Visual Paradigm

# Iterator Pattern Tutorial

Written Date : October 16, 2009

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

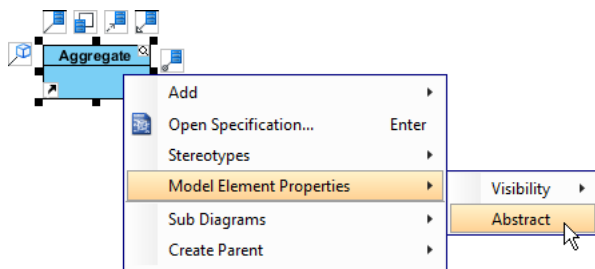## Modeling Design Pattern with Class Diagram

1.   Create a new project *Design Patterns*.

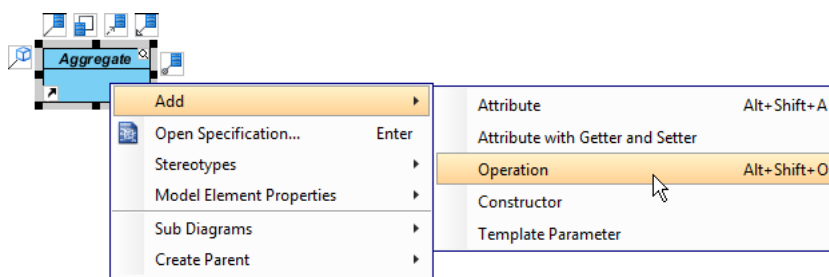2.   Create a class diagram *Iterator*.
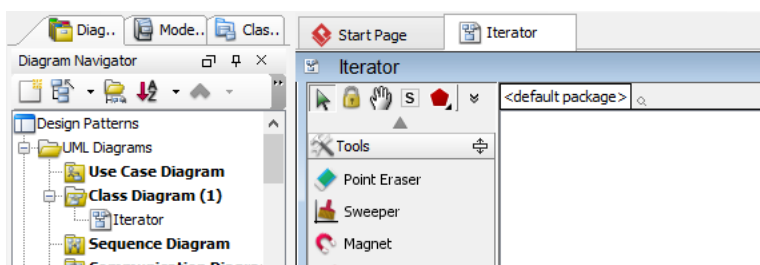
3.   Select **Class** from diagram toolbar. Click on the diagram to create a class. Name it as *Aggregate*.
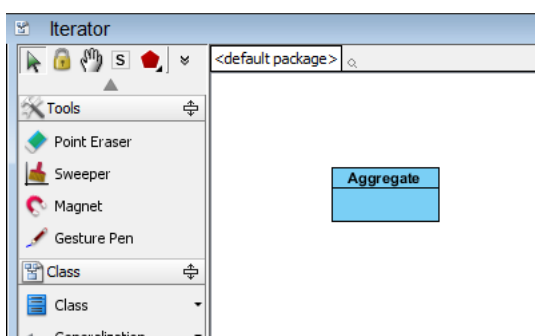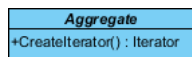
4. Right-click on *Aggregate*, and select **Model Element Properties** > **Abstract** to set it as abstract.
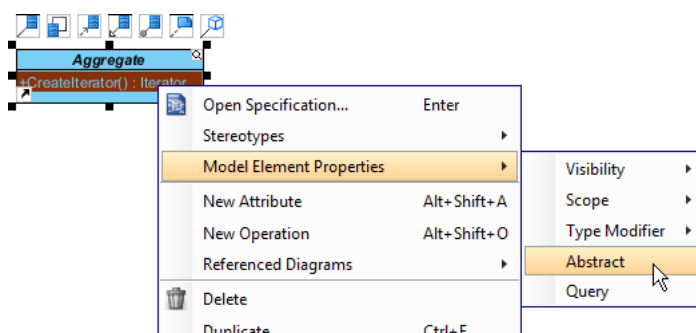

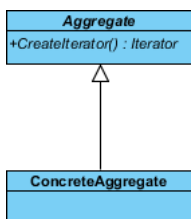5. Right-click on the *Aggregate* class, and select **Add** > **Operation** from the popup menu.

