



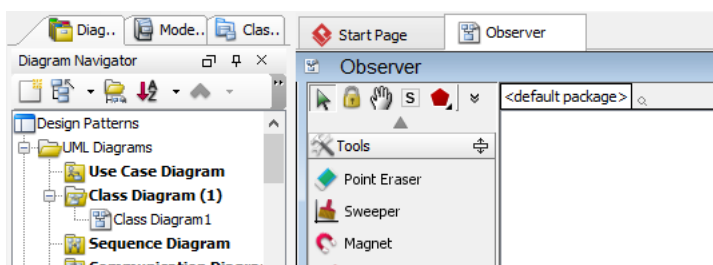
Observer Pattern Tutorial

Written Date : October 21, 2009

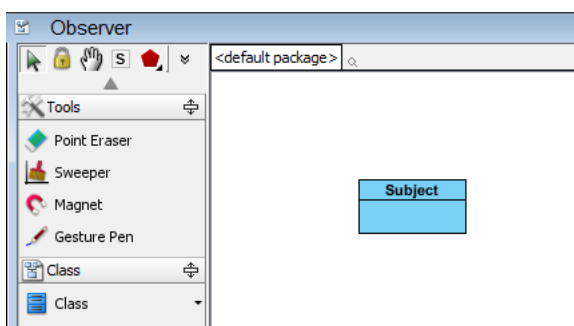
This tutorial is aimed to guide the definition and application of [Gang of Four \(GoF\)](#) observer [design pattern](#). By reading this tutorial, you will know how to develop a model for the observer pattern, and how to apply it in practice.

Modeling Design Pattern with Class Diagram

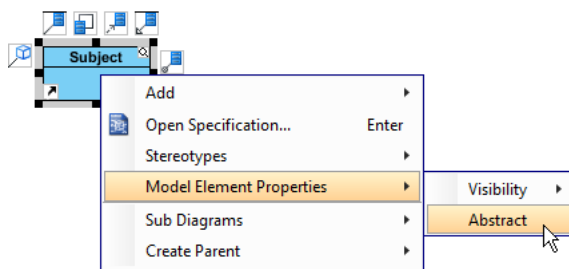
1. Create a new project *Design Patterns*.
2. Create a class diagram *Observer*.



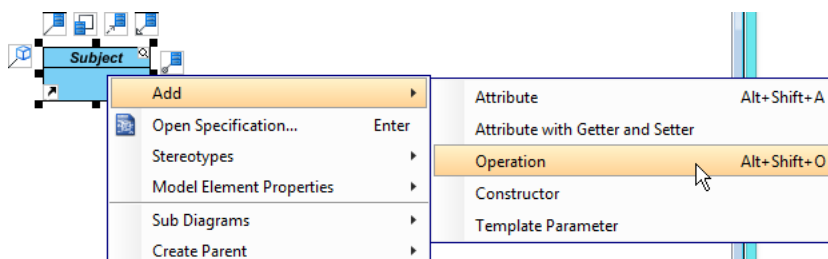
3. Select **Class** from diagram toolbar. Click on the diagram to create a class. Name it as *Subject*.



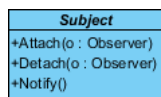
- Right click on *Subject*, and select **Model Element Properties > Abstract** to set it as abstract.



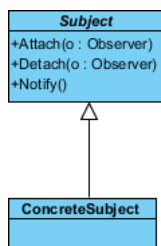
- Right click on *Subject* class, and select **Add > Operation** from the popup menu.



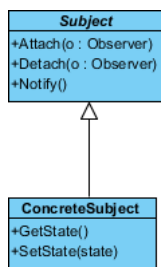
- Name the operation *Attach(o : Observer)*.
- Repeat step 5 and 6 to create the remaining two operations : *Detach (o : Observer)*, *Notify()*.



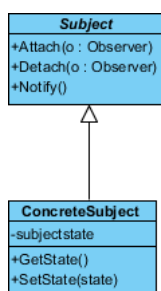
- Move the mouse cursor over the *Subject* class, and drag out **Generalization > Class** to create a subclass *ConcreteSubject*.



