Builder Pattern Tutorial

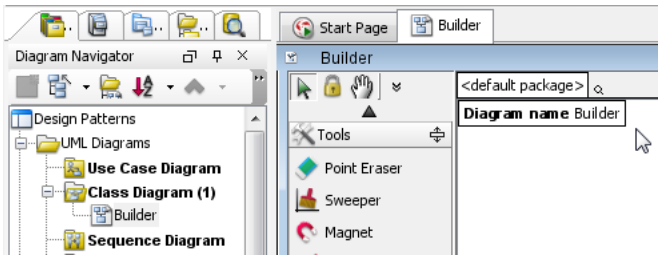
Written Date : September 28, 2009

This tutorial is aimed to guide the definition and application of Gang of Four (GoF) builder design pattern. By reading this tutorial, you will know how to develop a model for the builder pattern, and how to apply it in practice.

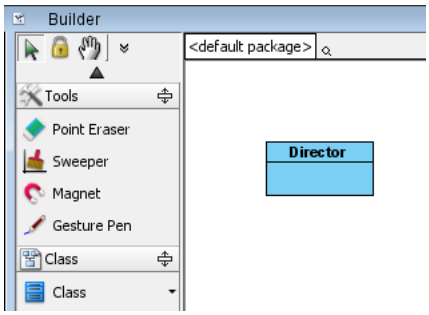
Modeling Design Pattern with Class Diagram

1. Create a new project *Design Patterns*.

2. Create a class diagram *Builder*.



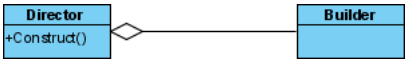
3. Select **Class** from diagram toolbar. Click on the diagram to create a class. Name it as *Director*.



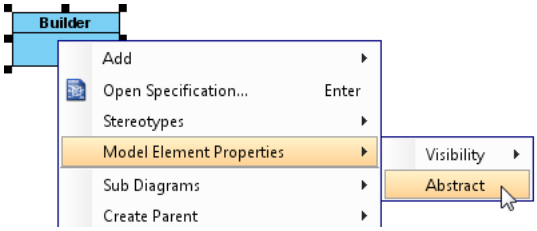
4. Right click on *Director* and select **Add** > **Operation** from the popup menu.

5. Name it as *Construct()*.

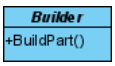
6. Move the cursor over *Director*, make use of resource icon **Aggregation** > **Class** to create an associated class *Builder*.



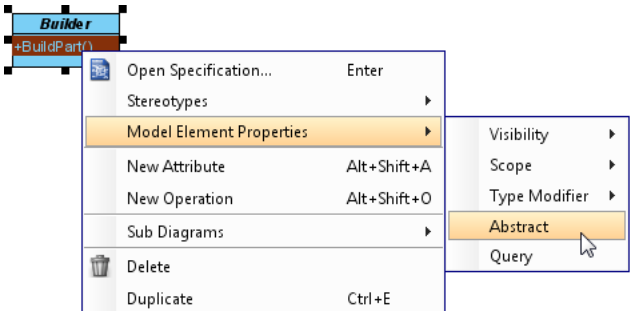
7. Set the *Builder* abstract by right clicking on *Builder* and selecting **Model Element Properties** > **Abstract** from the popup menu.



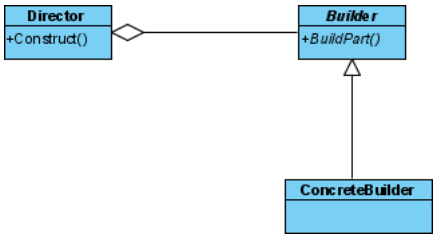
8. Create an operation *BuildPart()* in *Builder*.



