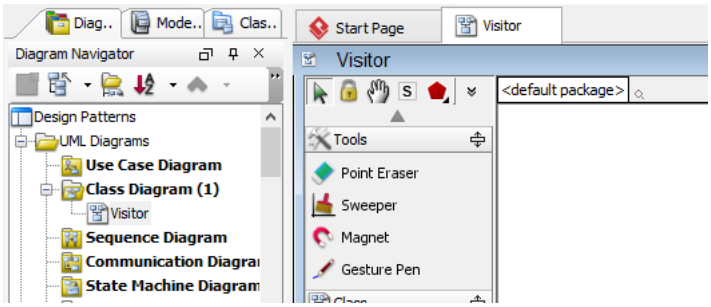
Visitor Pattern Tutorial

Written Date : October 28, 2009

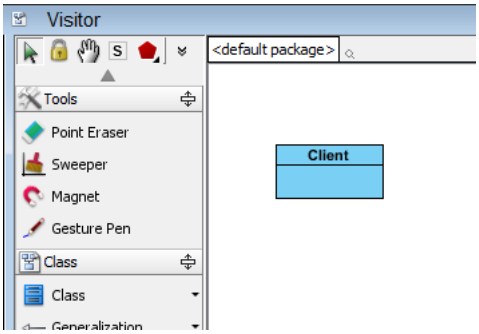
Modeling a Design Pattern with a Class Diagram

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

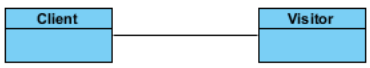
2. Create a class diagram named *Visitor*.



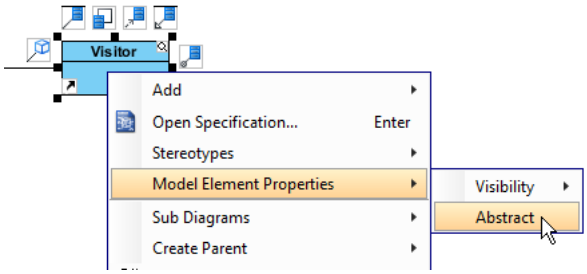
3. Select **Class** from the diagram toolbar. Click on the diagram to create a class and name it *Client*.



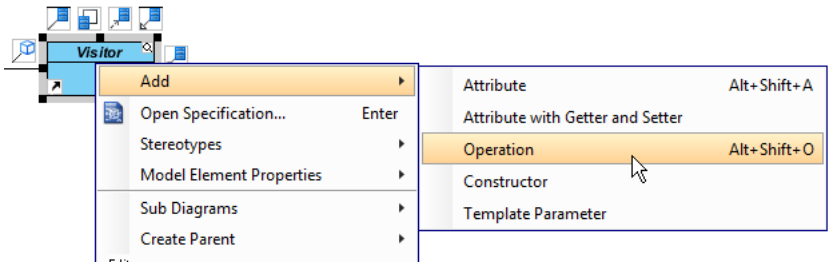
4. Move the mouse cursor over the *Client* class and drag out **Association** > **Class** to create an associated class named *Visitor*.



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

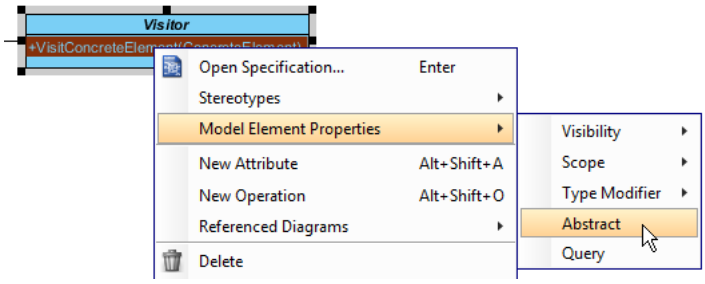


6. Right-click on the *Visitor* class and select **Add** > **Operation** from the popup menu.

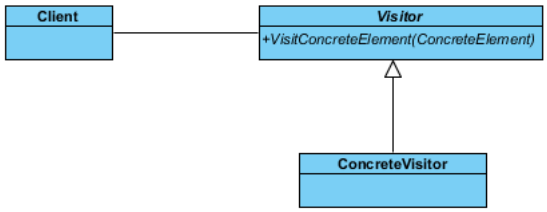


7. Name the operation `VisitConcreteElement(ConcreteElement)`.

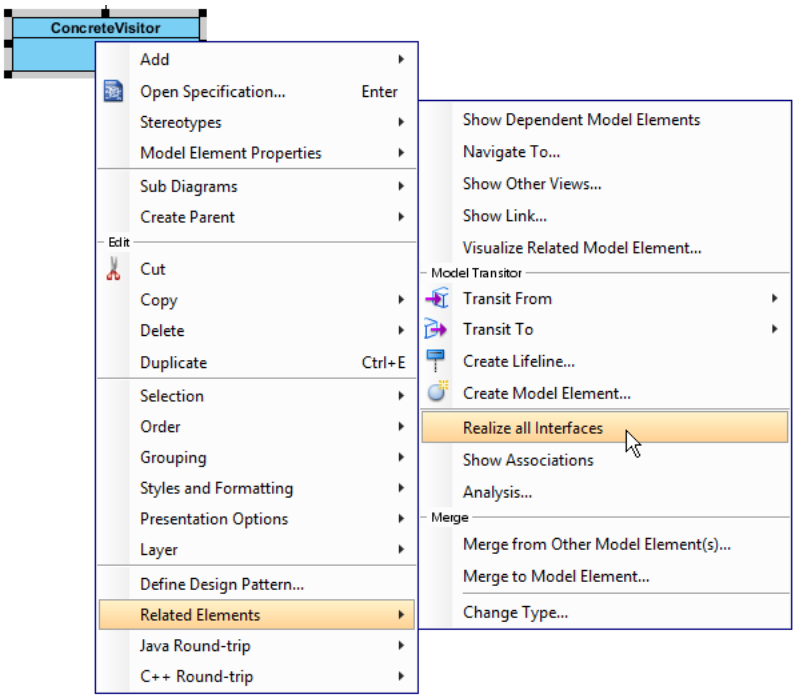
8. Right-click on `VisitConcreteElement()` and select **Model Element Properties** > **Abstract** to set it as abstract.



