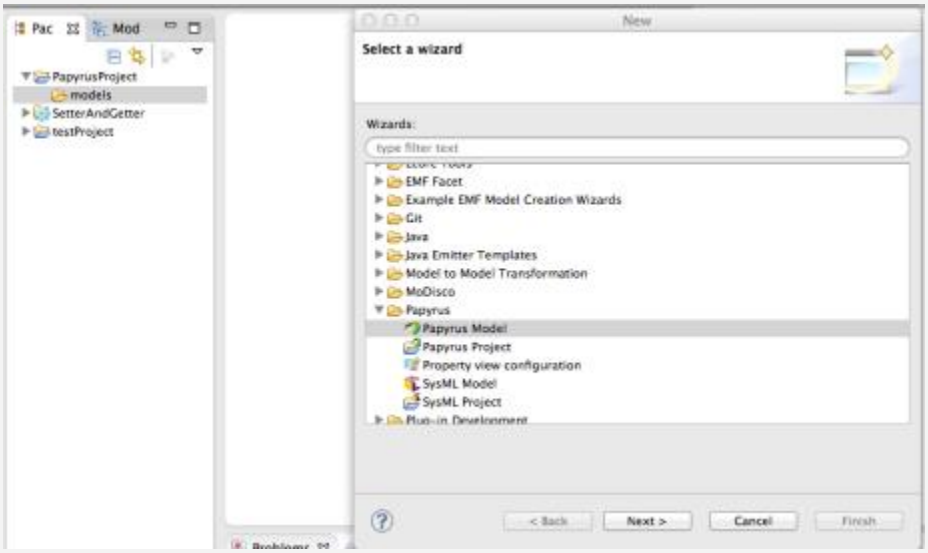
UML Diagrams (and Models) with Papyrus

Info This post has 14 comments, please enjoy the discussion !
In this post I am going to explain how to use Papyrus to create UML diagrams and models.

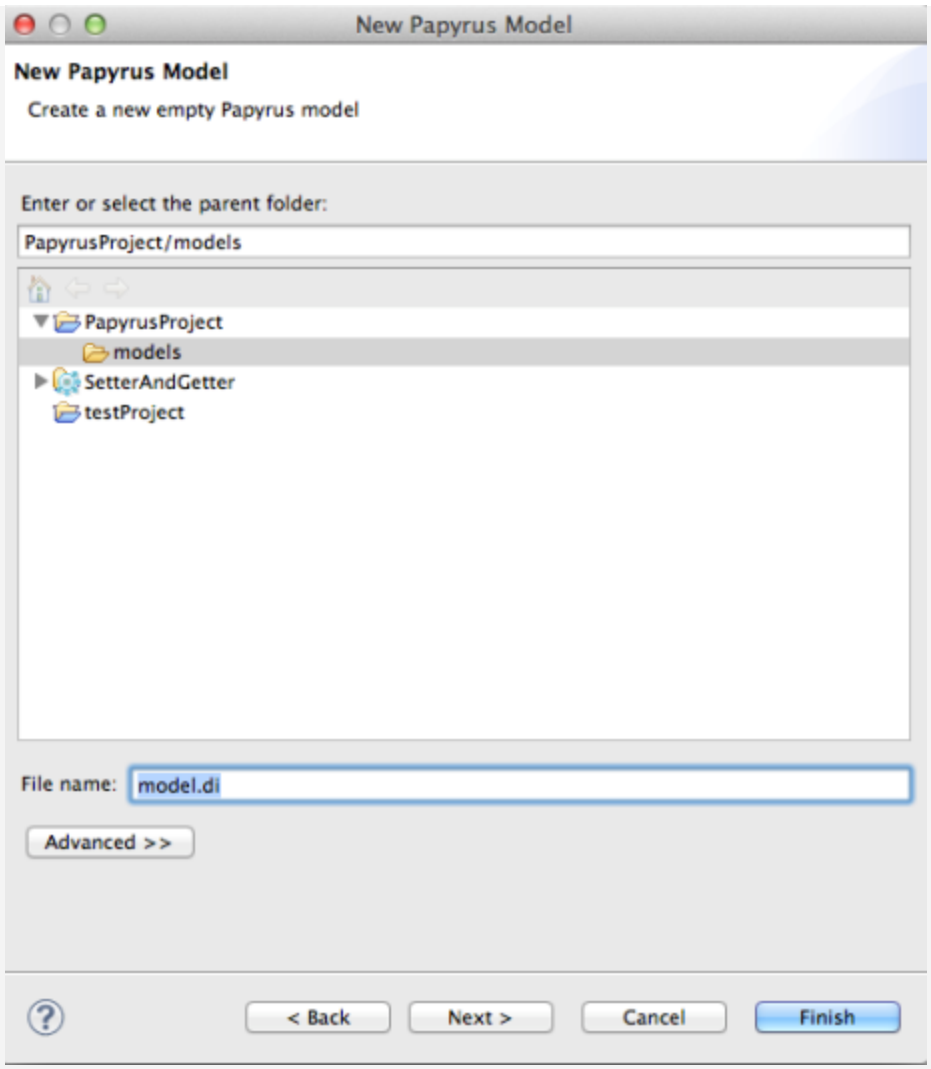
I want to stress this distinction because it is often hard for beginners to get the difference between these two artifacts. However this concept is quite important if you want to correctly manage your project. Basically in the EMF a model is a file containing an XMI description of an instance of a meta-model defined by means of the ECORE (the Eclipse MOF implementation). The Eclipse Modeling Framework comes with a tree based editor enabling an easier management of the models' contents. Obviously richer graphical editor are possible and this is what is for instance provided by Papyrus with respect to the UML meta-model. Diagram's contents are stored in a different file linked to the model and containing the data the editor needs to depict boxes and edges in the desired points.

