



## How to Keep Java and Class Diagram in Sync with Round-trip Engineering?

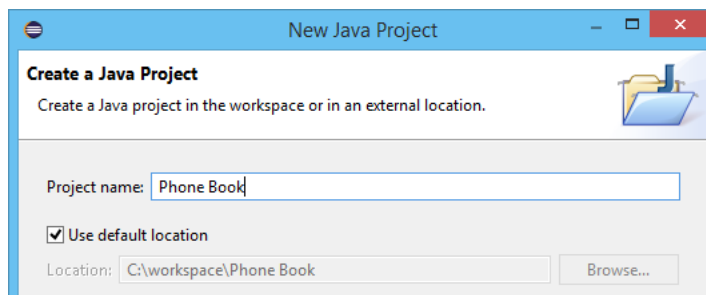
Written Date : March 03, 2016

### Preparation

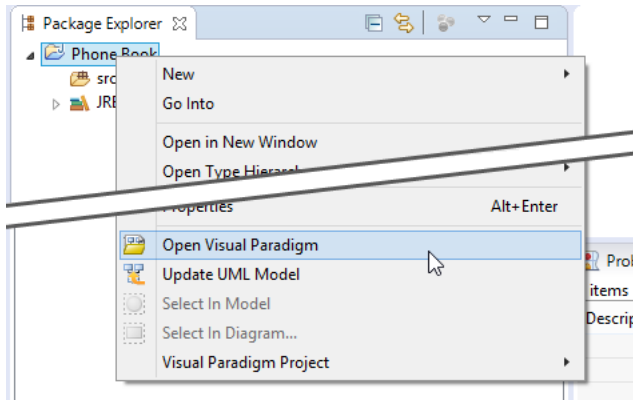
To follow and complete this tutorial, you must have Visual Paradigm installed, which can be downloaded from the Visual Paradigm [download page](#). Of course, you also need the Eclipse IDE. We assume you have already installed it, but if you haven't, please download it from [www.eclipse.org](http://www.eclipse.org) and install it. Additionally, ensure you have [installed the Eclipse integration from Visual Paradigm](#). Finally, to make the tutorial easier to follow, we will not describe every little step required to draw a class diagram in detail. We are assuming that you have the basic skills required to draw a [UML class diagram](#) in Visual Paradigm.

### Designing a System with a UML Class Diagram

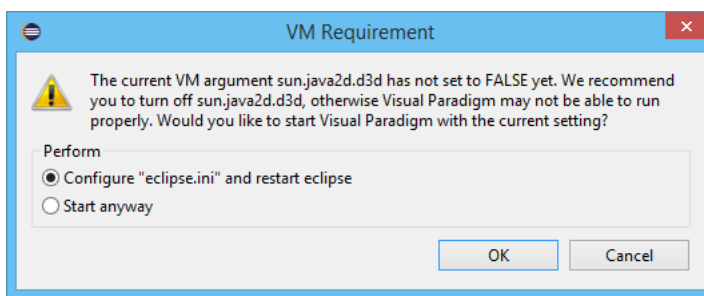
1. Create a Java project named *Phone Book* in Eclipse.



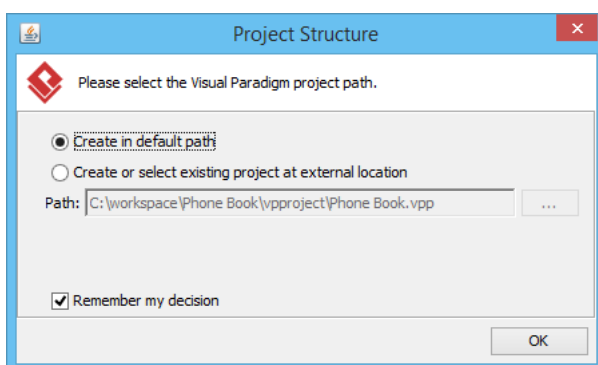
2. Right-click on the project node in **Package Explorer** and select **Open Visual Paradigm** from the popup menu.



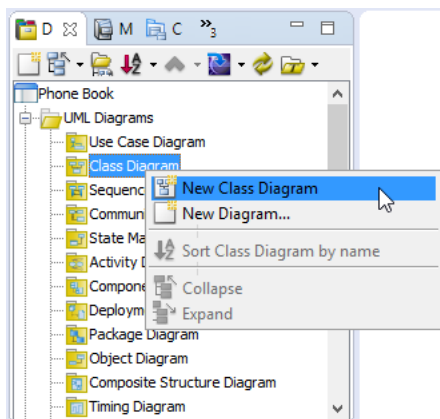
3. If you see the **VM Requirement** dialog box, please keep the option **Configure "eclipse.ini"** and **restart eclipse** selected and click **OK** to restart Eclipse. Then, re-perform the previous step to open Visual Paradigm.



