

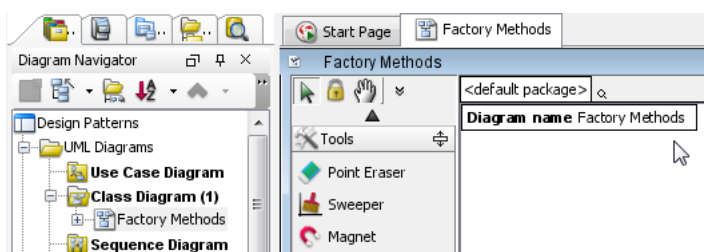


Factory Method Pattern Tutorial

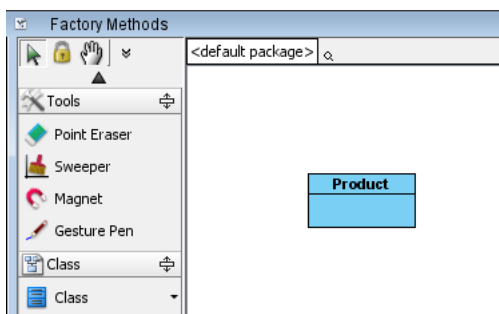
Written Date : September 28, 2009

Modeling a Design Pattern with a Class Diagram

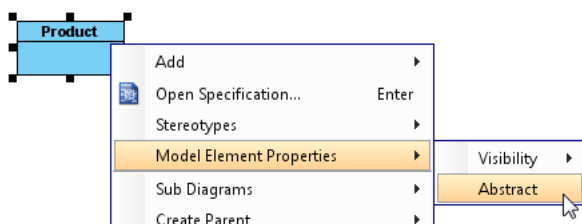
1. Create a new project named *Design Patterns*.
2. Create a class diagram named *Factory Method*.



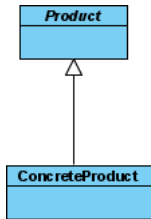
3. Select **Class** from the diagram toolbar. Click on the diagram to create a class. Name it *Product*.



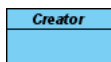
4. Set the *Product* class as abstract by right-clicking on it and selecting **Model Element Properties > Abstract** from the popup menu.



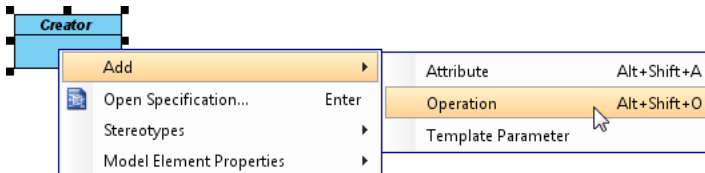
5. Move the mouse cursor over the *Product* class and drag out **Generalization > Class** to create a subclass named *ConcreteProduct*.



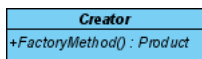
6. Create a class named *Creator* and set it as abstract.



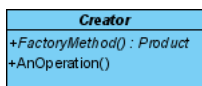
7. Right-click on the *Creator* class and select **Add > Operation** from the popup menu.



8. Name the operation *FactoryMethod()* and make it return *Product*.
9. Right-click on *FactoryMethod()* and select **Model Element Properties > Abstract** to set it as abstract.

