



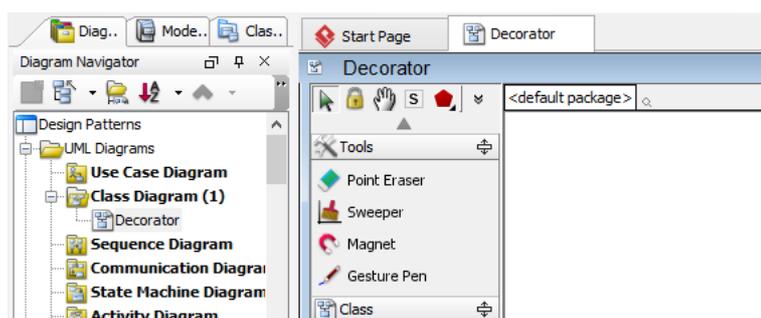
Decorator Pattern Tutorial

Written Date : October 8, 2009

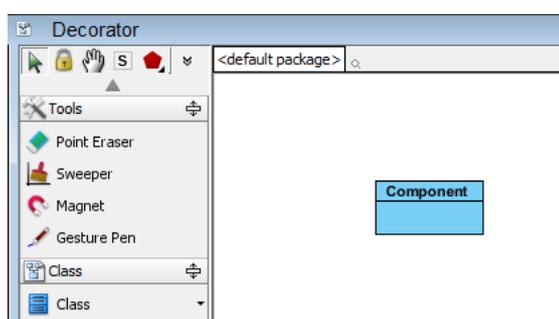
This tutorial is aimed to guide the definition and application of [Gang of Four \(GoF\)](#) decorator [design pattern](#). By reading this tutorial, you will know how to develop a model for the decorator pattern, and how to apply it in practice.

Modeling Design Pattern with Class Diagram

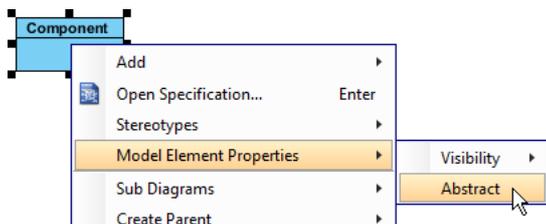
1. Create a new project *Design Patterns*.
2. Create a class diagram *Decorator*.



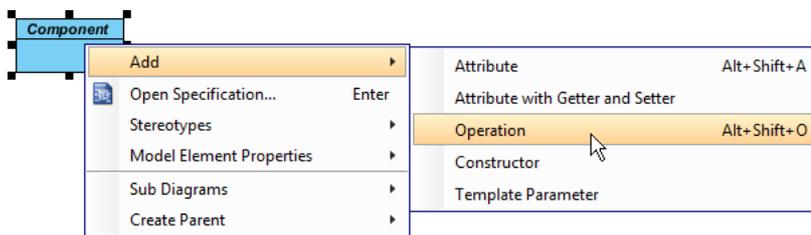
3. Select **Class** from diagram toolbar. Click on the diagram to create a class. Name it as *Component*.



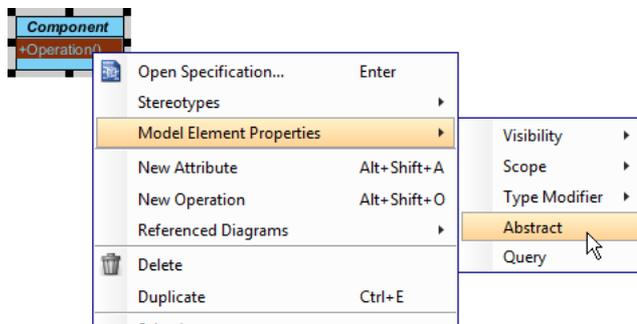
- Right-click on *Component*, and select **Model Element Properties** > **Abstract** to set it as abstract.



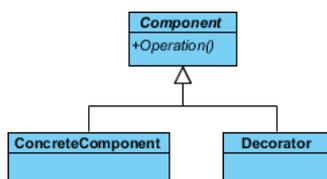
- Right-click on the *Component* class, and select **Add** > **Operation** from the popup menu.