6. Name the operation *CreateIterator()*, and make it return Iterator.
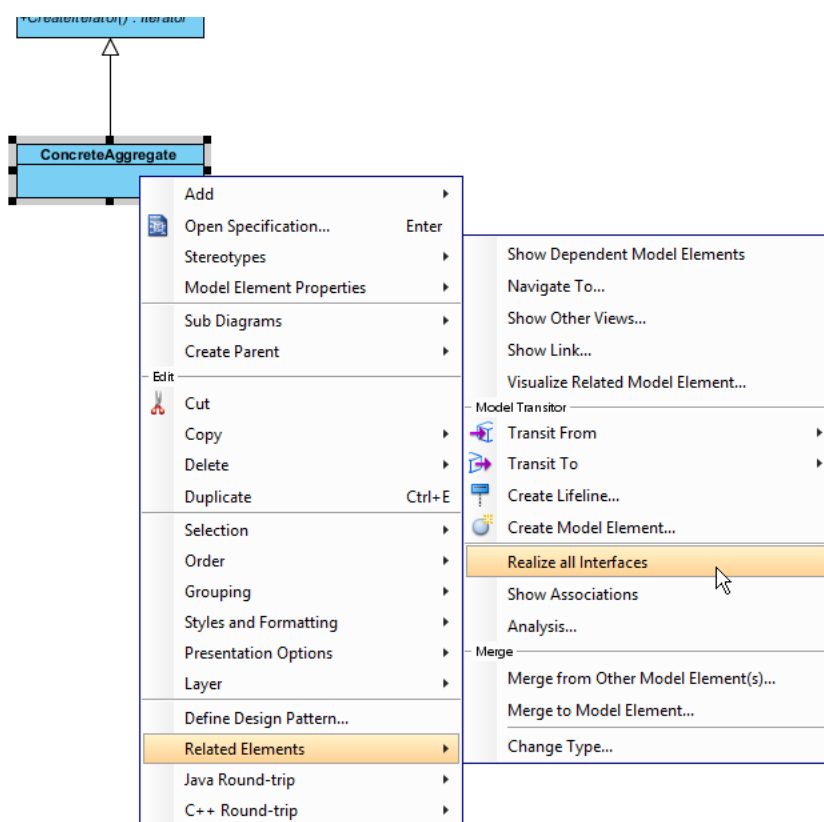

7. Right-click on *CreateIterator() : Iterator*, and select **Model Element Properties** > **Abstract** to set it as abstract.
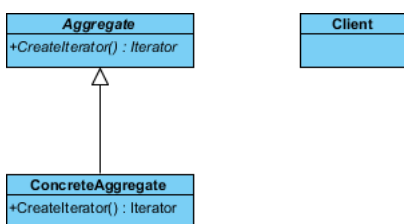
8. Move the mouse cursor over the *Aggregate* class, and drag out **Generalization** > **Class** to create a subclass *ConcreteAggregate*.
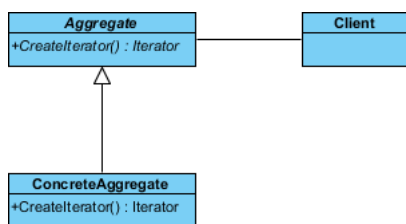
9. *ConcreteAggregate* will inherit the operations from *Aggregate*. Select *ConcreteAggregate*, right-click on them and select **Related Elements** > **Realize all Interfaces** from the popup menu.
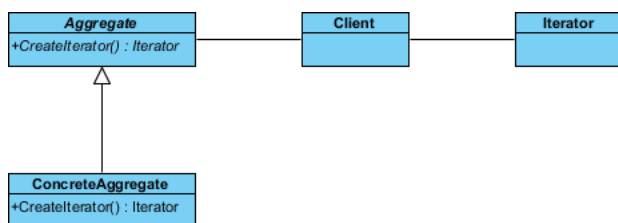
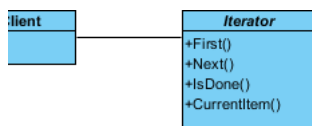10. Create a class *Client* near *Aggregate*.

