Bridge Pattern Tutorial

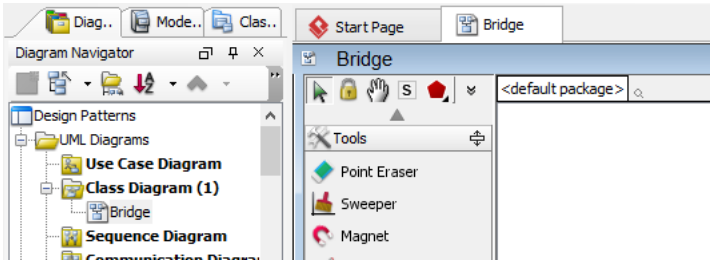
Written Date : October 8, 2009

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

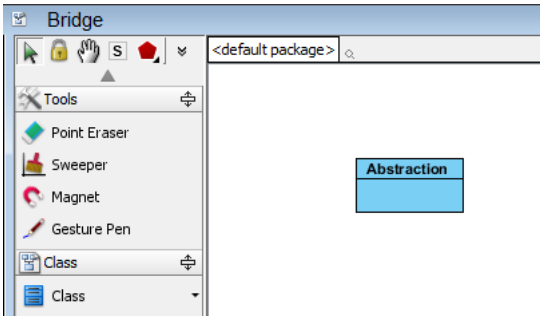
Modeling Design Pattern with Class Diagram

1. Create a new project *Design Patterns*.

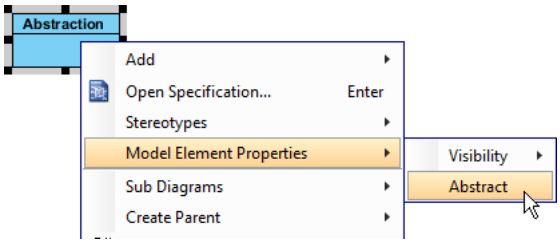
2. Create a class diagram *Bridge*.



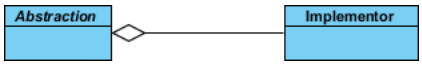
3. Select **Class** from diagram toolbar. Click on the diagram to create a class. Name it as *Abstraction*.



4. Right-click on *Abstraction*, and select **Model Element Properties** > **Abstract** to set it as abstract.

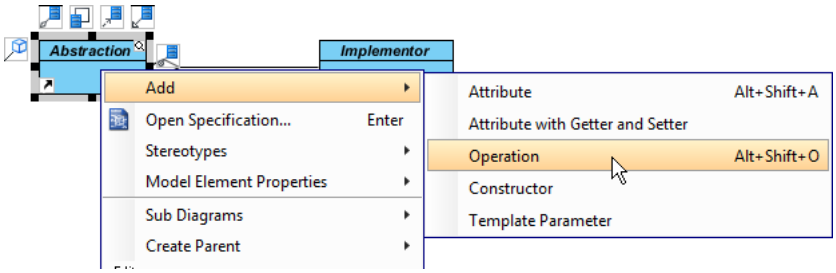


5. Move the mouse cursor over the *Abstraction* class, and drag out **Aggregation** > **Class** to create an associated class *Implementor*.



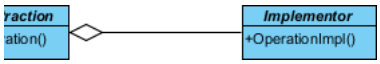
6. Right-click on *Implementor*, and select **Model Element Properties** > **Abstract** to set it as abstract.

7. Right-click on the *Abstract* class, and select **Add** > **Operation** from the popup menu.

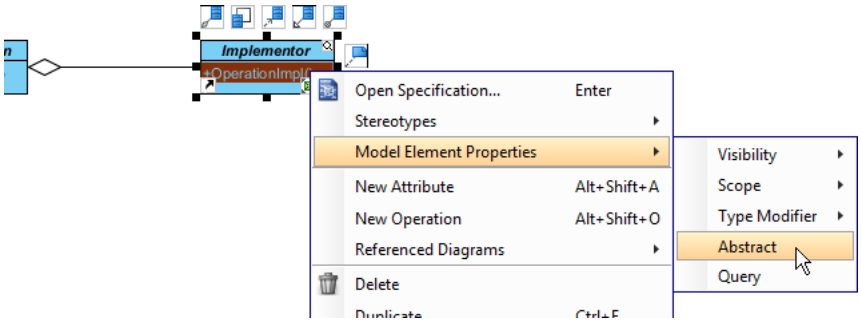


8. Name the operation *Operation()*.

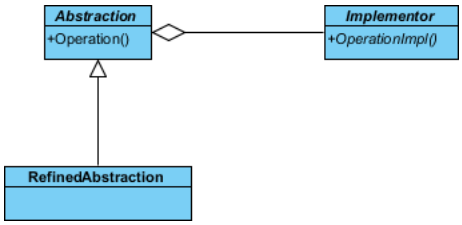
9. Right-click on the *Implementor* class, and select **Add** > **Operation** from the popup menu. Name the operation *OperationImpl()*.



