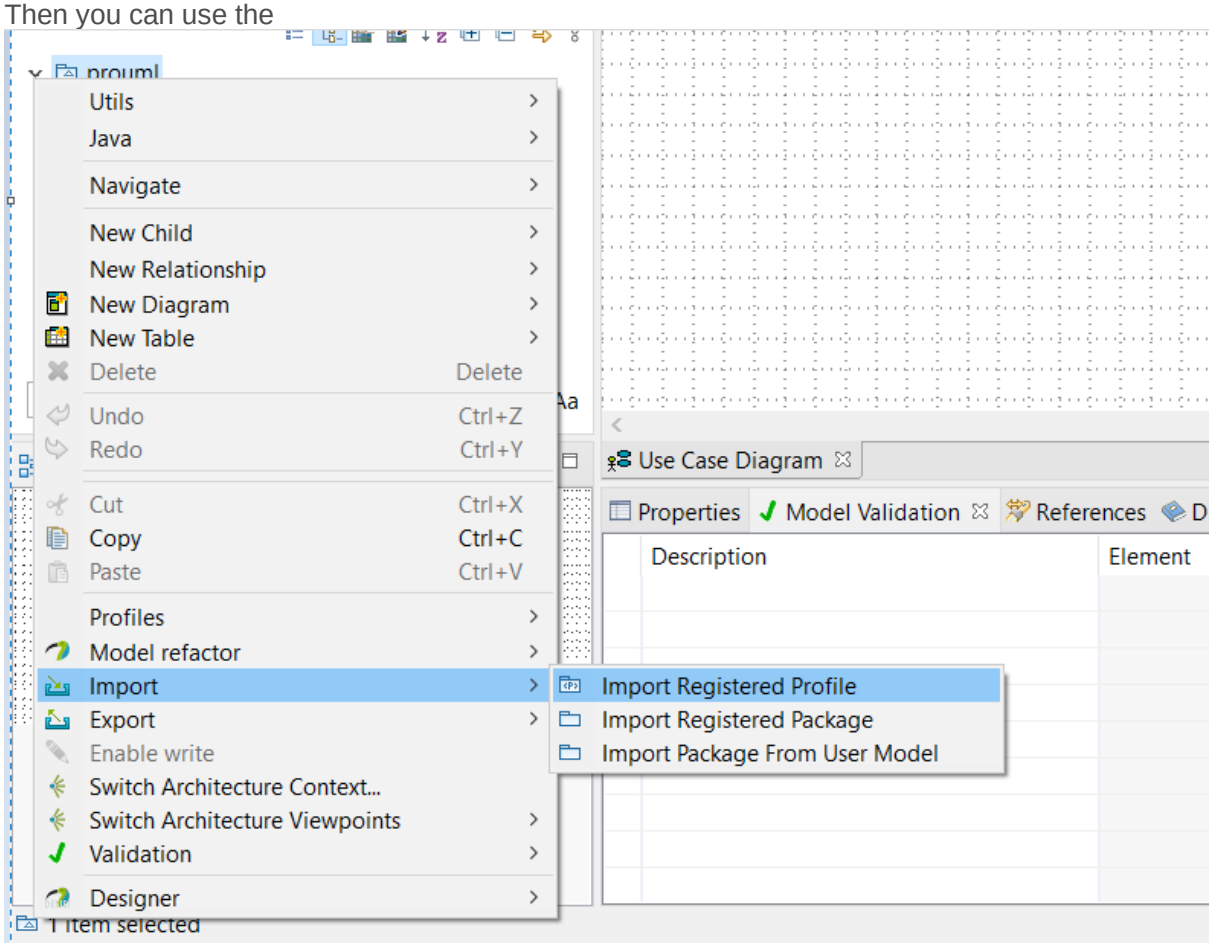


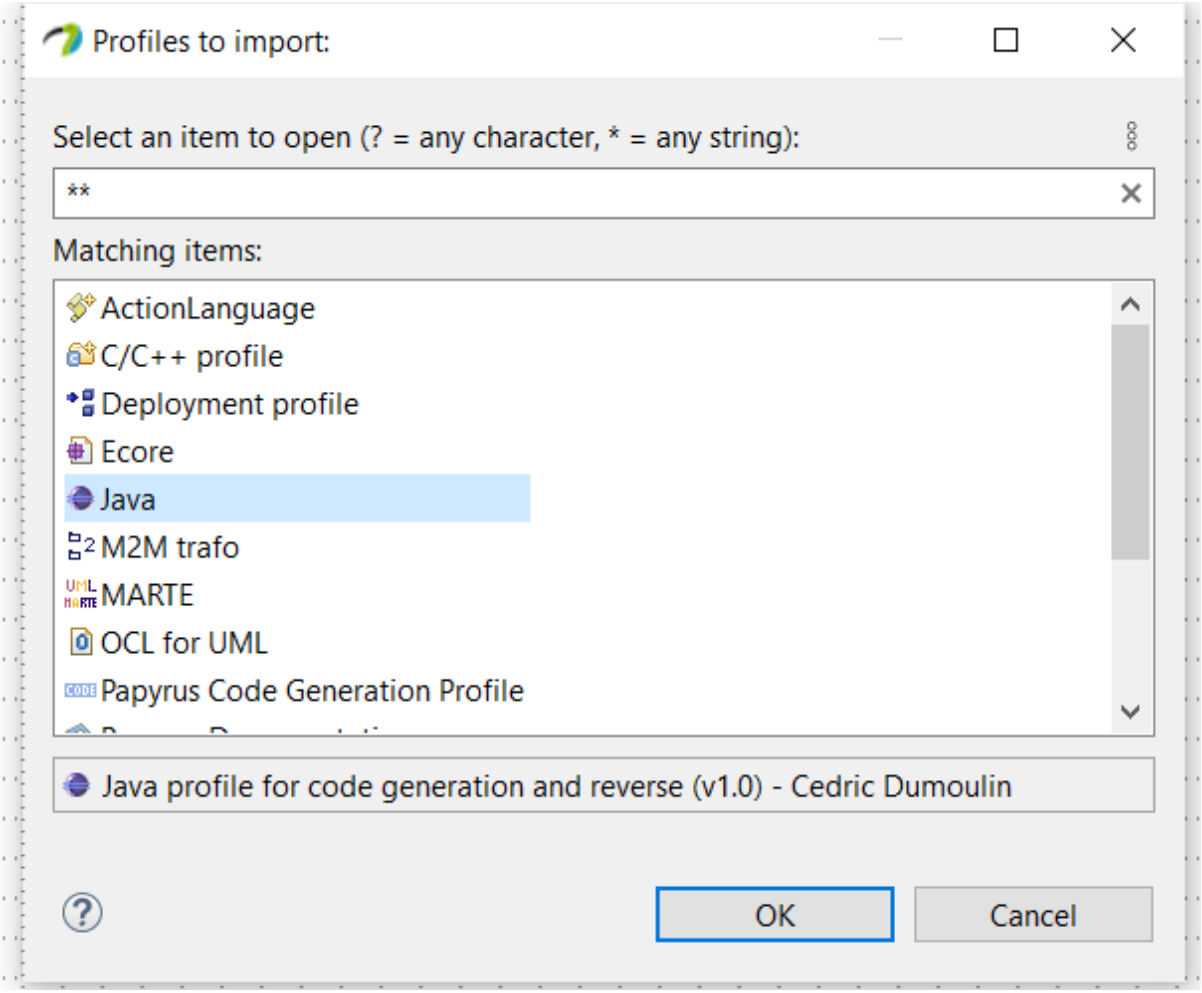
Java Profile

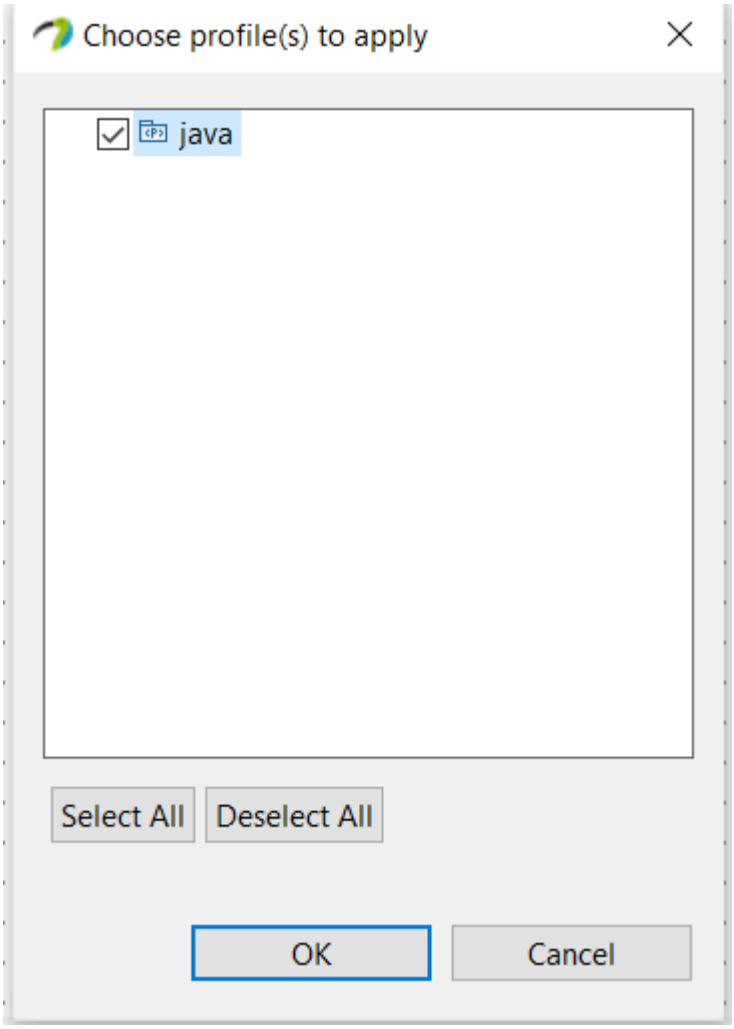
The Java library provides Java primitives and their wrapper classes. To import, this library:

- Right click on your model (in the Model Explorer view)
- Import > Import Registered Profile
- Choose the Java package
- This step is for adding stereotype