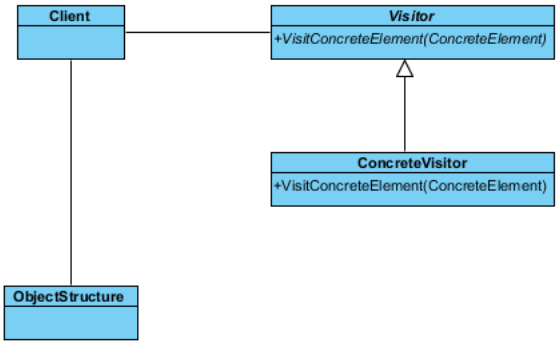
9. Move the mouse cursor over the *Visitor* class and drag out **Generalization** > **Class** to create a subclass named *ConcreteVisitor*.



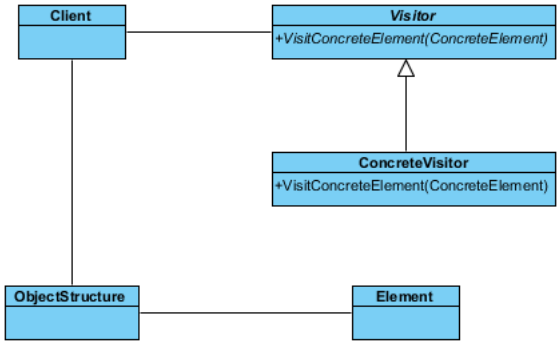
10. We need to make the concrete visitors inherit operations from the visitor class. Right-click on *ConcreteVisitor* and select **Related Elements** > **Realize all Interfaces** from the popup menu.



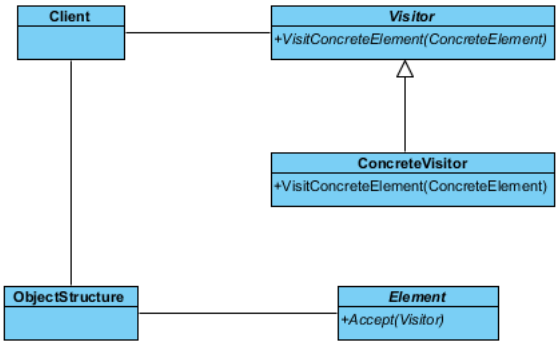
11. Move the mouse cursor over the *Client* class and drag out **Association** > **Class** to create an associated class named *ObjectStructure*.



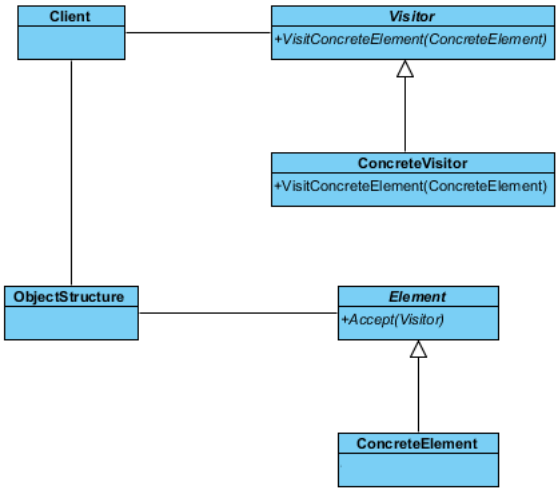
12. Move the mouse cursor over the *ObjectStructure* class and drag out **Association** > **Class** to create an associated class named *Element*.



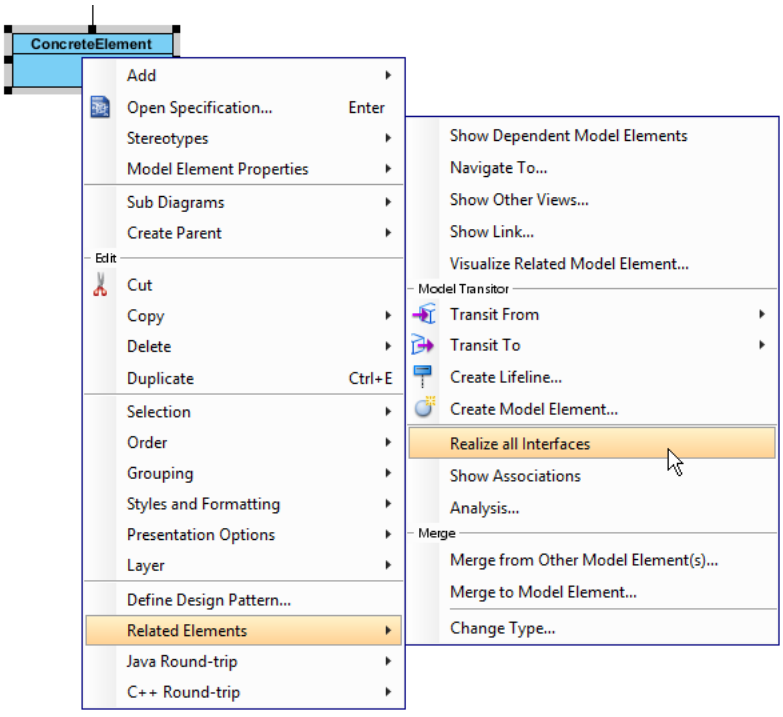
13. Right-click on *Element* and select **Model Element Properties** > **Abstract** to set it as abstract.
14. Right-click on the *Element* class, select **Add** > **Operation** from the popup menu, and name the operation `Accept(Visitor)`.
15. Right-click on `Accept(Visitor)` and select **Model Element Properties** > **Abstract** to set it as abstract. At this point, the diagram becomes:



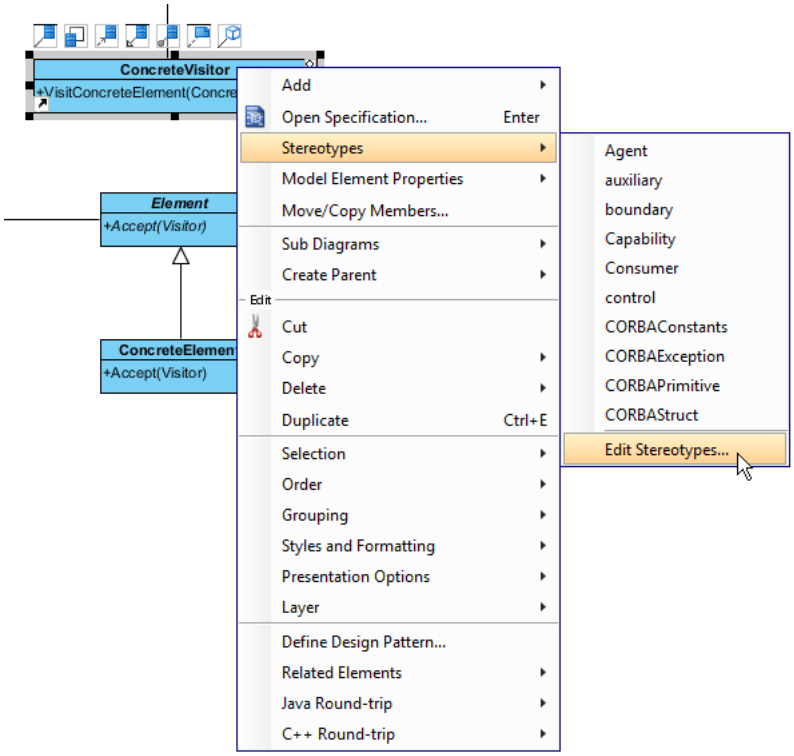
16. Move the mouse cursor over the *Element* class and drag out **Generalization** > **Class** to create a subclass named *ConcreteElement*.



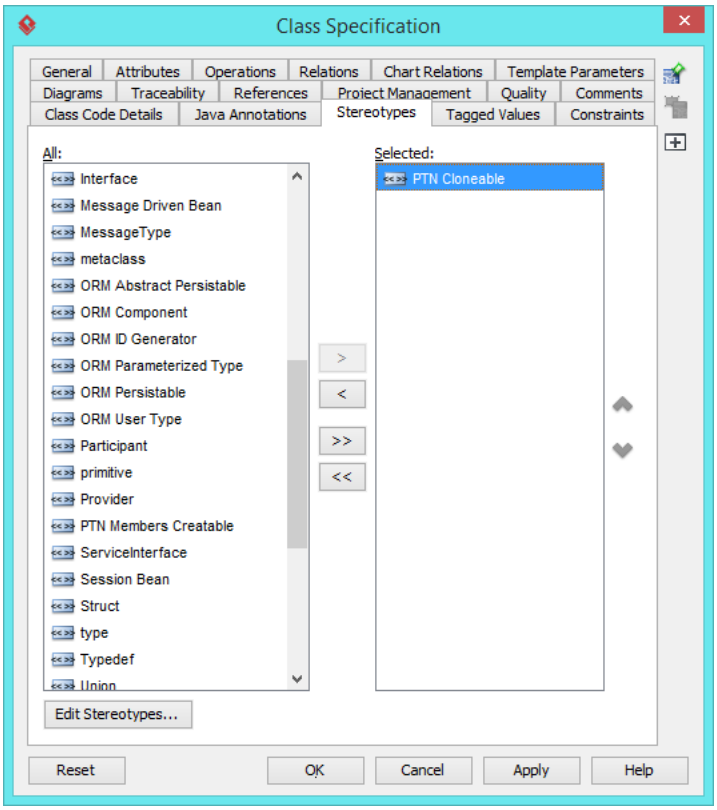
17. We need to make the concrete elements inherit operations from the element class. Right-click on *ConcreteElement* and select **Related Elements** > **Realize all Interfaces** from the popup menu.



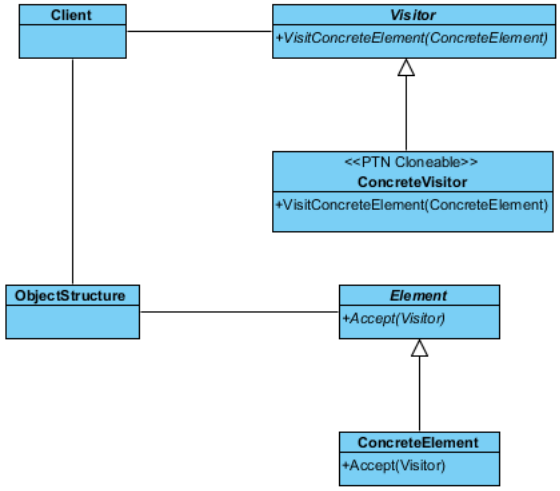
18. In practice, there may be multiple `ConcreteVisitor` classes. To represent this, stereotype the `ConcreteVisitor` class as **PTN Cloneable**. Right-click on the `ConcreteVisitor` class and select **Stereotypes** > **Stereotypes...** from the popup menu.



