Visual ◆ Paradigm

Template Pattern Tutorial

Written Date : October 27, 2009

This tutorial is aimed to guide the definition and application of Gang of Four (GoF) template design pattern. By reading this tutorial, you will know how to develop a model for the template pattern, and how to apply it in practice.
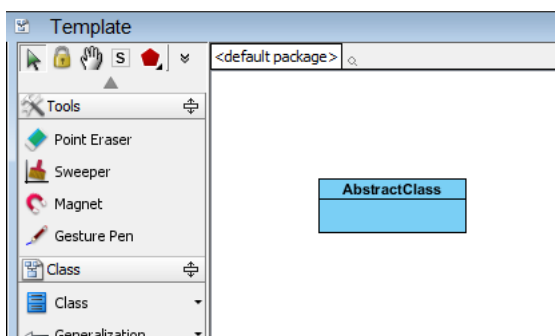
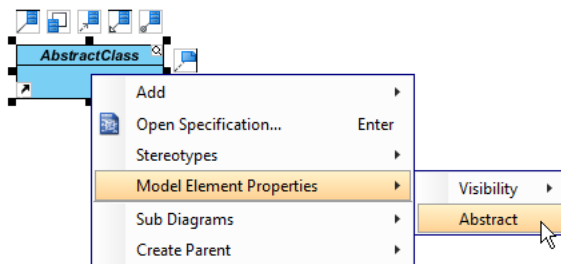## Modeling Design Pattern with Class Diagram

1.   Create a new project *Design Patterns*.

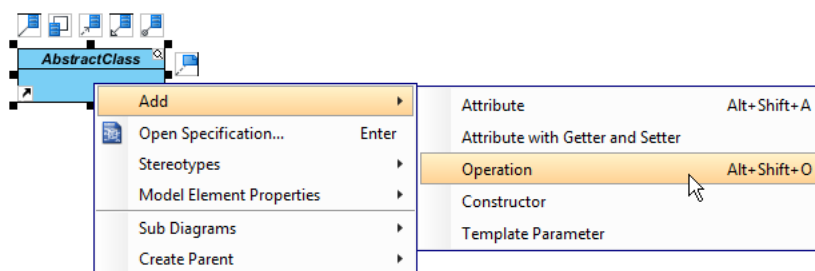2.   Create a class diagram *Template*.
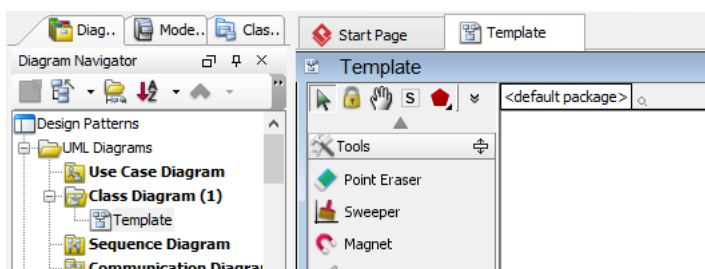
3.   Select **Class** from diagram toolbar. Click on the diagram to create a class. Name it as *AbstractClass*.
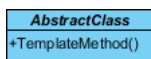
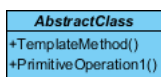4. Right-click on *AbstractClass*, and select **Model Element Properties** > **Abstract** to set it as abstract.

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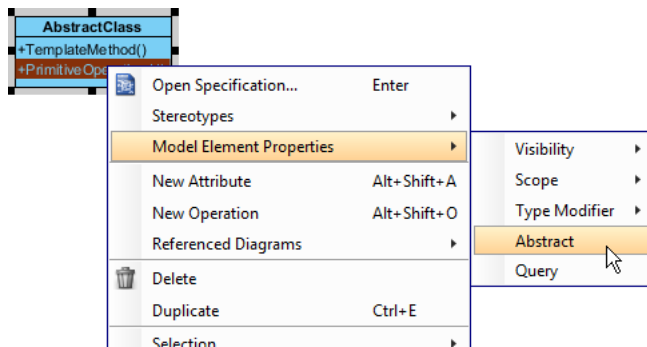

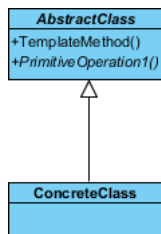6. Name the operation *TemplateMethod().*

7. Create another operation *PrimitiveOperation1().*

8. Right-click on *PrimitiveOperation1*, and select **Model Element Properties** > **Abstract** to set it as abstract.

9. Move the mouse cursor over the *AbstractClass* class, and drag out **Generalization** > **Class** to create subclasses *ConcreteClass*.