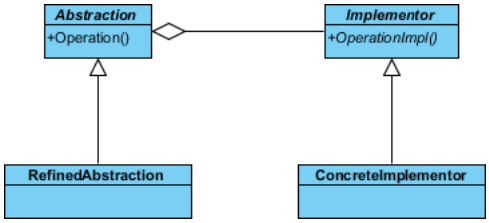
10. Right-click on *Implementor*, and select **Model Element Properties** > **Abstract** to set it as abstract.



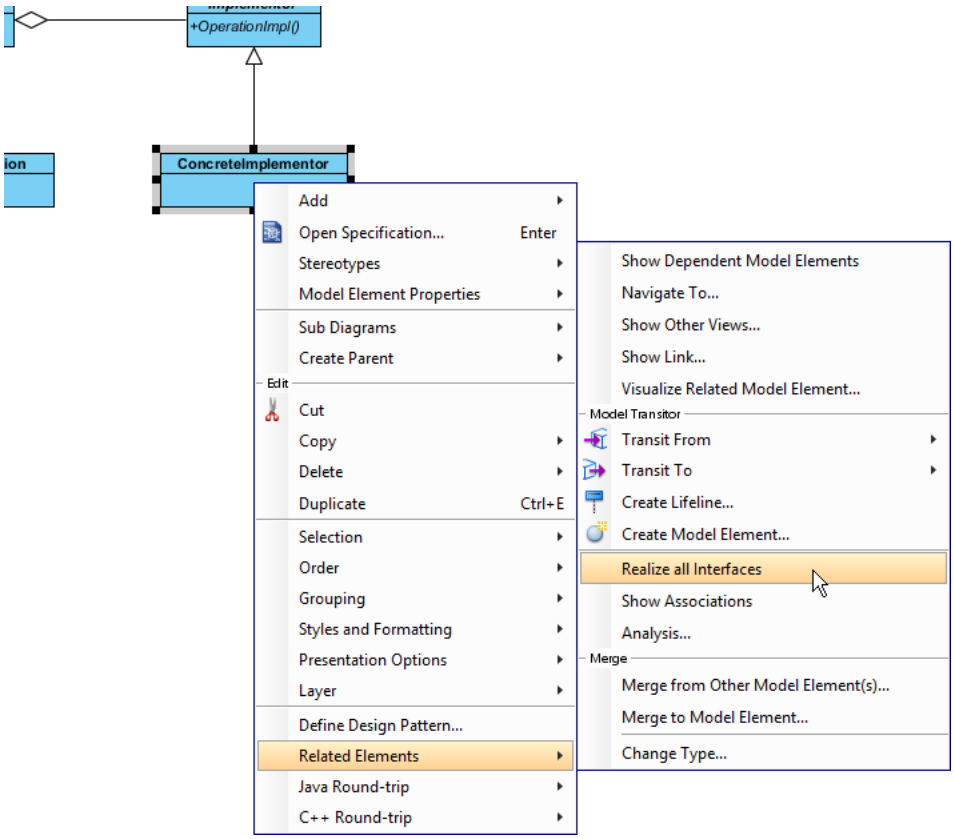
11. Move the mouse cursor over the *Abstraction* class, and drag out **Generalization** > **Class** to create a subclass *RefinedAbstraction*.



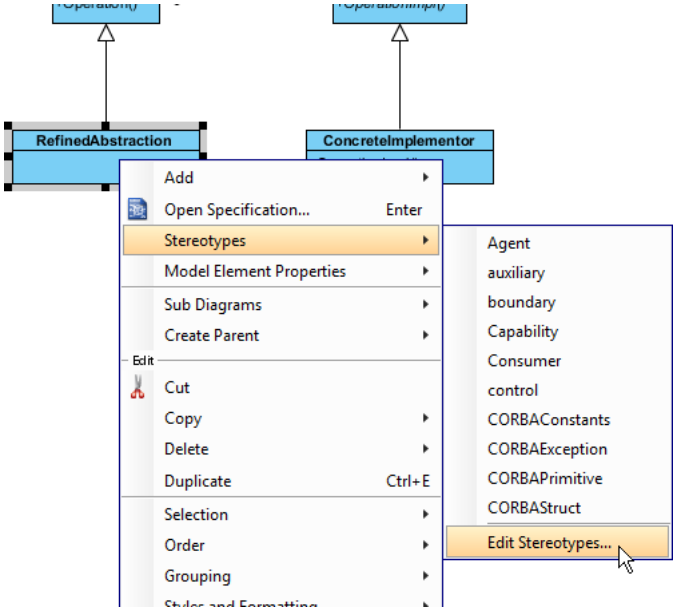
12. Repeat the previous step to create a subclass *ConcreteImplementor* from *Implementor*.