- Then you can use the

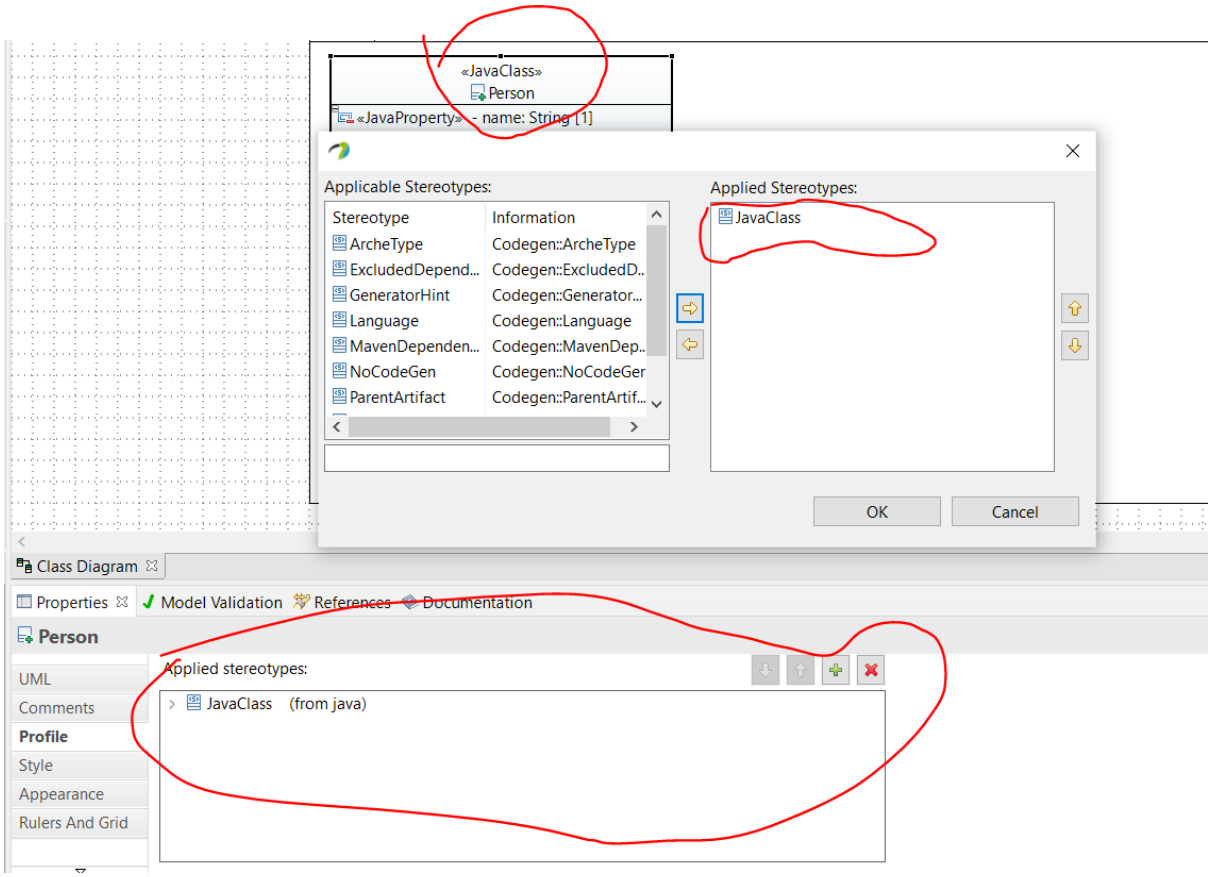



Profiles to import:
Select an item to open (? = any character, * = any string):
**
Matching items:
ActionLanguage
C/C++ profile
Deployment profile
Ecore
Java
M2M trafo
MARTE
OCL for UML
Papyrus Code Generation Profile
Java profile for code generation and reverse (v1.0) - Cedric Dumoulin
OK
Cancel