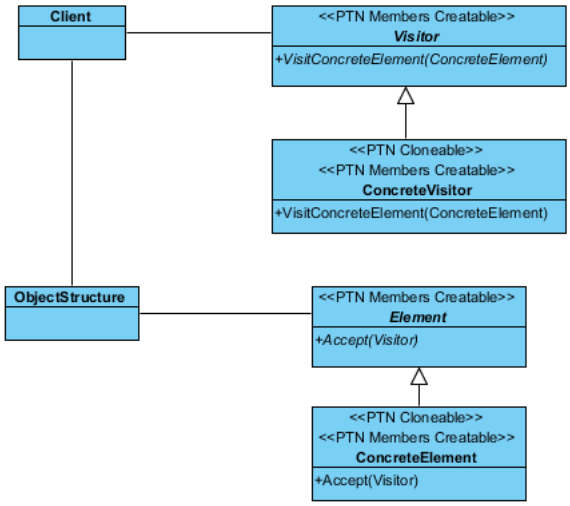
19. In the **Stereotypes** tab of the class specification, select **PTN Cloneable** and click > to assign it to the class. Click **OK** to confirm.



20. Repeat the stereotyping process on the `ConcreteElement` class.

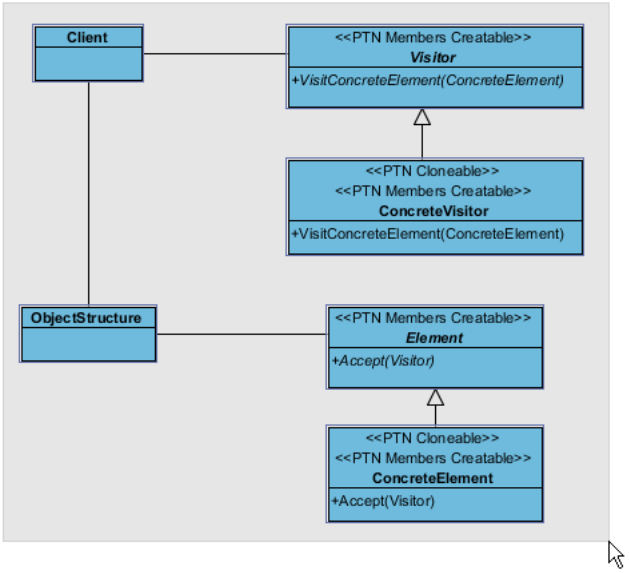


21. There may be multiple operations in `Visitor`, `Element`, and `ConcreteElement`. To represent this, stereotype them as **PTN Members Creatable** by following the same stereotyping steps as before. At this point, the pattern should look like this:

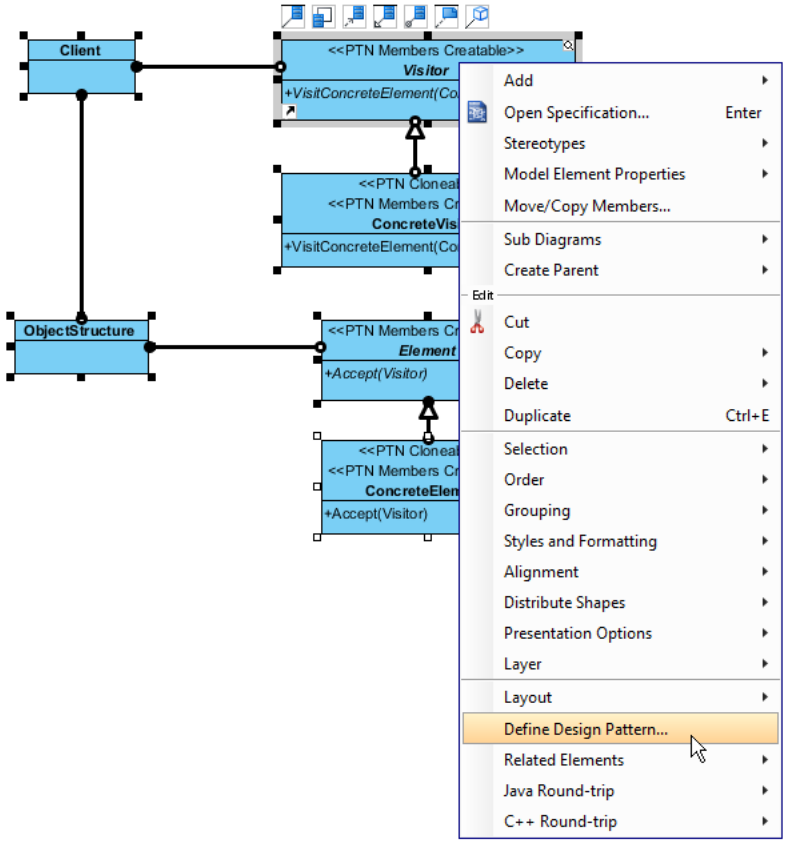


Defining the Pattern

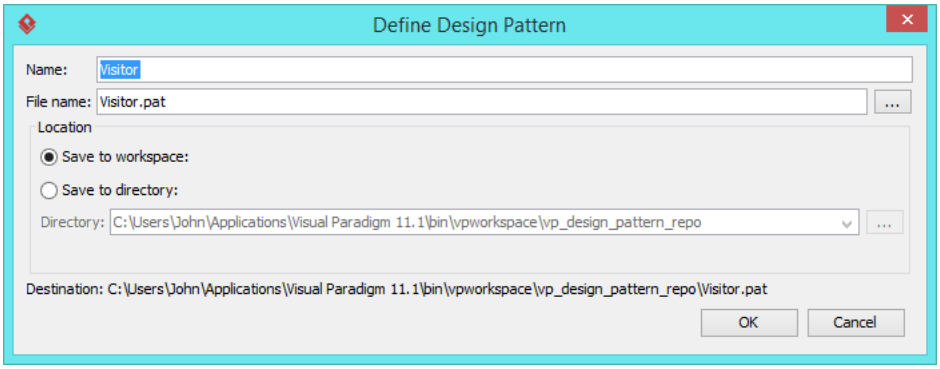
1. Select all classes on the class diagram.



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

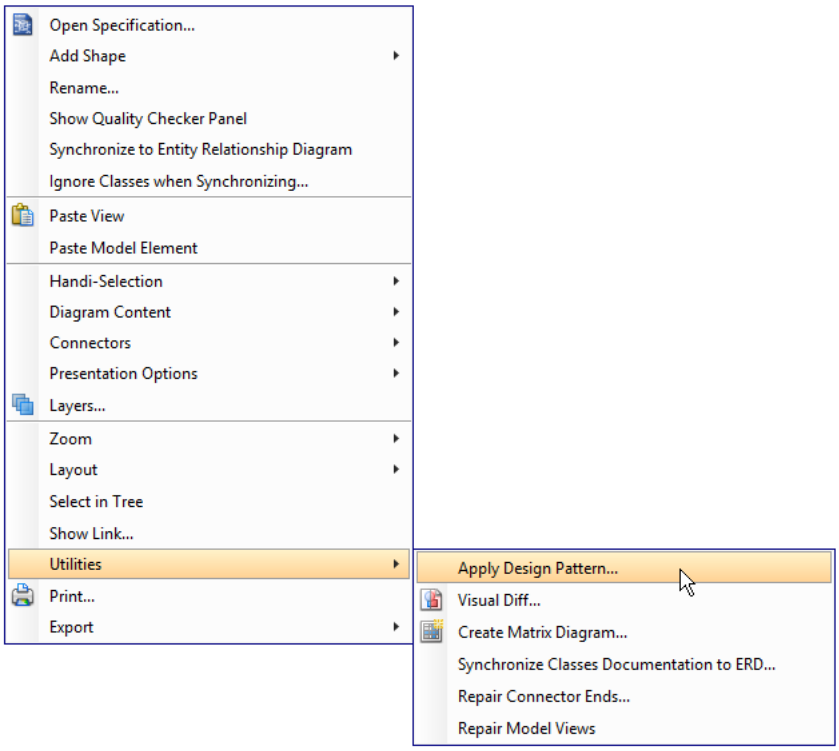


Applying a Design Pattern to a Class Diagram

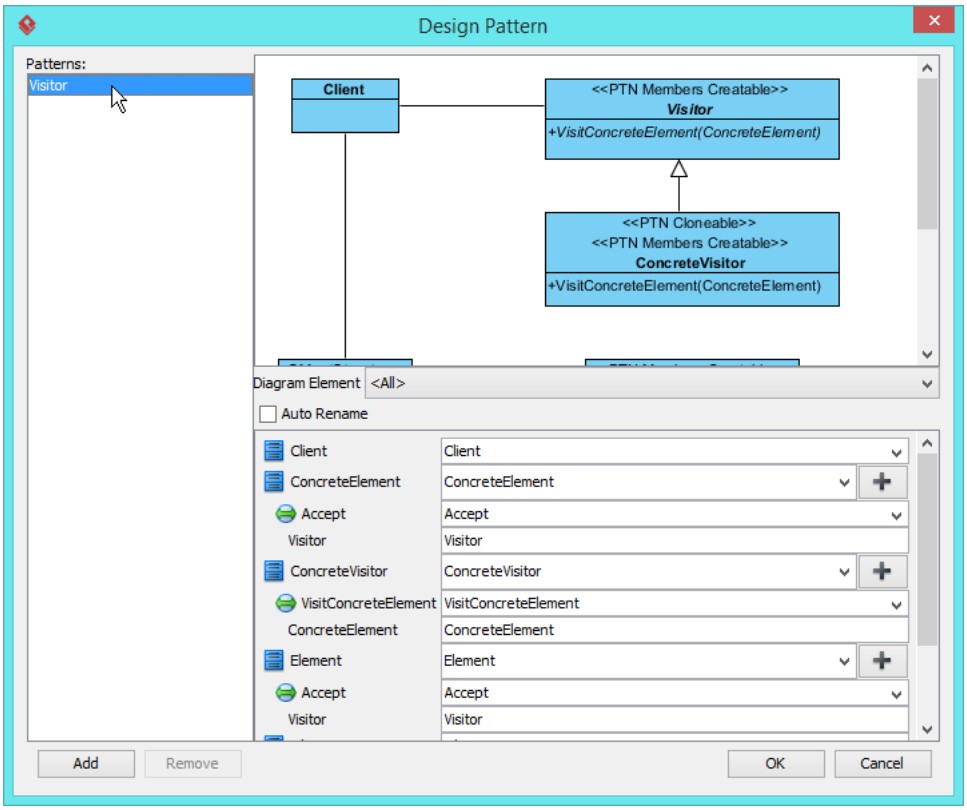
In this section, we will use the visitor pattern to model visiting elements in a room.