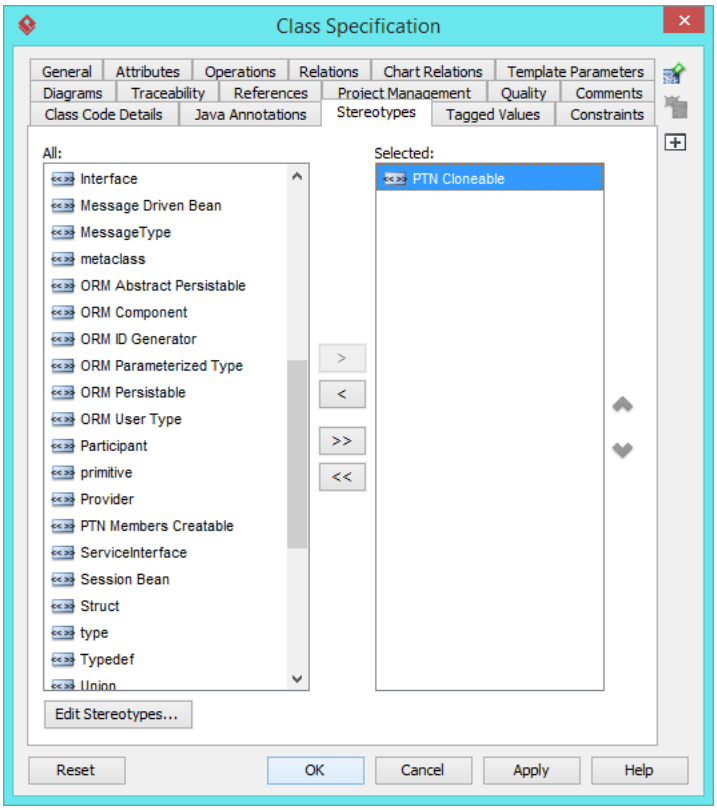
13. *ConcreteImplementor* will inherits the operations from *Implementor*. Right-click on *ConcreteImplementor* and select **Related Elements** > **Realize all Interfaces** from the popup menu.



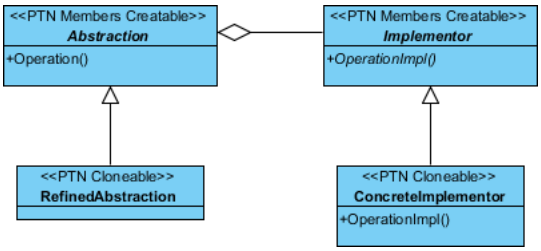
14. In practice, there may be multiple refined abstractions and/or concrete implementors. To represent this, stereotype the class *RefinedAbstraction* and *ConcreteImplementor* as **PTN Cloneable**. Right-click on *Abstraction* and select **Stereotypes** > **Stereotypes...** from the popup menu.



15. In the **Stereotypes** tab of the **Class Specification** dialog box, select **PTN Cloneable** and click > to assign it to *RefinedAbstraction* class. Click **OK** to confirm.



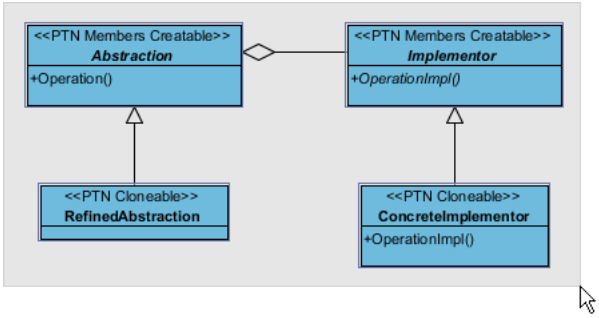
16. Repeat steps 14 and 15 on *concreteImplementor*.



