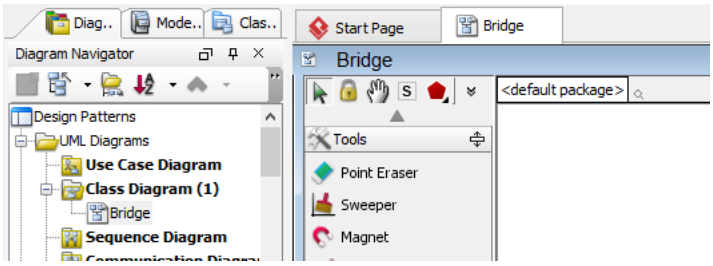
# Bridge Pattern Tutorial

Written Date : October 8, 2009

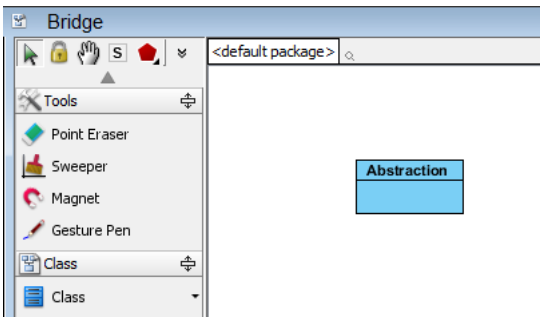
## Modeling a Design Pattern with a Class Diagram

1. Create a new project called *Design Patterns*.

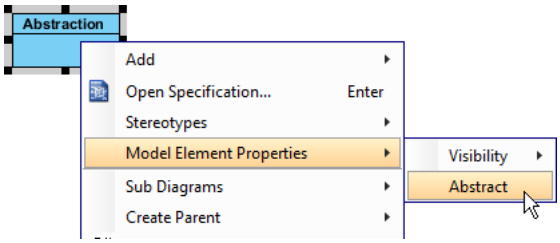
2. Create a class diagram named *Bridge*.



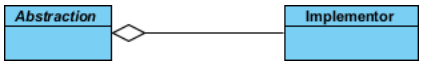

3. Select **Class** from the diagram toolbar. Click on the diagram to create a class. Name it *Abstraction*.



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

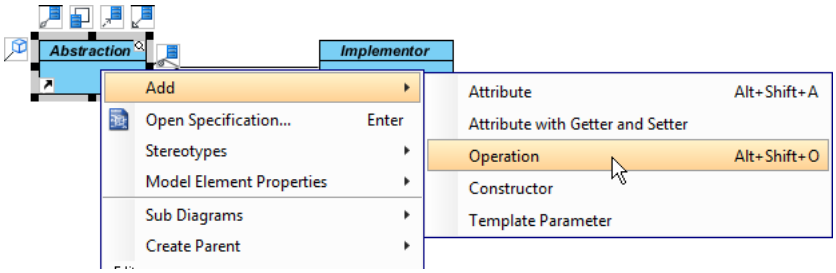



5. Move the mouse cursor over the *Abstraction* class and drag out **Aggregation > Class** to create an associated class named *Implementor*.



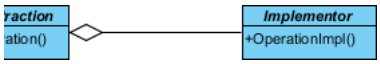

6. Right-click on *Implementor* and select **Model Element Properties > Abstract** to set it as abstract.

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

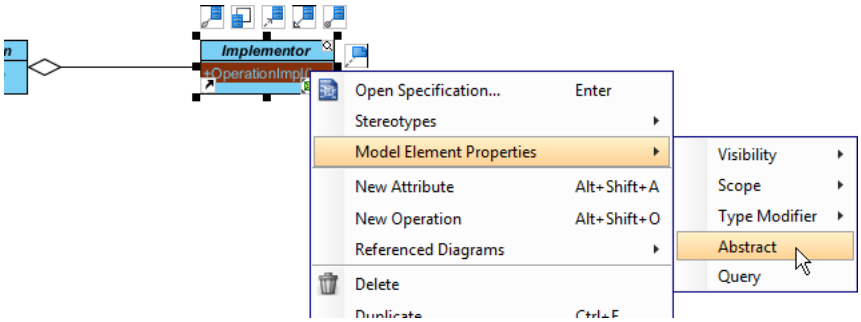



8. Name the operation *Operation()*.

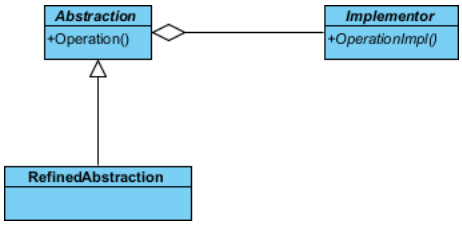
9. Right-click on the *Implementor* class and select **Add > Operation** from the popup menu. Name the operation *OperationImpl()*.