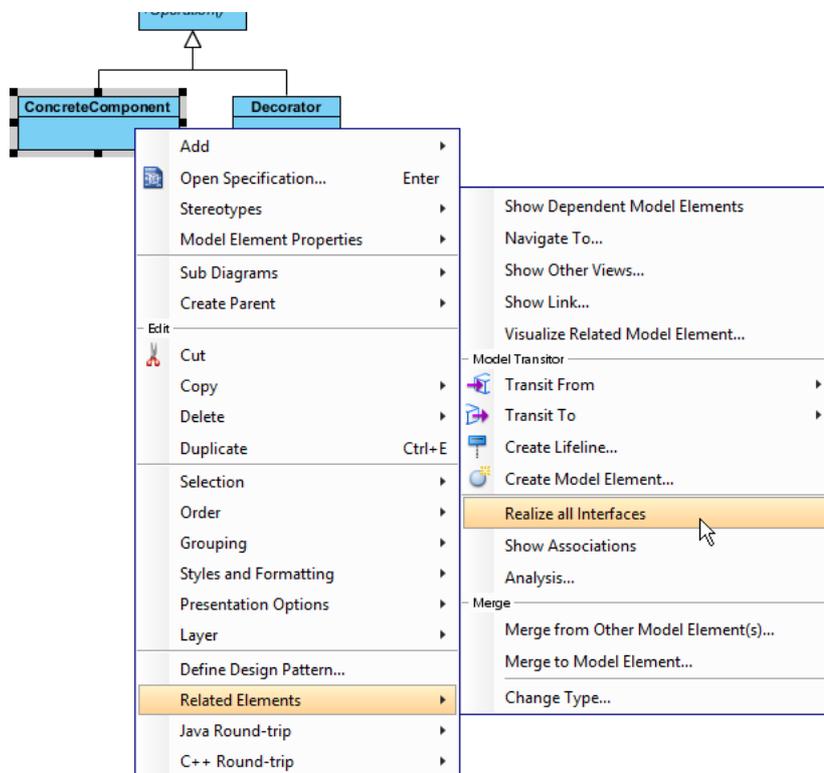
- Name the operation *Operation()*.
- Right-click on *Operation*, and select **Model Element Properties** > **Abstract** to set it as abstract.



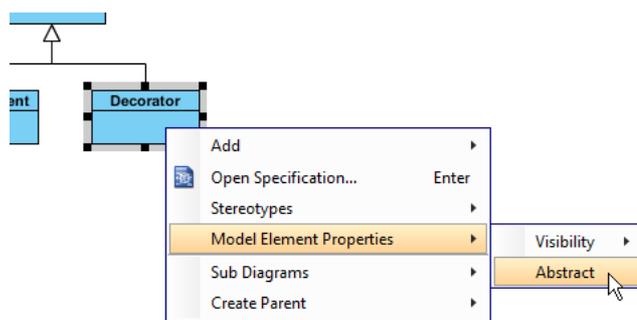
- Move the mouse cursor over the *Component* class, and drag out **Generalization** > **Class** to create a subclass *ConcreteComponent*. Repeat this step to create another subclass *Decorator*.



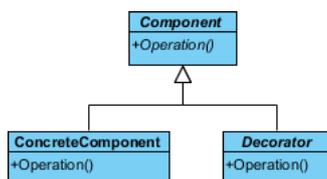
9. *ConcreteComponent* should inherit the operations from *Component*. Select *ConcreteComponent*, right-click on and select **Related Elements > Realize all Interfaces** from the popup menu.



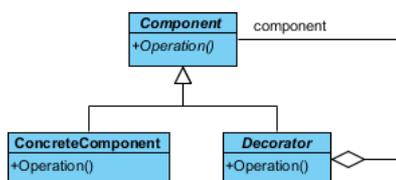
10. *Decorator* is an abstract class. Right-click on the *Decorator* class, and select **Model Element Properties > Abstract** to set it as abstract.



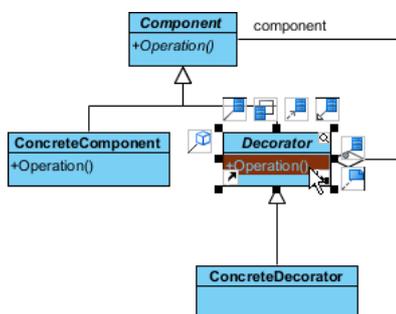
- Decorator should also inherit the operations from *Component*. Select *Decorator*, right-click on and select **Related Elements > Realize all Interfaces** from the popup menu.



