



How to Use Java Round-Trip Engineering?

Written Date : June 24, 2015

Visual Paradigm supports generating Java source code from UML class model, and keeping source code and UML model synchronized, through the Java round-trip engineering feature. In this tutorial you will be given a project file, and try to generate Java code from it. Then, make change in the code and have the UML class model updated from the code.

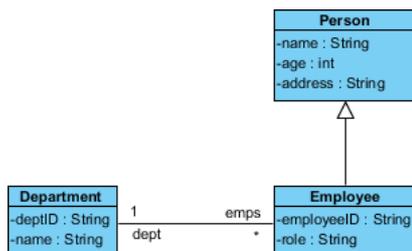
Overview of this Tutorial

In this tutorial, we will show you how to perform round-trip engineering of Java with a given project file.

In order to complete this tutorial you must have Visual Paradigm installed. You also need to have basic knowledge in [UML modeling](#) with Visual Paradigm.

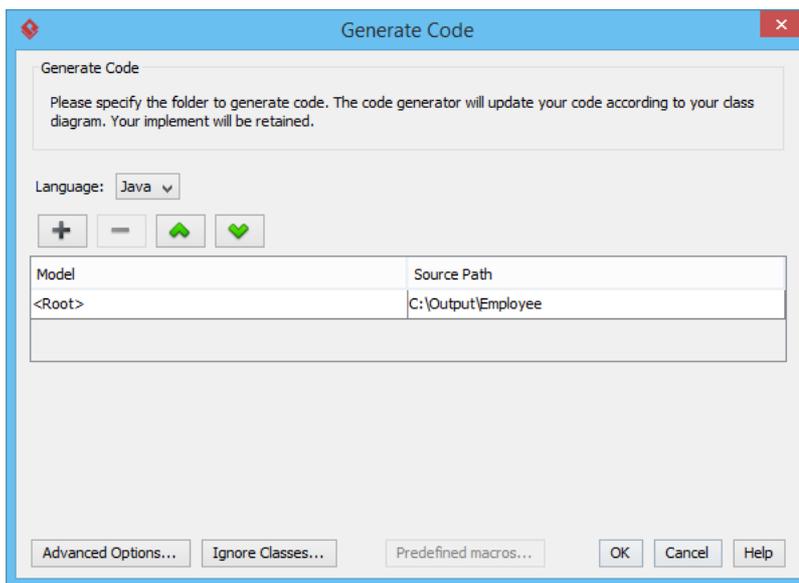
Generating Java Source Files from Class Diagram

1. Download [Employee.vpp](#). You can also find this file at the bottom of this tutorial, under the **Resources** section.
2. Open the downloaded project file in Visual Paradigm.
3. Open the class diagram *Employee*. You should see the class diagram below:

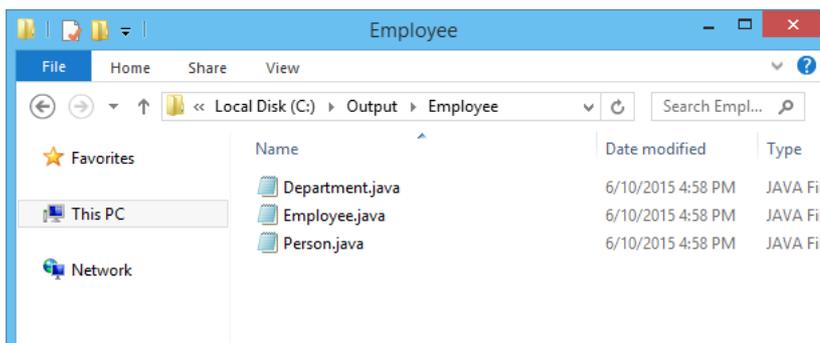


4. There are few ways you can take to generate Java code - generate code from diagram, from selected classes, or for the whole project. In this tutorial let's try to generate code for the whole project. Select **Tools > Code > Generate Java Code** from the toolbar.