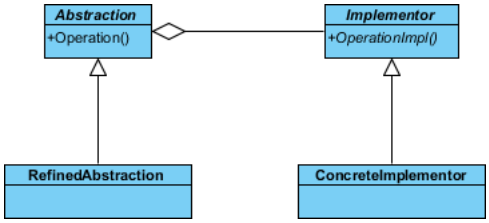
10. Right-click on *Implementor* and select **Model Element Properties > Abstract** to set it as abstract.



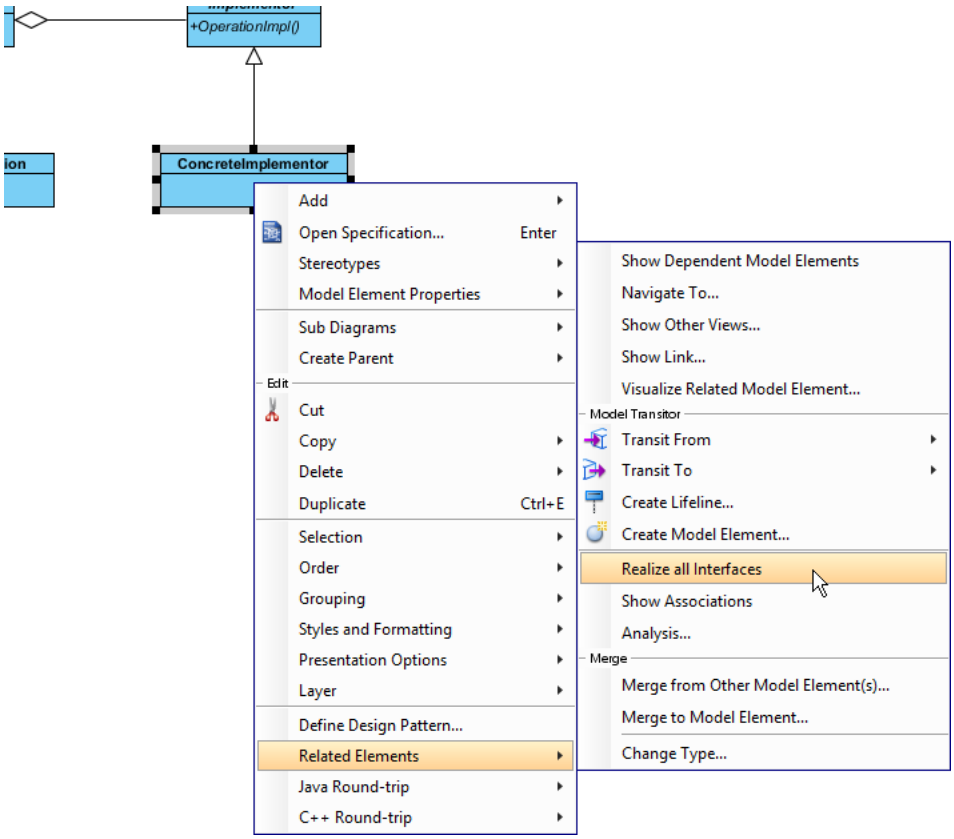

11. Move the mouse cursor over the *Abstraction* class and drag out **Generalization > Class** to create a subclass named *RefinedAbstraction*.