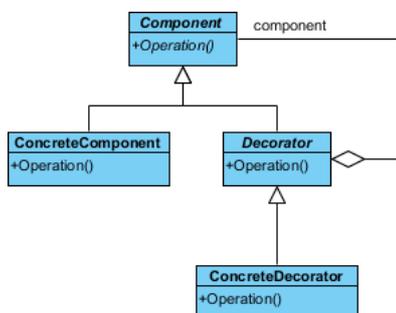
- Move the mouse cursor over the *Decorator* class, and drag out **Aggregation > Class** to *Component*. Name the *Component*'s role as *component*.



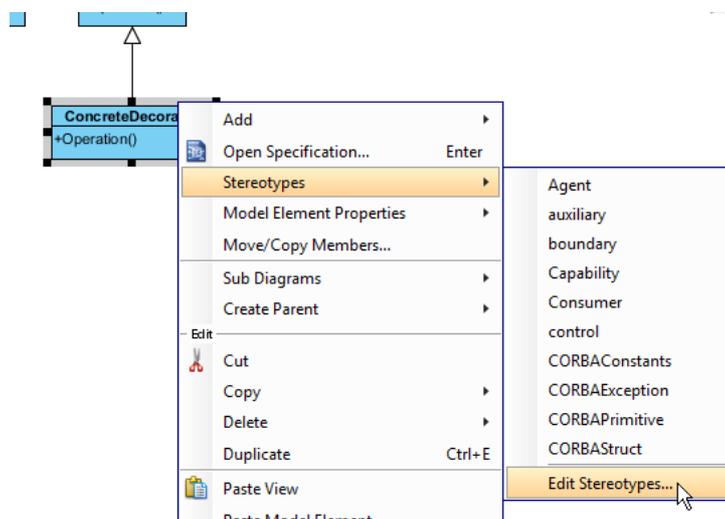
- Move the mouse cursor over the *Decorator* class, and drag out **Generalization > Class** to create a subclass *ConcreteDecorator*.
- We shall make *ConcreteDecorator* implements the decorator operation. Select *Operation* in *Decorator*.



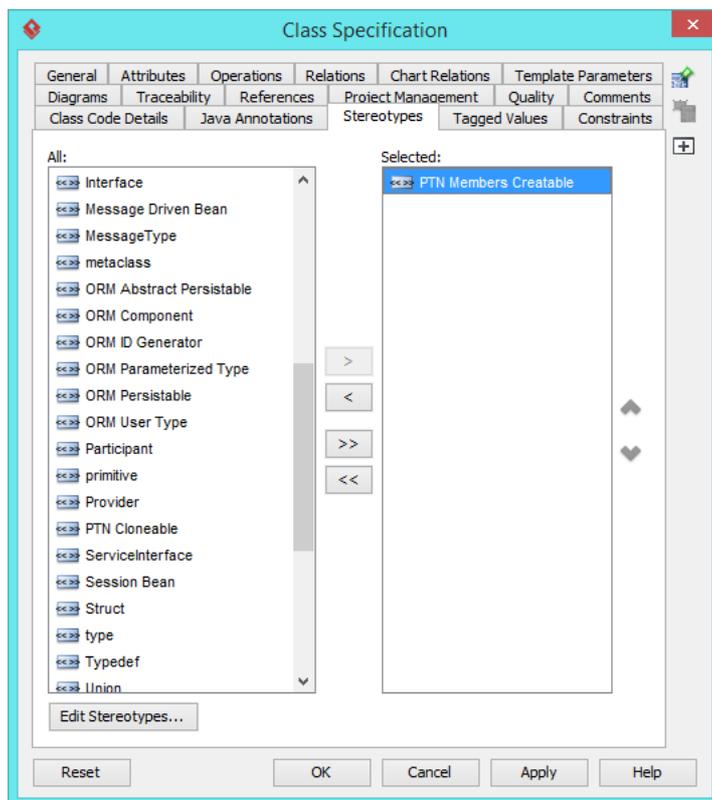
- Press the **Ctrl** key, and drag to *ConcreteDecorator*.



