

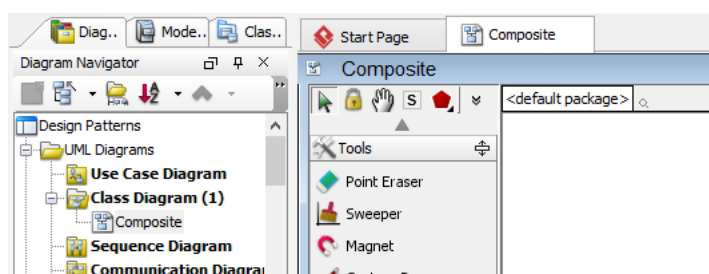


## Composite Pattern Tutorial

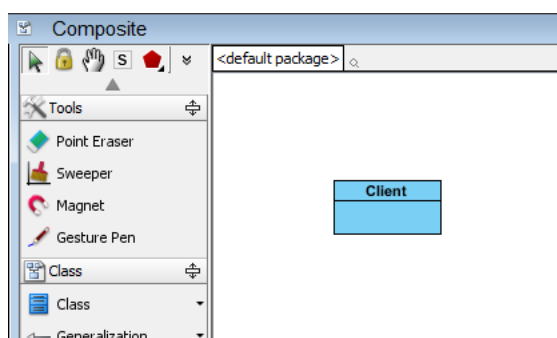
Written Date : October 7, 2009

### Modeling a Design Pattern with a Class Diagram

1. Create a new project named *Design Patterns*.
2. Create a class diagram named *Composite*.



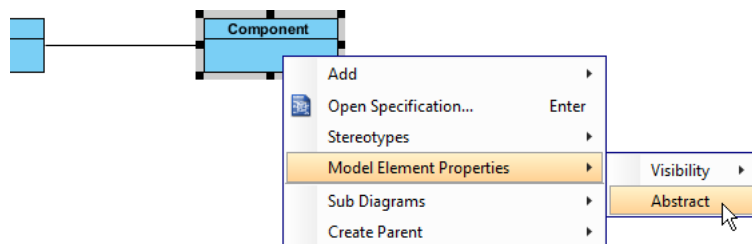
3. Select **Class** from the diagram toolbar. Click on the diagram to create a class and name it *Client*.



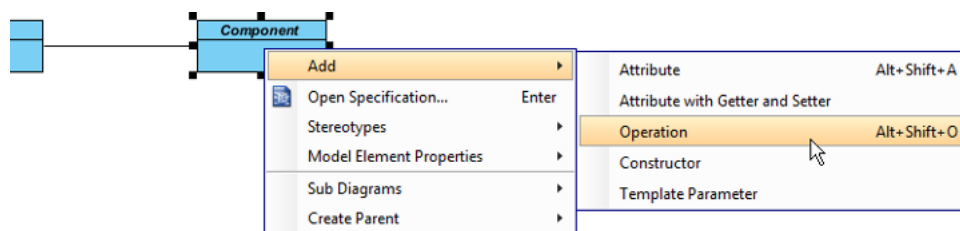
4. Move the mouse cursor over the *Client* class and drag out **Association > Class** to create an associated class named *Component*.



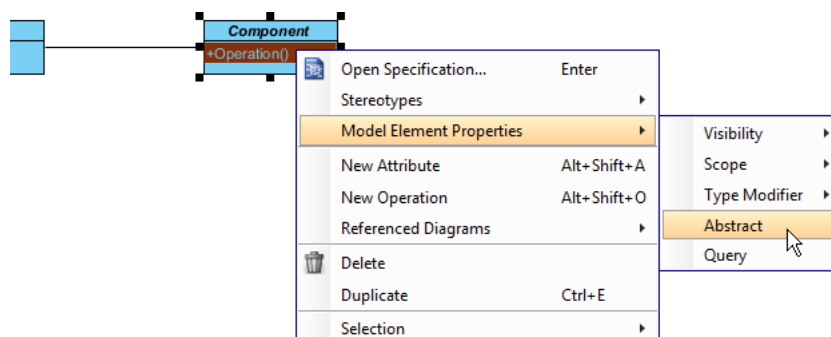
- Right-click on *Component* and select **Model Element Properties > Abstract** to set it as abstract.