Creating a new diagram and its related model

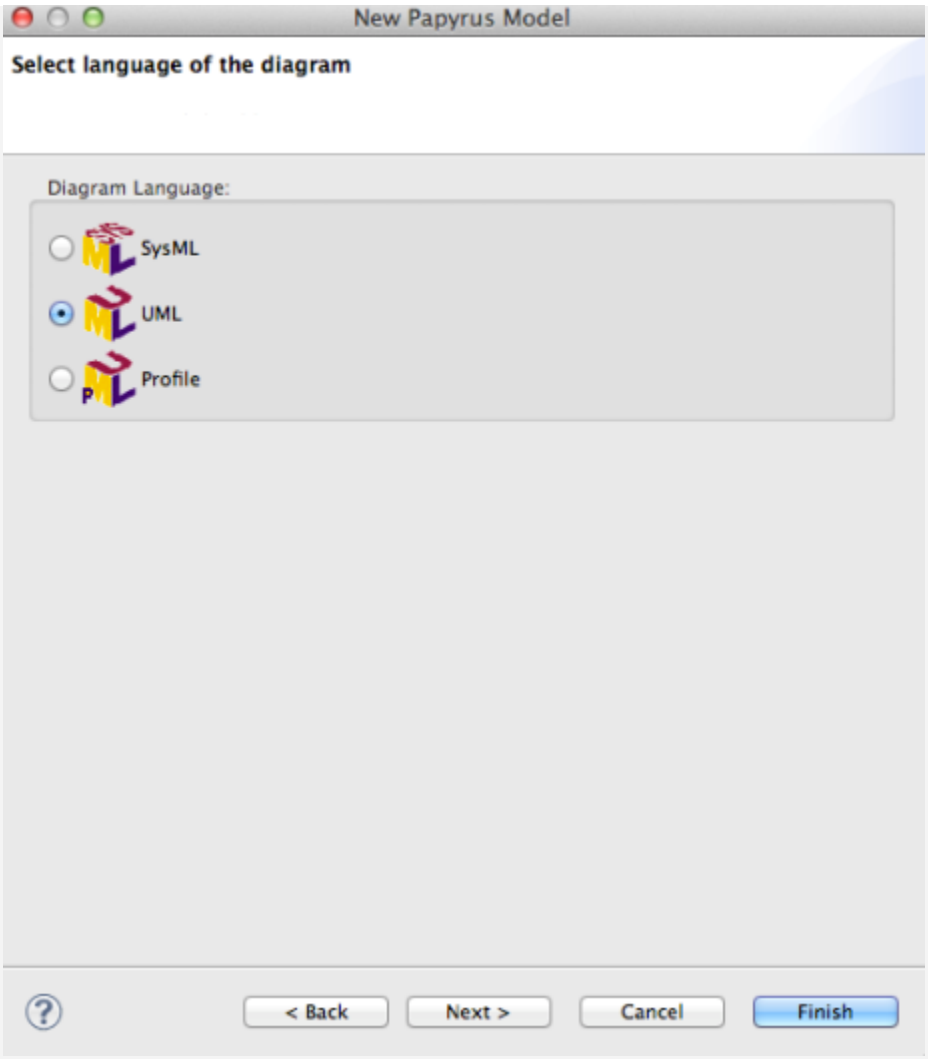
Create an empty project and a folder for your models Right click and select Net->Other… Choose Papyrus Model under the Papyrus folder in the wizard that will pop up



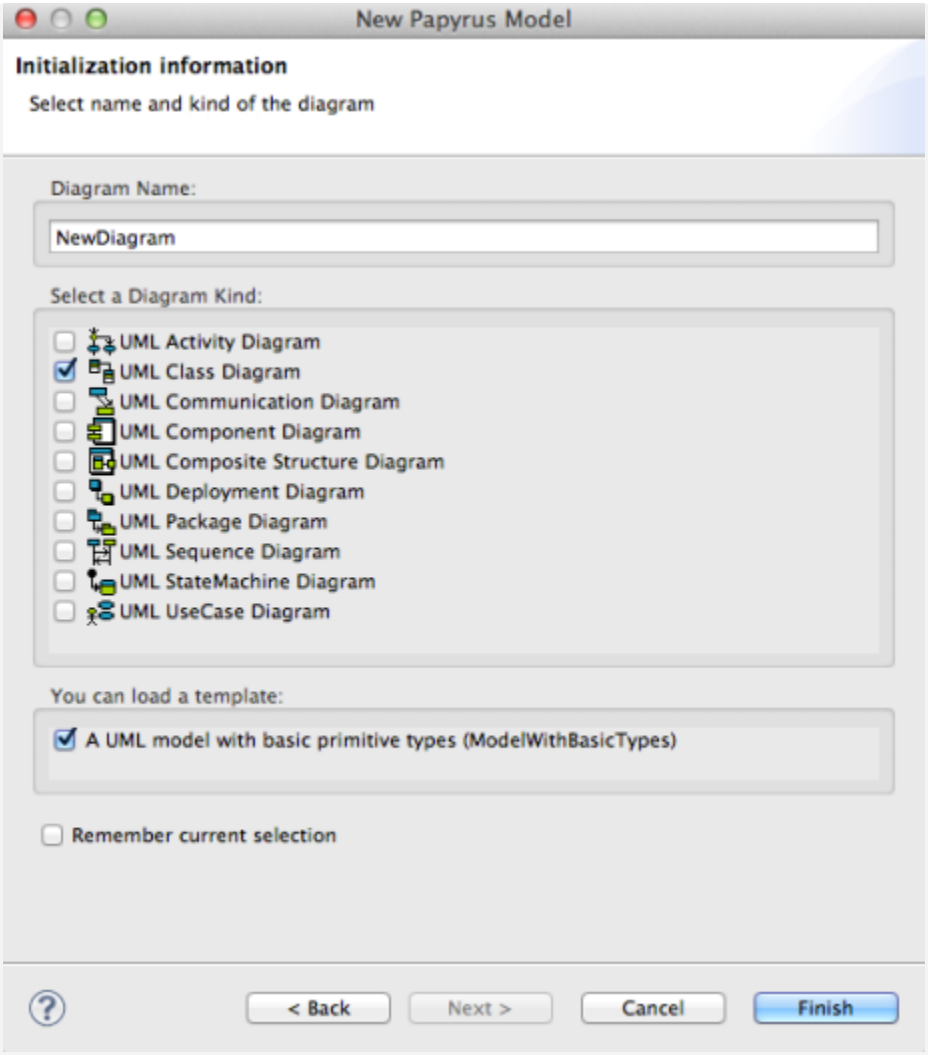
change the model name and press next (not finish)



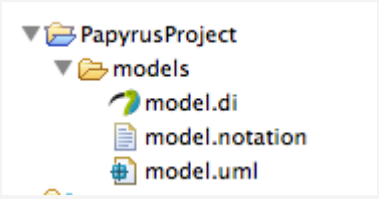
Choose UML and press next (not finish)



now you can choose with what kind to start. Also choose a name for such diagram. If you want yo use the PrimitiveType package flag the related checkbox. Note that this is due to the fact that UML does not come with predefined primitive types but, according to the UML Superstructure (http://www.omg.org/spec/UML) they are defined, as auxiliary construct, in the AuxiliaryConstructs package.