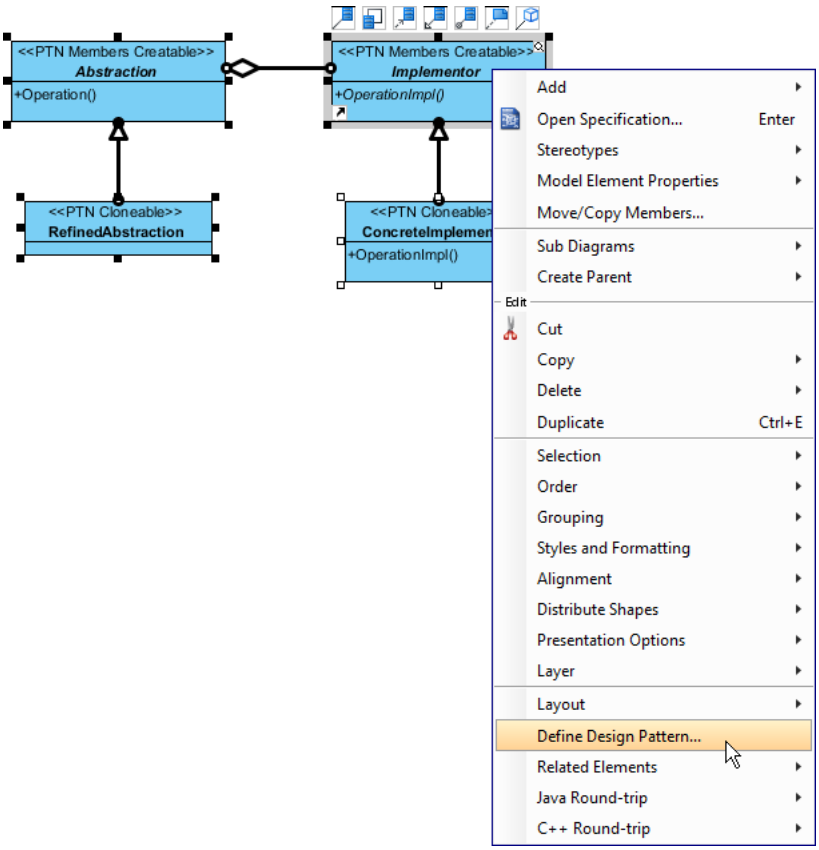
17. In practice, there may be multiple operations and/or operationImpls. To represent this, stereotype the class *Abstraction* and *Implementor* as **PTN Members Creatable**. Repeat steps 14 and 15 to stereotype *Abstraction* and *Implementor* as **PTN Members Creatable**.

Defining Pattern

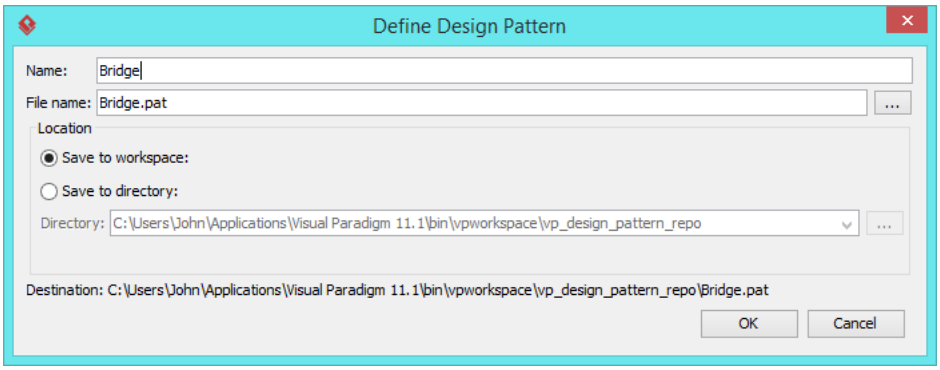
1. Select all classes on the class diagram.