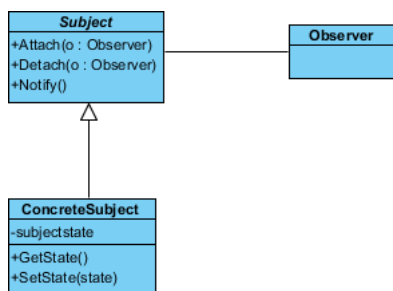
- Repeat steps 5 and 6 to create the following operations in *ConcreteSubject*: *GetState()*, *SetState(state)*.



- Right click on the *ConcreteSubject* class, and select **Add > Attribute** from the popup menu. Name the attribute *subjectstate*.

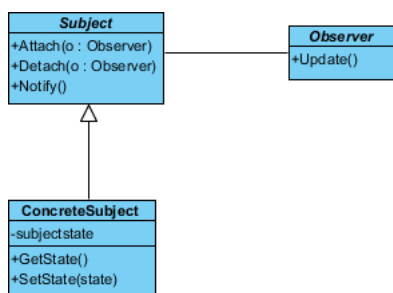


- Move the mouse cursor over the *Subject* class, and drag out **Association > Class** to create an associated class *Observer*.

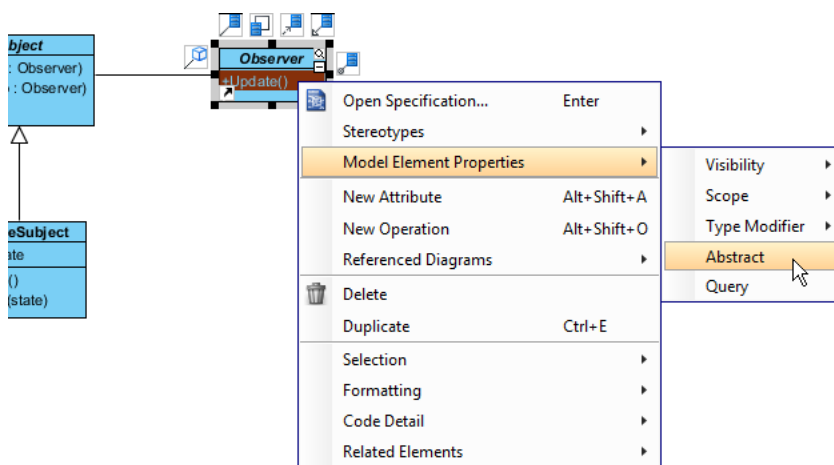


- Right click an *Observer*, and select **Model Element Properties > Abstract** to set it as abstract.

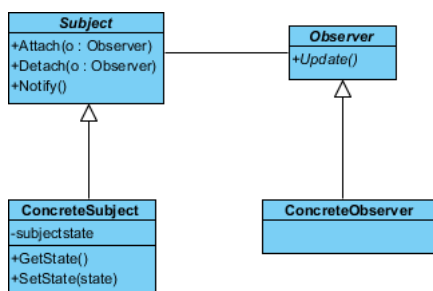
13. Repeat steps 5 and 6 to create the following operation in *Observer*: *Update()*.