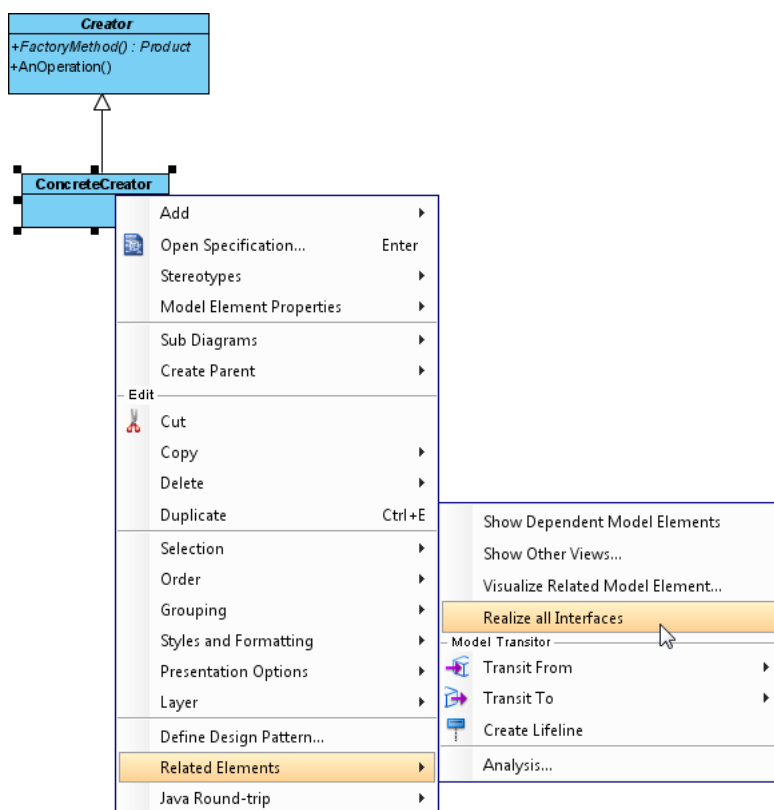


10. Add a non-abstract operation *AnOperation()* to *Creator*.

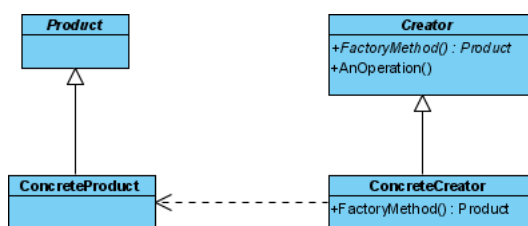


11. Move the mouse cursor over the *Creator* class and drag out **Generalization > Class** to create a subclass named *ConcreteCreator*.

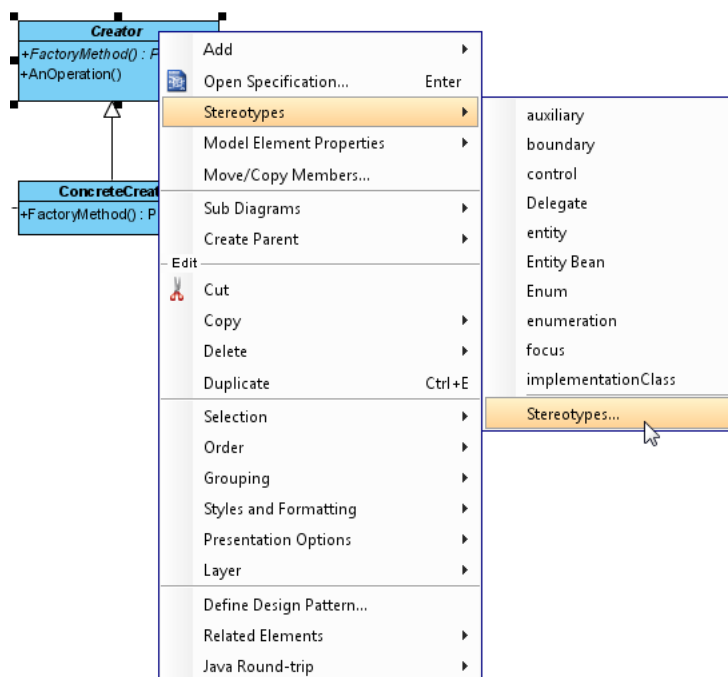
- Make *ConcreteCreator* inherit the abstract operations provided by *Creator* by right-clicking on *ConcreteCreator* and selecting **Related Elements > Realize all Interfaces** from the popup menu.