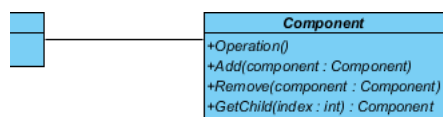
- Right-click on the *Component* class and select **Add > Operation** from the popup menu.



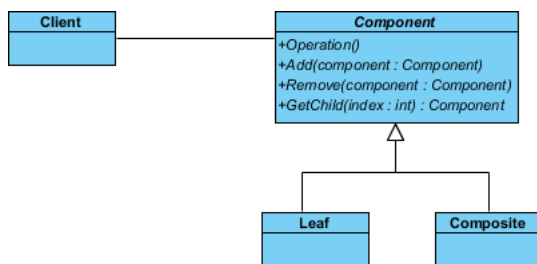
- Name the operation *Operation()*.
- Right-click on *Operation* and select **Model Element Properties > Abstract** to set it as abstract.



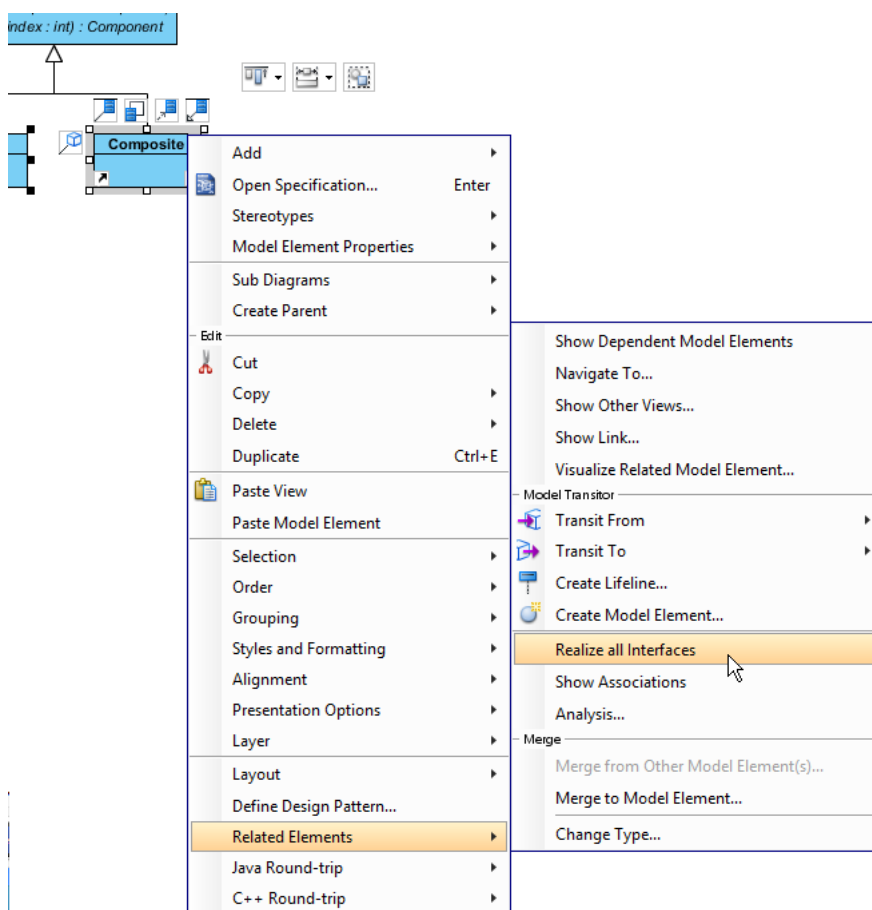
- Repeat steps 6 to 8 to create the operations *Add(component : Component)*, *Remove(component : Component)*, and *GetChild(index : int) : Component*.