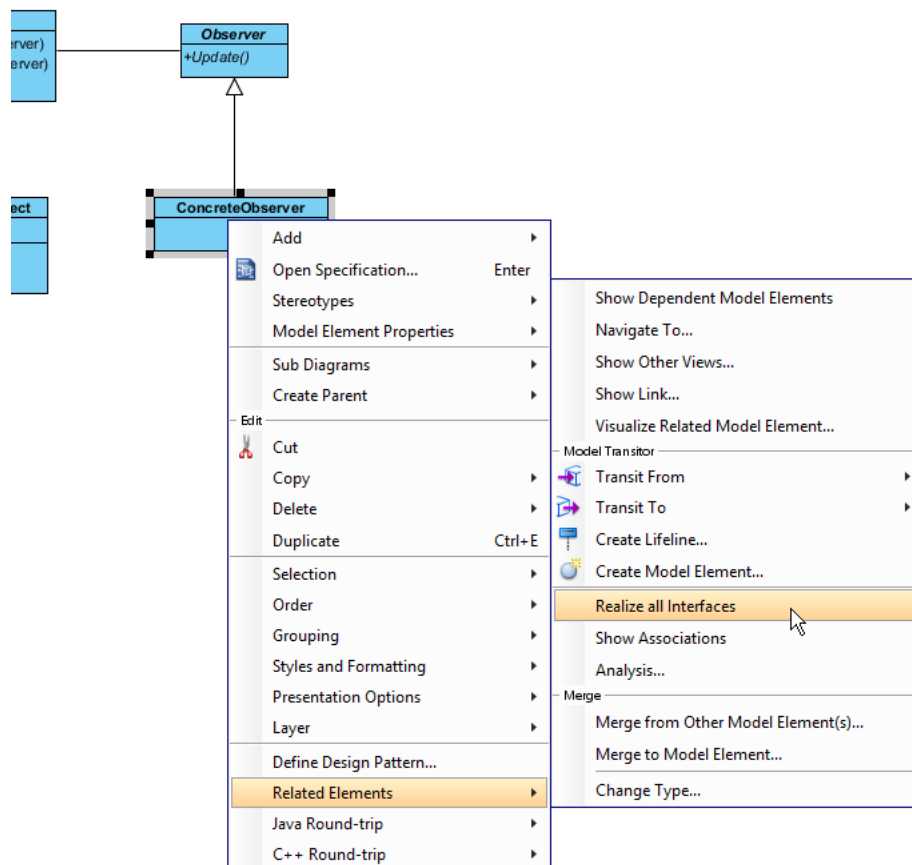
14. Right click on *Update()*, and select **Model Element Properties** > **Abstract** to set it as abstract.



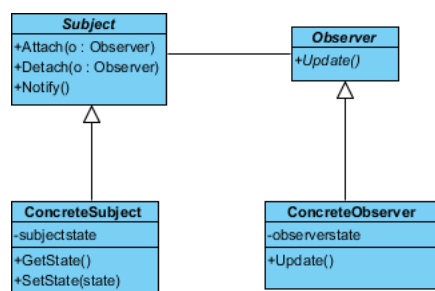
15. Move the mouse cursor over the *Observer* class, and drag out **Generalization** > **Class** to create a subclass *ConcreteObserver*.



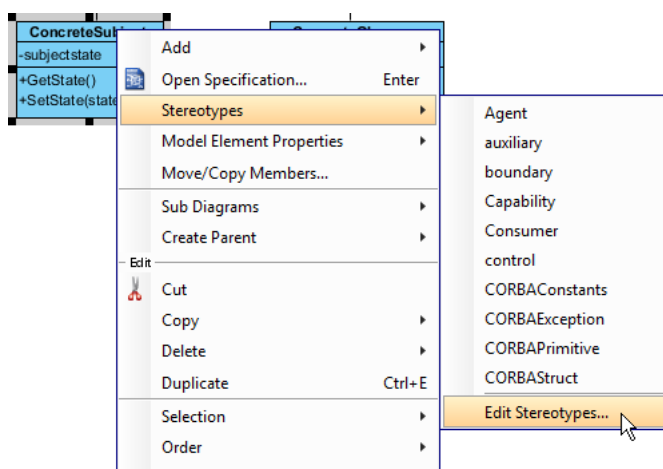
16. *ConcreteObserver* will inherit the operations from *Observer*. Right click on *ConcreteObserver* and select **Related Elements > Realize all Interfaces** from the popup menu.



