



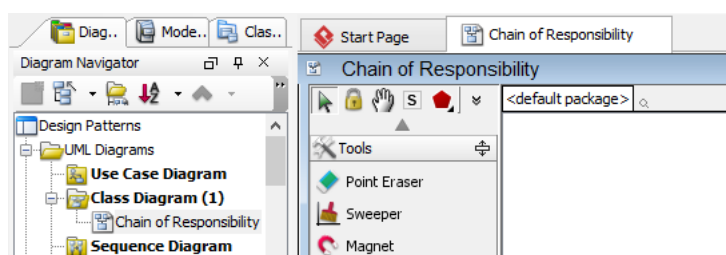
Chain of Responsibility Pattern Tutorial

Written Date : October 14, 2009

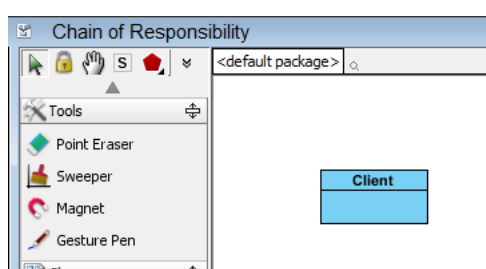
This tutorial is aimed to guide the definition and application of [Gang of Four \(GoF\)](#) chain of responsibility [design pattern](#). By reading this tutorial, you will know how to develop a model for the chain of responsibility pattern, and how to apply it in practice.

Modeling Design Pattern with Class Diagram

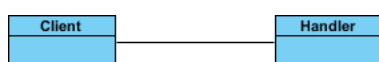
1. Create a new project *Design Patterns*.
2. Create a class diagram *Chain of Responsibility*.



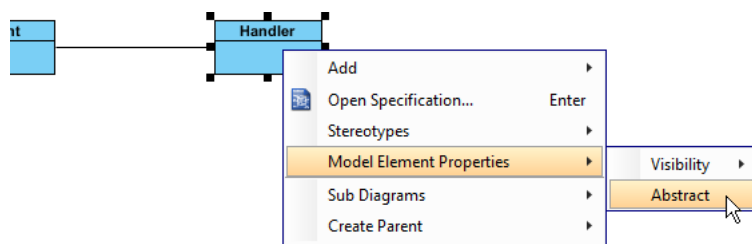
3. Select **Class** from diagram toolbar. Click on the diagram to create a class. Name it as *Client*.



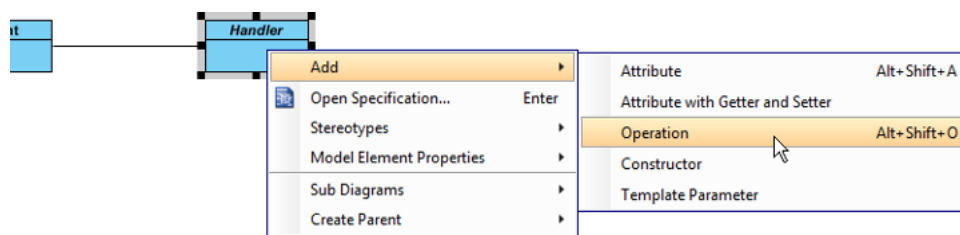
4. Move the mouse cursor over the *Client* class, and drag out **Association > Class** to create an associated class *Handler*.



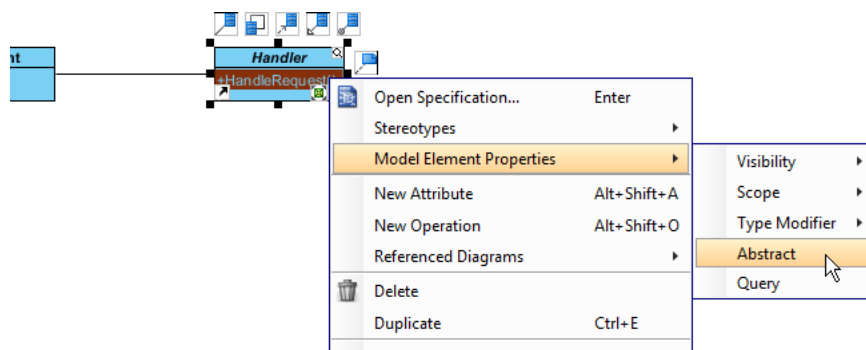
- Right-click on *Handler*, and select **Model Element Properties** > **Abstract** to set it as abstract.