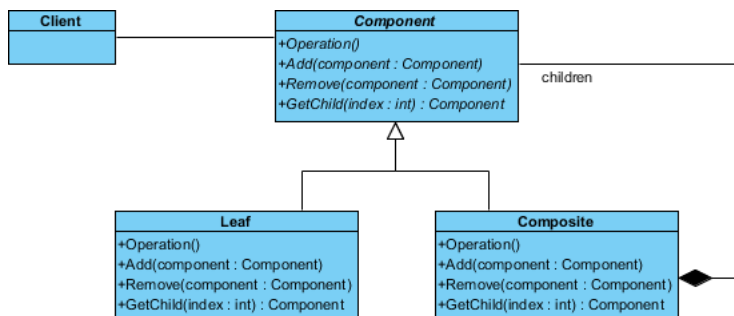
- Move the mouse cursor over the *Component* class and drag out **Generalization > Class** to create a subclass named *Leaf*. Repeat this step to create another subclass named *Composite* from *Component*.



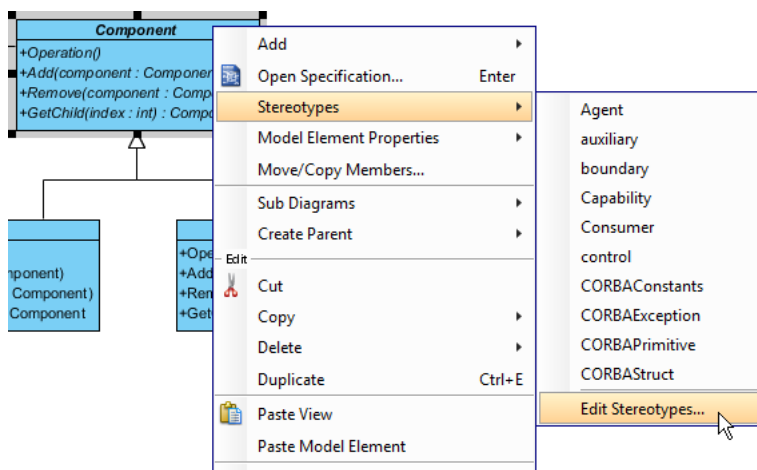
- Leaf* and *Composite* inherit the operations from *Component*. Select both *Leaf* and *Composite*, then right-click on them and select **Related Elements > Realize all Interfaces** from the popup menu.



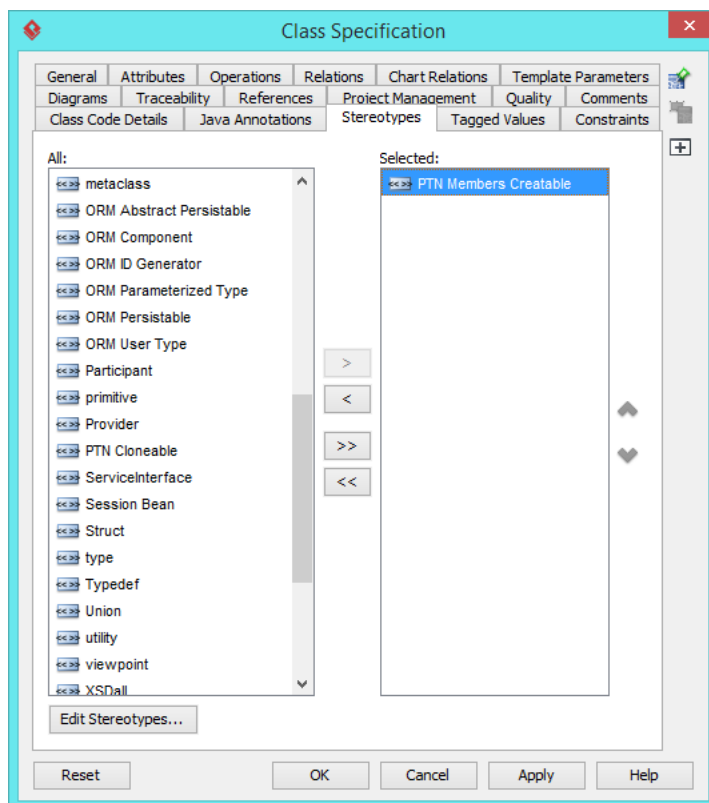
12. Move the mouse cursor over the *Component* class and drag out **Composition > Class** to *Component*. Name the *Component*'s role "children."