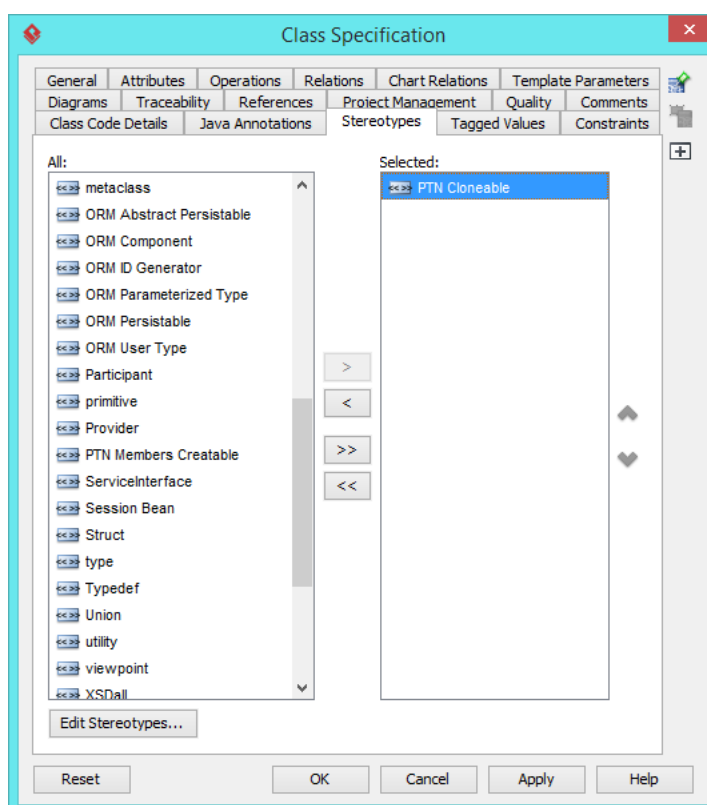
17. Right click on the *ConcreteObserver* class, and select **Add > Attribute** from the popup menu. Name the attribute *observerstate*. Up to now, the diagram should look like:



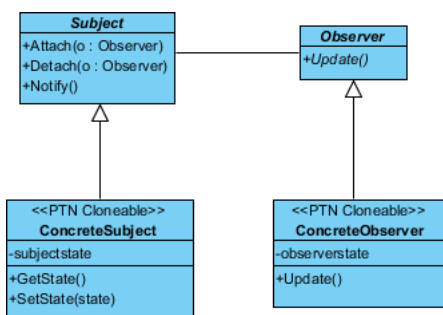
18. In practice, there may be multiple concrete subjects and observers. To represent this, stereotype the class *ConcreteSubject* and *ConcreteObserver* as **PTN Cloneable**. Right click on *ConcreteSubject* and select **Stereotypes > Stereotypes...** from the popup menu.



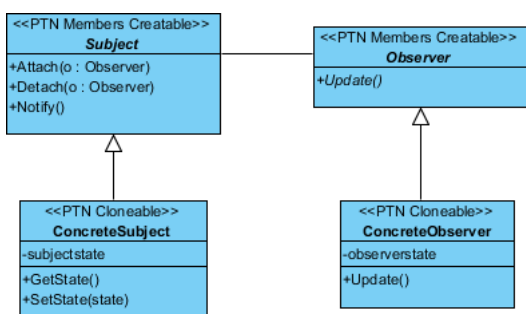
19. In the **Stereotypes** tab of the **Class Specification** dialog box, select **PTN Cloneable** and click > to assign it to *ConcreteSubject* class. Click **OK** to confirm.



20. Repeat steps 18 and 19 on *ConcreteObserver*.

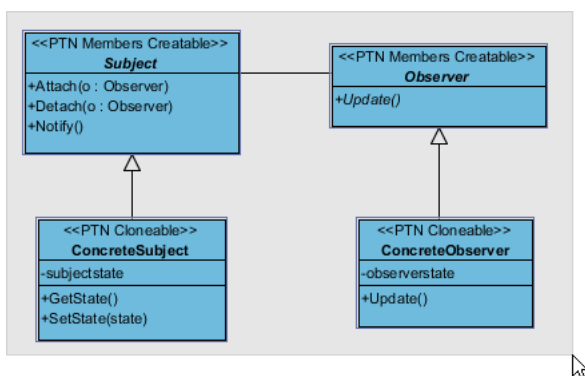


21. In practice, there are domain specific operations in subject and observer. To represent this, stereotype the class *Subject* and *Observer* as **PTN Members Creatable**. Repeat steps 18 and 19 to stereotype *Subject* and *Observer* as **PTN Members Creatable**.