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



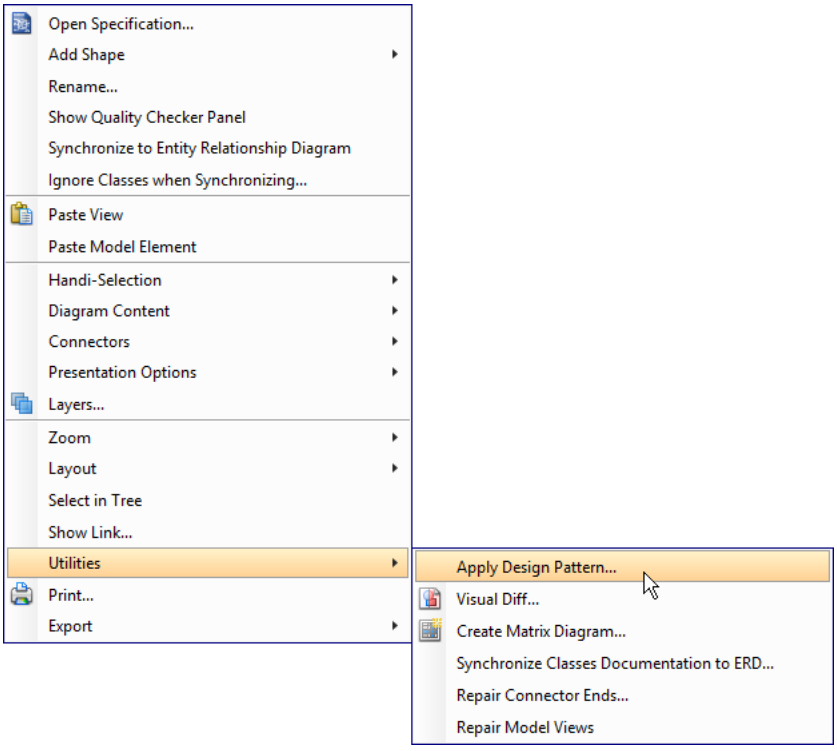
3. In the **Define Design Pattern** dialog box, specify the pattern name *Bridge*. Keep the file name as is. Click **OK** to proceed.



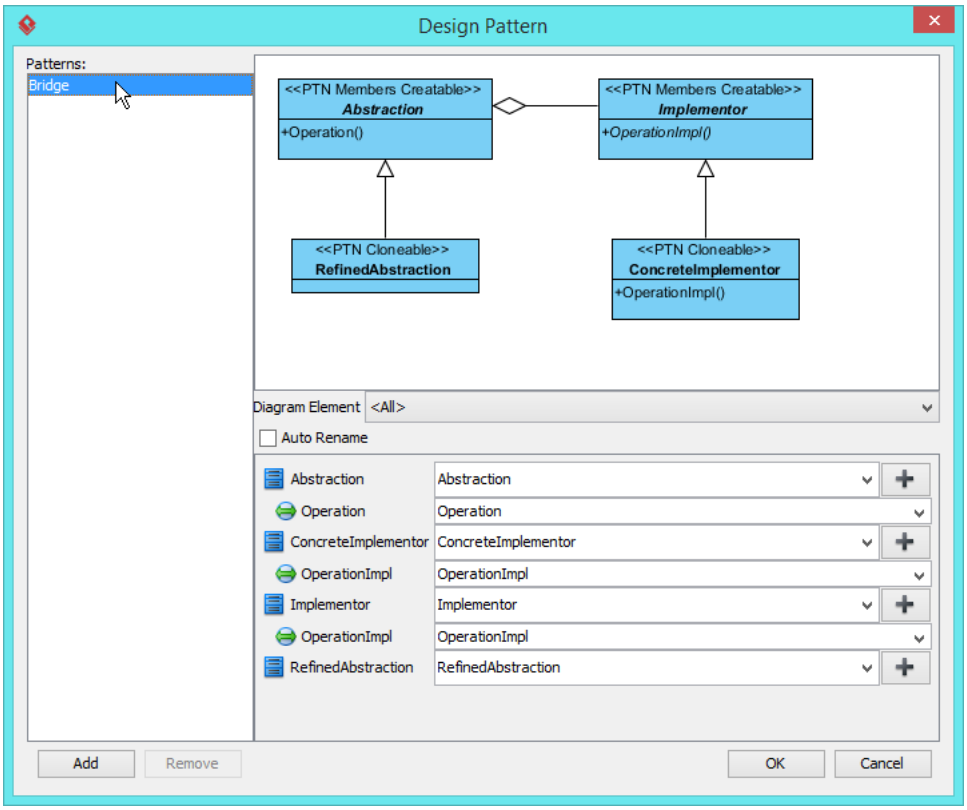
Applying Design Pattern on Class Diagram

In this section, we are going to apply the bridge pattern to model a report generator for various report types.

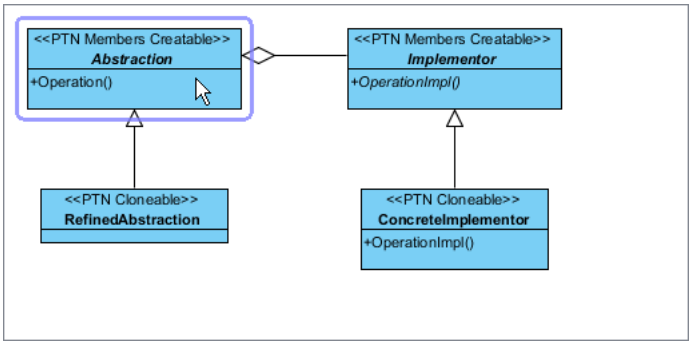
1. Create a new project *Diagram Editor*.
2. Create a class diagram *Domain Model*.
3. Right-click on the class diagram and select **Utilities** > **Apply Design Pattern...** from the popup menu.