11. Move the mouse cursor over the *Client* class, and drag out **Association** > **Class** to *Aggregate.*
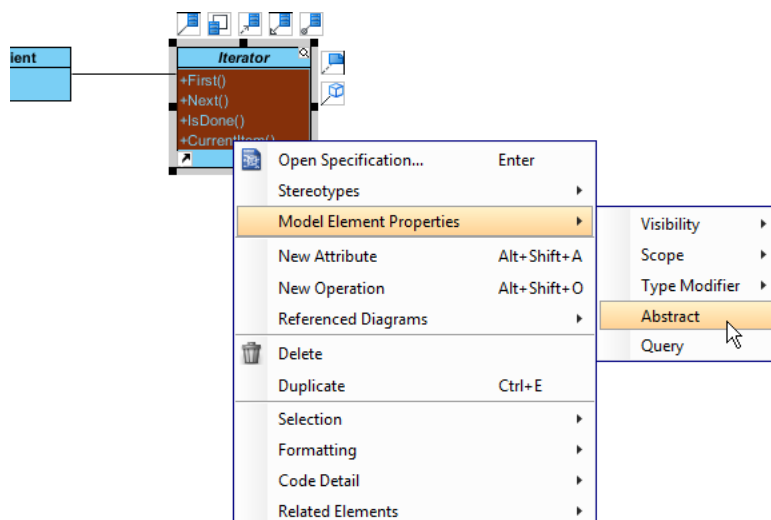

12. Move the mouse cursor over the *Client* class, and drag out **Association** > **Class** to create an associated class *Iterator*.


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

14. Right-click on the *Iterator* class, and select **Add** > **Operation** from the popup menu to add operations: *First()*, *Next()*, *IsDone()*, *CurrentItem().*

15. Select all operations in *Iterator*. Right-click and select **Model Element Properties** > **Abstract** to set it as abstract.


16. *ConcreteIterator* will inherit the operations from *Iterator*. Select *ConcreteIterator*, right-click on them and select **Related Elements** > **Realize all Interfaces** from the popup menu.
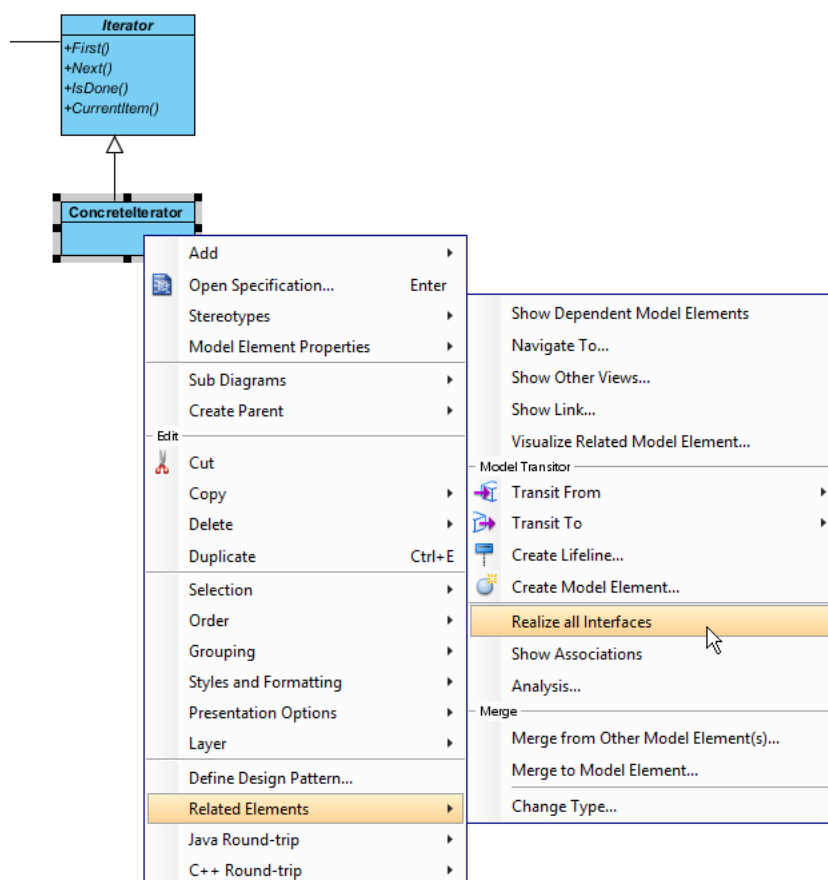
17. Move the mouse cursor over the *ConcreteAggregate* class, and drag out **Dependency** > **Class** to class *ConcreteIterator*.
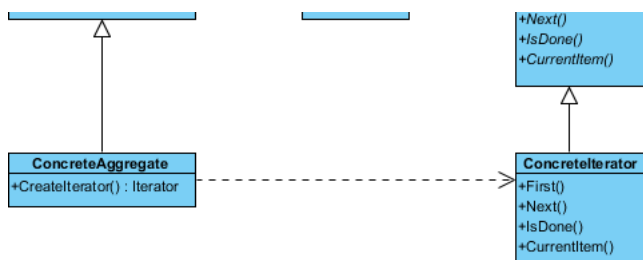