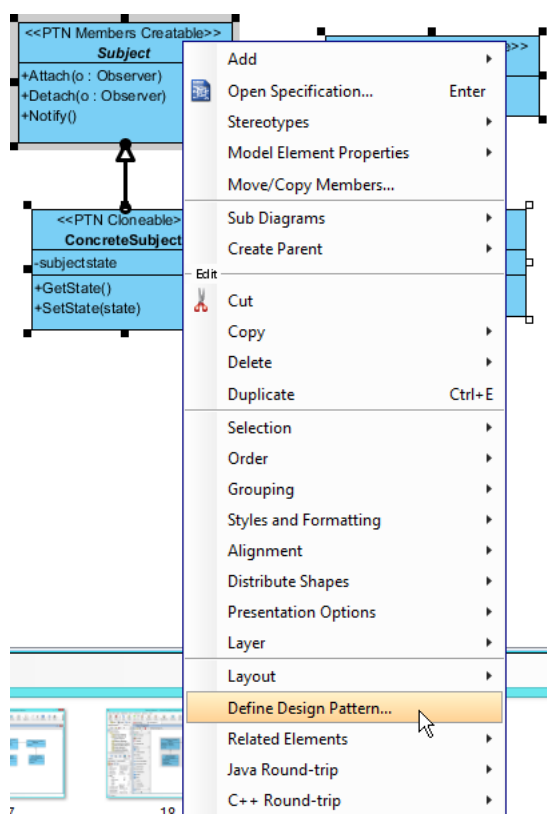


Defining Pattern

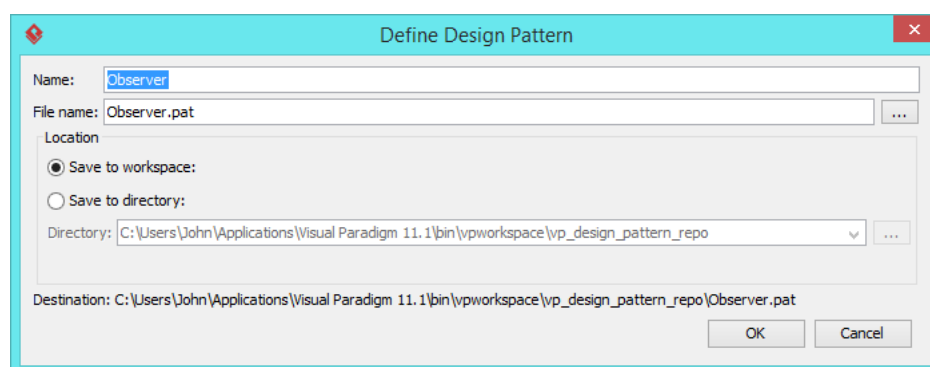
1. Select all classes on the class diagram.



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



- In the **Define Design Pattern** dialog box, specify the pattern name *Observer*. Keep the file name as is. Click **OK** to proceed.