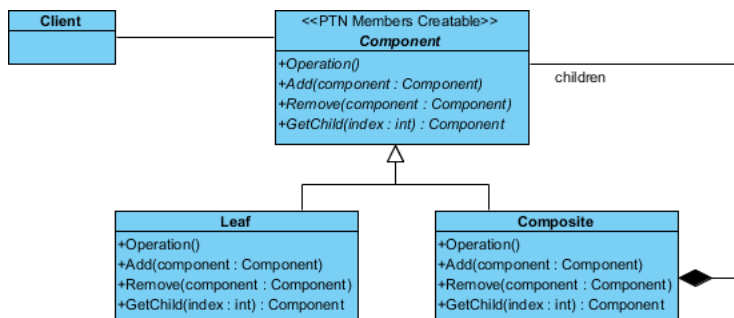
13. In practice, there may be multiple operations in *Component*. To represent this, stereotype the *Component* class as **PTN Members Creatable**. Right-click on *Component* and select **Stereotypes > Stereotypes...** from the popup menu.



- In the **Stereotypes** tab of the **Class Specification** dialog box, select **PTN Members Creatable** and click > to assign it to the *Component* class. Click **OK** to confirm.

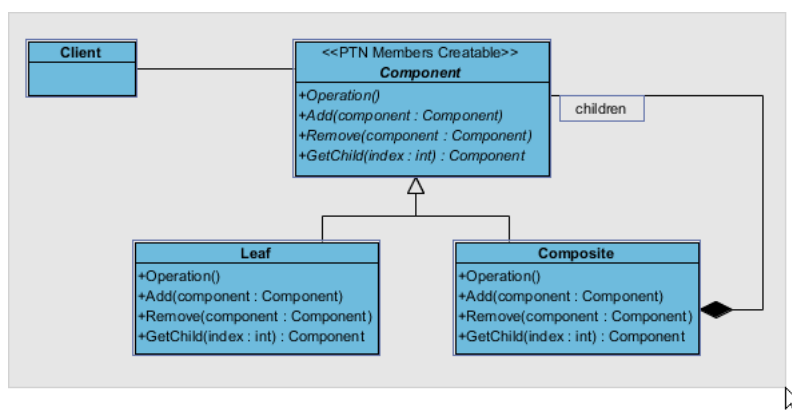


Up to now, the diagram should look like this:

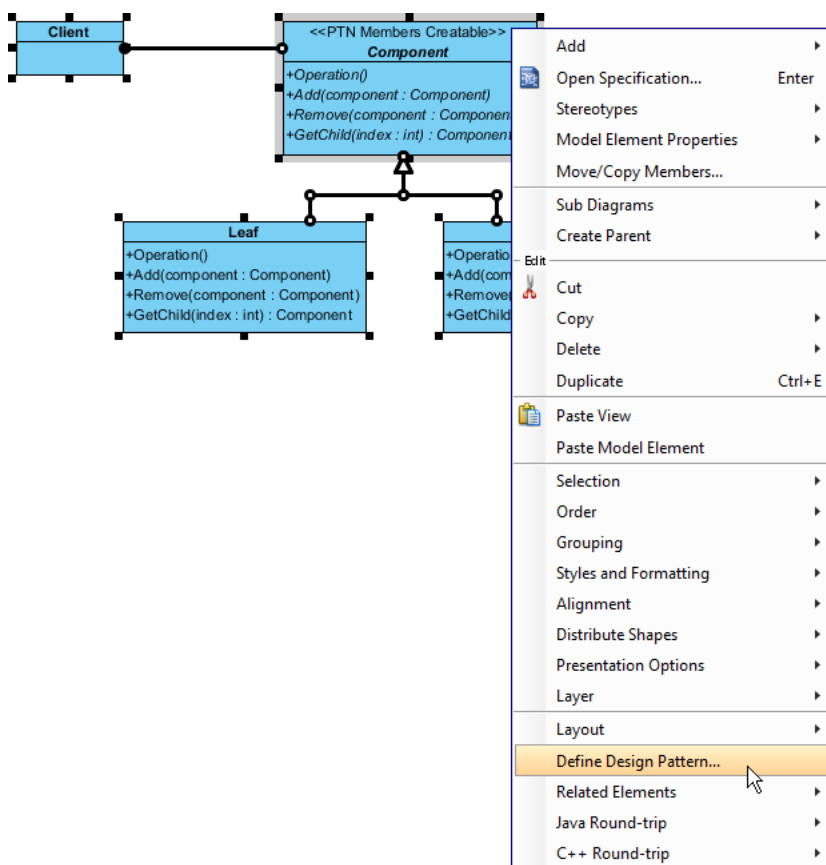


## Defining a Pattern

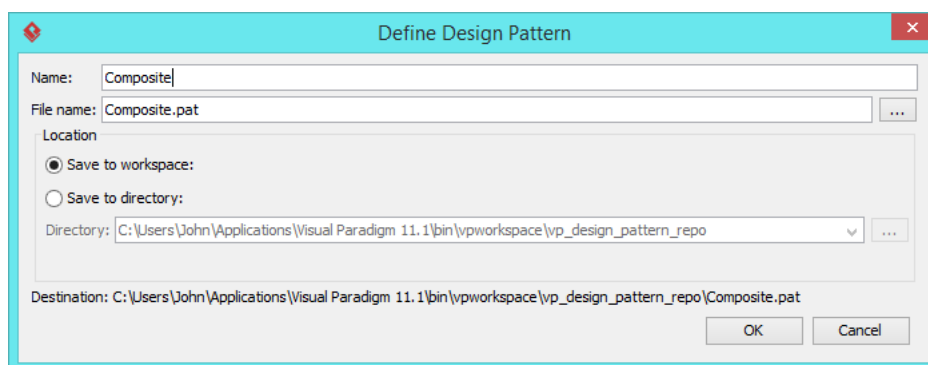
1. Select all classes on the class diagram.



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



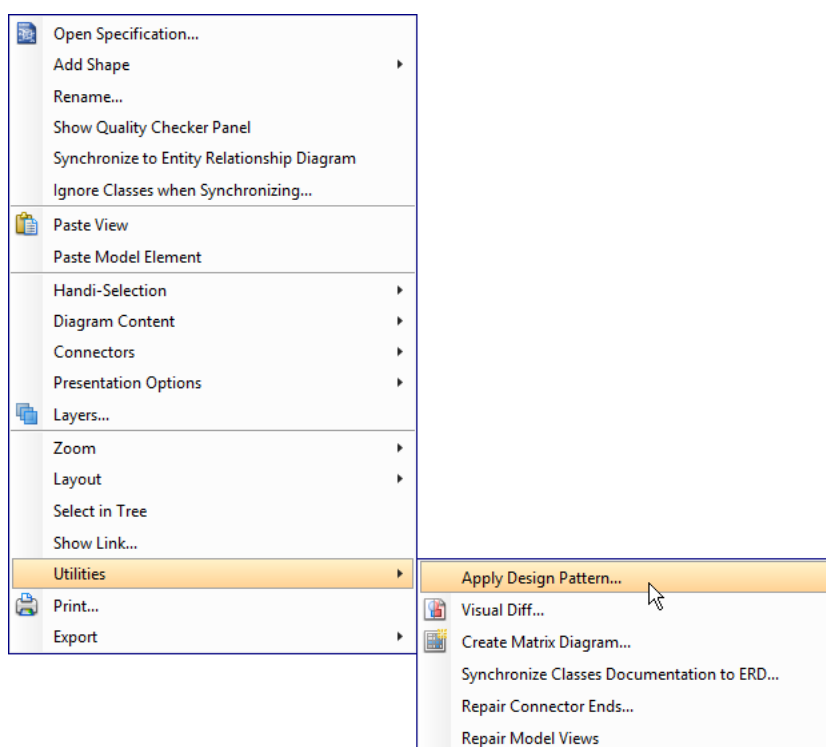
3. In the **Define Design Pattern** dialog box, specify the pattern name as *Composite*. Keep the file name as is and click **OK** to proceed.