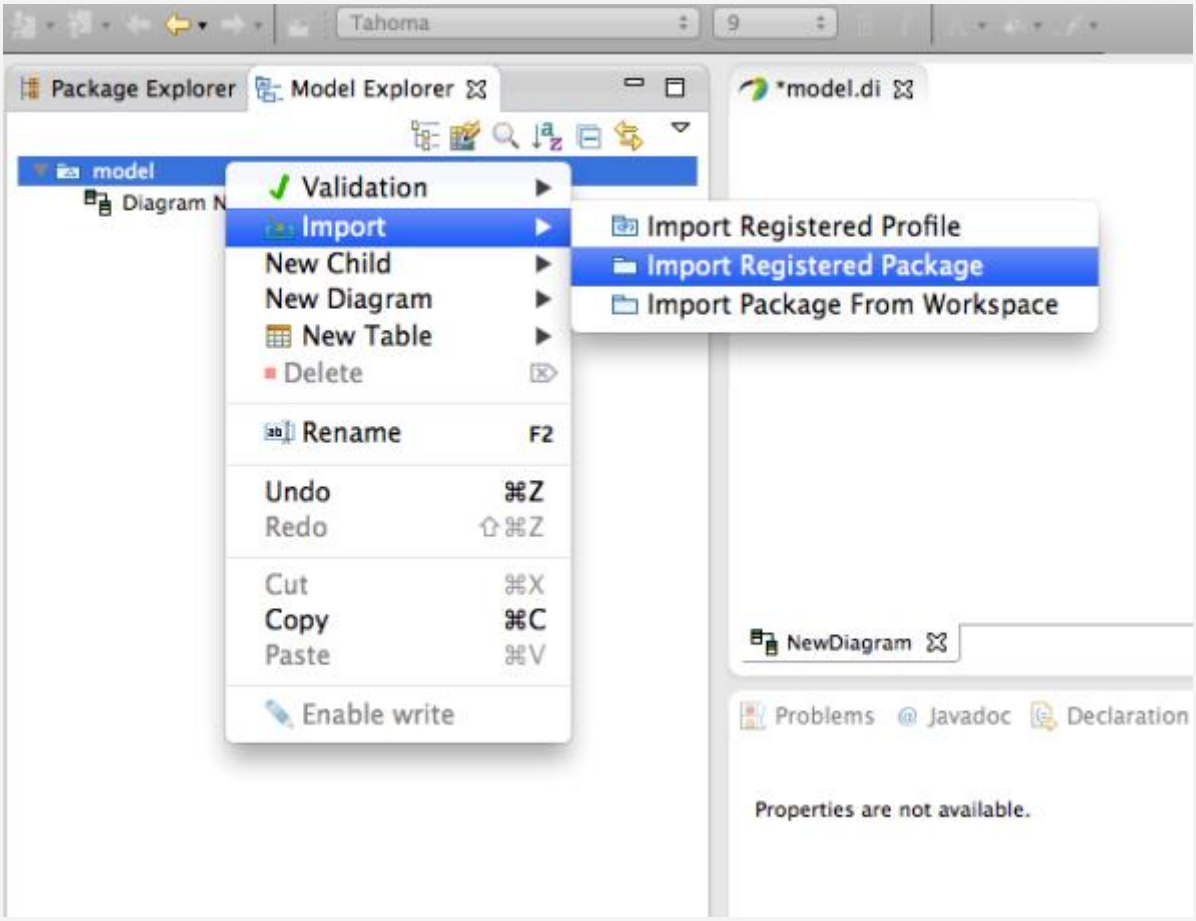
As you can easily note, in your Package Explorer tab three files have been created. The model.uml is the actual model while the model.di and model.notation files contain the information used by the editor to depict diagrams.