BeforeCode generation add stereotypes

Add stereotypes for class name, properties and for operations as well

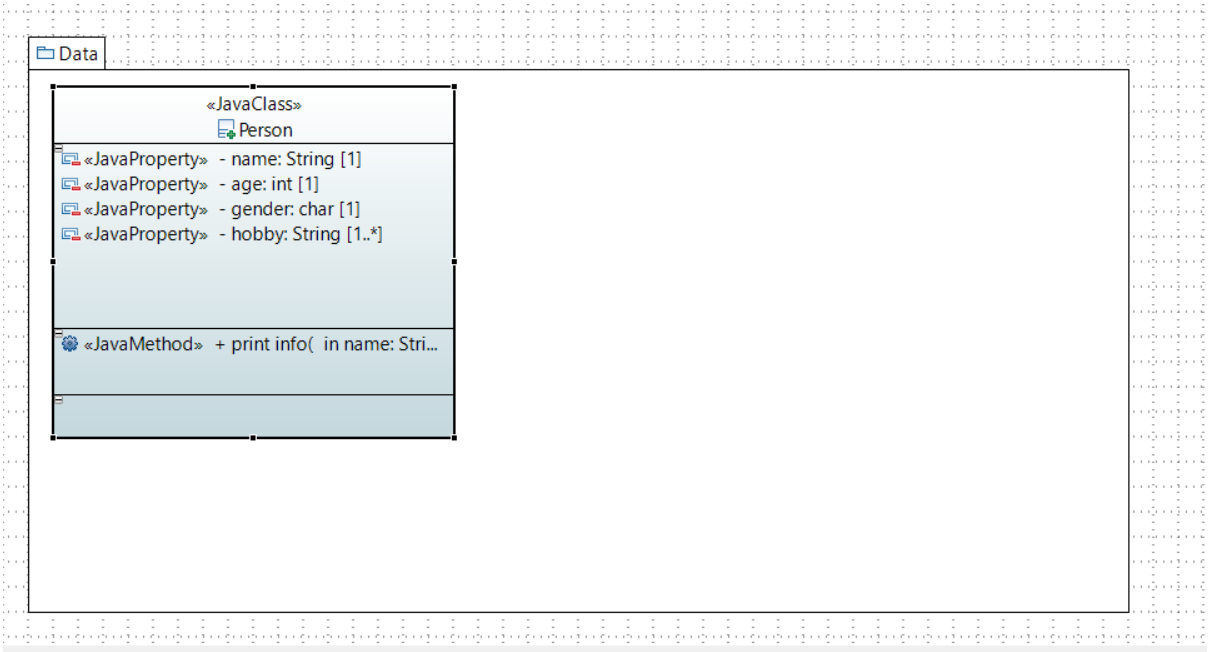


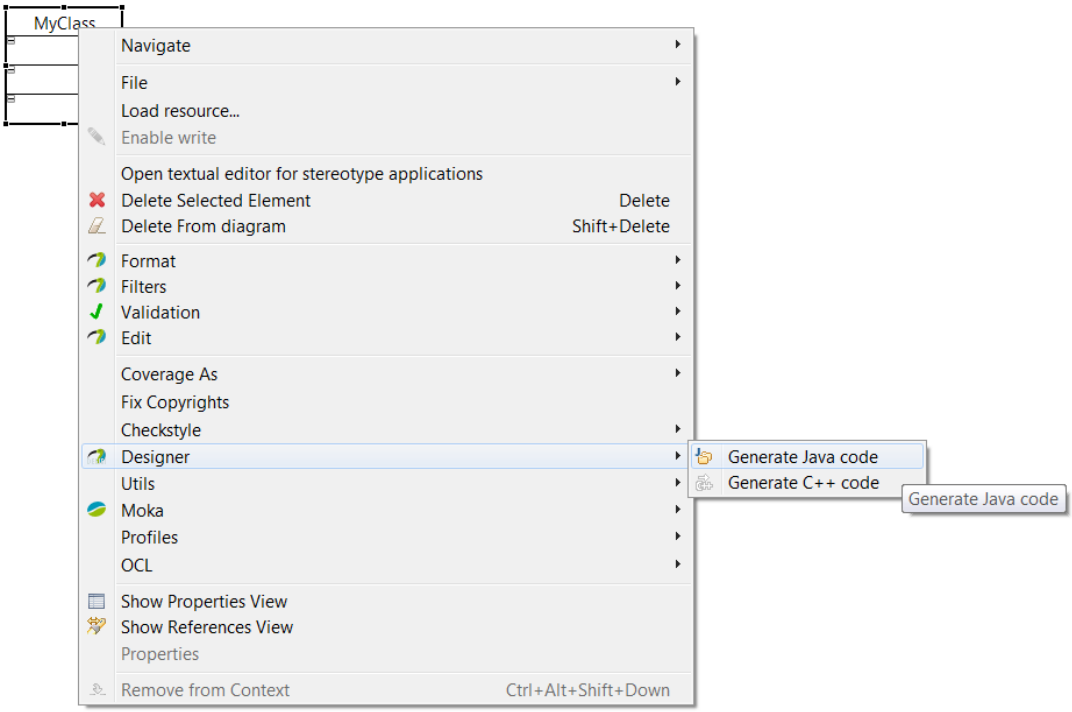
Code generation from UML elements

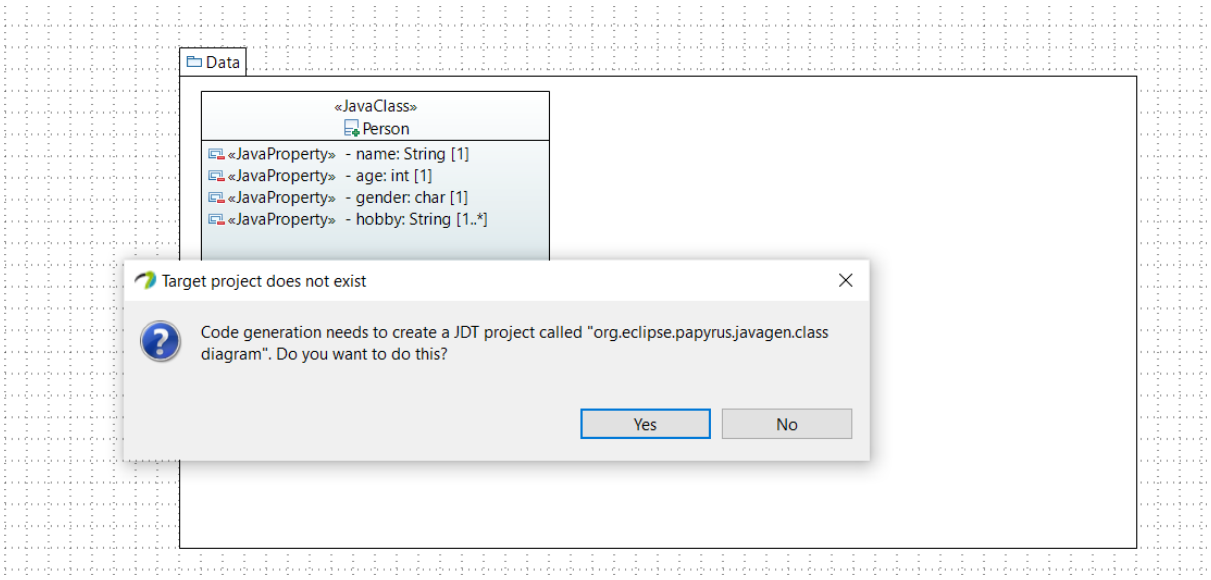
You can generate code for a specific classifier or package in your UML model. To generate code, do the following:

- Right click on classifier or package, either in a diagram or in the model explorer
- Designer > Generate Java code
- Follow the JDT dialogs that let you create a new JDT project in Eclipse, where code will be generated, if there is no JDT project associated with your model (e.g. the first time you generate code from your model)