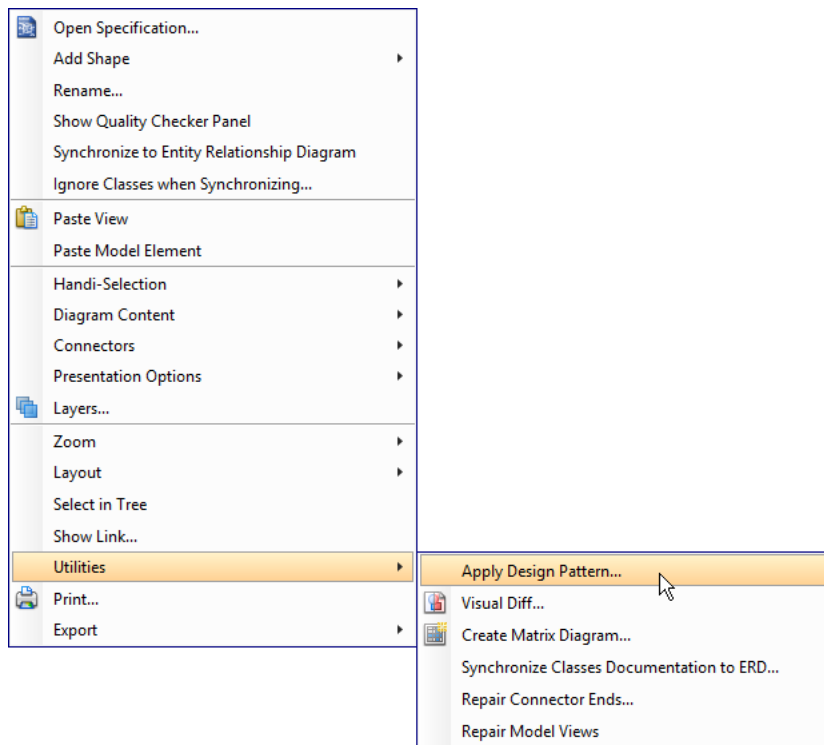


Applying Design Pattern on Class Diagram

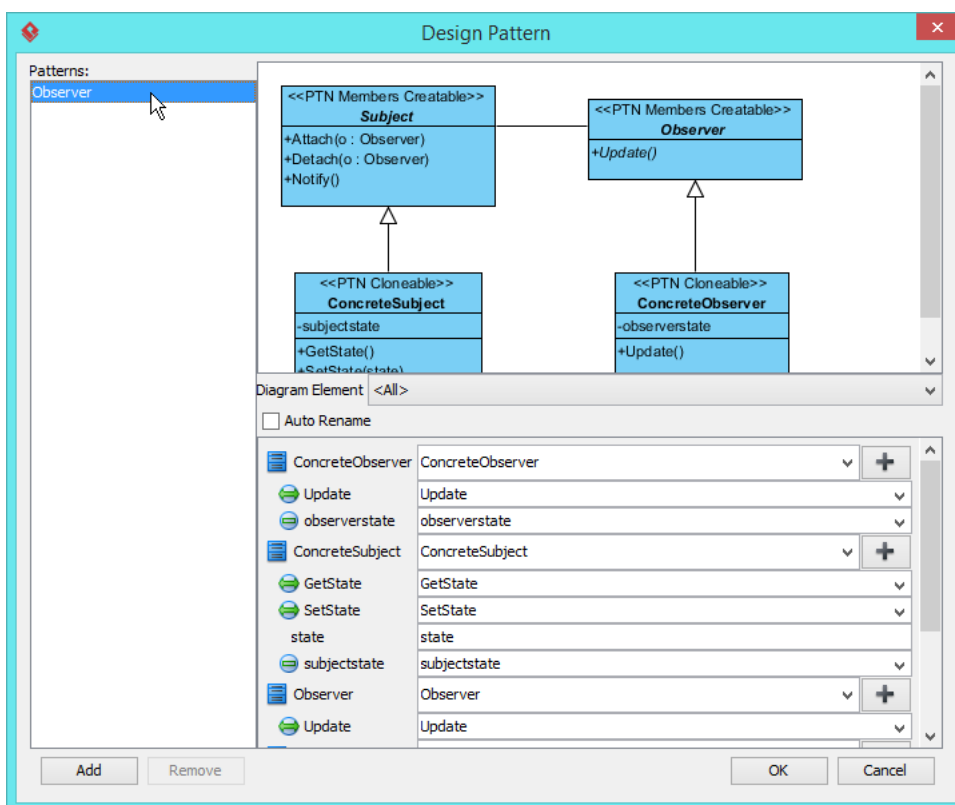
In this section, we are going to apply the observer pattern to model a diagram editor for observing changes of model, and calling various panes like the property and overview panes to update their content.

- Create a new project *Diagram Editor*.
- Create a class diagram *Domain Model*.