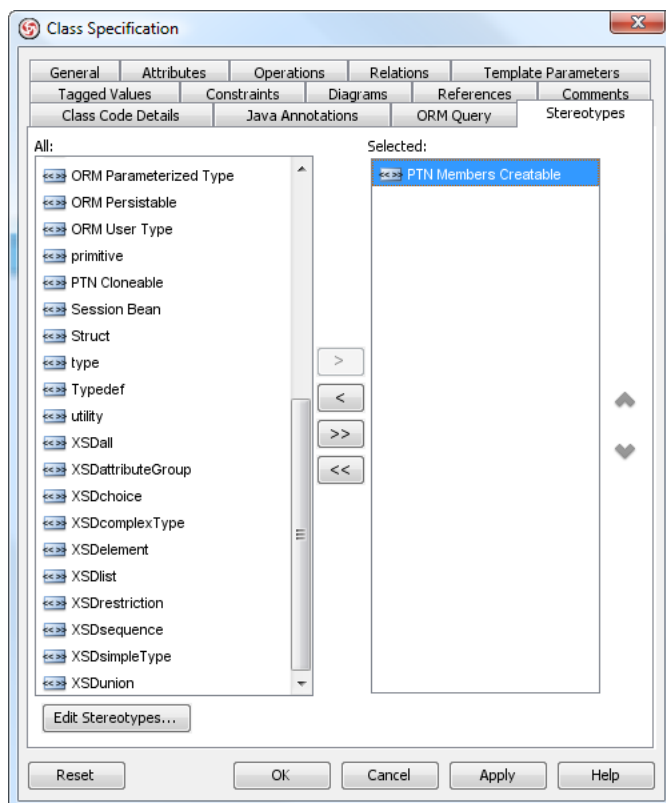
- In practice, the *FactoryMethod* in *ConcreteCreator* is expected to return an instance of *ConcreteProduct*. Therefore, add a dependency between *ConcreteCreator* and *ConcreteProduct*. Move the mouse cursor over the *ConcreteCreator* class and drag out **Dependency > Class** to *ConcreteProduct*. Up to now, the diagram should look like this:



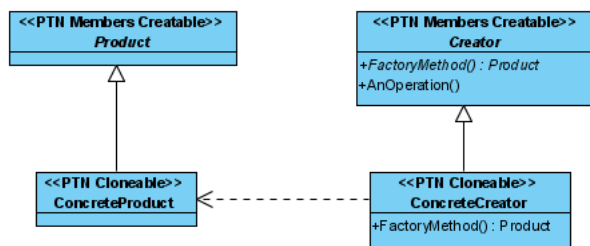
14. There may be more than one operation in the *Creator* class. To represent this, stereotype the *Creator* class as **PTN Members Creatable**. Right-click on the **Creator** class and select **Stereotypes > Stereotype** from the popup menu.



- In the **Stereotypes** tab of the class specification, select **PTN Members Creatable** and click > to assign it to the class. Click **OK** to confirm.

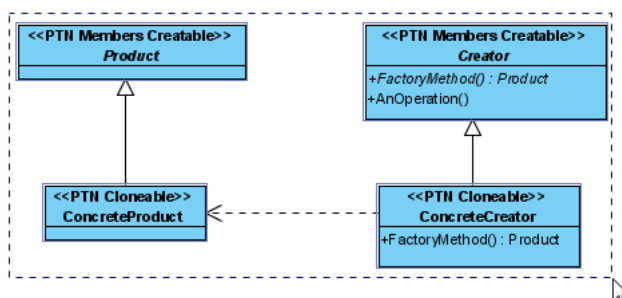


- The *Product* class should also have its own operations. Repeat steps 14 and 15 to stereotype it as **PTN Members Creatable**.
- There may be multiple concrete products and creators. Let's repeat steps 14 and 15 to stereotype *ConcreteProduct* and *ConcreteCreator* as **PTN Cloneable**. The diagram should look like this:

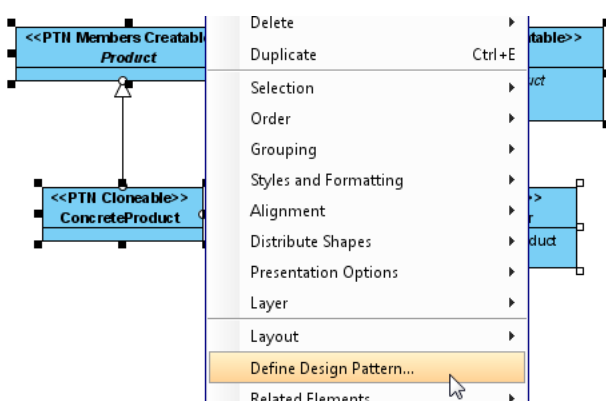


Defining a Pattern

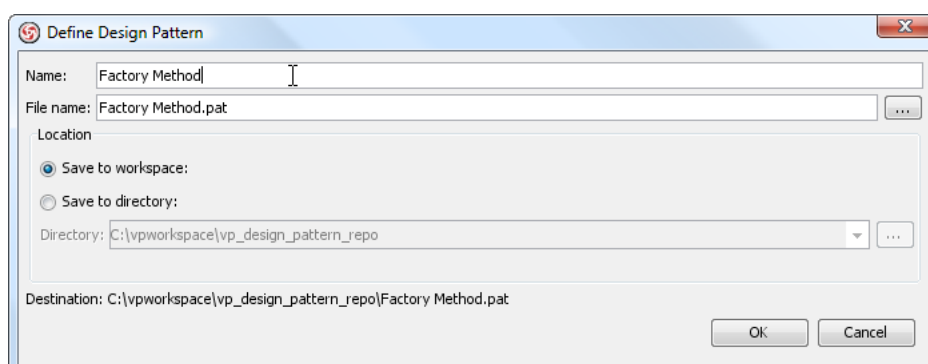
1. Select all classes on the class diagram.



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

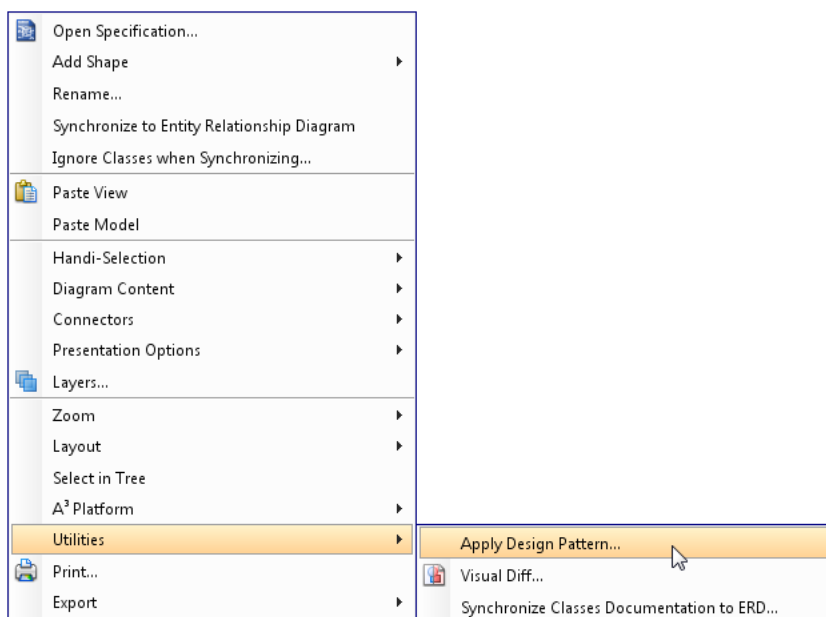


Applying a Design Pattern to a Class Diagram

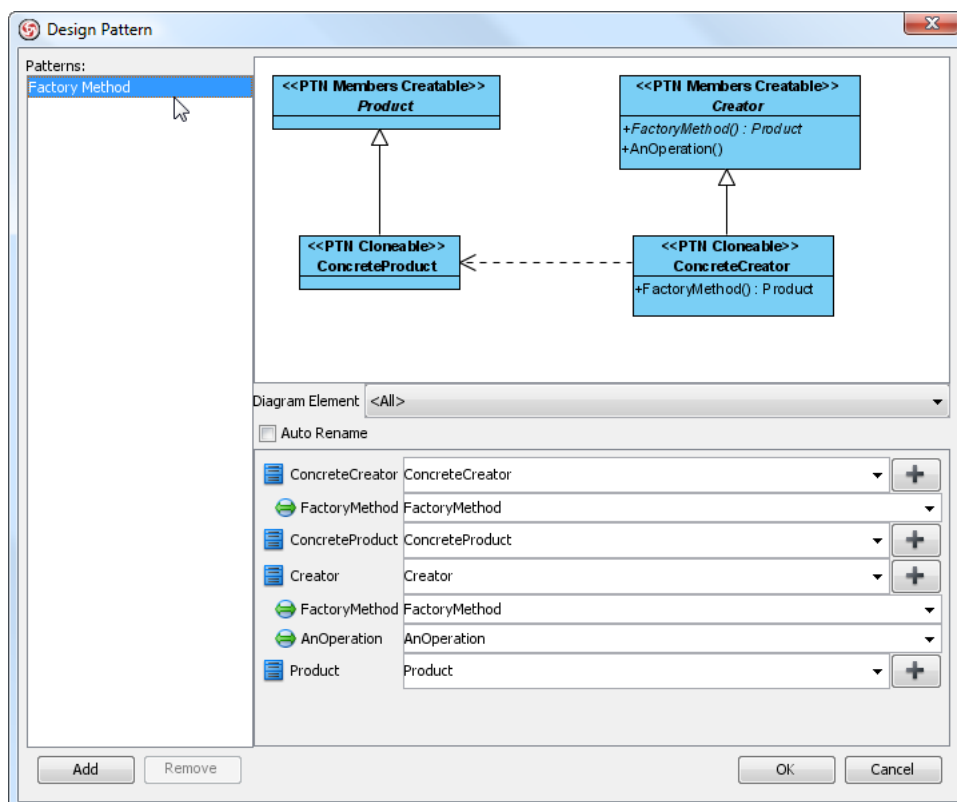
In this section, we will try to use the factory method pattern to model a part of a text editor.