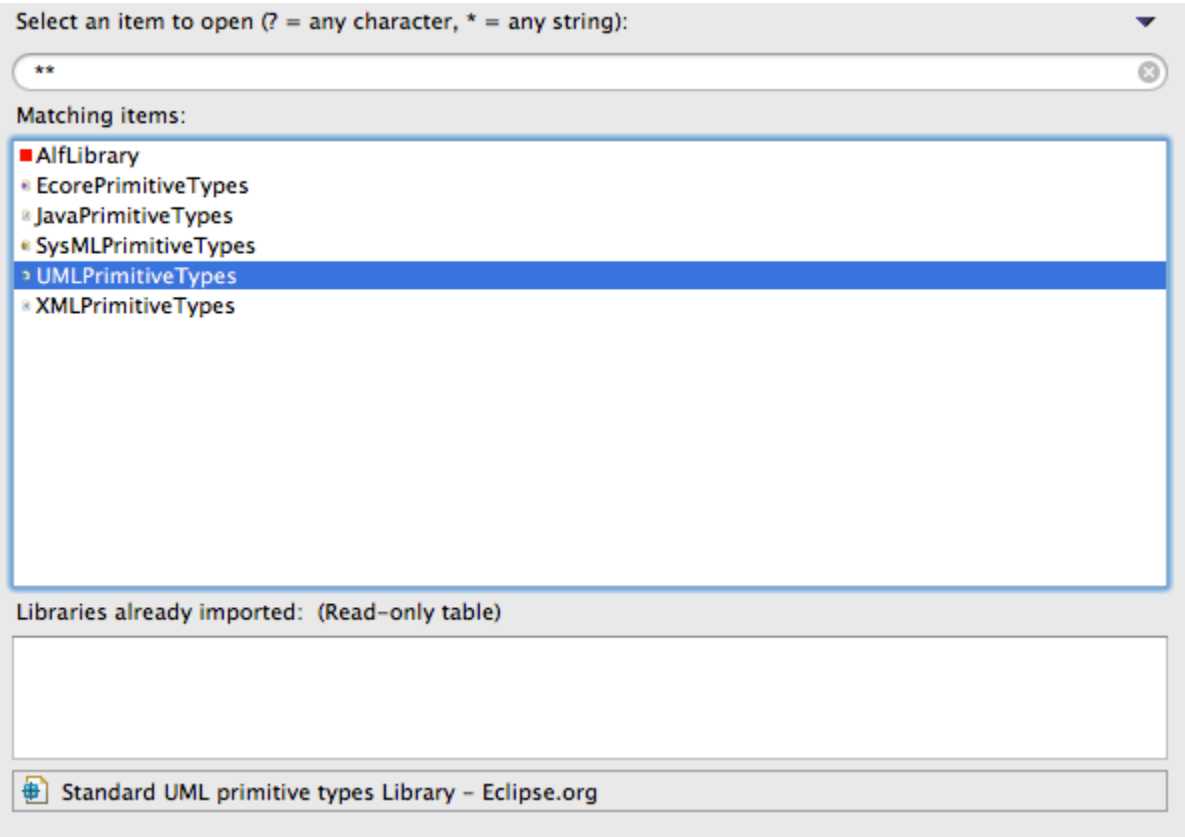
Importing primitive types into an already existing model

If you need to add the primitive type package into an already existing model you just have to right click on the model element from the Model Explorer tab. If you can’t see it select the Window menu then Show

View->Other… then write model explorer in the search pop up and choose it. Choose Import->Import Registered Package as depicted in the following figure

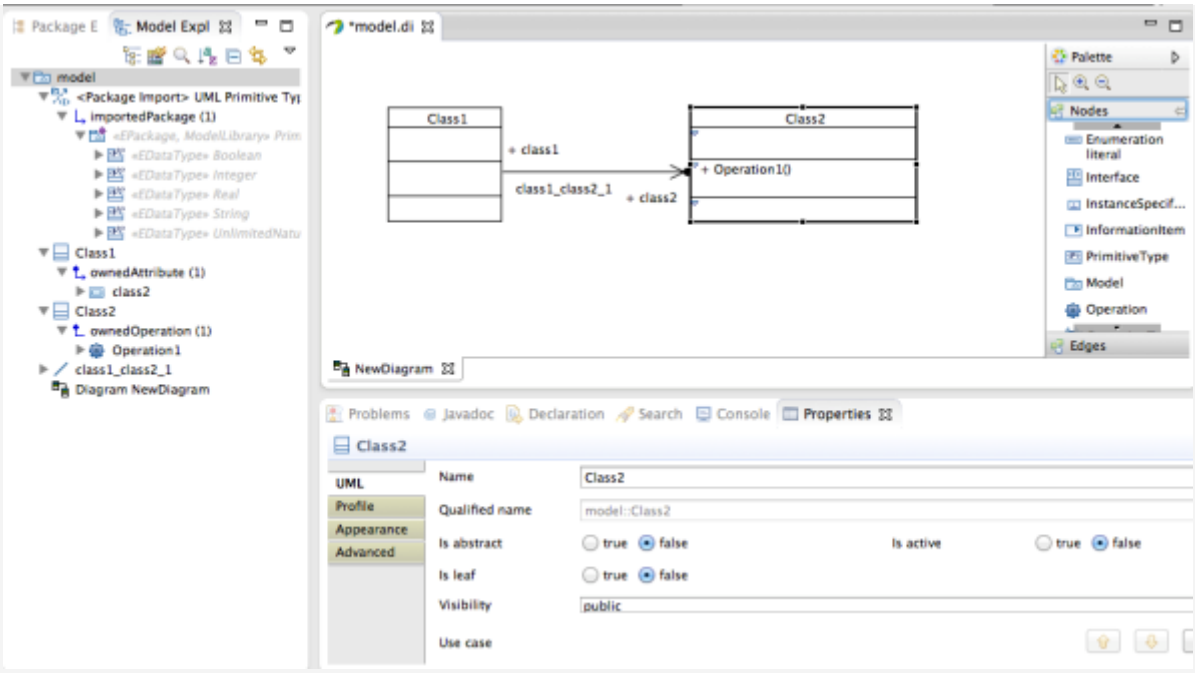


The following window will pop up and. Select UMLPrimitiveTypes and press the Finish button