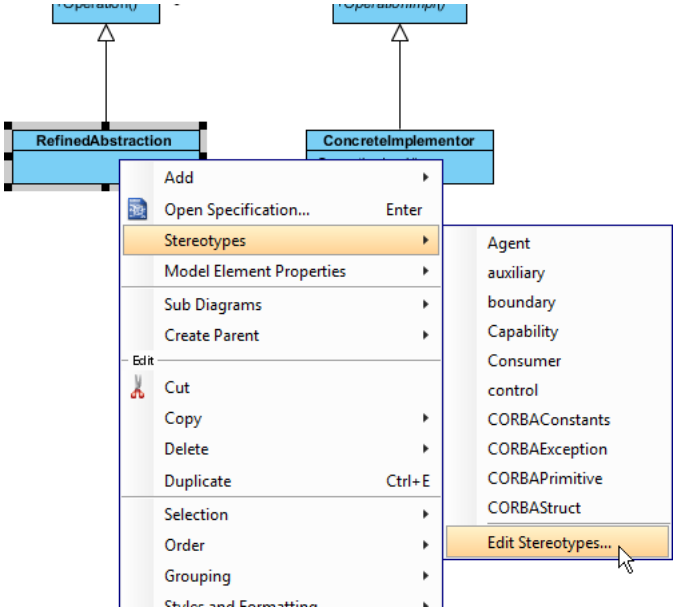

12. Repeat the previous step to create a subclass named *ConcreteImplementor* from *Implementor*.



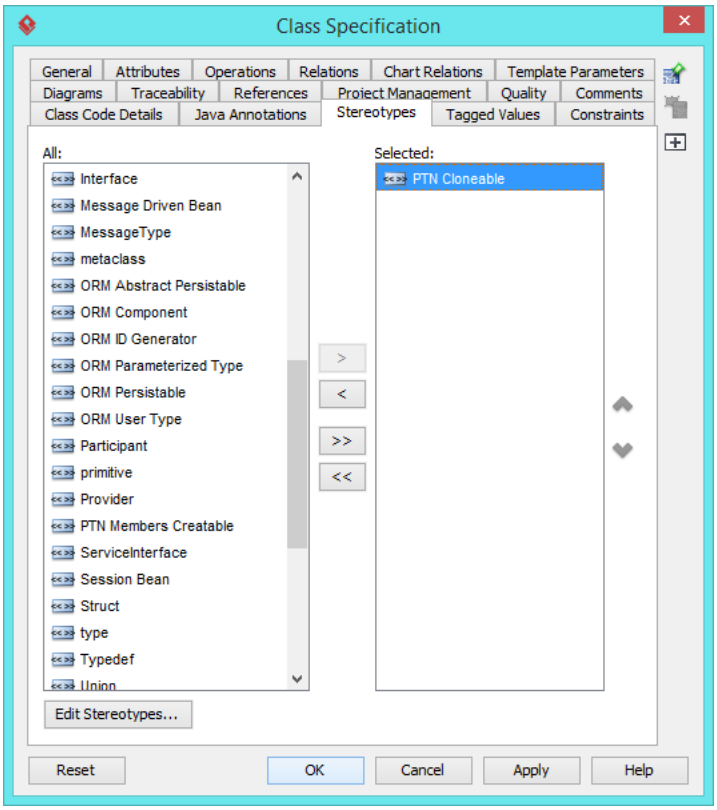
13. *ConcreteImplementor* will inherit the operations from *Implementor*. Right-click on *ConcreteImplementor* and select **Related Elements > Realize all Interfaces** from the popup menu.



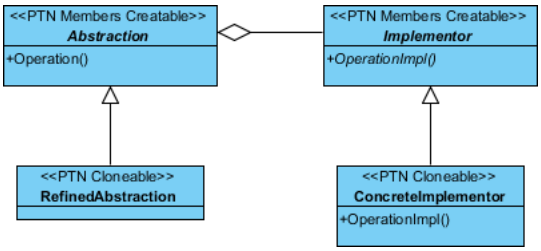
14. In practice, there may be multiple refined abstractions and/or concrete implementors. To represent this, stereotype the *RefinedAbstraction* and *ConcreteImplementor* classes as **PTN Cloneable**. Right-click on *Abstraction* and select **Stereotypes > Stereotypes...** from the popup menu.