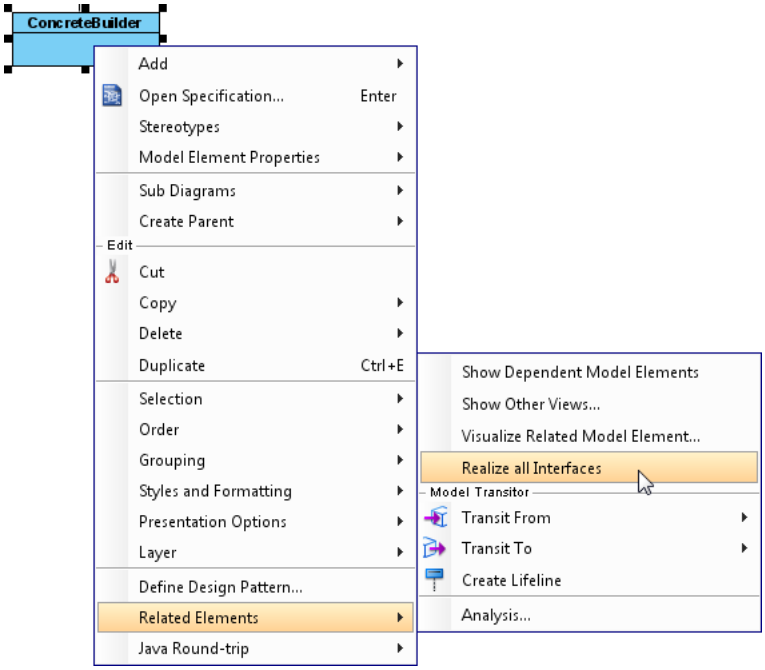
9. Set *BuildPart()* abstract by right clicking on it and selecting **Model Element Properties** > **Abstract** from the popup menu.



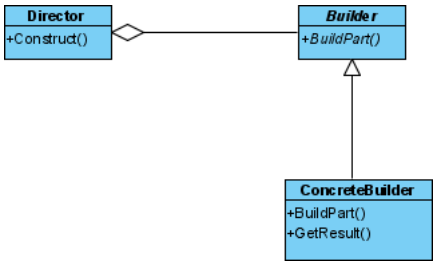
10. Move the cursor over *Builder*, make use of resource icon **Generalization** > **Class** to create a subclass *ConcreteBuilder*.



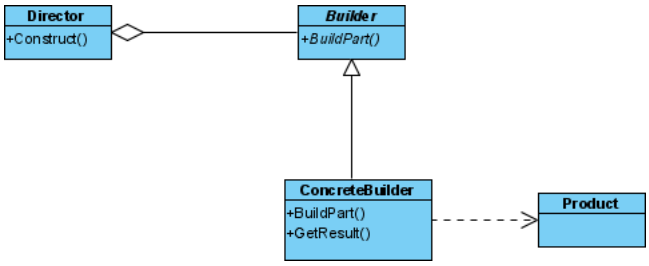
11. Inherit operations from *Builder* by right clicking on *ConcreteBuilder* and selecting **Related Elements** > **Realize all Interfaces** from the popup menu.



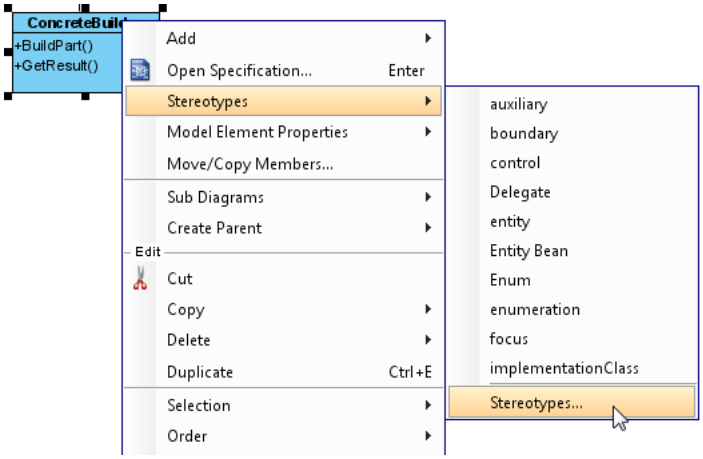
12. Add an operation *GetResult()* to *ConcreteBuilder*. Up to now, the diagram should look like this:



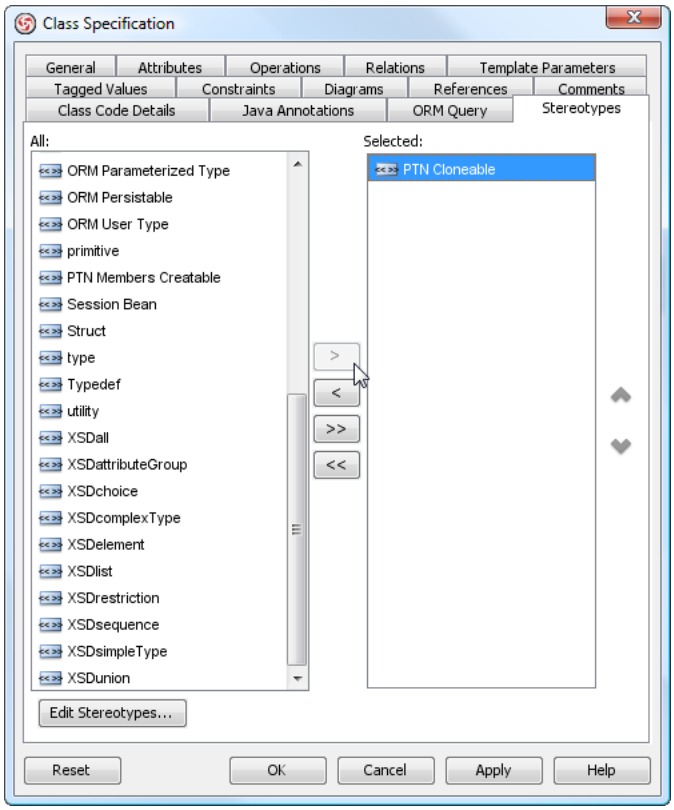
13. Move the cursor over *ConcreteBuilder*, make use of resource icon **Dependency** > **Class** to create a depending class *Product*.



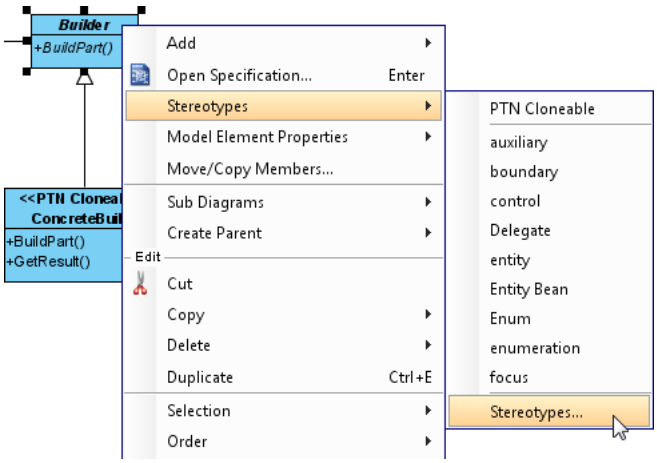
14. In practice, there can be multiple concrete builder classes. To represent this, we need to assign a special stereotype to the *ConcreteBuilder* class. Right click on *ConcreteBuilder* and select **Stereotypes** > **Stereotypes...** from the popup menu.



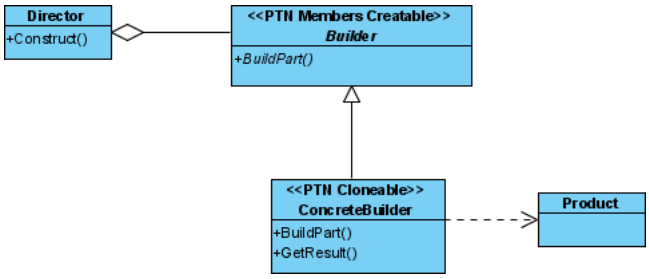
15. Select **PTN Cloneable**, click > to assign it to the **Selected Stereotype** list. Click **OK** to confirm.



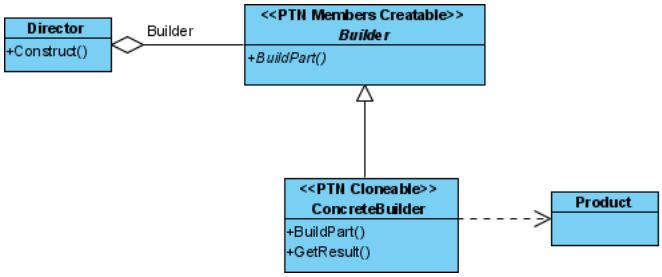
16. In practice, there may be multiple to operations in the *Builder* class for creating parts. To represent this, we need to assign a special stereotype to the *Builder* class. Right click on *Builder* and select **Stereotypes** > **Stereotypes...** from the popup menu.