18. Move the mouse cursor over the *ConcreteIterator* class, and drag out **Association** > **Class** to class *ConcreteAggregate*.

19. In practice, there may be multiple *ConcreteAggregate* classes. To represent this, stereotype the *ConcreteAggregate* class as **PTN Cloneable**. Right-click on *ConcreteAggregate* class and select **Stereotypes** > **Stereotypes...** from the popup menu.
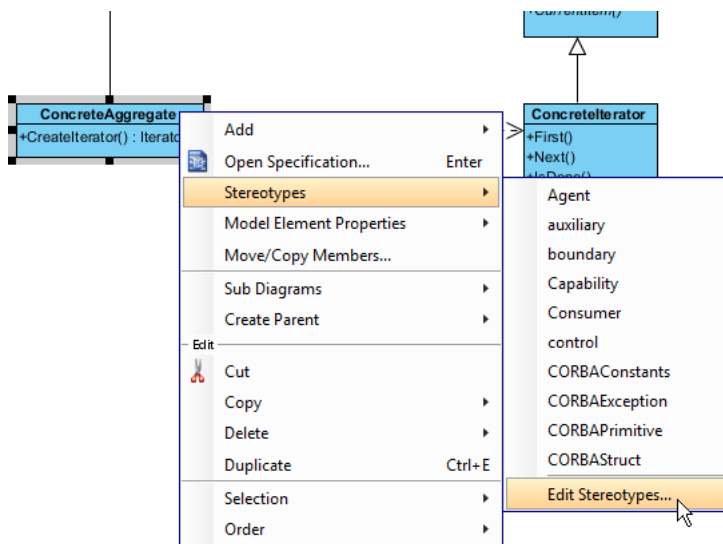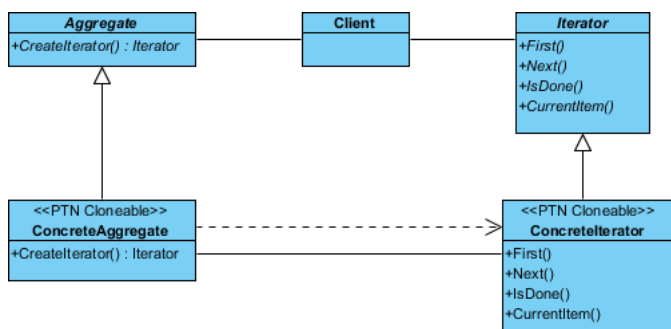