10. We need to make the concrete classes inherit operations from the abstract class. Right-click on *ConcreteClass* and select **Related Elements** > **Realize all Interfaces** from the popup menu.
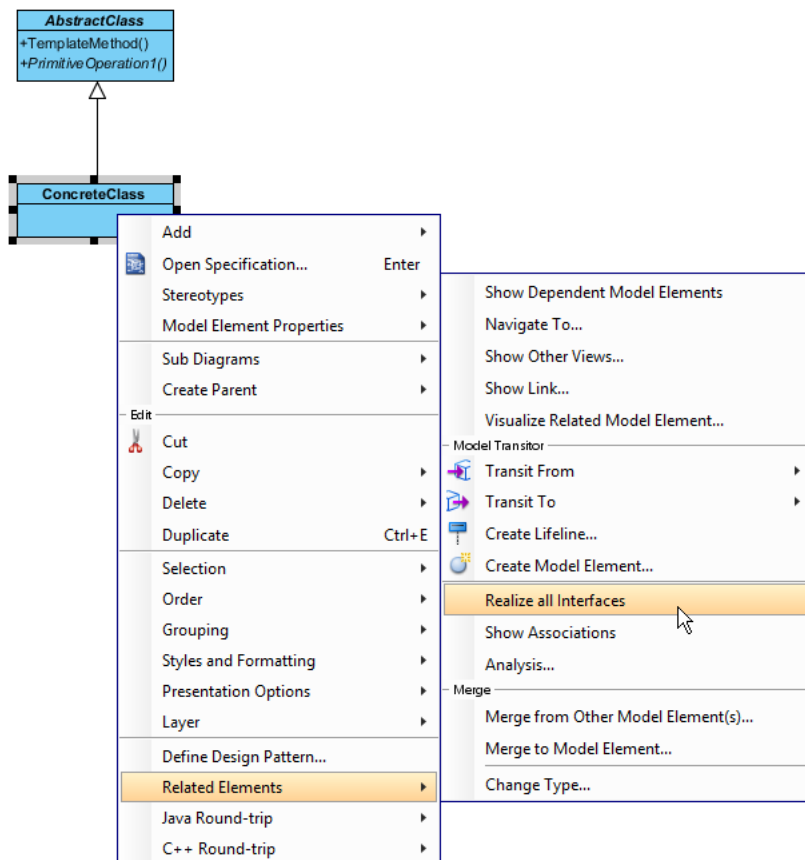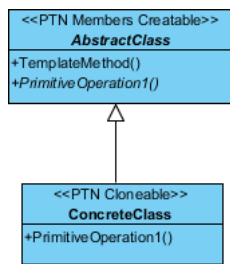
11. In practice, there may be multiple concrete strategies. To represent this, stereotype the class *ConcreteClass*, as **PTN Cloneable**. Right-click on *ConcreteClass* and select **Stereotypes** > **Stereotypes...** from the popup menu.
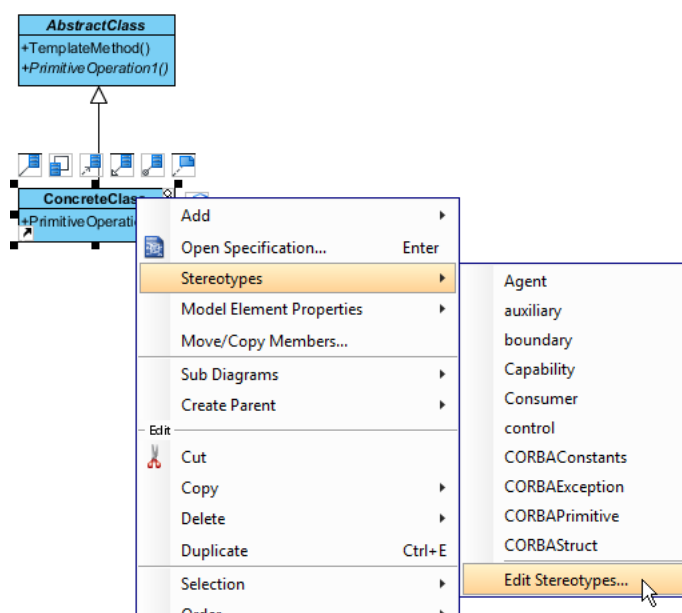