17. Assign stereotype **PTN Members Creatable** to the class. Click **OK** to confirm.

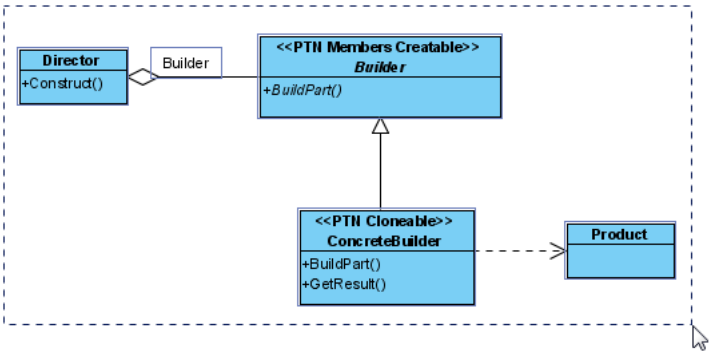


18. For the association end of the association between the *Director* and *Builder* classes, there is a role named *Builder*. Double click on the *Director* end and enter *Builder* as name. The diagram should now become:

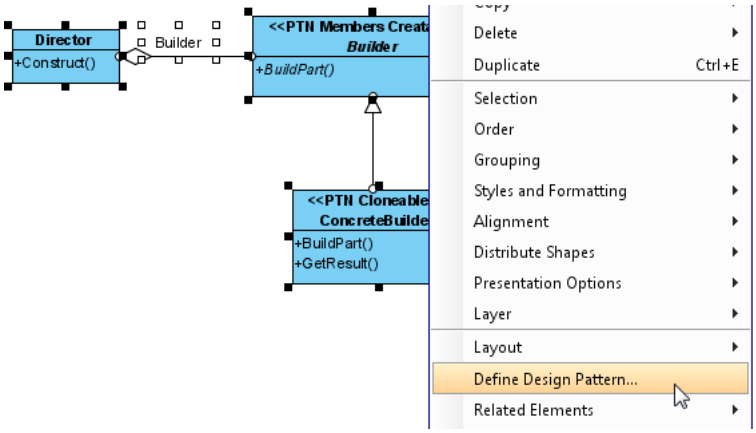


Defining Pattern

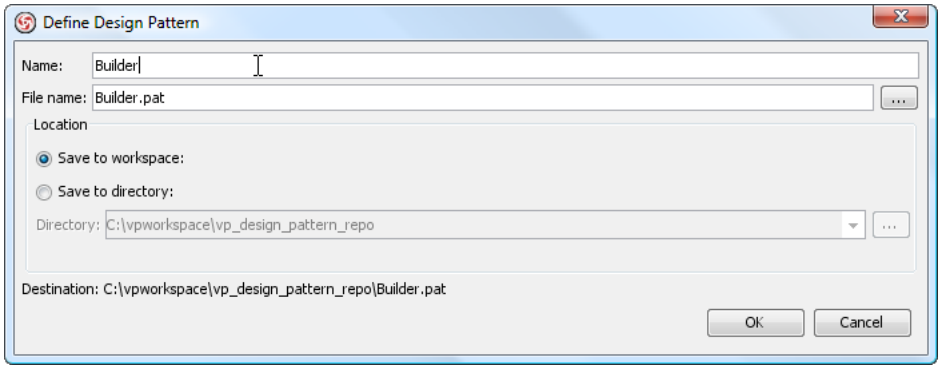
1. Select all classes on the class diagram.



2. Right click on the selection and select **Define Design Pattern...** from the popup menu.



3. In the **Define Design Pattern** dialog box, specify the pattern name *Builder*. Keep the file name as is. Click **OK** to proceed.