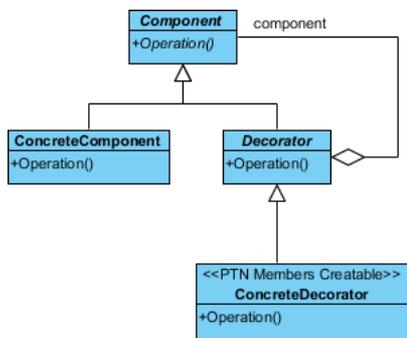
16. In practice, there may be added behaviors in concrete decorators. To represent this, stereotype the class *ConcreteDecorator* as **PTN Members Creatable**. Right-click on *ConcreteDecorator* and select **Stereotypes > Stereotypes...** from the popup menu.



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

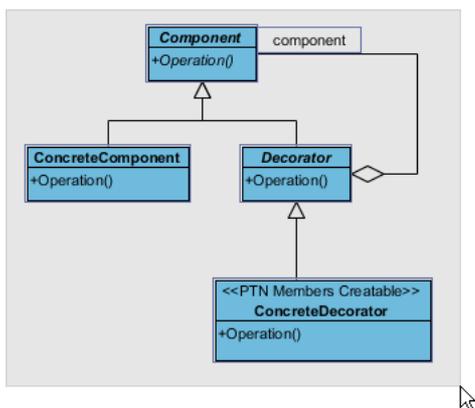


Up to now, the diagram should look like this:

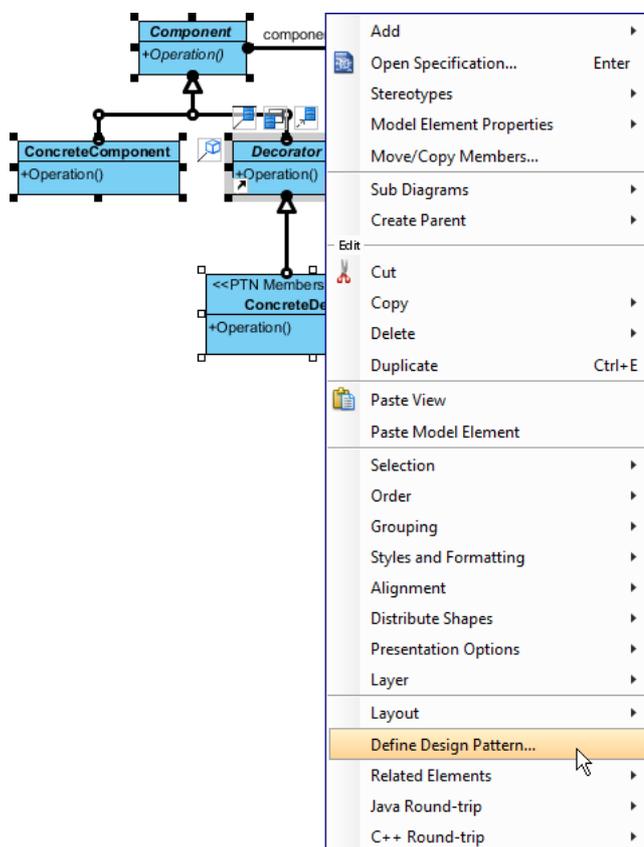


Defining Pattern

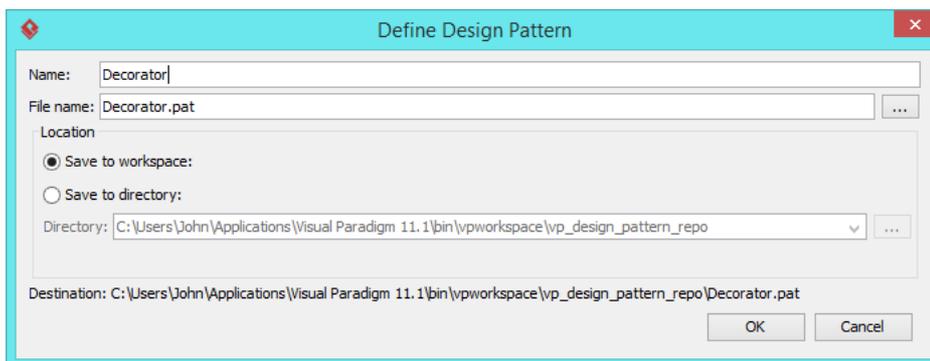
1. Select all classes on the class diagram.