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



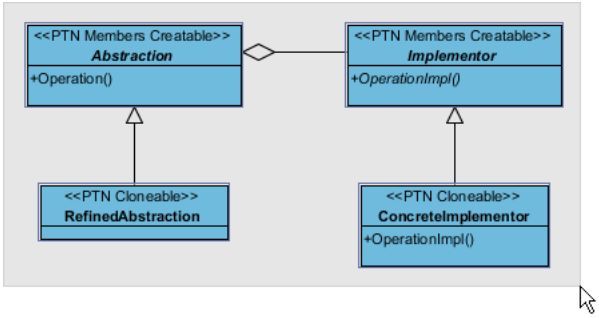

16. Repeat steps 14 and 15 for *ConcreteImplementor*.



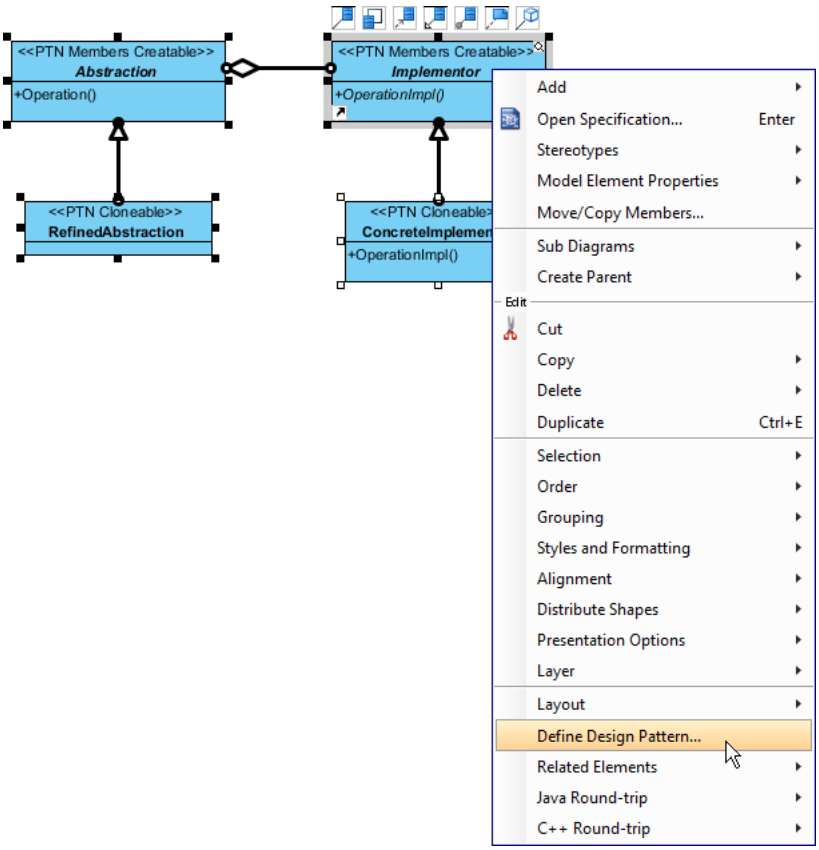

17. In practice, there may be multiple operations and/or operationImpls. To represent this, stereotype the *Abstraction* and *Implementor* classes as **PTN Members Creatable**. Repeat steps 14 and 15 to stereotype *Abstraction* and *Implementor* as **PTN Members Creatable**.

## Defining a Pattern

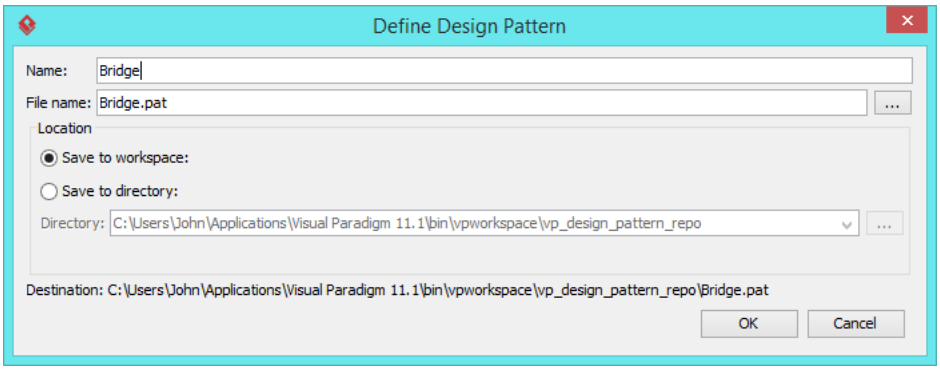
1. Select all classes on the class diagram.