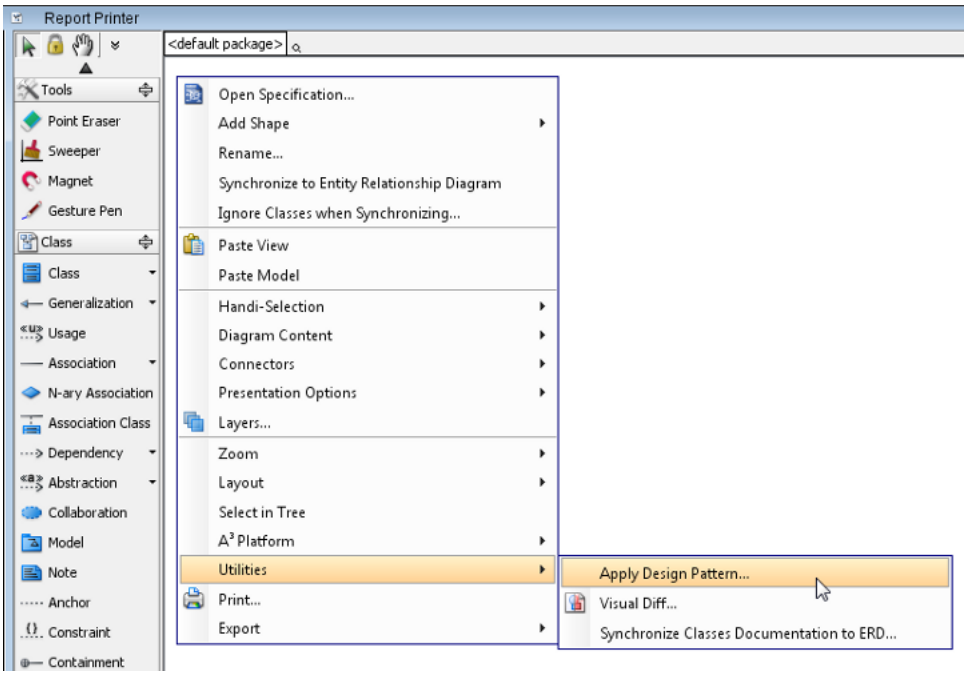


Applying Design Pattern on Class Diagram

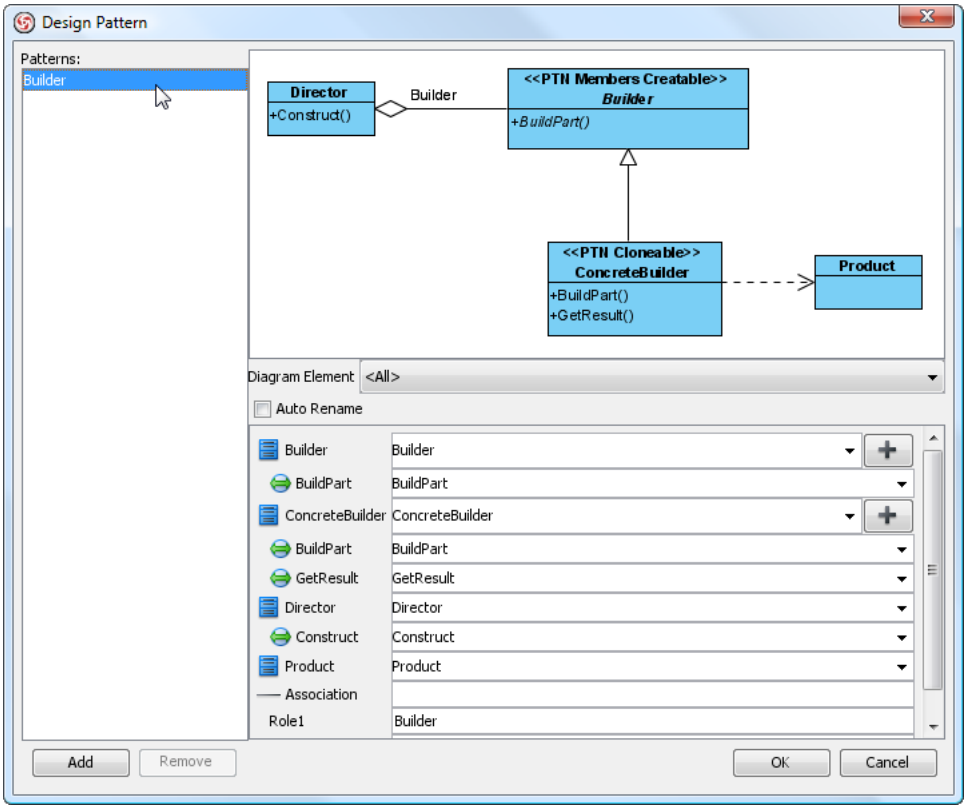
In this section, we are going to apply the builder pattern in modeling a report printing system that offers printing PDF, RTF, and HTML report one after the other, with same content in them.