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



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



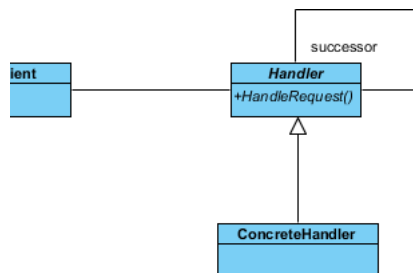
- Move the mouse cursor over the *Handler* class, and click on the resource icon **Self Association** to create a self association.



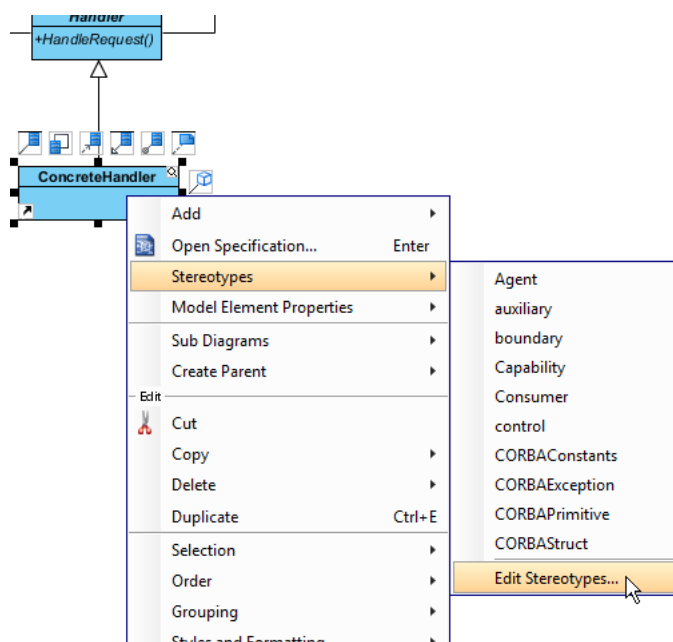
10. Name the association end successor.



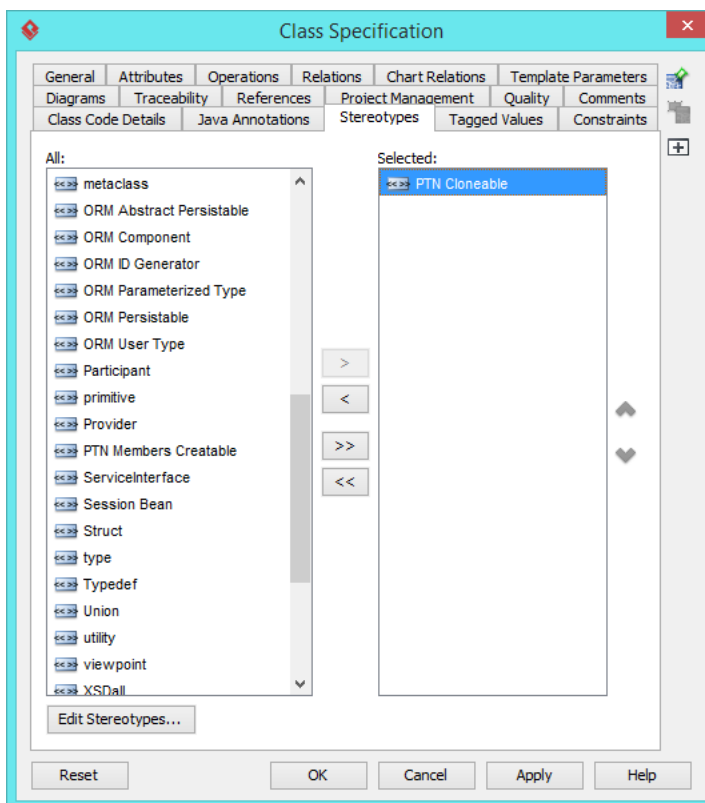
11. Move the mouse cursor over the *Handler* class, and drag out **Generalization > Class** to create subclasses *ConcreteHandler*.