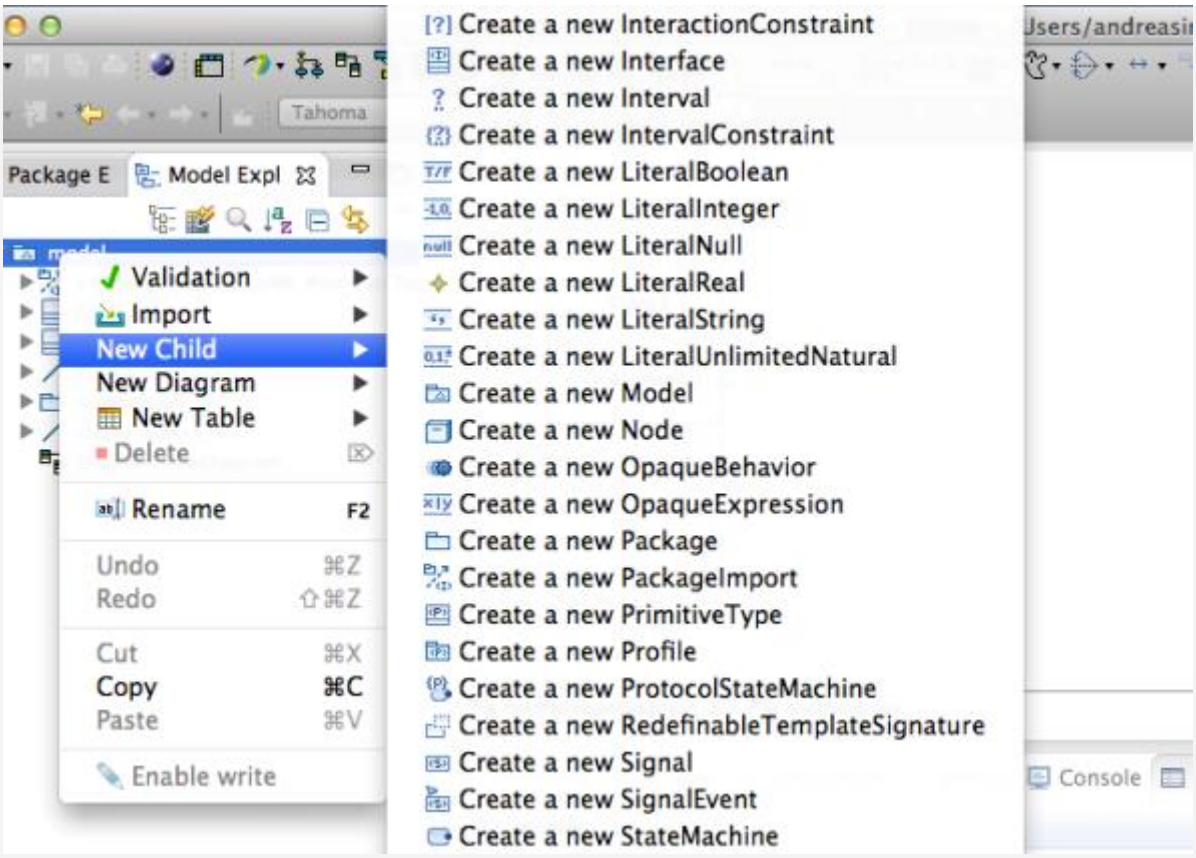
A Simple Class Diagram

The following figure depicts an example of Class Diagram. It just consists of two classes with an association (see http://lowcoupling.wordpress.com/2012/09/19/understanding-uml-association/ for a detailed description on this construct). Entities can be dragged and dropped from the palette at the right side of the screen. I like to have the model contents depicted in the left side. To this end you should select the model explorer tab. If you can't see it select the Window menu then Show View->Other… then write model explorer in the search pop up and choose it. UML constructs have many properties you may need to edit. That is why I like to have the properties view open on the bottom of the screen. If you can't see it right click on a model element (i.e. Class2) and choose "Show Properties View".