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



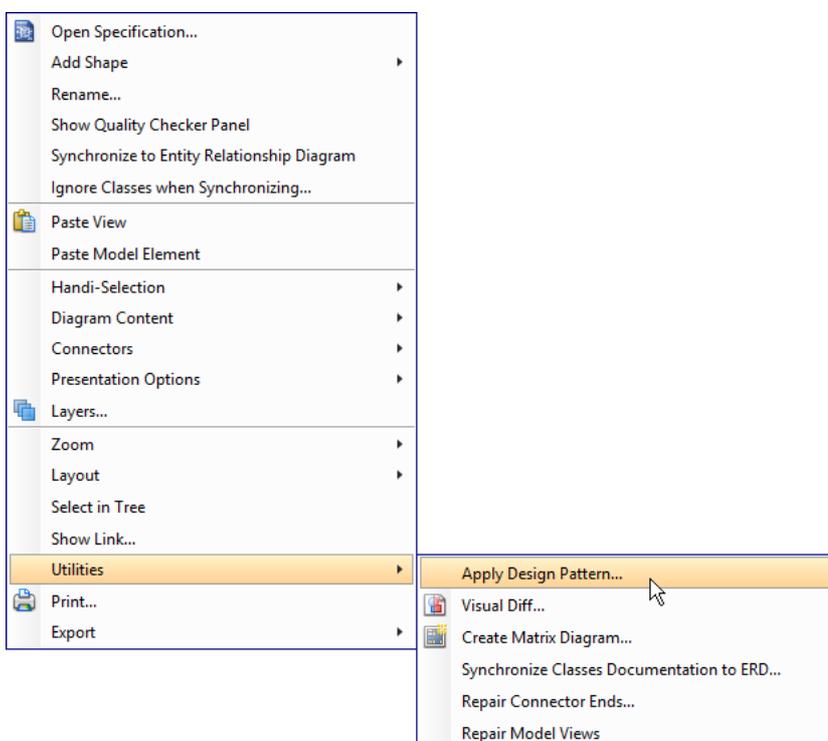
3. In the **Define Design Pattern** dialog box, specify the pattern name *Decorator*. Keep the file name as it. Click **OK** to proceed.



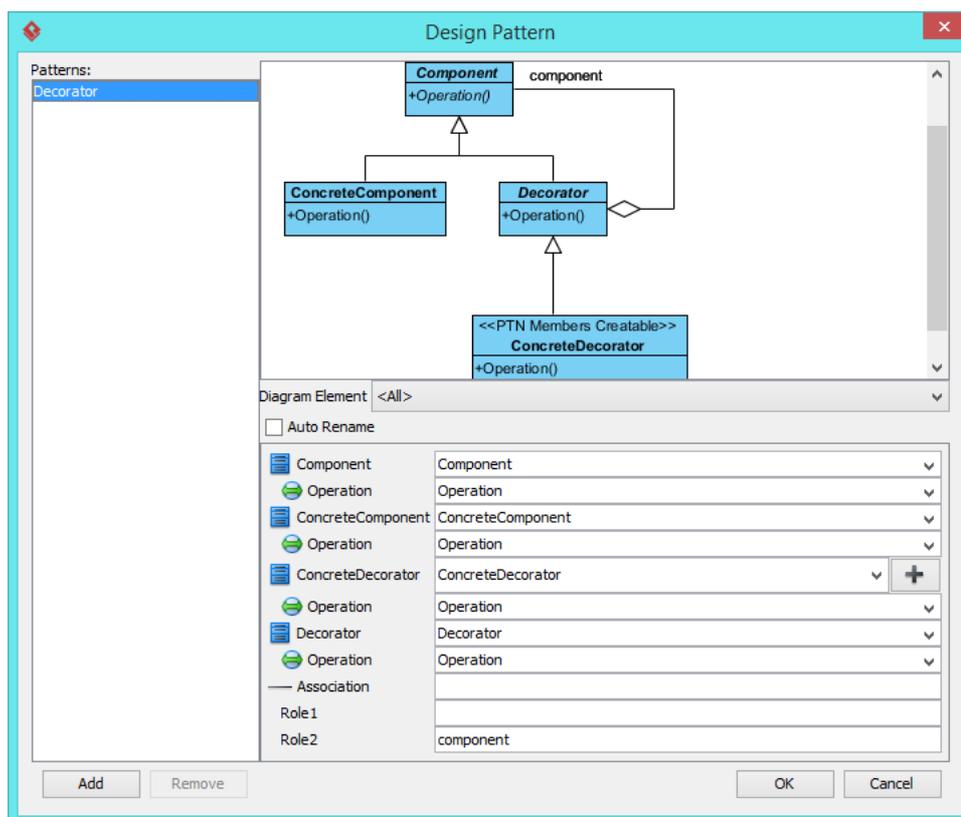
Applying Design Pattern on Class Diagram