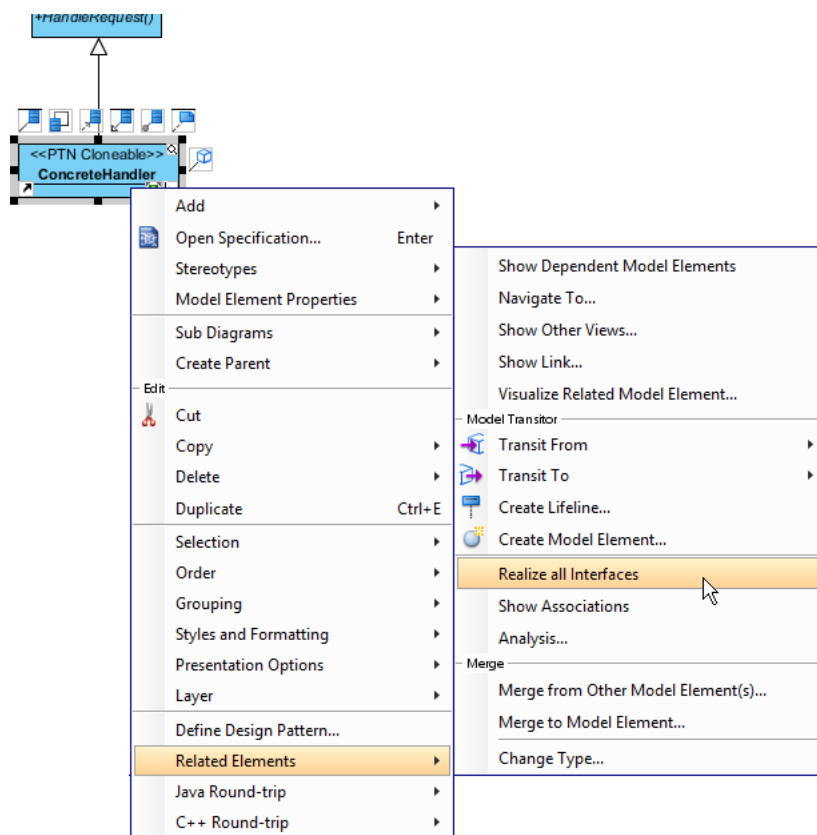
12. In practice, there may be multiple concrete handlers. To represent this, stereotype the class *ConcreteHandler* as **PTN Cloneable**. Right-click on *ConcreteHandler* and select **Stereotypes > Stereotype...** from the popup menu.



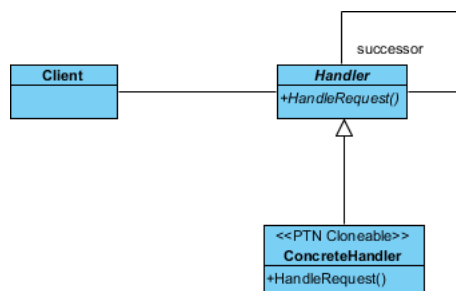
13. In the **Stereotypes** tab of the **Class Specification** dialog box, select **PTN Cloneable** and click **>** to assign it to *ConcreteHandler* class. Click **OK** to confirm.



14. We need make the concrete handlers inherit operations from the handle class. Right-click on *ConcreteHandler* and select **Related Elements > Realize all Interfaces** from the popup menu.

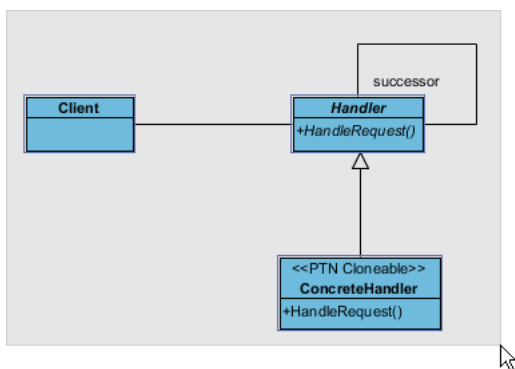


Up to now, the diagram should look like this:

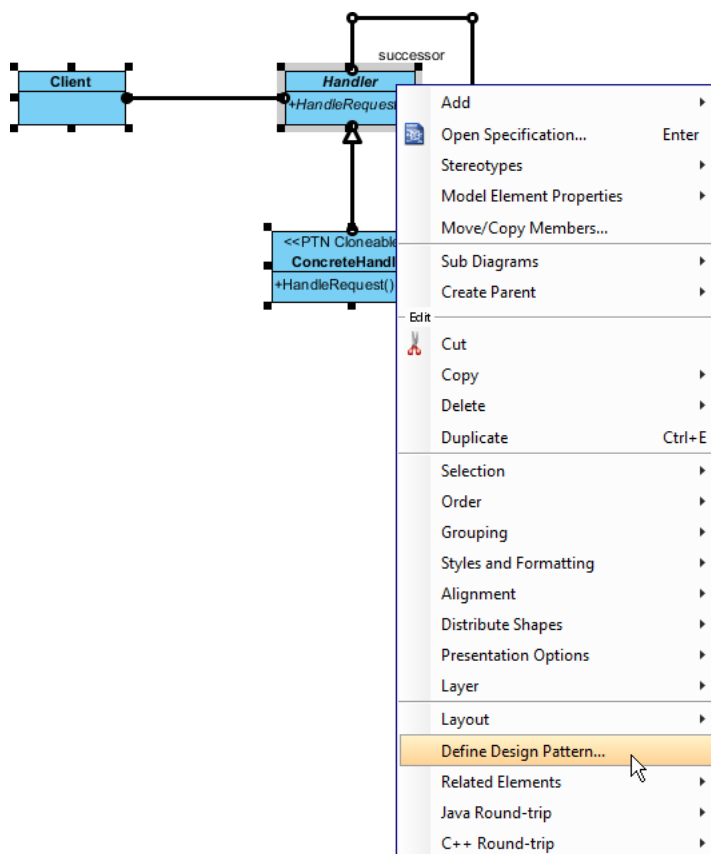


Defining Pattern

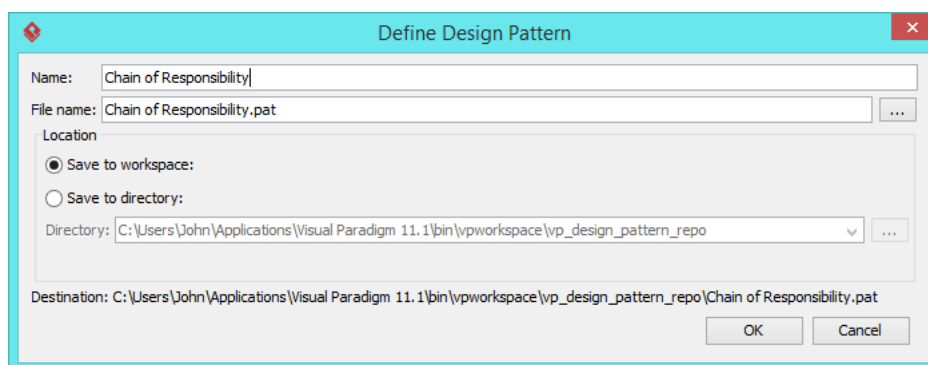
1. Select all classes on the class diagram.



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



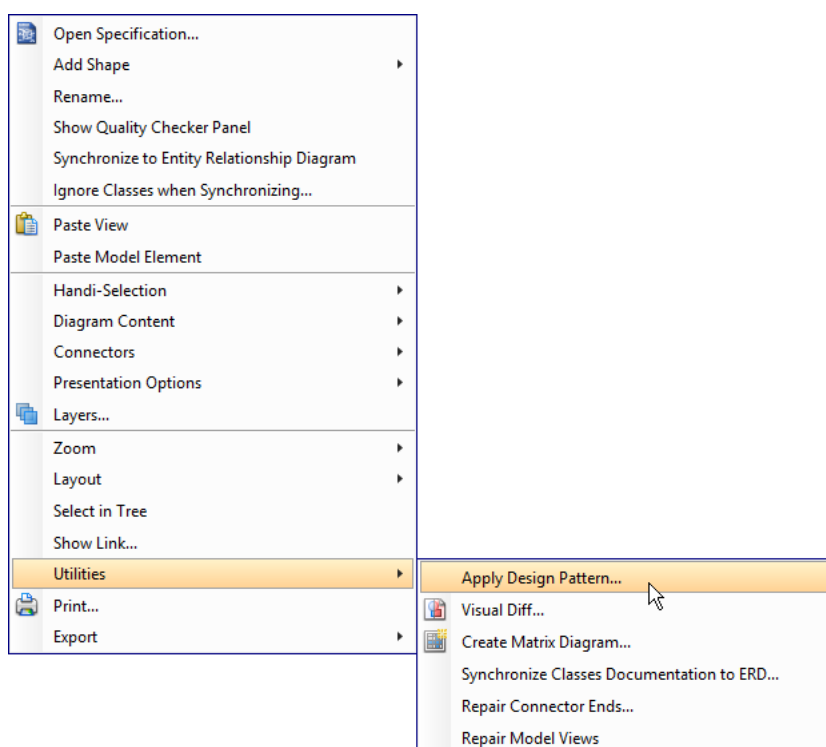
3. In the **Define Design Pattern** dialog box, specify the pattern name *Chain of Responsibility*. Keep the file name as is. Click **OK** to proceed.