20. In the **Stereotypes** tab of class specification, select **PTN Cloneable** and click > to assign it to the class. Click **OK** to confirm.
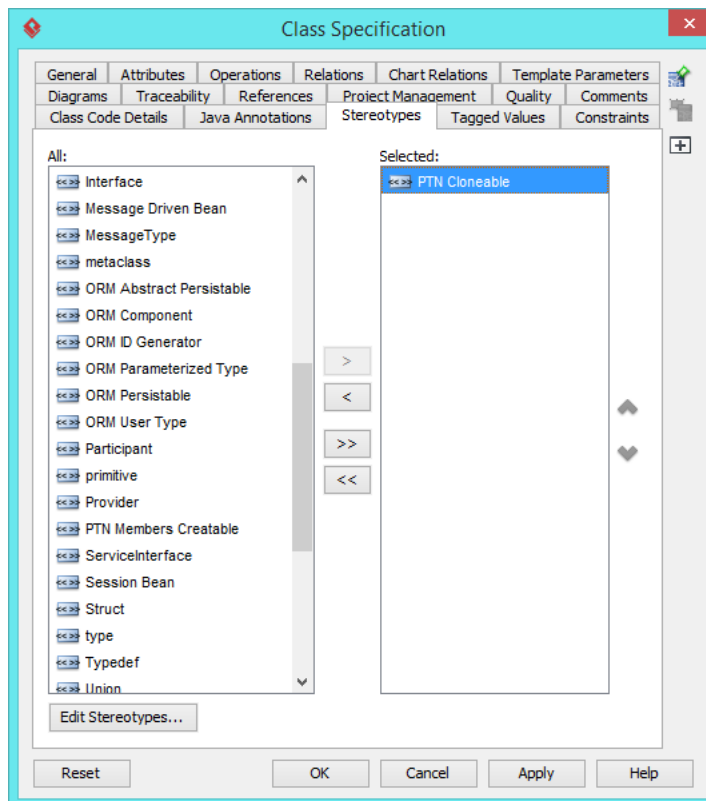
21. There may be multiple *ConcreteIterator* classes. Right-click on *ConcreteIterator* class and select **Stereotypes** > **PTN Cloneable** from the popup menu.
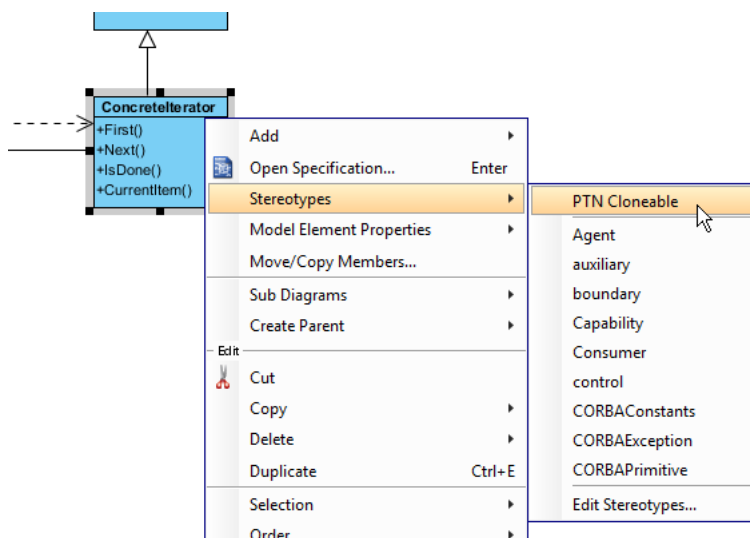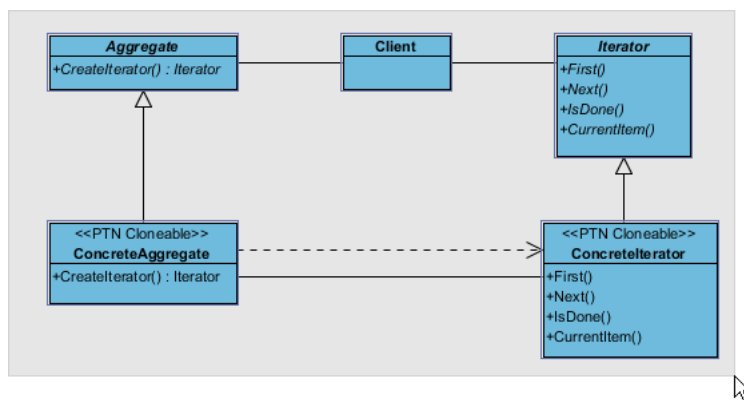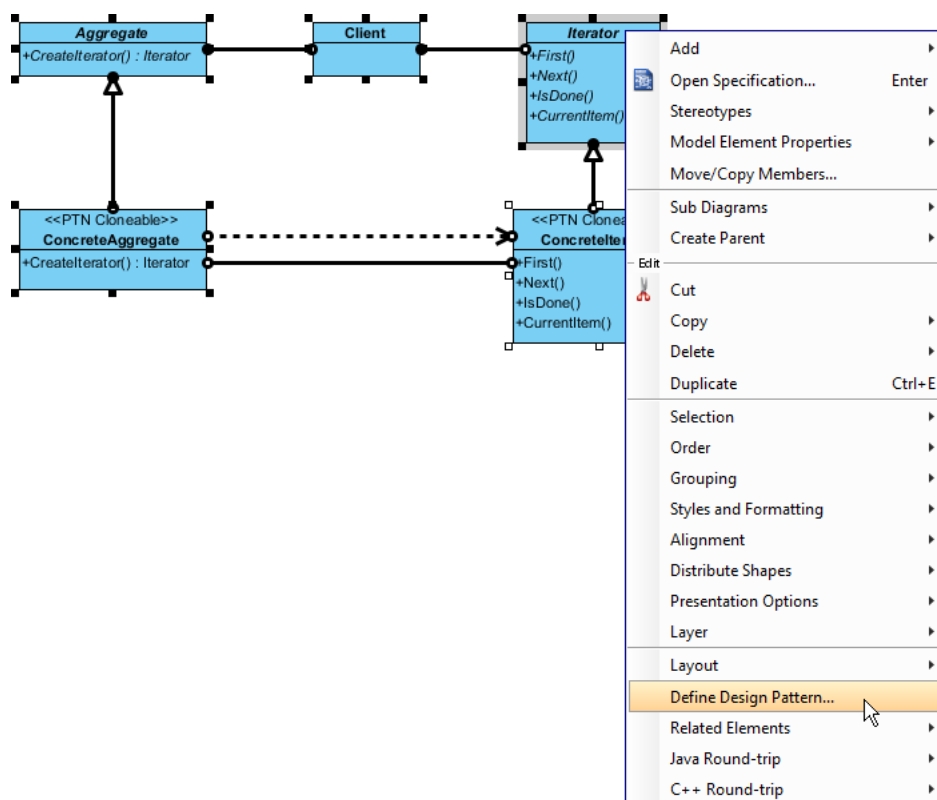
The diagram become:
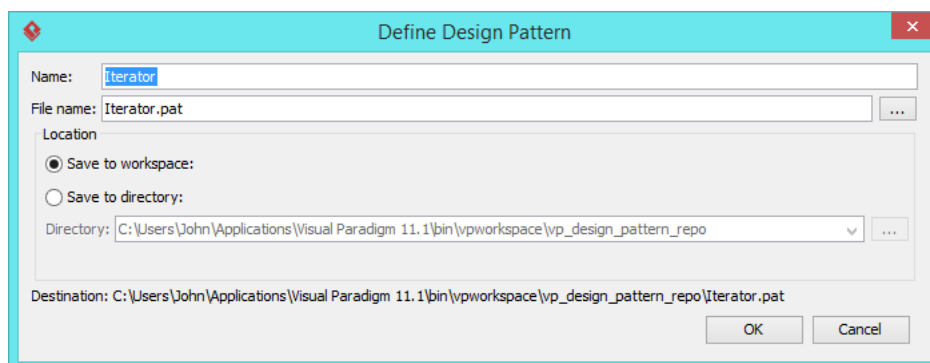
Defining Pattern