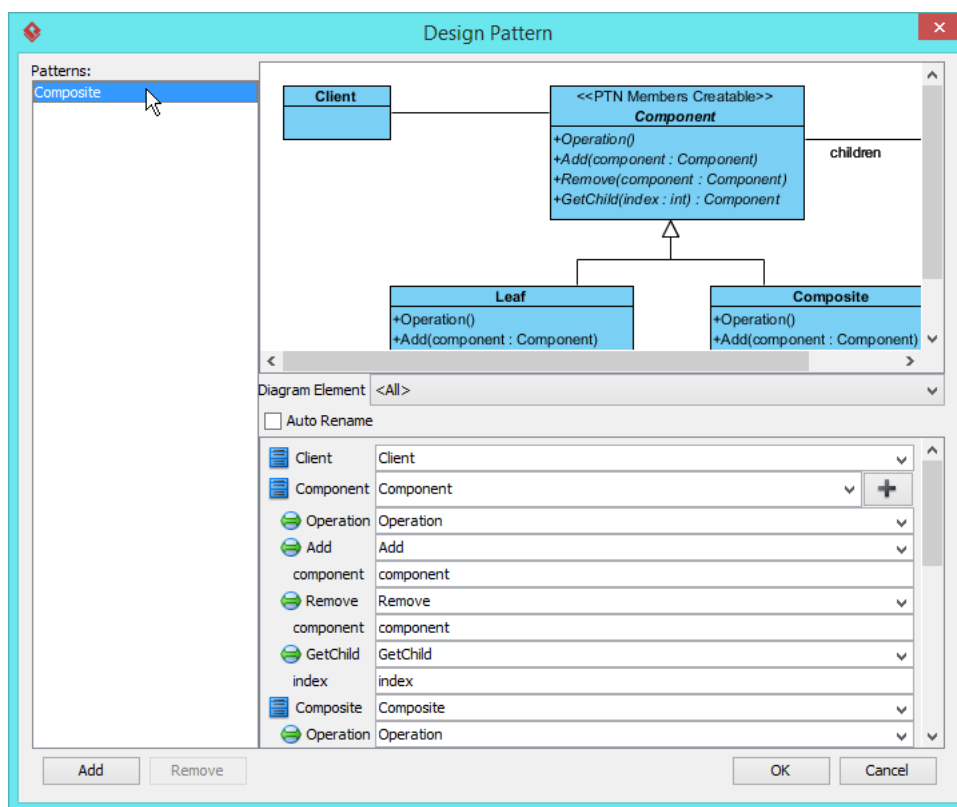
## Applying a Design Pattern to a Class Diagram

In this section, we will apply the composite pattern to model a furniture shop's furniture catalog.