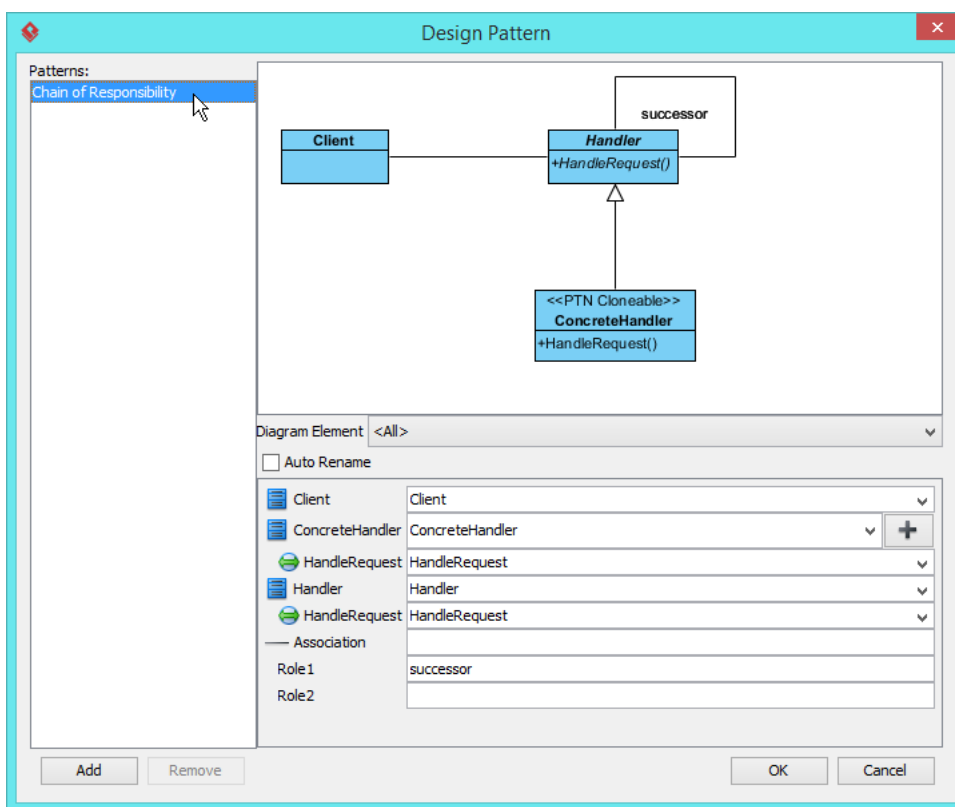
Applying Design Pattern on Class Diagram

In this section, we are going to apply the chain of responsibility pattern in modeling a coin dispenser.

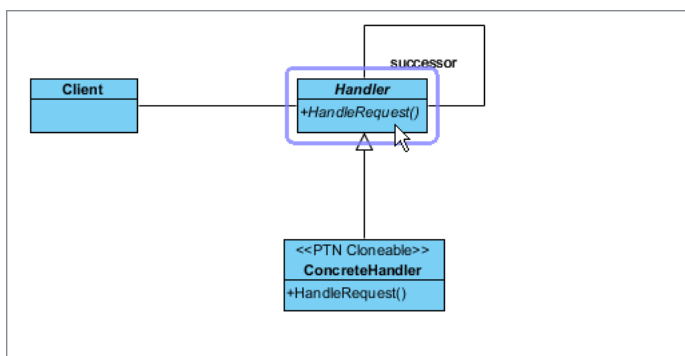
1. Create a new project *Coin Dispenser*.
2. Create a class diagram *Domain Model*.
3. Right-click on the class diagram and select **Utilities > Apply Design Pattern...** from the popup menu.