1. Create a new project named *My Room*.
2. Create a class diagram named *Domain Model*.

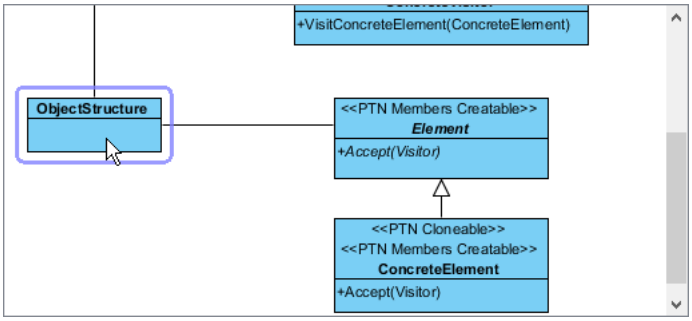
3. Right-click on the class diagram and select **Utilities** > **Apply Design Pattern...** from the popup menu.



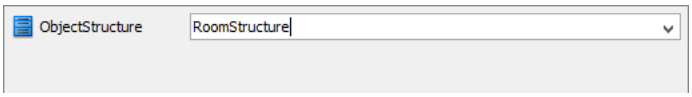
4. In the **Design Pattern** dialog box, select *Visitor* from the list of patterns.



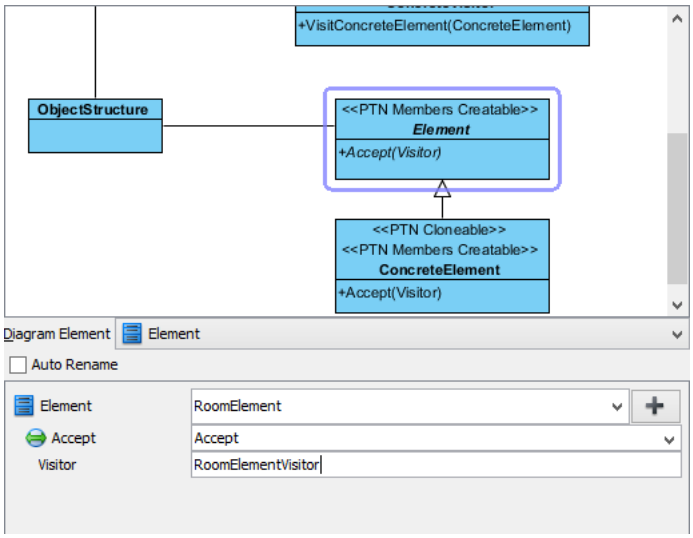
5. Click on *ObjectStructure* in the overview.



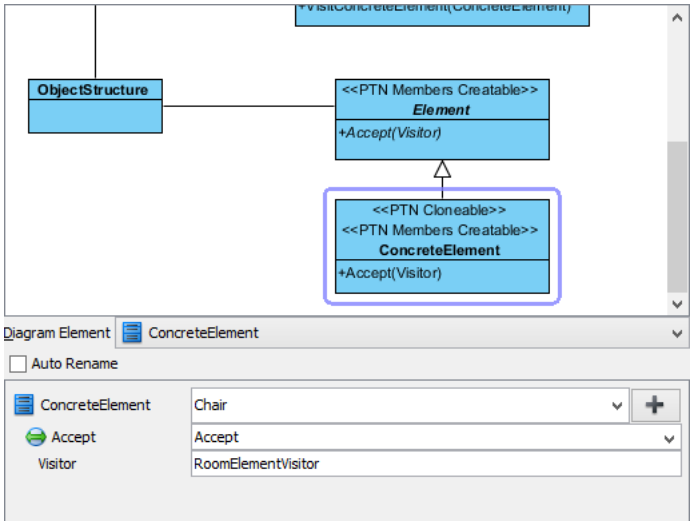
6. Rename it to *RoomElements* in the bottom pane.



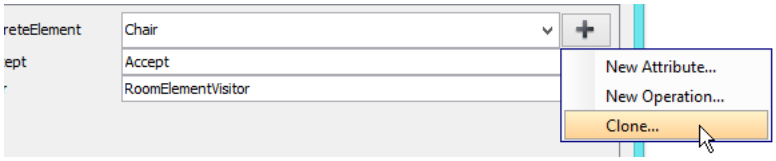
7. Select *Element* in the overview. In the bottom pane, rename it to *RoomElement*. Rename the parameter *Visitor* in `Accept()` to *RoomElementVisitor*.



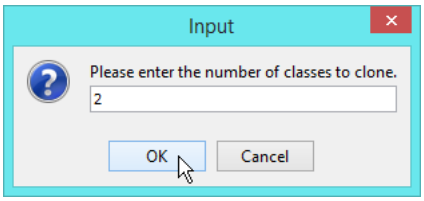
8. Select *ConcreteElement* in the overview. In the bottom pane, rename it to *Chair*. Rename the parameter *Visitor* in `Accept()` to *RoomElementVisitor*.