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



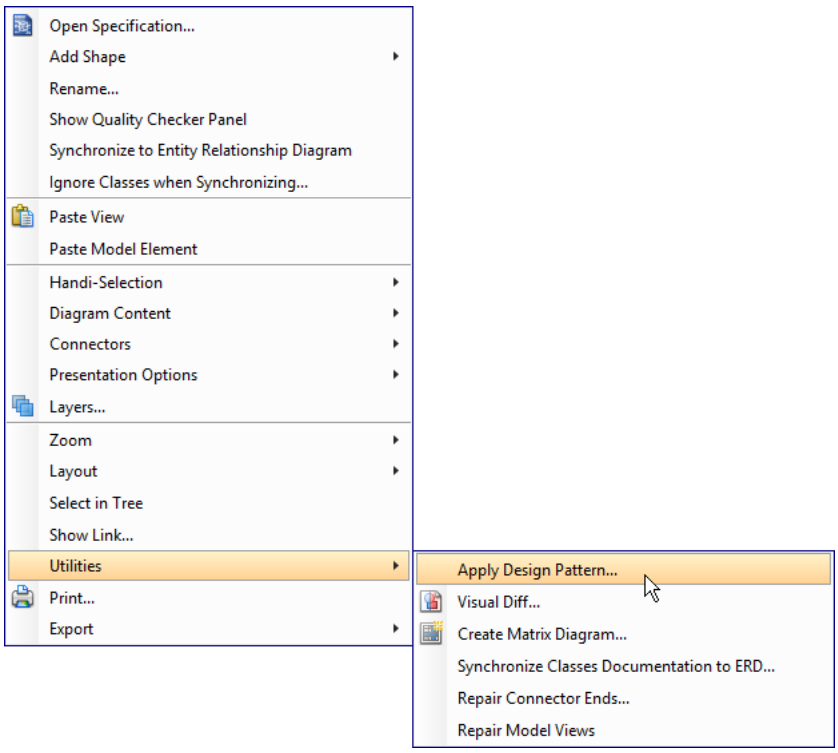
3. In the **Define Design Pattern** dialog box, specify the pattern name as *Bridge*. Keep the file name as is and click **OK** to proceed.



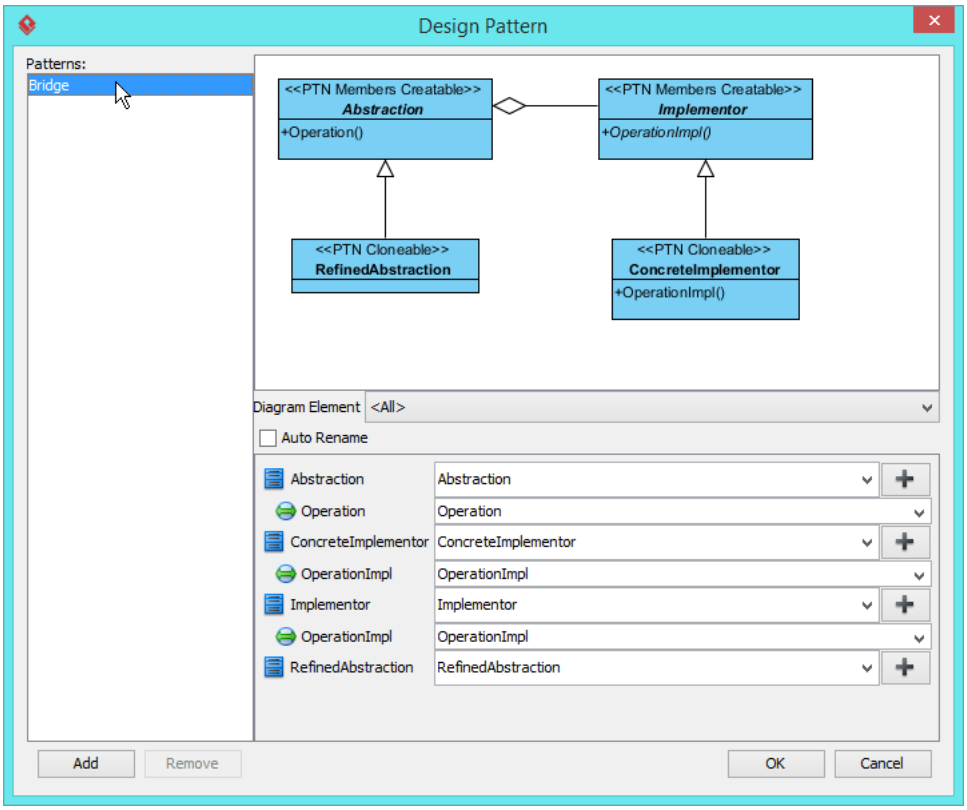

## Applying a Design Pattern to a Class Diagram

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

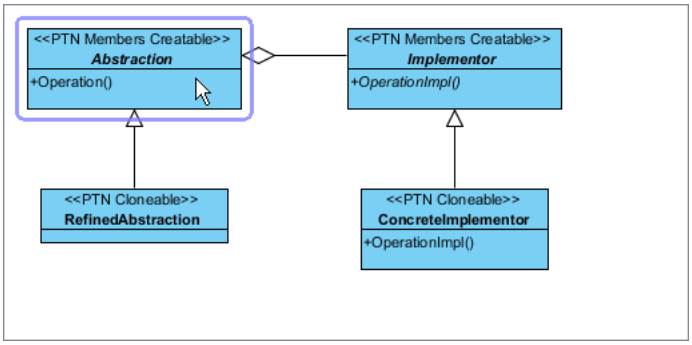
1. Create a new project named *Diagram Editor*.
2. Create a class diagram named *Domain Model*.
3. Right-click on the class diagram and select **Utilities > Apply Design Pattern...** from the popup menu.