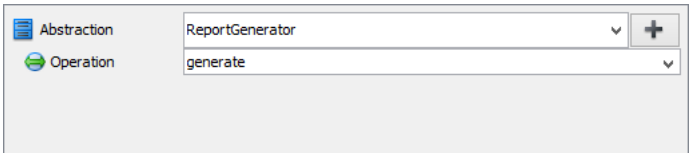
4. In the **Design Pattern** dialog box, select *Bridge* from the list of patterns.



5. Click on *Abstraction* in the overview.

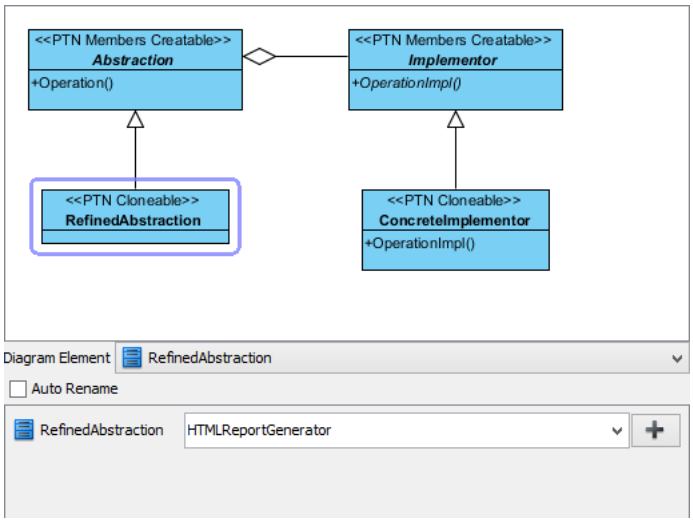


6. Rename *Abstraction* to *ReportGenerator* and operation *Operation* to *generate* at the bottom pane.



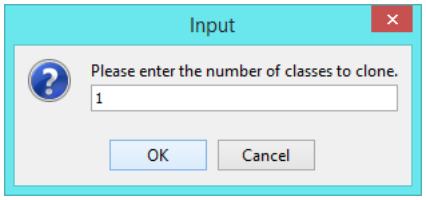
7. Select *RefinedAbstraction* in the overview pane.

8. Rename *RefinedAbstraction* to *HTMLReportGenerator*.

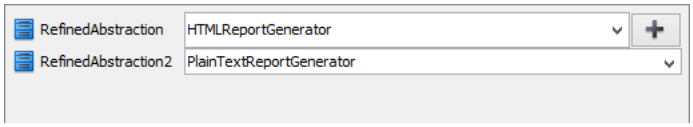


9. Besides HTML report generator, we need also a Plain Text report generator. Click on the + button at the bottom pane, beside *Abstraction*, and select **Clone...**