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 *Iterator*. Keep the file name as is. Click **OK** to proceed.
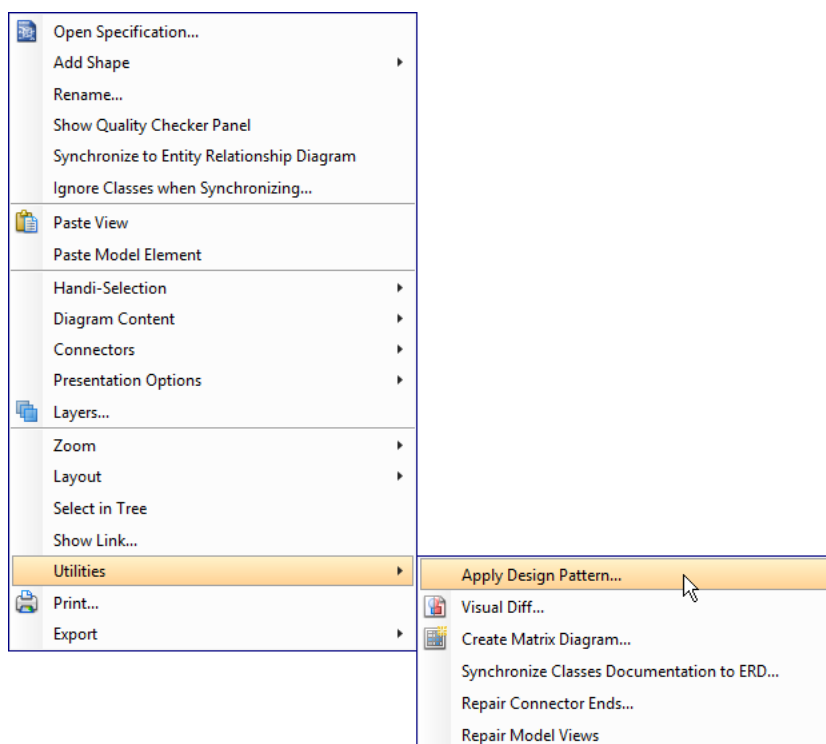