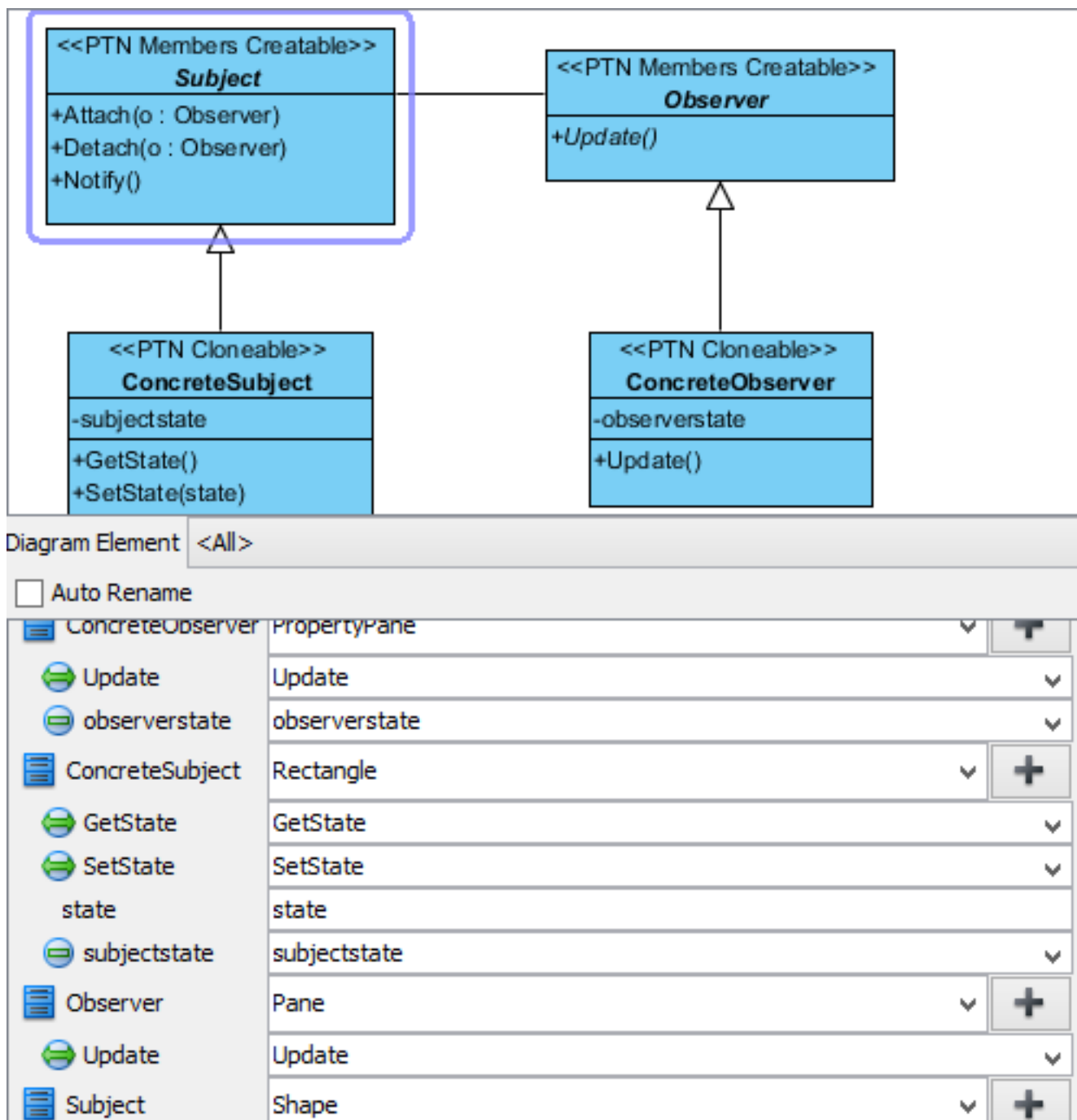
3. Right click on the class diagram and select **Utilities > Apply Design Pattern...** from the popup menu.



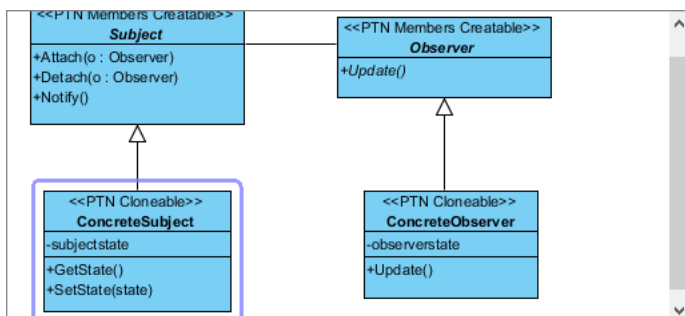
4. In the **Design Pattern** dialog box, select *Observer* from the list of patterns.



- At the bottom pane, rename classes *Subject*, *Observer*, *ConcreteSubject* and *ConcreteObserver* to *Shape*, *Pane*, *Rectangle* and *PropertyPane* respectively.



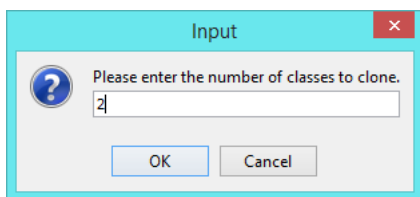
- Besides rectangle, there are more types of shape like circle and triangle. Select *ConcreteSubject* from the overview pane.