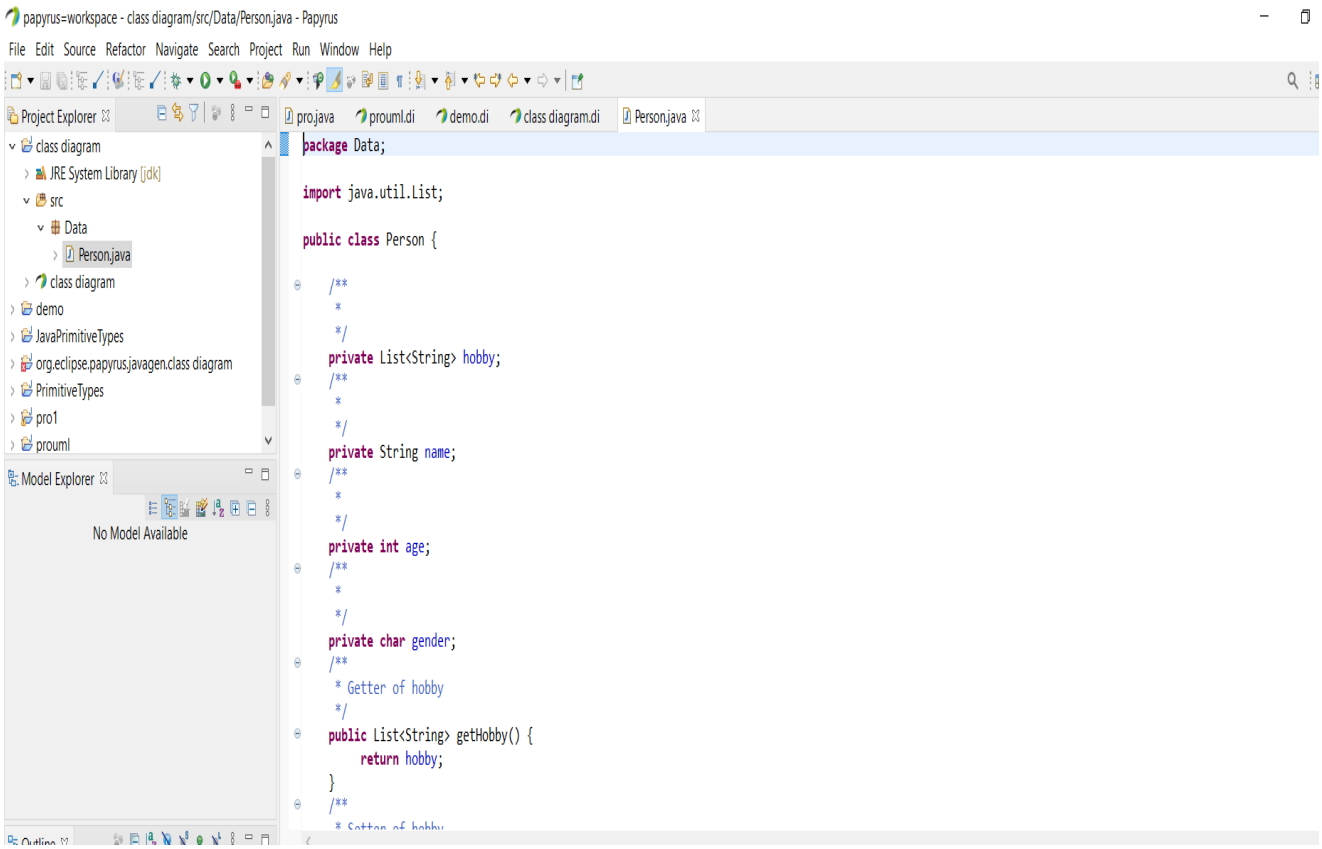
When you generate code from a classifier, its required classifiers are also generated. Required classifiers are classifiers related for the generated classifier, e.g. typing one of its attributes, inheritance relationship, dependency relationship. When you generate code from a package (e.g. the root of your model), all of its classifiers, and their required classifiers, will be generated.


Data
«JavaClass»
Person
«JavaProperty» - name: String [1]
«JavaProperty» - age: int [1]
«JavaProperty» - gender: char [1]
«JavaProperty» - hobby: String [1..*]
«JavaMethod» + print info(in name: Stri...


MyClass
Navigate
File
Load resource...
Enable write
Open textual editor for stereotype applications
Delete Selected Element Delete
Delete From diagram Shift+Delete
Format
Filters
Validation
Edit
Coverage As
Fix Copyrights
Checkstyle
Designer
Utils
Moka
Profiles
OCL
Show Properties View
Show References View
Properties
Remove from Context Ctrl+Alt+Shift+Down
Generate Java code
Generate C++ code
Generate Java code


Data
«JavaClass»
Person
«JavaProperty» - name: String [1]
«JavaProperty» - age: int [1]
«JavaProperty» - gender: char [1]
«JavaProperty» - hobby: String [1..*]
Target project does not exist
Code generation needs to create a JDT project called "org.eclipse.papyrus.javagen.class diagram". Do you want to do this?
Yes
No

Click on yes


papyrus=workspace - class diagram/src/Data/Person.java - Papyrus
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
class diagram
JRE System Library [jdk]
src
Data
Person.java
class diagram
demo
JavaPrimitiveTypes
org.eclipse.papyrus.javagen.class diagram
PrimitiveTypes
pro1
prouml
Model Explorer
No Model Available
pro.java
prouml.di
demo.di
class diagram.di
Person.java
package Data;
import java.util.List;
public class Person {
/**
*
*/
private List<String> hobby;
/**
*
*/
private String name;
/**
*
*/
private int age;
/**
*
*/
private char gender;
/**
* Getter of hobby
*/
public List<String> getHobby() {
return hobby;
}
/**