12. In the **Stereotypes** tab of the **Class Specification** dialog box, select **PTN Cloneable** and click > to assign it to *ConcreteClass* class. Click **OK** to confirm.
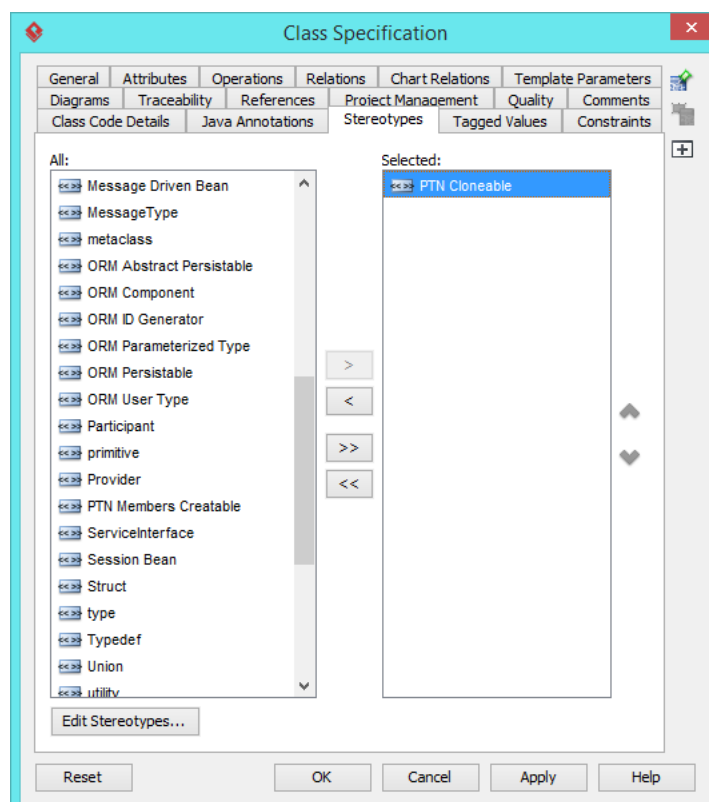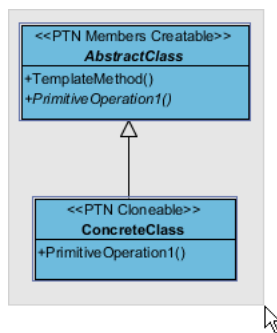
13. There may be multiple primitive operations. To represent this, stereotype the class *AbstractClass* as **PTN Members Creatable**. Repeat steps 11 and 12 to stereotype *AbstractClass* as **PTN Members Creatable**.
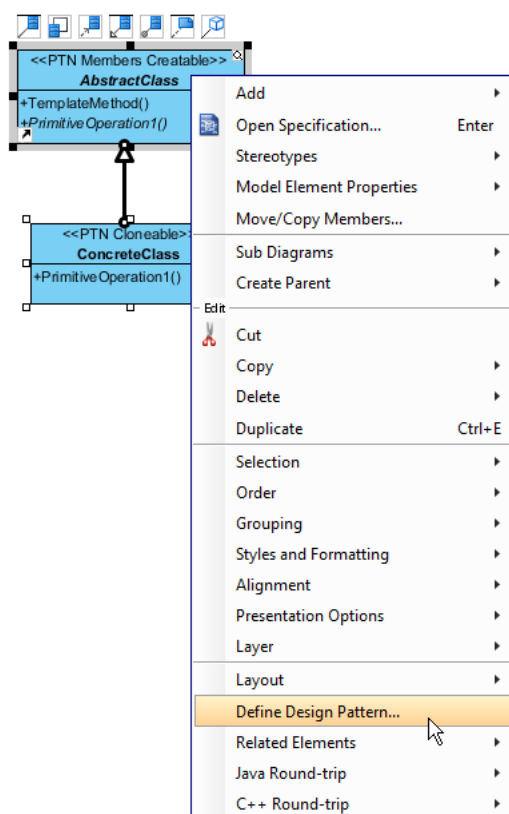
```
    <<PTN Members Creatable>>
         AbstractClass
    +TemplateMethod()
    +PrimitiveOperation1()

              △
              │

      <<PTN Cloneable>>
         ConcreteClass
    +PrimitiveOperation1()
```