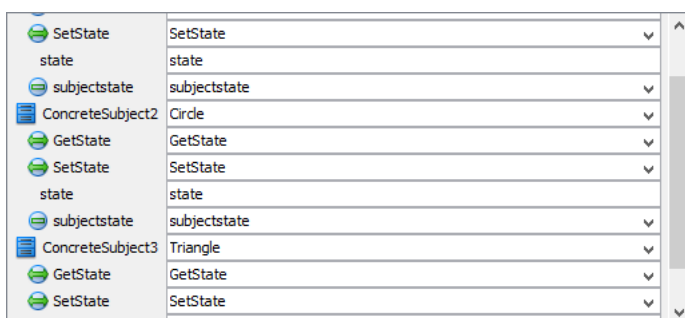
- Click on the + button next to the class name and select **Clone...** from the popup menu.



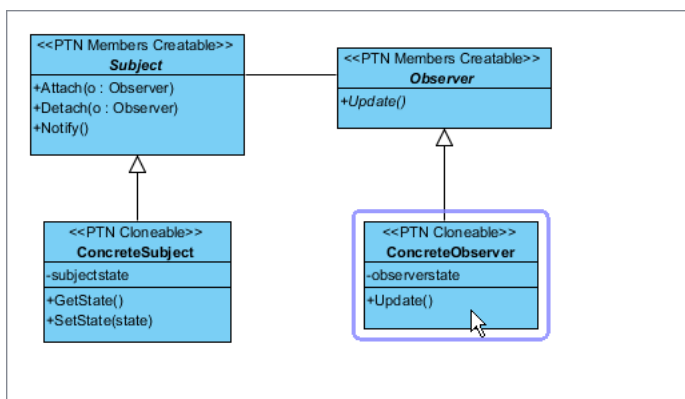
- Enter 2, which is the number of classes to clone, and click **OK** to confirm.



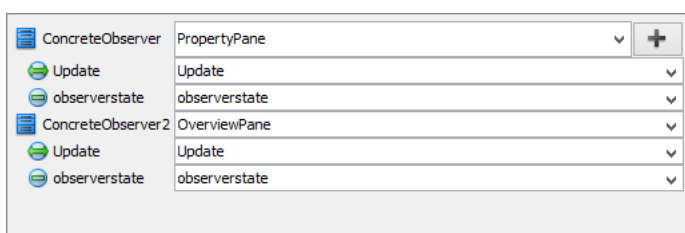
- At the bottom pane, rename *ConcreteSubject2* and *ConcreteSubject3* to *Circle* and *Triangle*.



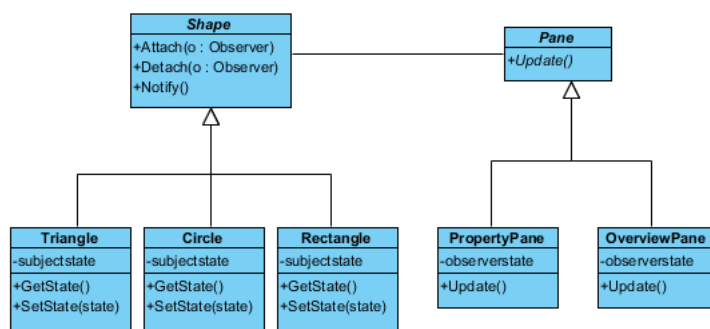
- For observers, there are also panes like overview pane. Select *ConcreteObserver* from the overview pane.



- Click on the + button next to the class name and select **Clone...** from the popup menu.
- Enter 1, which is the number of classes to clone, and click **OK** to confirm.
- At the bottom pane, rename *ConcreteObserver2* to *OverviewPane*.



- Click **OK** to confirm. Here is the diagram formed:



Resources

- [Design Patterns.vpp](#)
- [Observer.pat](#)

Related Links

- [Full set of UML tools and UML diagrams](#)



[Visual Paradigm home page](https://www.visual-paradigm.com/)
(<https://www.visual-paradigm.com/>)

[Visual Paradigm tutorials](https://www.visual-paradigm.com/tutorials/)
(<https://www.visual-paradigm.com/tutorials/>)