In this section, we are going to apply the decorator pattern to model a domain model of diagram editor.

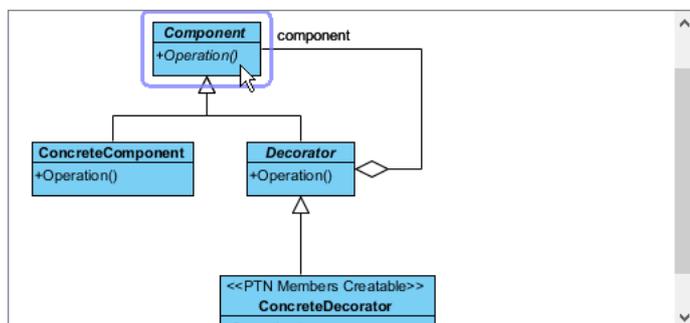
1. Create a new project *Diagram Editor*.
2. Create a class diagram *Domain Model*.
3. Right-click on the class diagram and select **Utilities > Apply Design Pattern...** from the popup menu.



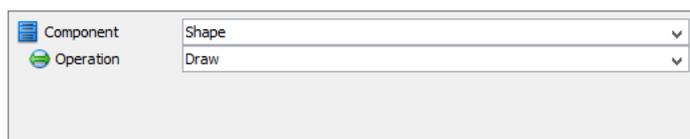
- In the **Design Pattern** dialog box, select *Decorator* from the list of patterns.



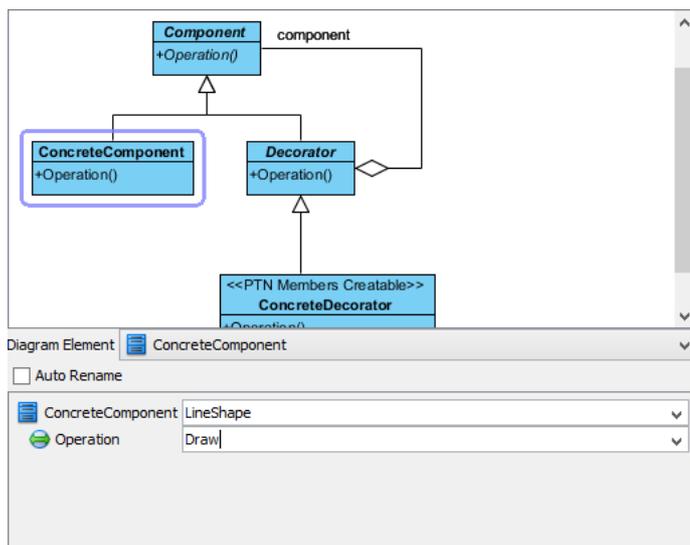
- Click on *Component* in the overview.



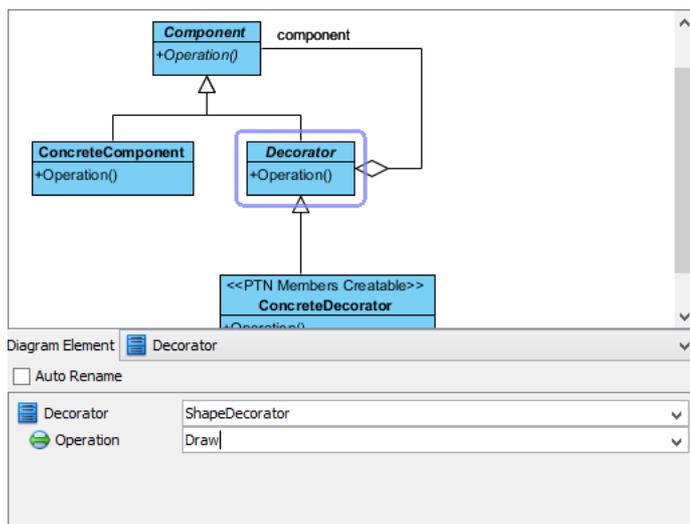
- Rename *Component* to *Shape* at the bottom pane, and operation *Operation* to *Draw*.