10. Enter *1* to be the number of classes to clone. Click **OK** to confirm.

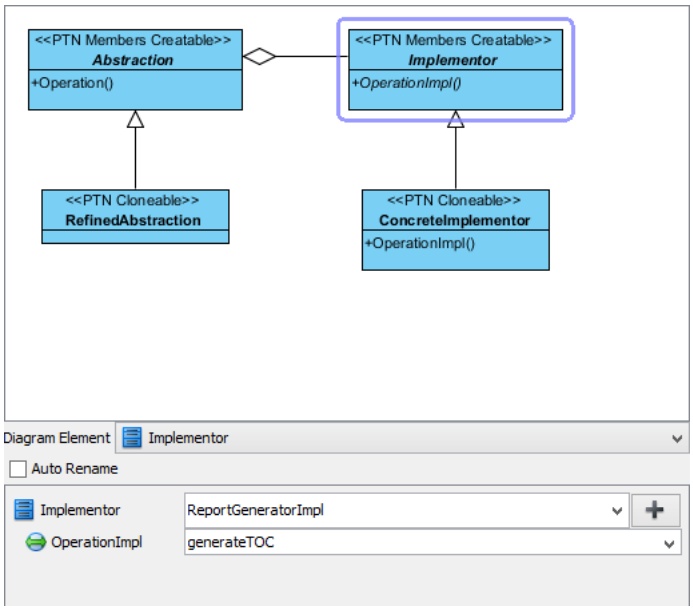


11. Rename *RefinedAbstraction2*, the cloned class, to *PlainTextReportGenerator*.

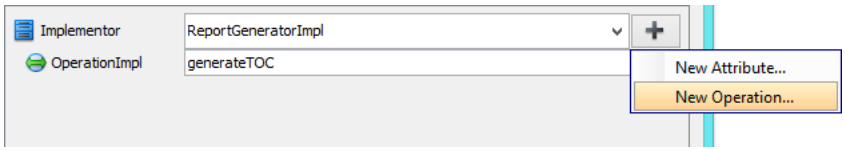


12. Select Implementor in the overview pane.

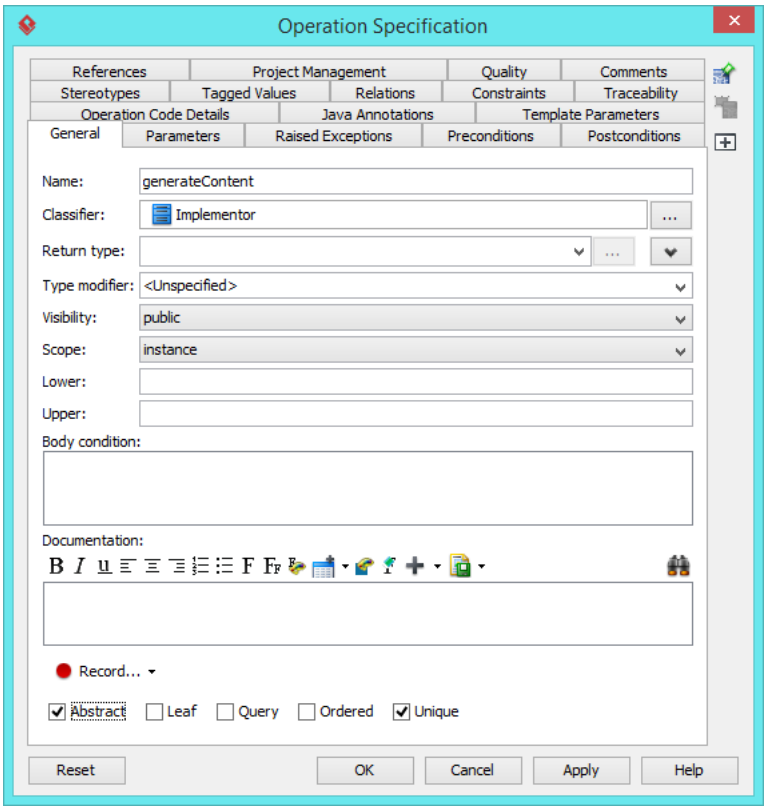
13. Rename *Implementor* to *ReportGeneratorImpl*, and *OperationImpl* to *generateTOC*.



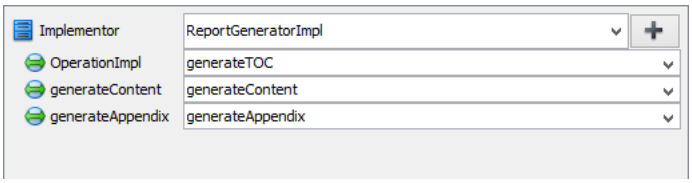
14. We need 2 more operations for generating content and appendix. Click on the + button and select **New Operation...** from the popup meun.



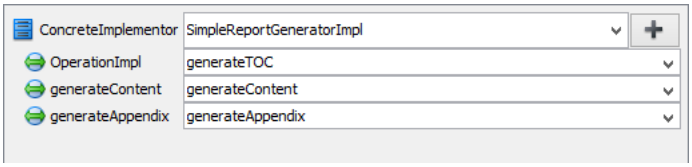
15. In the **Operation Specification** dialog box, name the operation *generateContent*. Check **Abstract** at the bottom of the dialog box.



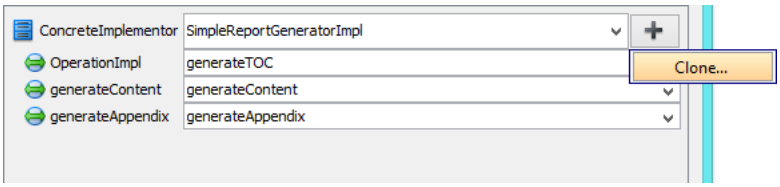
16. Repeat the previous steps to create another abstract operation *generateAppendix*.



17. Select *ConcreteImplementor* in overview. Rename *ConcreteImplementor* to *SimpleReportGeneratorImpl*, and operation *OperationImpl* to *generateTOC*.