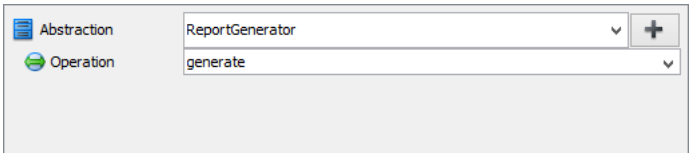
4. In the **Design Pattern** dialog box, select *Bridge* from the list of patterns.




5. Click on *Abstraction* in the overview.

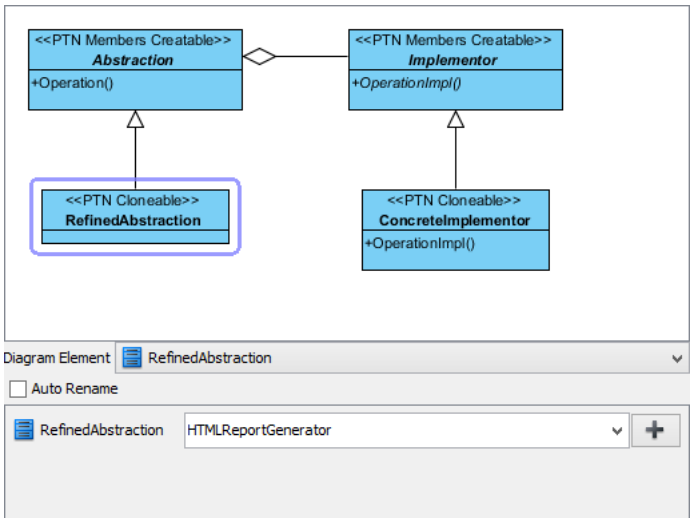



6. Rename *Abstraction* to *ReportGenerator* and the *Operation* operation to *generate* in the bottom pane.



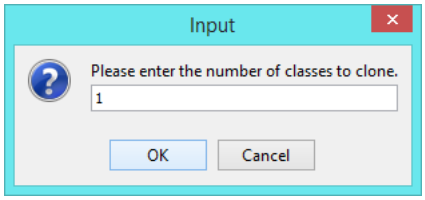

7. Select *RefinedAbstraction* in the overview pane.

8. Rename *RefinedAbstraction* to *HTMLReportGenerator*.

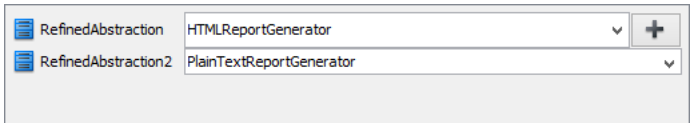



9. In addition to the HTML report generator, we also need a Plain Text report generator. Click the + button in the bottom pane, next to *Abstraction*, and select **Clone...**