Defining Pattern

1. Select all classes on the class diagram.

```
    <<PTN Members Creatable>>
         AbstractClass
    +TemplateMethod()
    +PrimitiveOperation1()

              △
              │

      <<PTN Cloneable>>
         ConcreteClass
    +PrimitiveOperation1()
```

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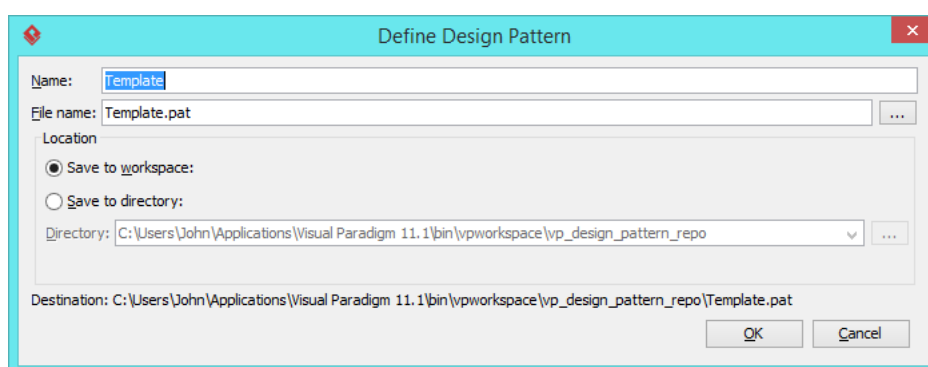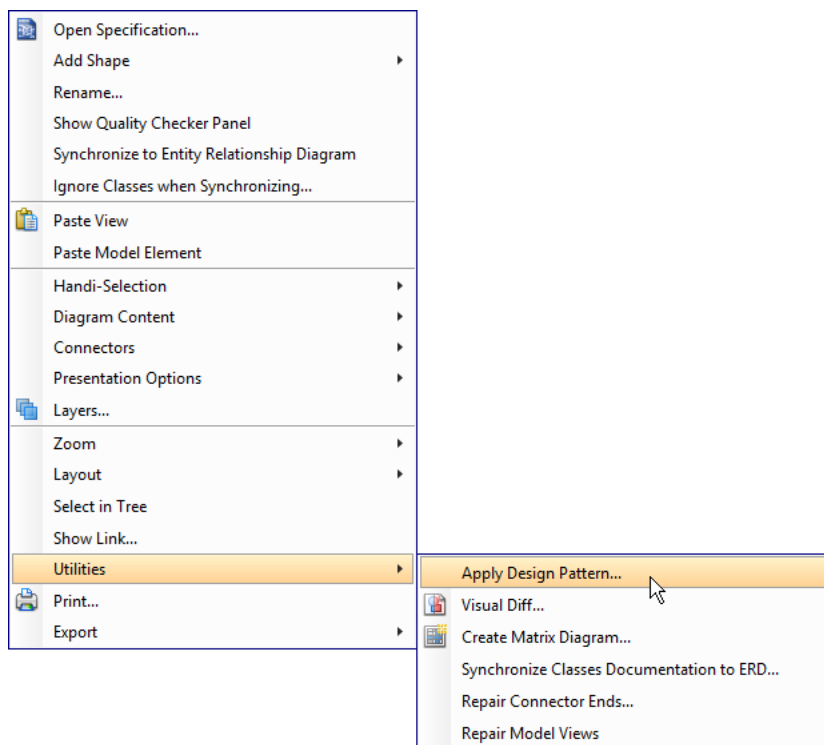