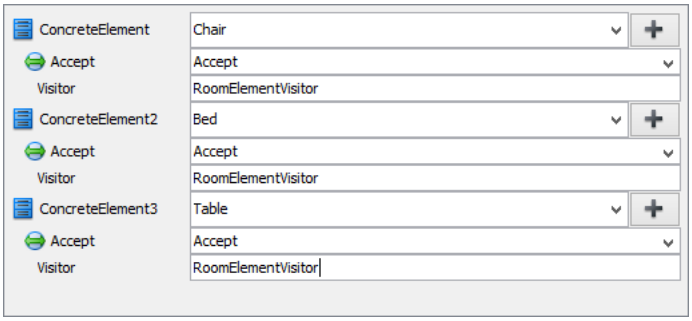
9. We need two more concrete elements for a bed and a table. Keep `ConcreteElement` selected, click the **+** button, and select **Clone...** from the popup menu.



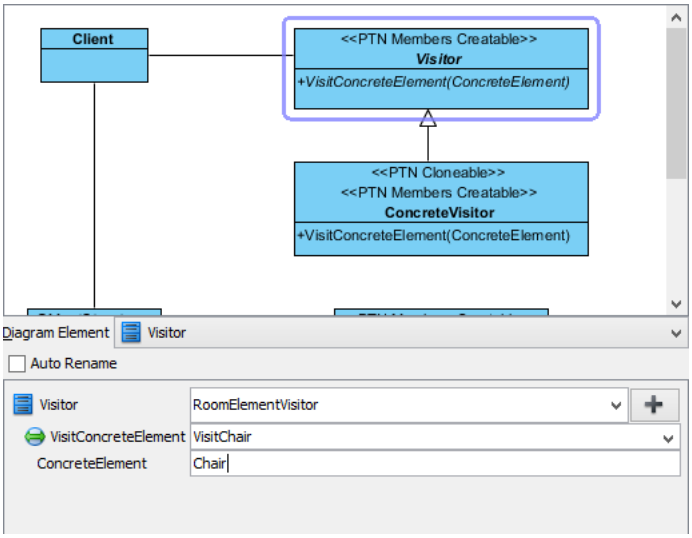
10. Enter `2` as the number of classes to clone. Click **OK** to confirm.



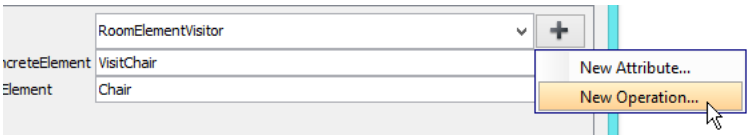
11. In the bottom pane, rename *ConcreteElement2* and *ConcreteElement3* to *Bed* and *Table*. For each, rename the parameter *Visitor* to *RoomElementVisitor*.



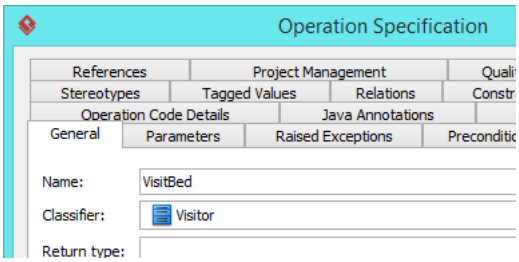
12. Select *Visitor* in the overview. In the bottom pane, rename it to *RoomElementVisitor*. Rename the operation *VisitConcreteElement* to *VisitChair*, and its parameter *ConcreteElement* to *Chair*.



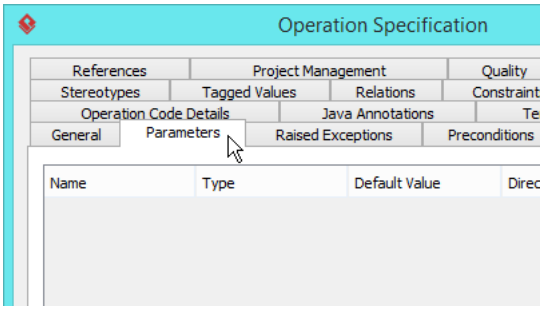
13. We need two more operations for visiting the bed and table. Keep *Visitor* selected, click the **+** button, and select **New Operation...** from the popup menu.



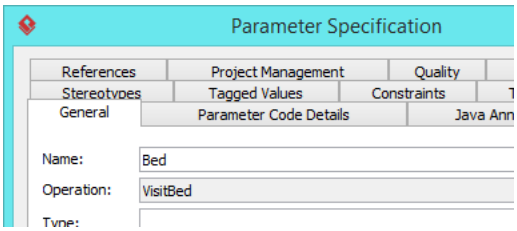
14. In the **Operation Specification** dialog box, name the operation `VisitBed`.



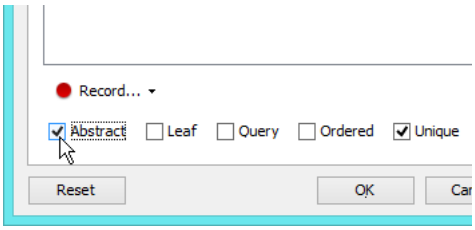
15. Open the **Parameters** tab.



16. Click **Add...** at the bottom, and create a parameter named `Bed` of type `Bed` in the **Parameter Specification** dialog box. Click **OK** to go back to the **Operation Specification** dialog box.