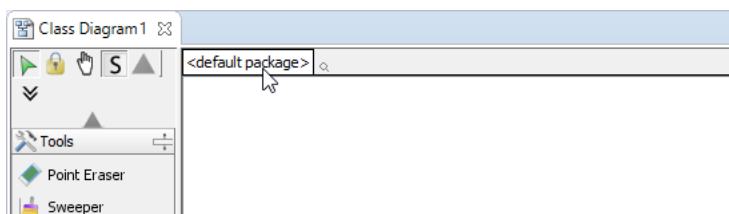
4. Click **OK** when you are prompted to select a path to store the .vpp file.



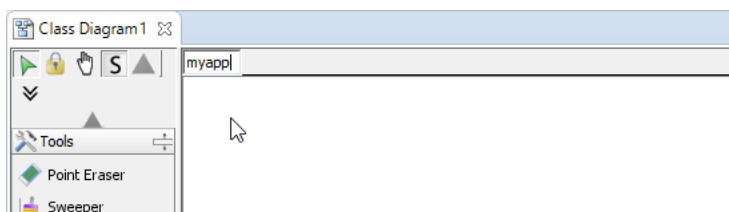
5. In **Diagram Navigator**, right-click on the **Class Diagram** node and select **New Class Diagram** from the popup menu.



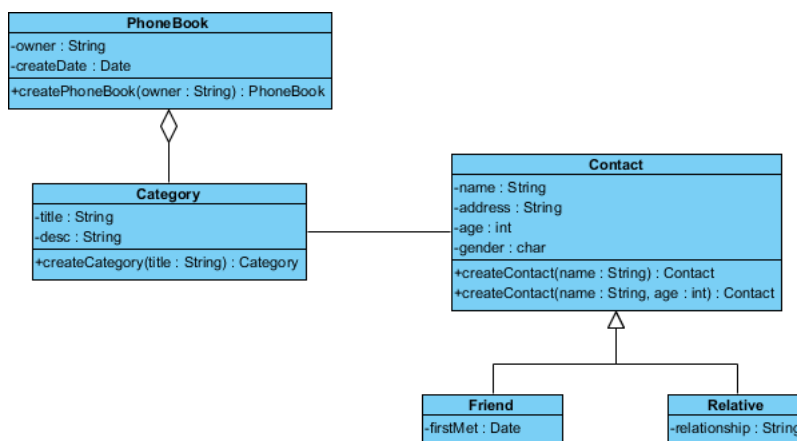
6. A new diagram is created. Double-click on the package header field at the top left corner of the diagram, labeled **<default package>**.



7. Enter *myapp* and press **Enter**. From now on, classes drawn in this diagram will be placed in a (new) package named *myapp*. In practice, you can also enter a nested package structure like *com.vp.myapp*.



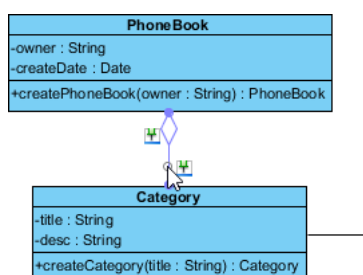
8. Draw the following class diagram.



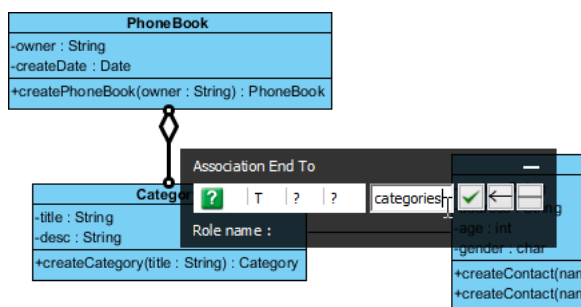
As mentioned earlier, we want to focus on integration with Eclipse in this tutorial and will not detail the steps related to diagramming. If you have difficulties drawing this diagram, here are some tips that may help:

- To create a class, select the **Class** tool from the diagram toolbar and click on the diagram.
- To add an attribute, right-click on the class and select **Add > Attribute** from the popup menu. Type the name and type inline (e.g., *owner : String*) and press **Enter** to confirm creation.
- To add an operation, right-click on the class and select **Add > Operation** from the popup menu. Type the method signature inline (e.g., *createPhoneBook(owner:String):PhoneBook*) and press **Enter** to confirm creation.
- To create a related class (e.g., an associated class), use the Resource Catalog. Move your mouse pointer over a class, then press and drag out the **Resource Catalog** icon at the top right of the class.