5. In the **Generate Code** window, specify the **Source Path**. You can specify a path by clicking on the cell and then click on the ... to popup the file chooser for selecting a folder.



6. Click **OK** to generate. Source files are generated to the output path specified.

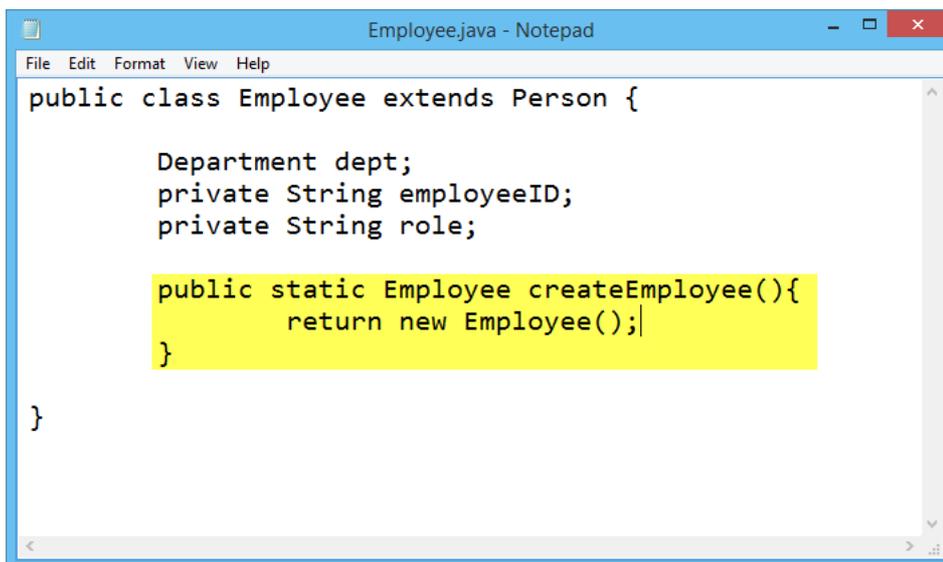


Updating UML Class Model from Java

Once you have finished generating code, you can modify it with your IDE or with any text editor. When you've finished the modification, you can reverse the code-base back to the Visual Paradigm for an update of the UML class model. Let's have a try.

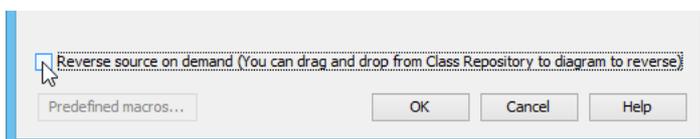
1. Open the generated file *Employee.java* in a text editor.

- Let's say we want to add a static operation for creating an Employee object. Let's write the createEmployee() method in the code as shown in the image below.

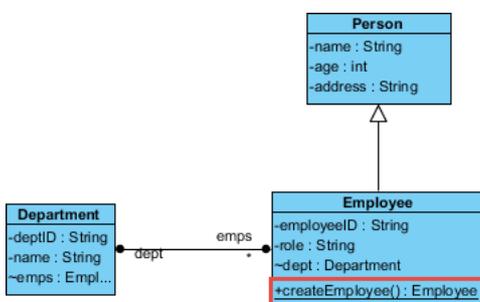


```
public class Employee extends Person {  
  
    Department dept;  
    private String employeeID;  
    private String role;  
  
    public static Employee createEmployee(){  
        return new Employee();  
    }  
  
}
```

- Save the changes.
- Let's update our UML model to reflect the change just made. Select **Tools > Code > Reverse Java Code...** from the toolbar.
- Uncheck **Reverse source on demand** at the bottom of the **Reverse Code** window. If you keep it checked, the UML model won't be updated directly. Instead, an index tree will be formed from the source files specified in the source folder. We want our model updated, so let's uncheck it.



- Click **OK**. Now, the class diagram is updated and you should see the operation createEmployee() added to the Employee class.



Resources

1. [Employee.vpp](#)

Related Links

- [Code Engineering features in Visual Paradigm](#)
- [State Machine Diagram feature description](#)



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

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