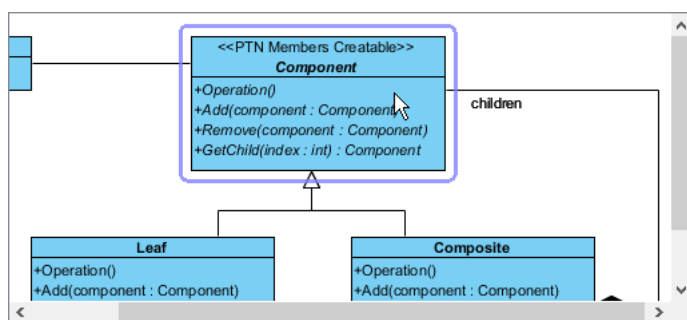
1. Create a new project named *Furniture Shop*.
2. Create a class diagram named *Domain Model*.
3. Right-click on the class diagram and select **Utilities > Apply Design Pattern...** from the popup menu.



- In the **Design Pattern** dialog box, select *Composite* from the list of patterns.



- Click on *Component* in the overview.



6. Rename *Component* to *Furniture*, and the "component" parameters in various operations to "furniture" in the bottom pane.



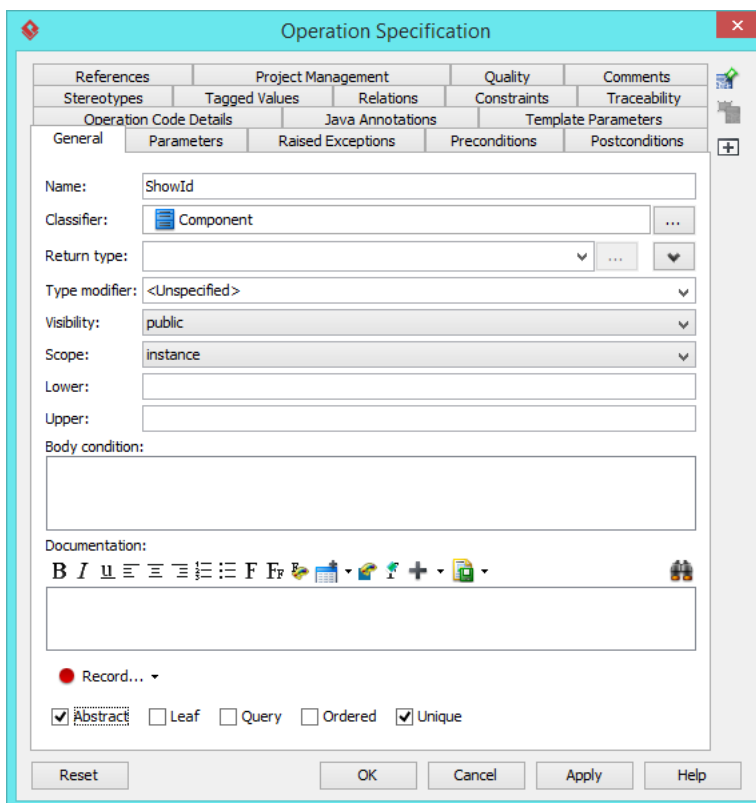
7. Rename *Operation* to *ShowPrice*.



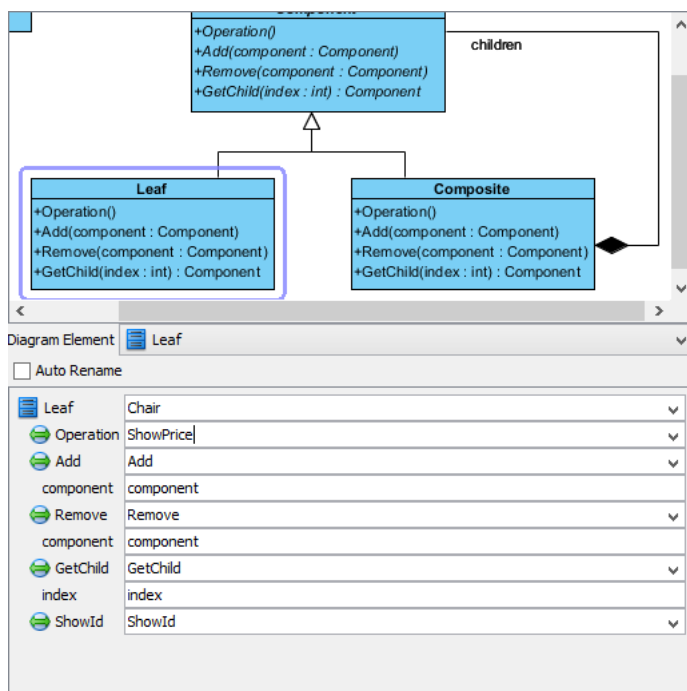
8. Besides the *ShowPrice* operation, we also need one more for *ShowId*. Keep *Component* selected, click the + button in the bottom pane, and select **New Operation...** from the popup menu.