10. Enter *1* as the number of classes to clone and click **OK** to confirm.

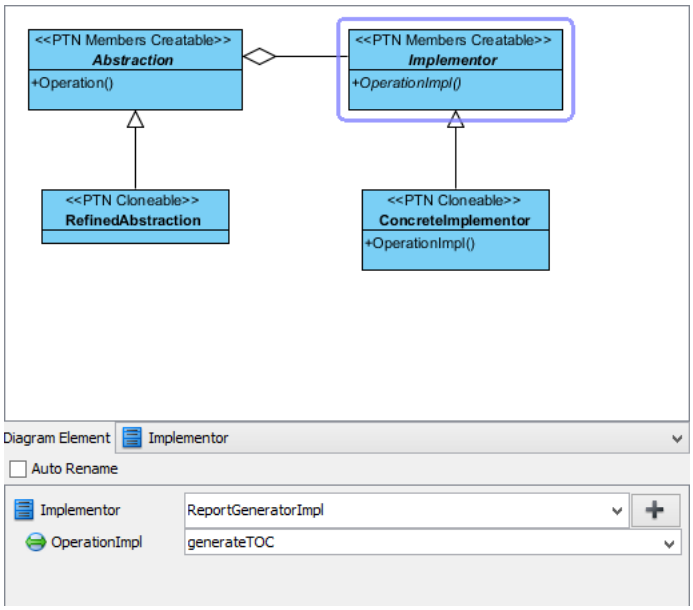



11. Rename the cloned class, *RefinedAbstraction2*, to *PlainTextReportGenerator*.

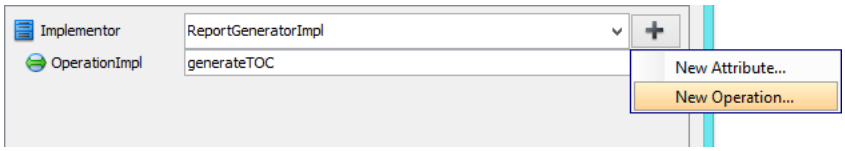



12. Select *Implementor* in the overview pane.

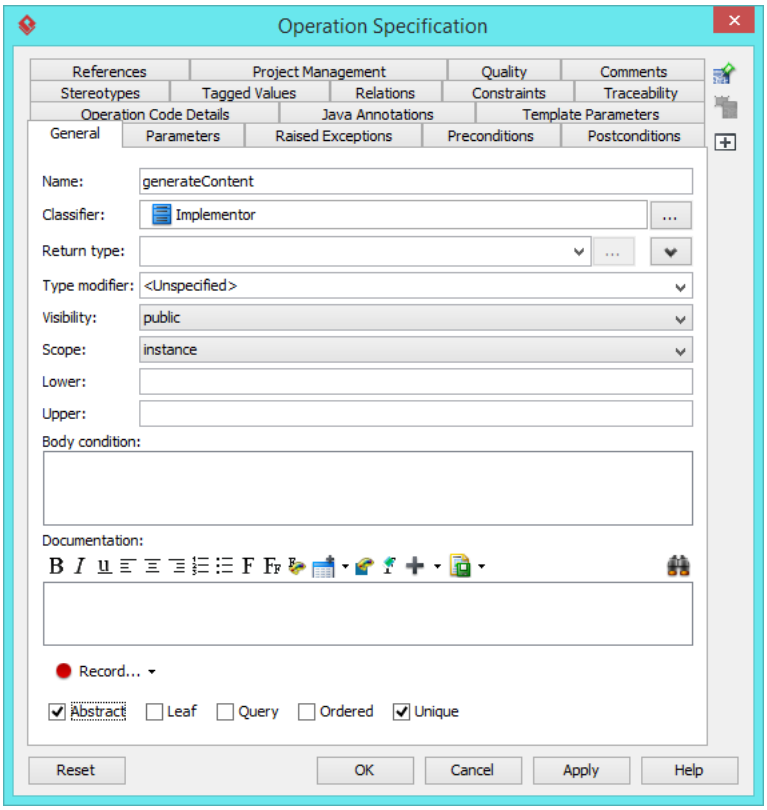
13. Rename *Implementor* to *ReportGeneratorImpl* and *OperationImpl* to *generateTOC*.



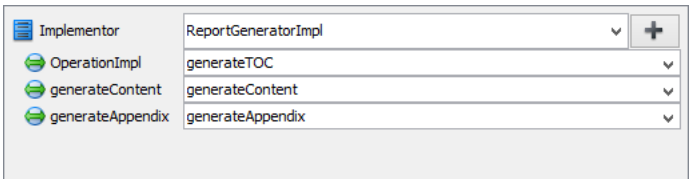

14. We need two more operations for generating the content and appendix. Click the + button and select **New Operation...** from the popup menu.



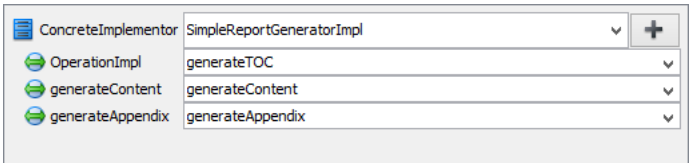
15. In the **Operation Specification** dialog box, name the operation *generateContent*. Check the **Abstract** box at the bottom of the dialog.