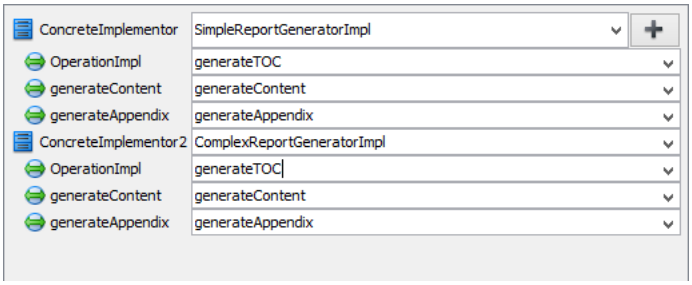


18. Similar to *RefinedAbstraction*, here we need to have another concrete implementor for generating complex report. Click on the + button and select **Clone...** from the popup menu.

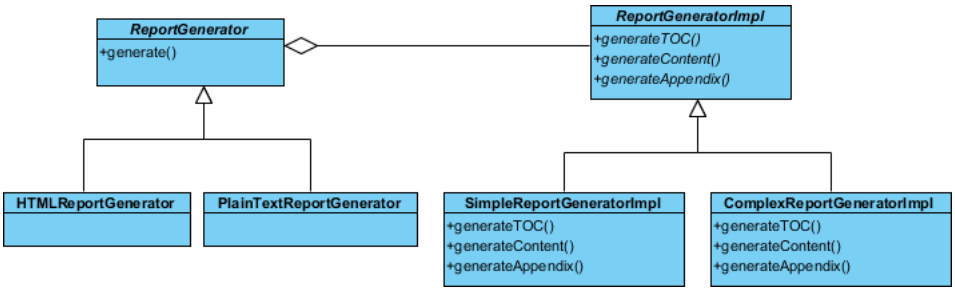


19. Enter *1* to be the number of classes to clone. Click **OK** to confirm.

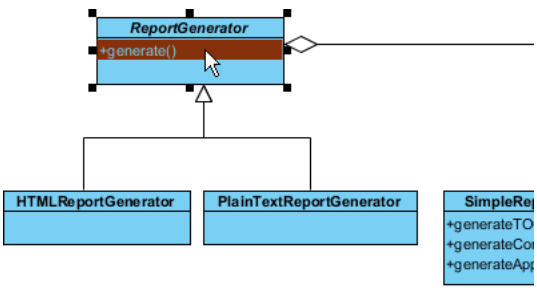
20. Rename *ConcreteImplmentor2*, the cloned class, to *ComplexReportGeneratorImpl*, and operation *OperationImpl* to *generateTOC*.



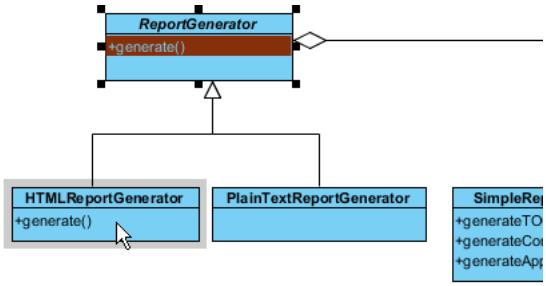
21. Click **OK** to confirm. Here is the diagram formed:



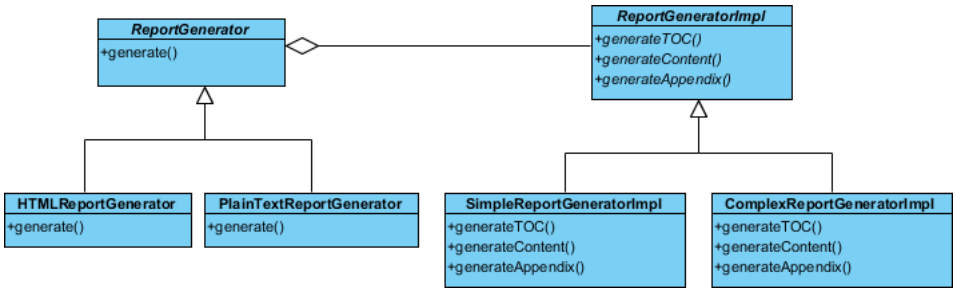
22. We want *HTMLReportGenerator* and *PlainTextReportGenerator* to implement their own way of generating repot. Select the generate operation is *ReportGenerator*.



23. Press the **Ctrl** key, and drag to *HTMLReportGenerator*. Replease the mouse button afterwards.



24. Repeat the previous steps to create the generate method in *PlainTextReportGenerator*. Here is the completed diagram:



Resources

1. Bridge.pat
2. Design Patterns.vpp

Related Links

- Full set of UML tools and UML diagrams

Visual Paradigm

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

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