1. Create a new project *Report Printer*.
2. Create a class diagram *Report Printer*.

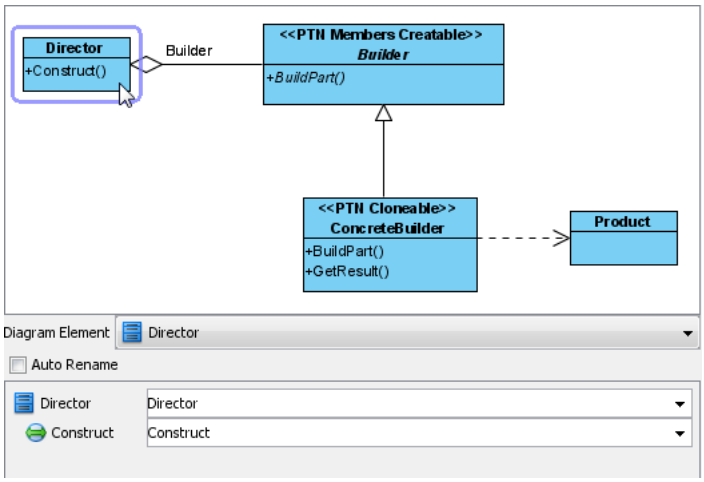
3. Right click on the class diagram and select **Utilities** > **Apply Design Pattern...** from the popup menu.



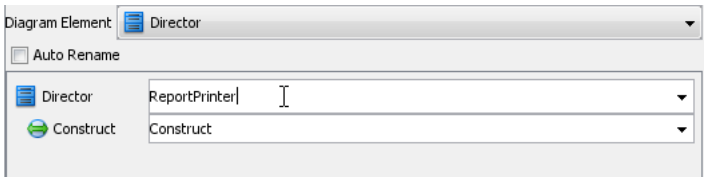
4. In the **Design Pattern** dialog box, select *Builder* from the list of patterns.



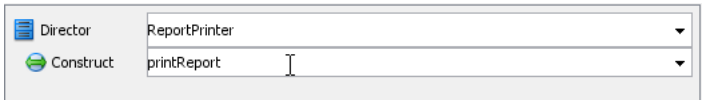
5. Click on *Director* in the preview.



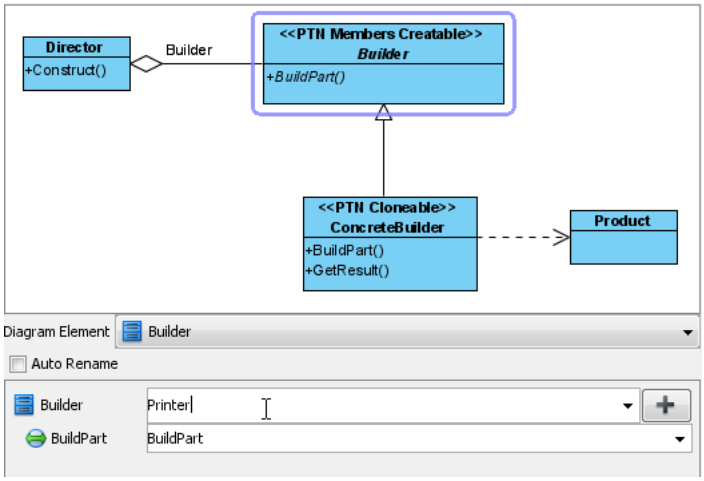
6. Rename *Director* to *ReportPrinter* at the bottom pane.



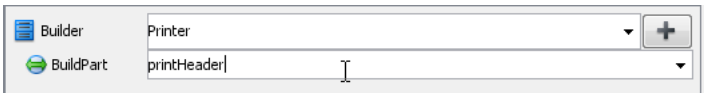
7. Rename the operation *Construct* to *printReport*.