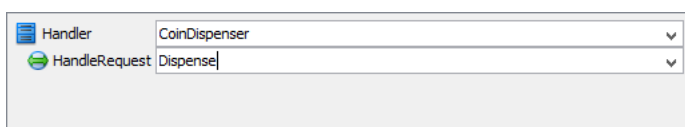
- In the **Design Pattern** dialog box, select *Chain of Responsibility* from the list of patterns.



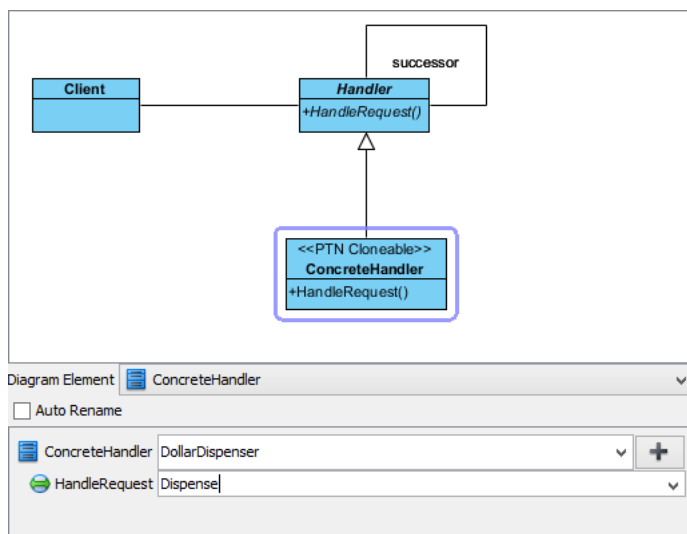
- Click on *Handler* in the overview.



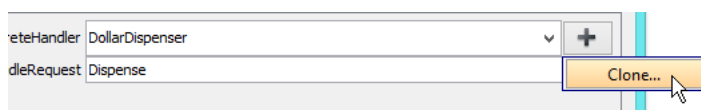
- Rename *Handler* to *CoinDispenser*, and operation *HandleRequest* to *Dispense* at the bottom pane.



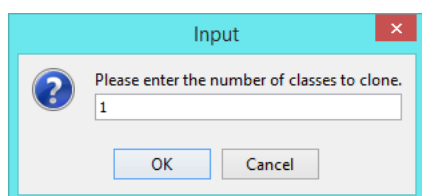
- Click on *ConcreteHandler* in overview, and rename it to *DollarDispenser*, and operation *HandleRequest* to *Dispense*.



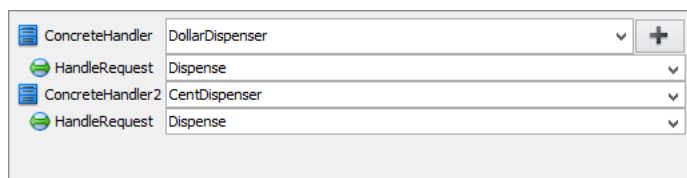
- We need one more concrete handler for dispensing cents. Keep *ConcreteHandler* selected, click on + and select **Clone...** from the popup menu.



- Enter 1 to be the number of classes to clone. Click **OK** to confirm.

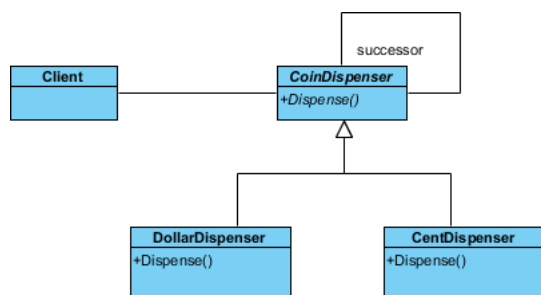


- Rename *ConcreteHandler2* to *CentDispenser*, and operation *HandleRequest* to *Dispense*.



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

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



Resources

1. [Chain of Responsibility.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/>)