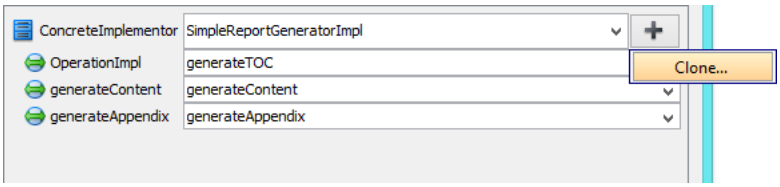

16. Repeat the previous steps to create another abstract operation named *generateAppendix*.




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

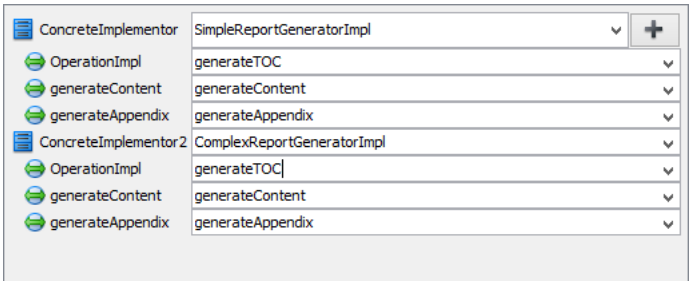


18. Similar to *RefinedAbstraction*, we need another concrete implementor for generating complex reports. Click the + button and select **Clone...** from the popup menu.

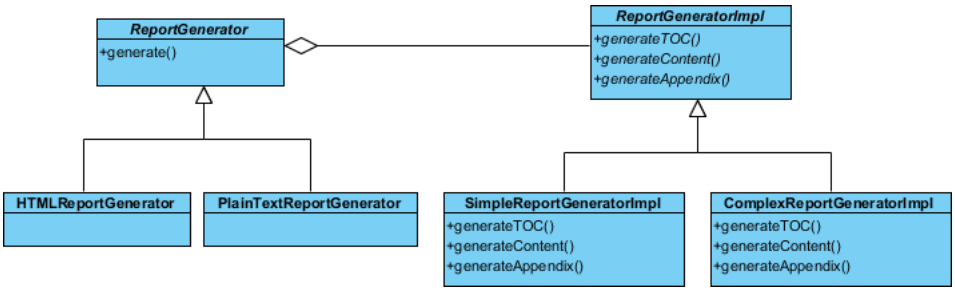



19. Enter *1* as the number of classes to clone and click **OK** to confirm.

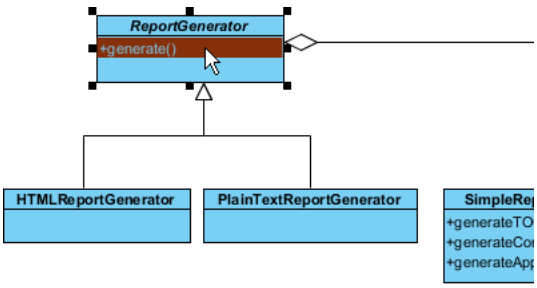
20. Rename the cloned class, *ConcreteImplementor2*, to *ComplexReportGeneratorImpl*, and the *OperationImpl* operation to *generateTOC*.



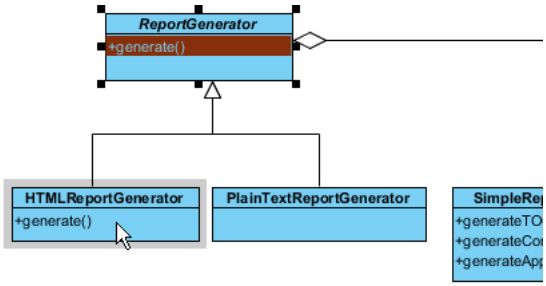

21. Click **OK** to confirm. The diagram should now look like this:



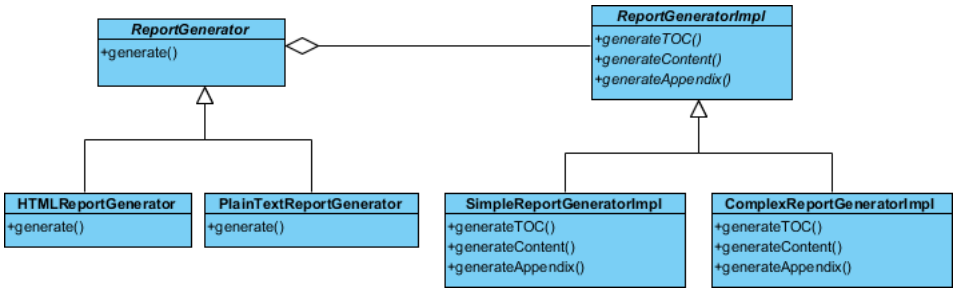

22. We want *HTMLReportGenerator* and *PlainTextReportGenerator* to implement their own way of generating reports. Select the "generate" operation in *ReportGenerator*.



23. Press the **Ctrl** key and drag it to *HTMLReportGenerator*. Release the mouse button afterward.




24. Repeat the previous steps to create the "generate" method in *PlainTextReportGenerator*. The completed diagram should look like this:




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/)