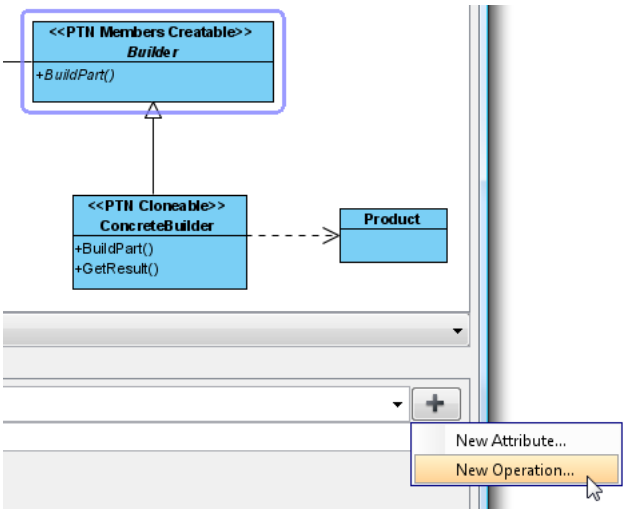
8. Rename the class *Builder* to *Printer*.



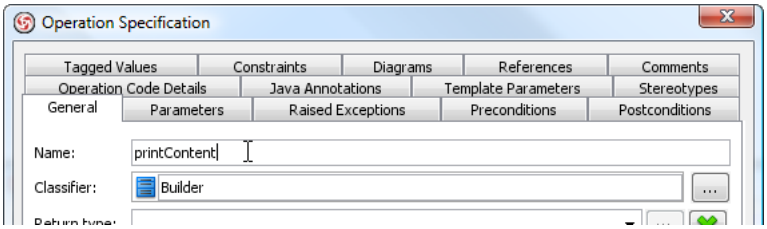
9. Rename the operation *BuildPart* to *PrintHeader*.



10. We need three operations in the *Printer* class for printing header, content and footer. Press the + button, and select **New Operation...** to add more operations.

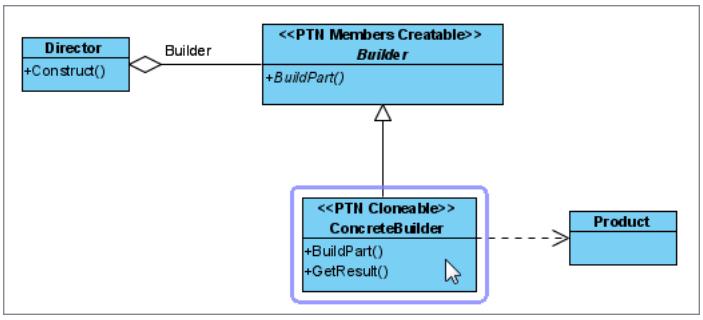


11. In the **Operation Specification**, name the operation printContent. Check **Abstract** at the bottom of dialog box. Click **OK** to confirm.

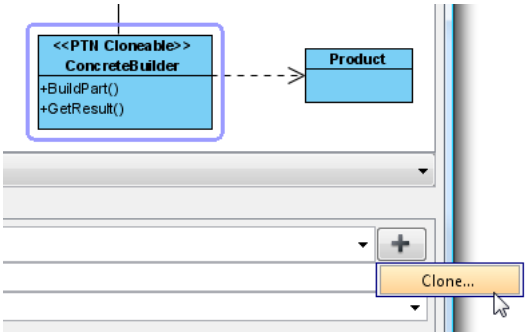


12. Repeat steps 10 and 11 to create the *printFooter* method.

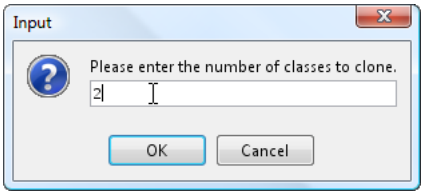
13. Now, clone the *ConcreteBuilder* class to make it have 3 subclasses as printers for PDF, RTF and HTML reports. Select *ConcreteBuilder* in preview pane first.



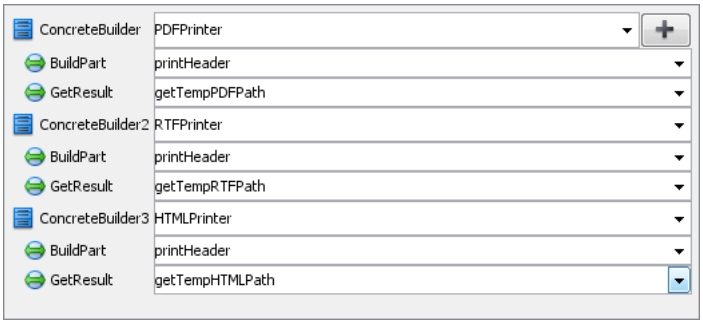
14. Click on the + button and select **Clone**.