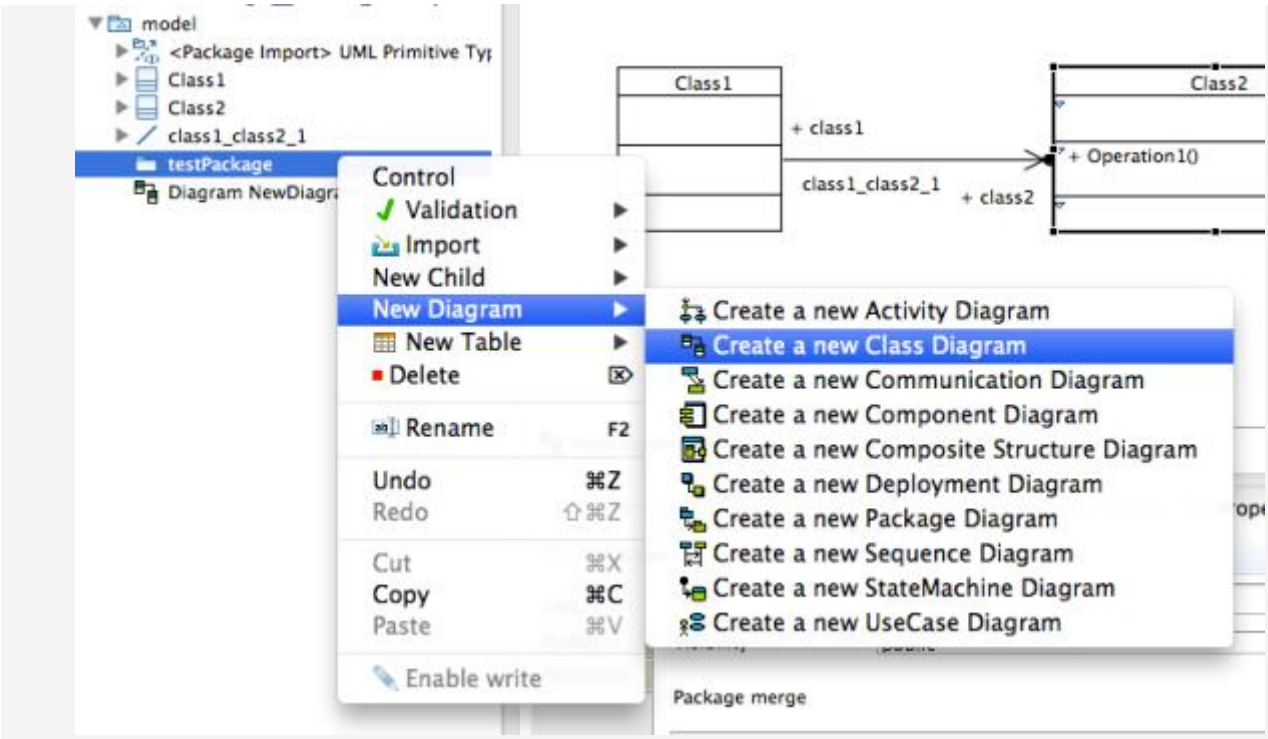


Making things a bit more interesting …

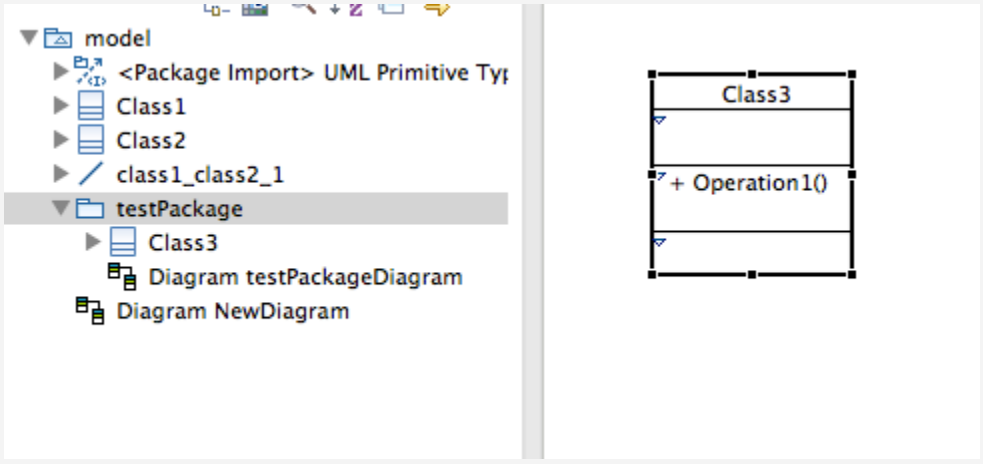
Now let’s try to make things more interesting. Trying for instance to add a package to your model. To this end you may right click on the model element on the right model explorer tab, select New Child->Create a New Package. I this example I have called the new package “testPackage”.



Now right click on the just created package, always in the right model explorer tab and select New Diagram->Create a New Class Diagram. I have called the new diagram testPackageDiagram.

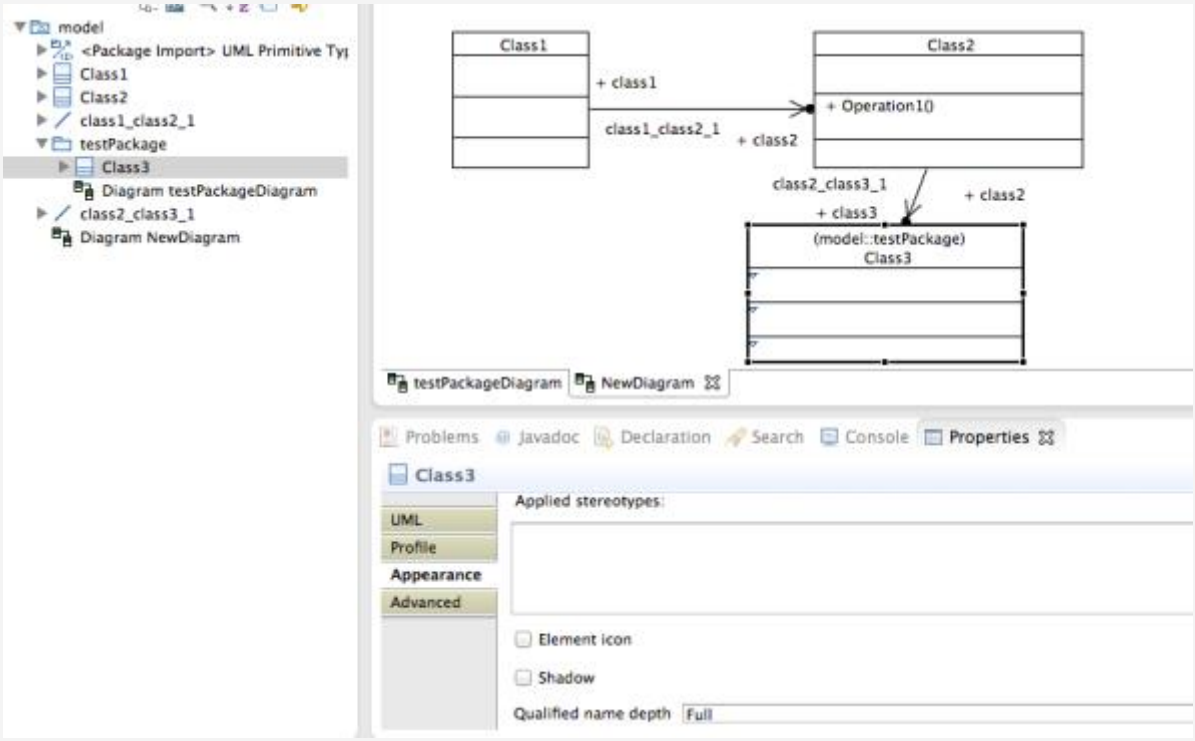


Now in the model explorer you can see the contents plus the two diagrams you have created. You can switch from a diagram to another by clicking on the one you want to depict. Create a new class (i.e. Class3) in the testPackageDiagram. As you can see the classes is created under the testPackage.