Applying Design Pattern on Class Diagram

In this section, we are going to apply the iterator pattern to model the use of iterator in a diagram editor.

1.   Create a new project *Diagram Editor*.

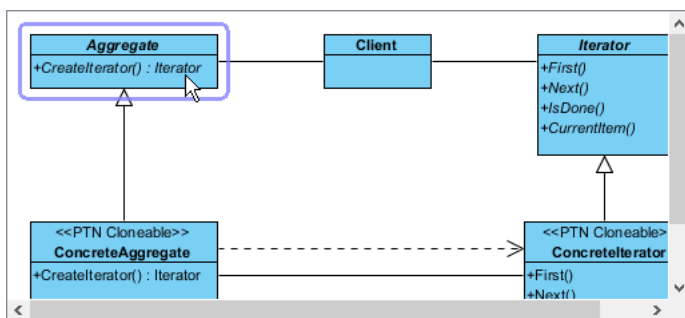2.   Create a class diagram *Domain Model*.

3. Right-click on the class diagram and select **Utilities** > **Apply Design Pattern...** from the popup menu.
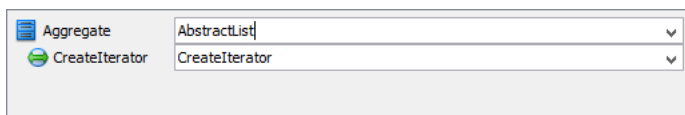
4. In the **Design Pattern** dialog box, select *Iterator* from the list of patterns.
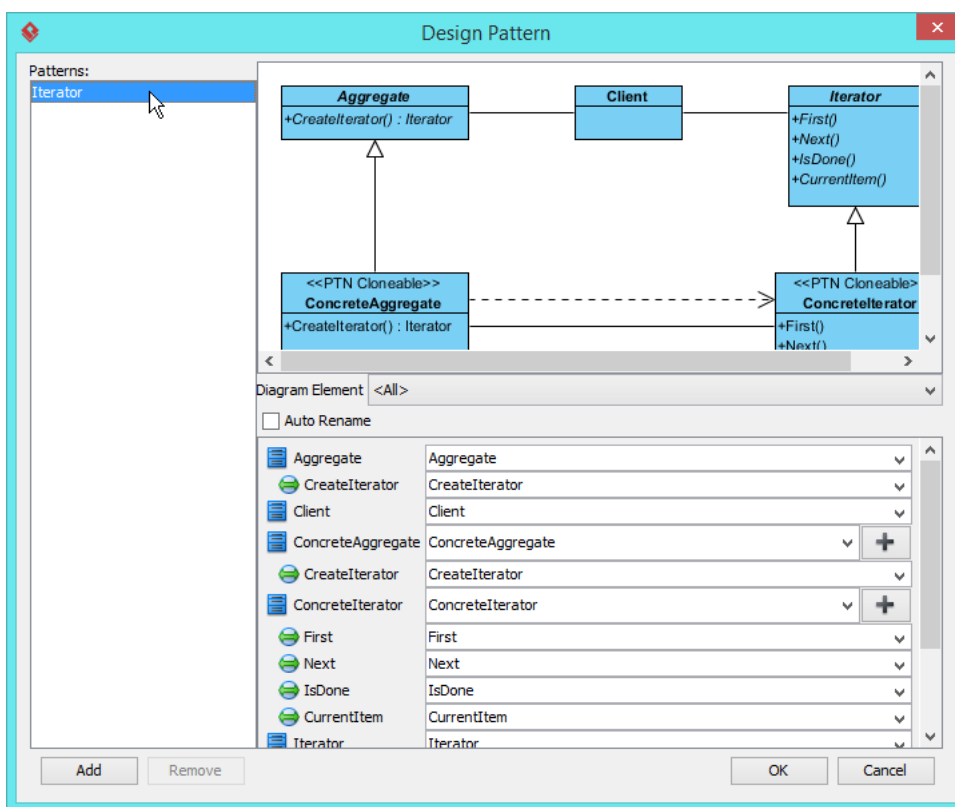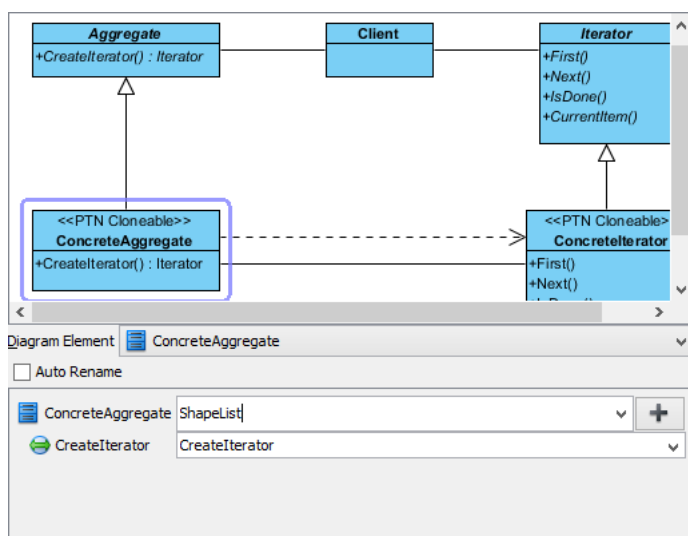