Applying Design Pattern on Class Diagram

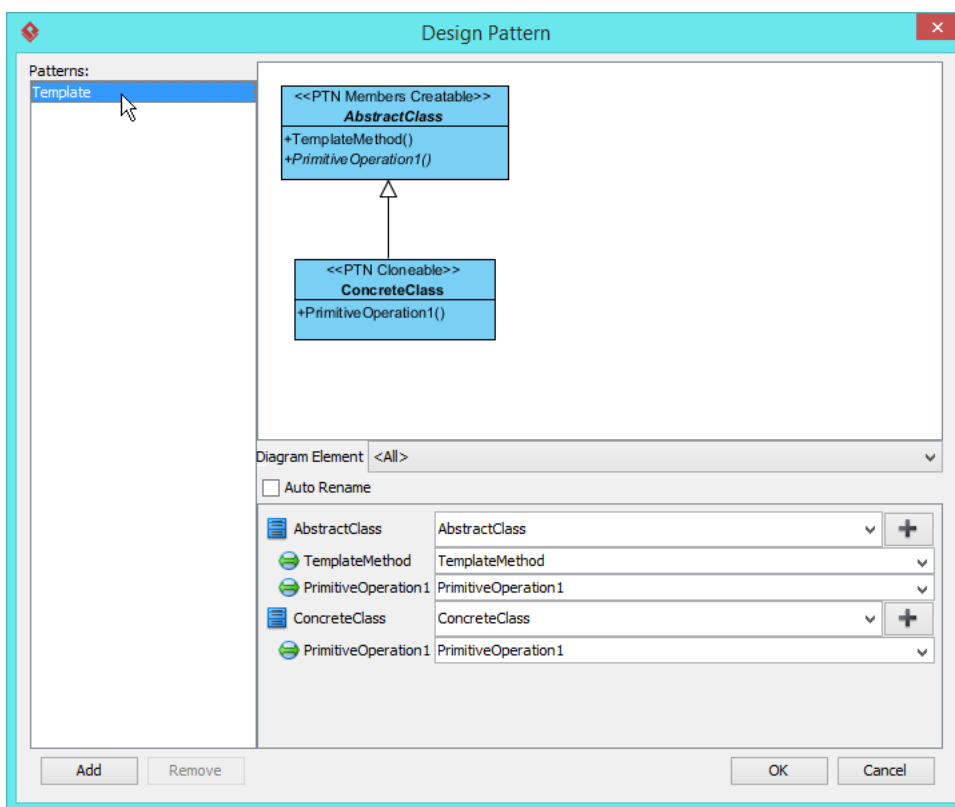In this section, we are going to apply the template pattern in modeling a diagram editor.