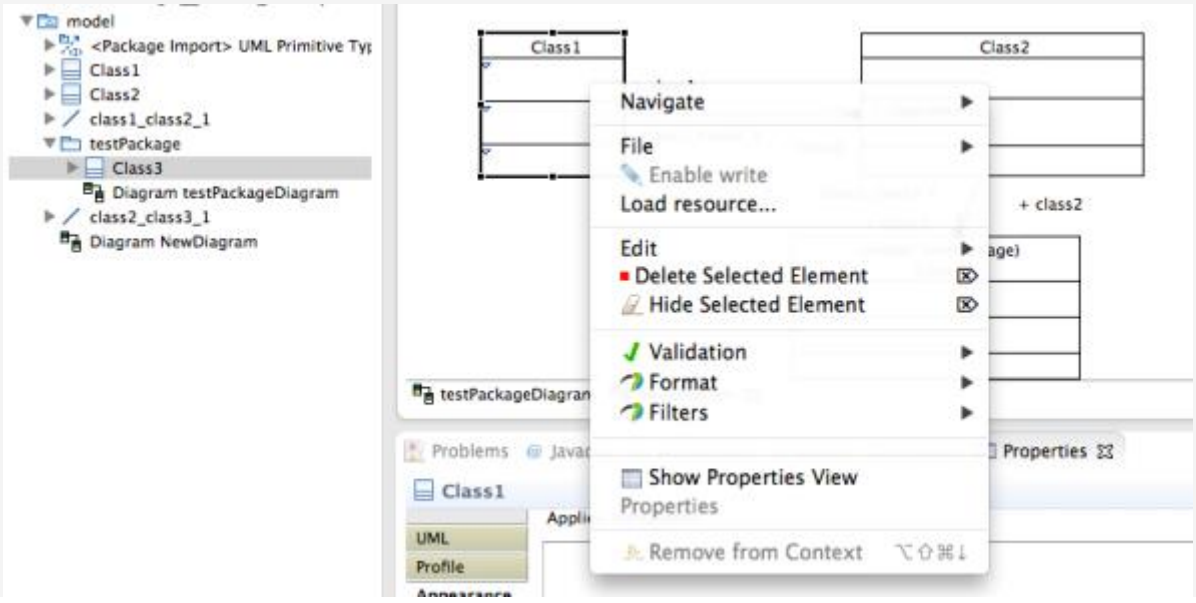


Now we try something harder. Go back to the first diagram, you should know how…, and drag the Class3 from the model to the diagram. As depicted in the following figure. If you want to depict the namespace of the class show the properties view tab, you should know how.., select the Appearance sub-tab of the Properties tab, and choose full for the Qualified Name depth menu. You can now see, in the diagram also,

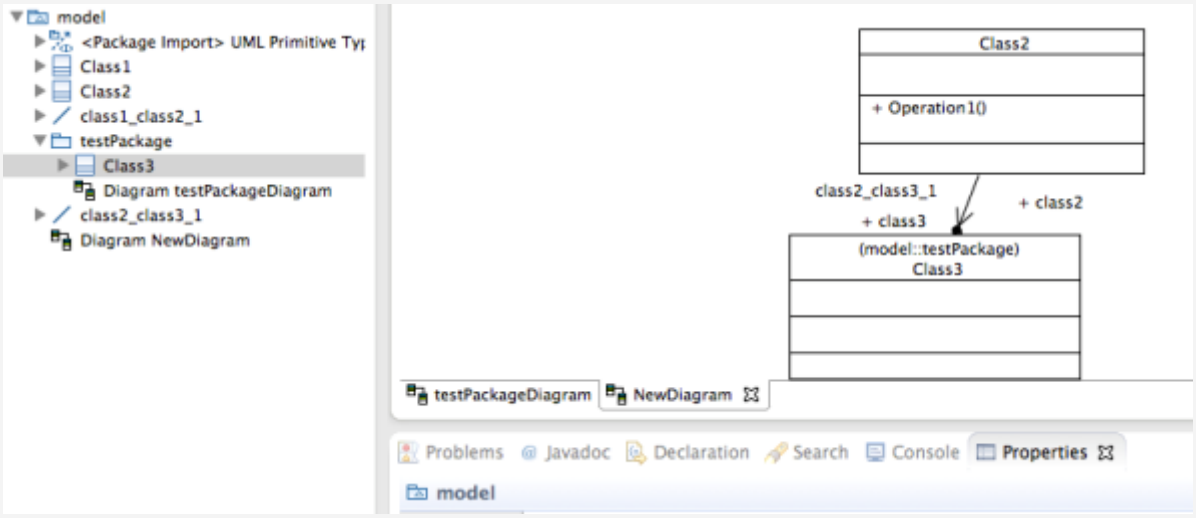
the Class3 belongs to the package called testPackage. You just realized an important thing. The diagrams are not the model.



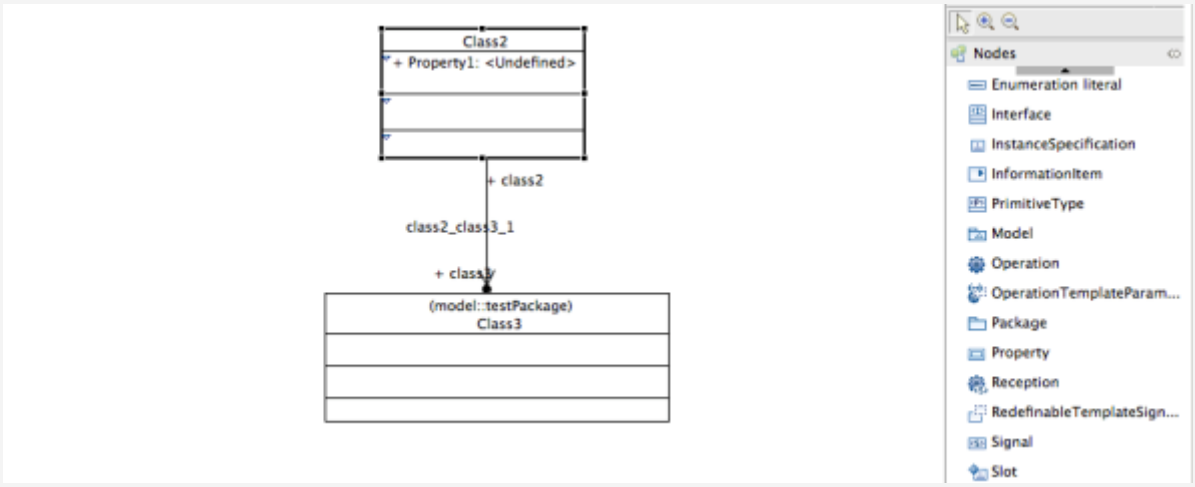
Diagrams are in fact just means to visualize interesting aspects of the models. nothing more. You can choose to hide things from a diagram by right clicking on an element to hide and choosing Hide Selected Element.