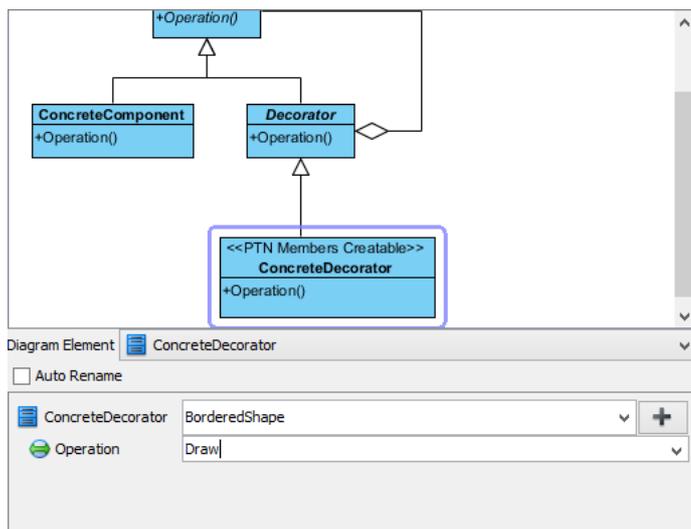
7. Select *ConcreteComponent* in overview, and rename it to *LineShape*, and its operation *Operation* to *Draw* at the bottom pane.



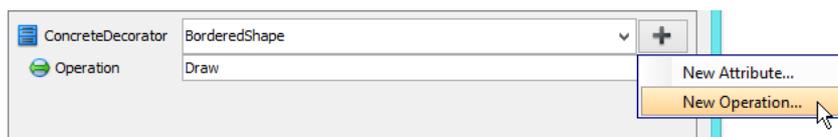
8. Select *Decorator* in overview, and rename it to *ShapeDecorator*, and its operation *Operation* to *Draw* at the bottom pane.



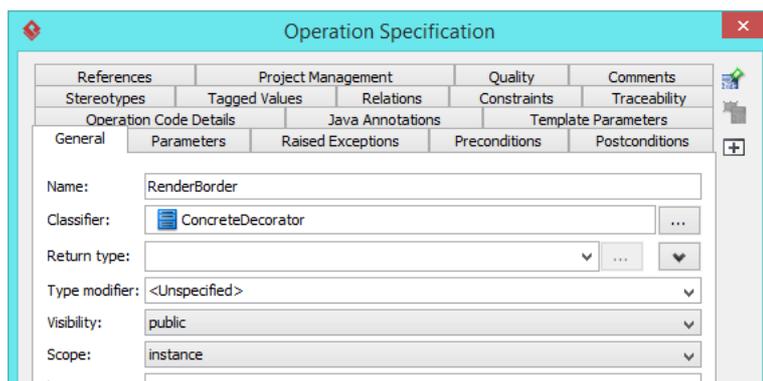
9. Select *ConcreteDecorator* in overview, and rename it to *BorderedShape*, and its operation *Operation* to *Draw* at the bottom pane.



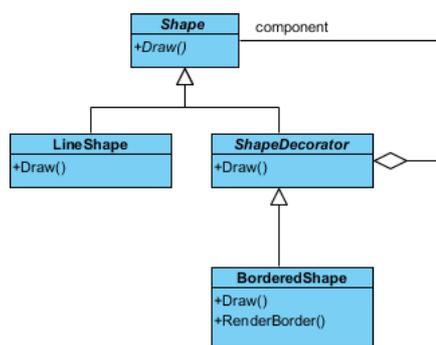
10. We need several operations for additional behaviors in *BorderedDecorator*. Click on the + button at the bottom pane and select **New Operation...** from the popup menu.



11. In the **Operation Specification** dialog box, enter *RenderBorder* as operation name. Click **OK** to confirm.



12. Click **OK** to apply the pattern to diagram. This is the final diagram:



Resources

1. [Decorator.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/>)