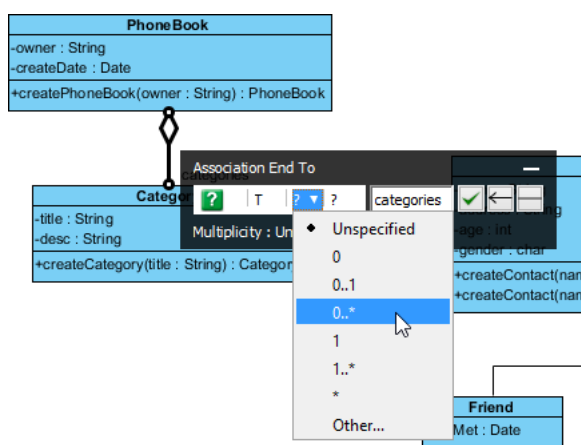
9. Let's set the role names. Double-click on the *Category* end of the connector between class *PhoneBook* and *Category*.



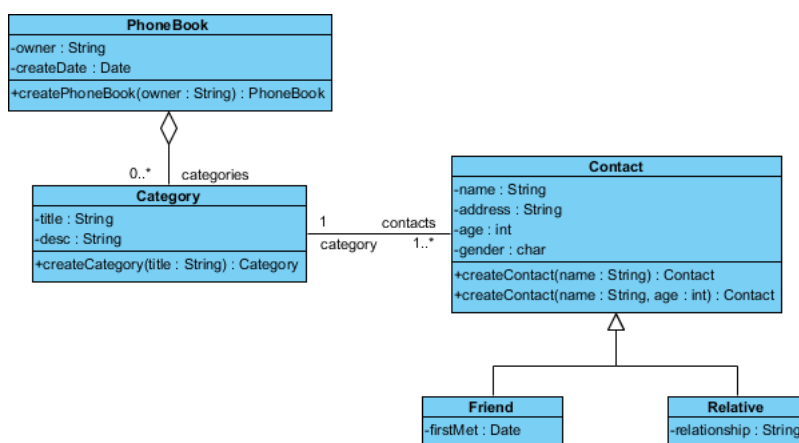
10. Name the role *categories*.



11. Set multiplicity as *0..\**. These changes will affect the category attribute that will be generated in the *PhoneBook* class.



12. Similarly, edit the connector between *Category* and *Contact* by setting the role names of the two ends to "category" and "contacts," and multiplicities to *1* and *1..\**, respectively. This is the completed diagram.



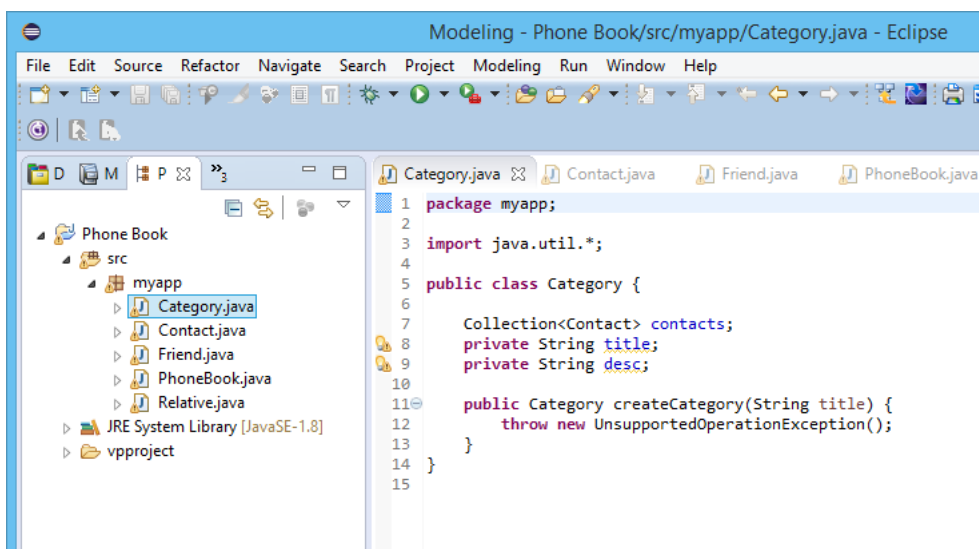
13. Press **Ctrl + S** to save the project.

## UML to Java Code Generation in Eclipse

1. Now comes the code generation. Click the **Update Code** button on the Eclipse toolbar.



2. Check the **Package Explorer**. You should see a list of generated files. You can open them to fill in the code body.



3. This is the end of the tutorial. Instead of closing Eclipse now, you may try something more by editing the code (e.g., adding, renaming, or deleting classes, attributes, and operations) and clicking the **Update UML Model** button, which is next to the **Update Code** button, and observing the changes that will be made in the class model. Enjoy!

### Related Links

- [User's Guide - Overview and installation of Eclipse integration](#)
- [Tutorial - Begin UML Modeling in Eclipse](#)
- [Tutorials on Visual Paradigm IDE Integration](#)
- [Full set of UML tools and UML diagrams](#)



Visual Paradigm home page  
(<https://www.visual-paradigm.com/>)

Visual Paradigm tutorials  
(<https://www.visual-paradigm.com/tutorials/>)