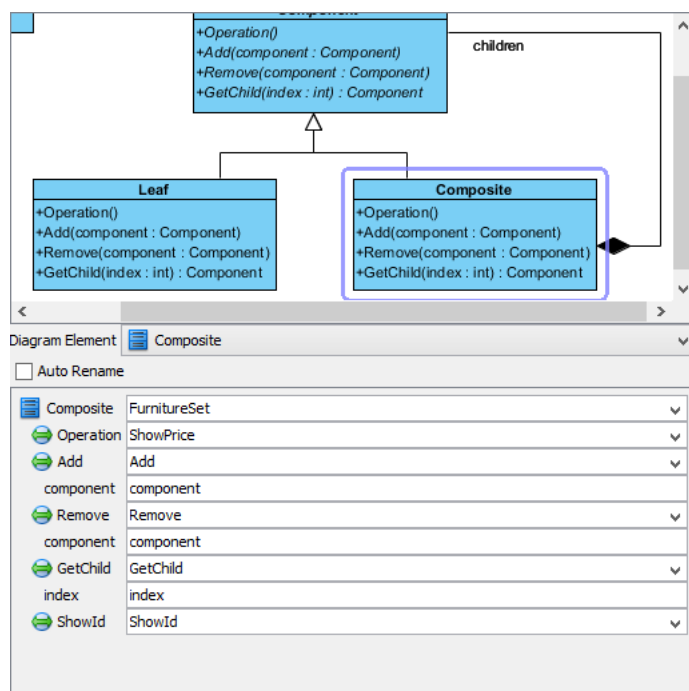
9. In the **Operation Specification** dialog box, name the operation *ShowId*. Check the **Abstract** box at the bottom of the dialog.



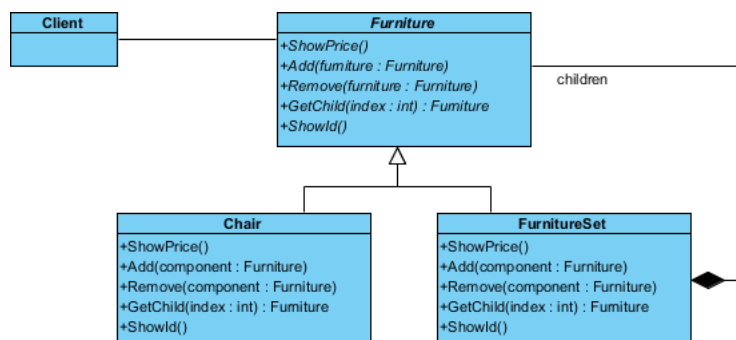
10. Select *Leaf* in the overview and rename it to *Chair* in the bottom pane. Also, rename the *Operation* operation to *ShowPrice*. Note that if the **Auto Rename** option is on, renaming the operation is not necessary as this will be done automatically.



11. Select *Composite* in the overview and rename it to *FurnitureSet* in the bottom pane. Also, rename *Operation* to *ShowPrice*. Click **OK** to apply the pattern to the diagram.



This is the result:



### Resources

1. [Composite.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/>)