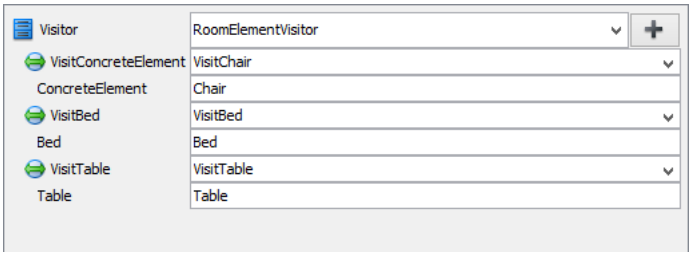


17. In the **General** page, check **Abstract**.

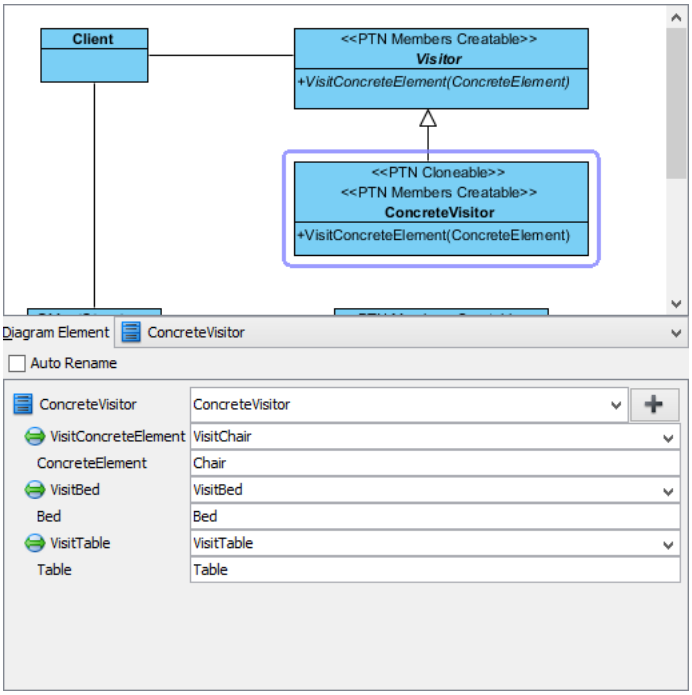


18. Click **OK** to confirm editing.

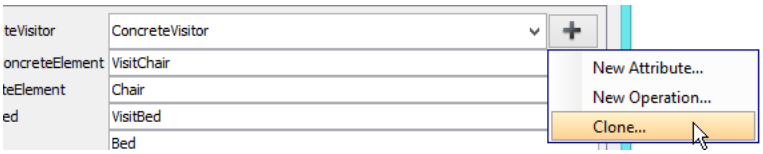
19. Repeat the previous steps to create one more abstract operation, `VisitTable`, which has a `Table` parameter.



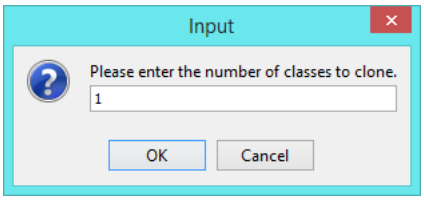
20. Select *ConcreteVisitor* in the overview. In the bottom pane, rename it to *RoomElementPaintVisitor*. Rename the operation *VisitConcreteElement* to *VisitChair*, and its parameter *ConcreteElement* to *Chair*.



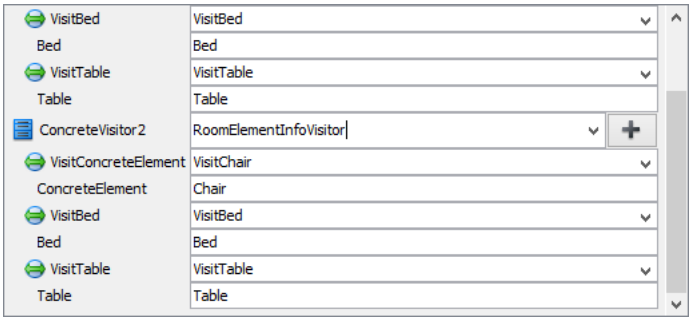
21. We need one more visitor for printing elements' information. Keep `ConcreteVisitor` selected, click the **+** button, and select **Clone...** from the popup menu.



22. Enter `1` as the number of classes to clone. Click **OK** to confirm.

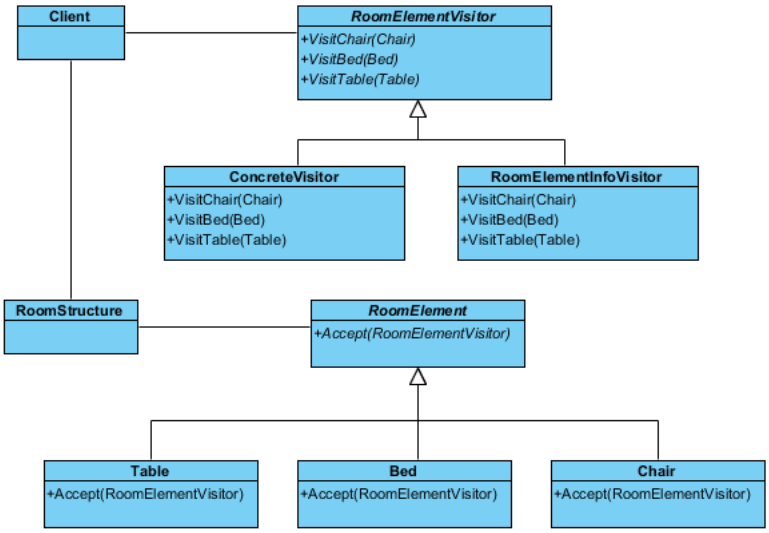


23. In the bottom pane, rename *ConcreteVisitor2* to *RoomElementInfoVisitor*. Rename the operation *VisitConcreteElement* to *VisitChair*, and its parameter *ConcreteElement* to *Chair*.



24. Click **OK** to confirm editing and apply the pattern to the diagram.

25. Tidy up the diagram. It should become:



Resources

1. Design Patterns.vpp
2. Visitor.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/)