15. Select **2** as the number of classes to clone.

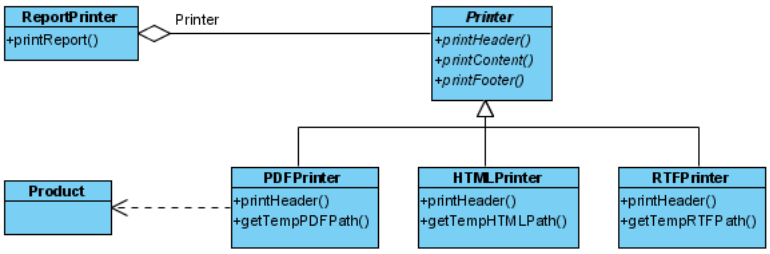


16. Rename the classes and operations as shown below:

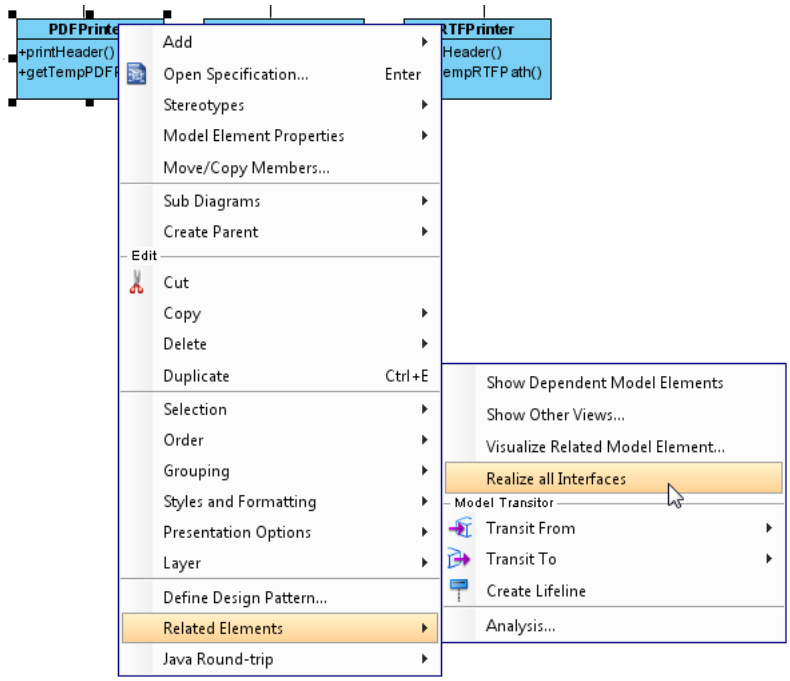


17. Click **OK** to confirm the changes and apply the pattern on diagram.

18. Tidy up the diagram to make it looks like below:

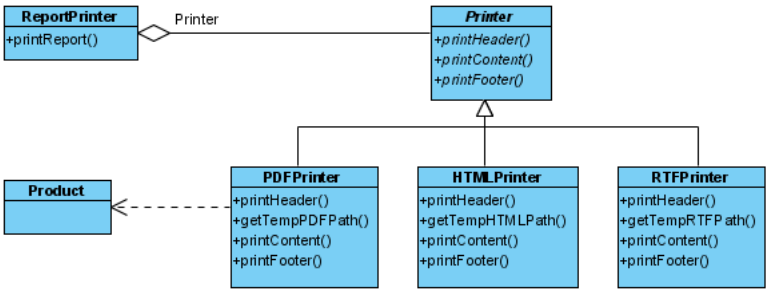


19. Inherits the operations from the *Printer* class to the concrete printer classes. Right click on *PDFPrinter* class and select **Related Elements** > **Realize all Interfaces** from the popup menu.



20. Do the same for classes *HTMLPrinter* and *RTFPrinter*.

21. Rename the role *Builder* to *Printer*. The result should look like this:



Resources

1. Builder.pat
2. Design Patterns.vpp

Related Links

- 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/)