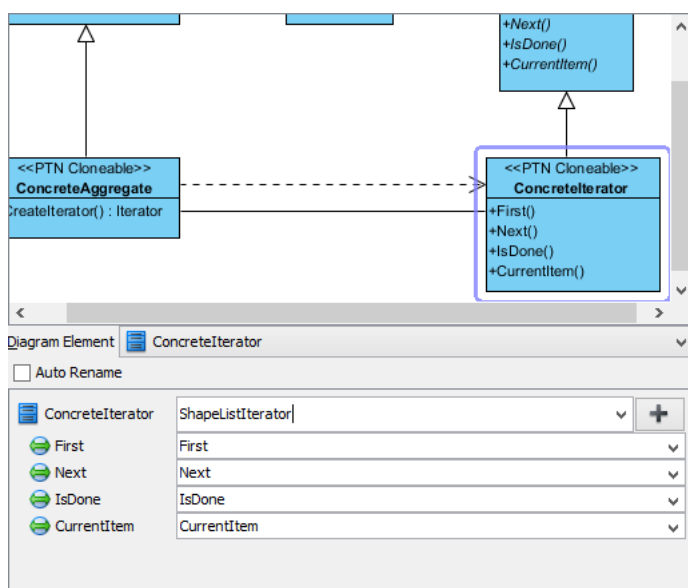5. Click on *Aggregate* in the overview.

6. Rename *Aggregate* to *AbstractList* at the bottom pane.

7. Select *ConcreteAggregate* in overview, and rename it to *ShapeList* at the bottom pane.
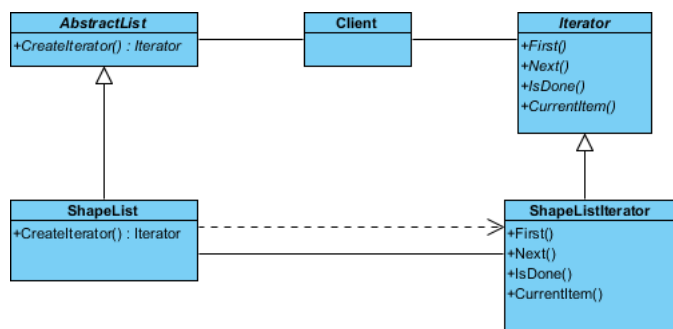

8. Select *ConcreteIterator* in overview, and rename it to *ShapeListIterator* at the bottom pane.

9. Click **OK** to apply the pattern to diagram.

10. Tidy up the diagram. Here is the result:

Resources

1. Design Patterns.vpp

2. Iterator.pat

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/)