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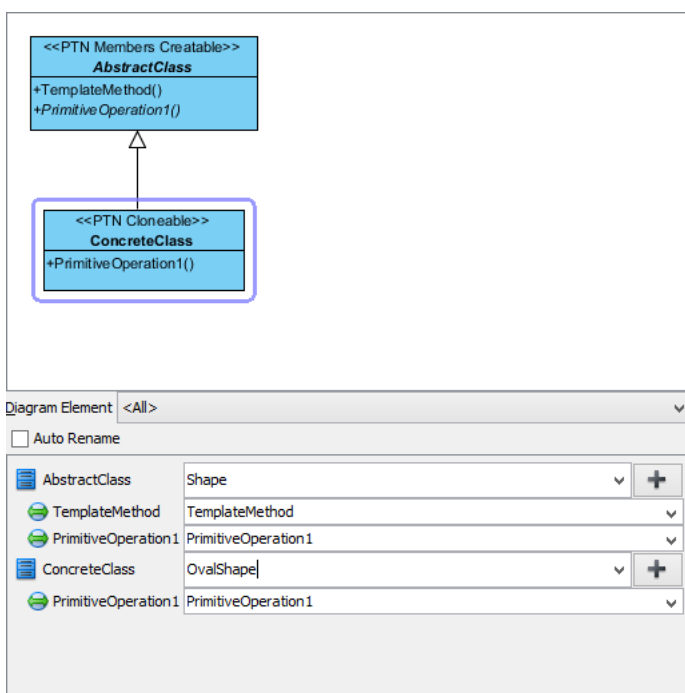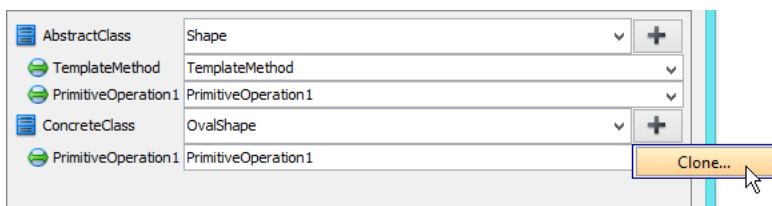
4.   In the **Design Pattern** dialog box, select *Template* from the list of patterns.
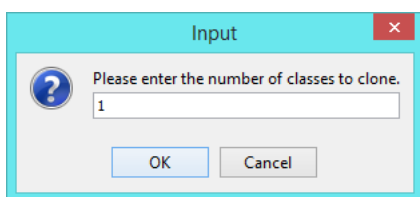

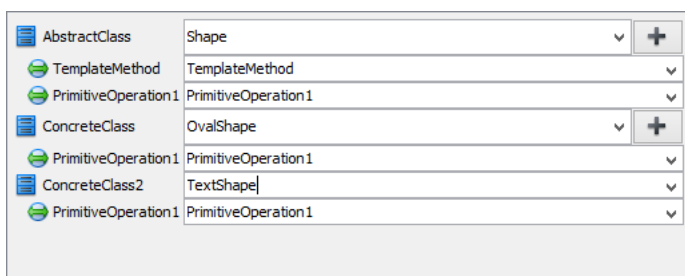5.   At the bottom pane, rename *AbstractClass* and *ConcreteClass* to *Shape* and *OvalShape*.

6. We need to have one more concrete class for text shape, click on the + button next to *ConcreteClass* and select **Clone...** from the popup menu.
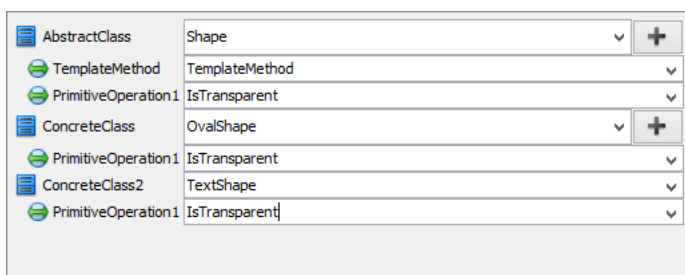
7. Enter 1 to be the number of classes to clone. Click **OK** to confirm.

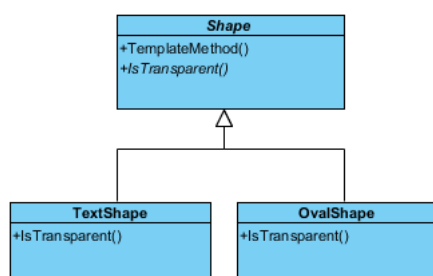8. Rename *ConcreteClass2* to *TextShape*.

9. Rename *TemplateMethod* and *PrimitiveOperation1* to *Render* and *IsTransparent* respectively.

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

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

Resources

1.   Design Patterns.vpp

2.   Template.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/)