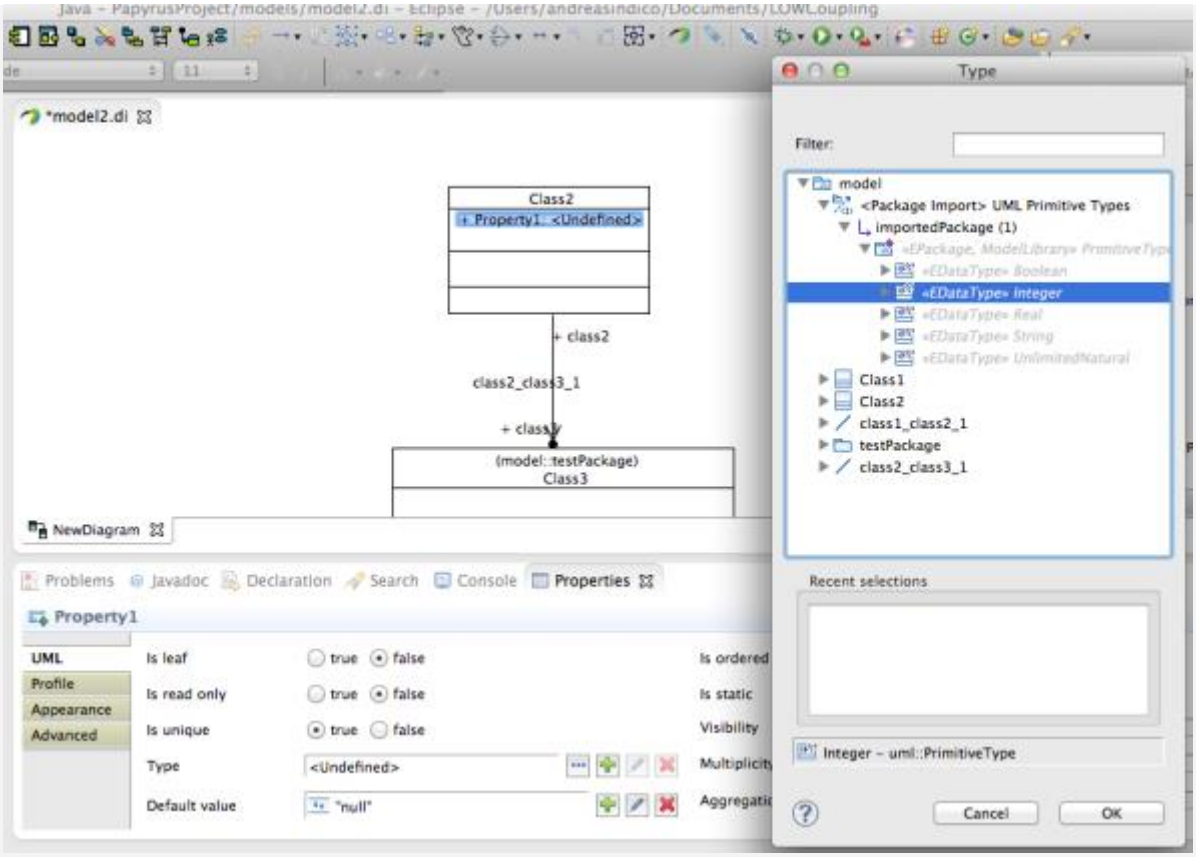
As you can see in the following figure the element will disappear from the diagram but not from the model. If you want to have it back in the diagram just drag and drop it.



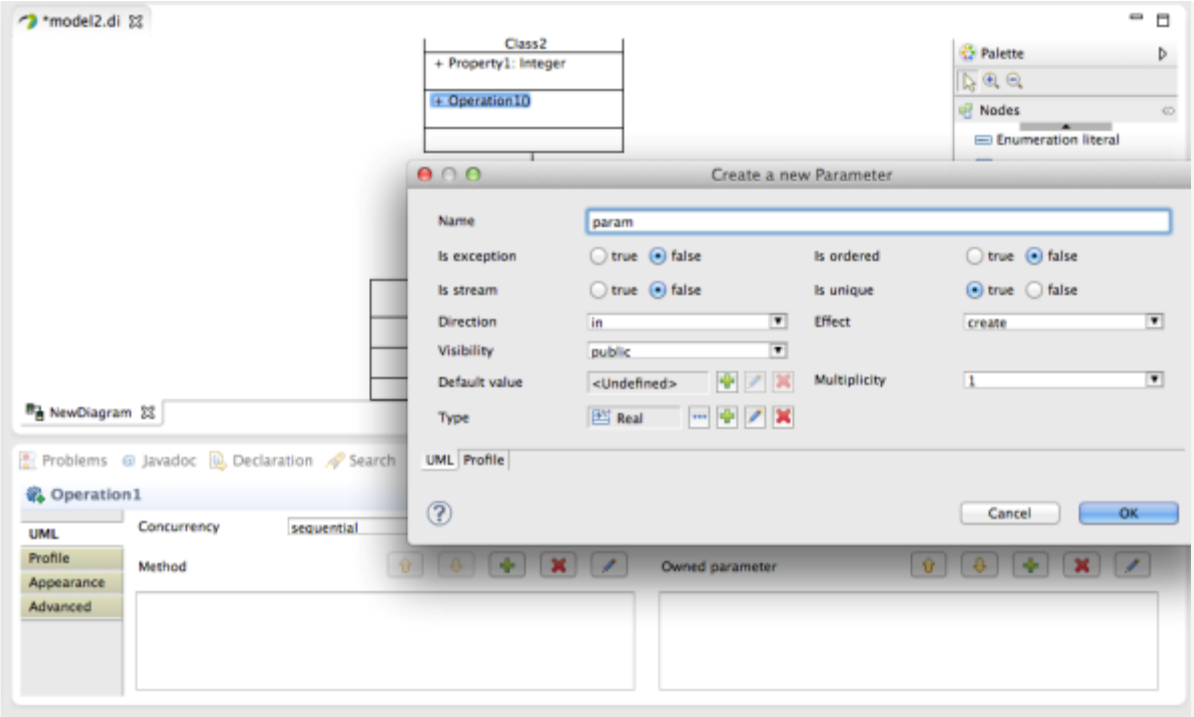
If you instead select Delete Selected Element the element will be both deleted from the diagram and the model. Now try to add a Property to a Class and set the name you want for it



On the properties tab you may find the Type box. Press the … button and a tree dialog will popup. If you want to give the property a primitive type expand the related imported package as show in the figure ahead and choose the primitive type you want.



For Operation's parameters is basically the same. To add a parameter to an Operation select it and, in the Properties tab, press the + button on the top of the Owned parameter box. Then in the dialog that will popup set all the parameter characteristics included its type



Heads Up!

If you have found this post useful please consider to share it with your friends

inShare

and to reccomend it on Google

thank you !!!