1. Create a new project named *Text Editor*.
2. Create a class diagram named *Domain Model*.

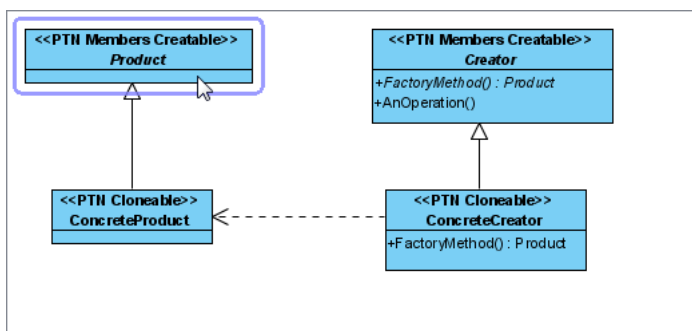
3. Right-click on the class diagram and select **Utilities > Apply Design Pattern...** from the popup menu.



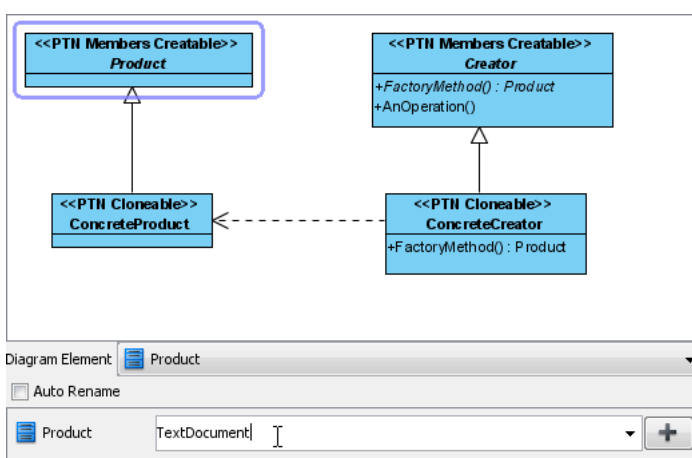
4. In the **Design Pattern** dialog box, select *Factory Method* from the list of patterns.



- Click on **Product** in the overview.



- Rename it to *TextDocument* in the bottom pane.



- Click on +, and select **New Operation...** from the popup menu. We will create the operations available in the *TextDocument* class.
- In the **Operation Specification** dialog box, enter *Open* as the operation name.
- Repeat steps 7 and 8 to create operations *close*, *save*, and *reopen*.
- Click on *ConcreteProduct* in the overview.
- Rename *ConcreteProduct* to *PlainTextDocument*.

12. We need to process one more document type for RTF documents. Keep *ConcreteProduct* selected and click the + button, then select **Clone...** from the popup menu.

13. Enter *1* as the number of classes to clone. Click **OK** to confirm.

14. Enter *RTFDocument* as the class name.

15. Select the *Creator* class in the overview.
16. Rename *Creator* to *TextEditor*, the operation *FactoryMethod* to *createDoc*, and *AnOperation* to *newDoc*.

17. We need more operations. Click on +, then select **New Operation...** from the popup menu.

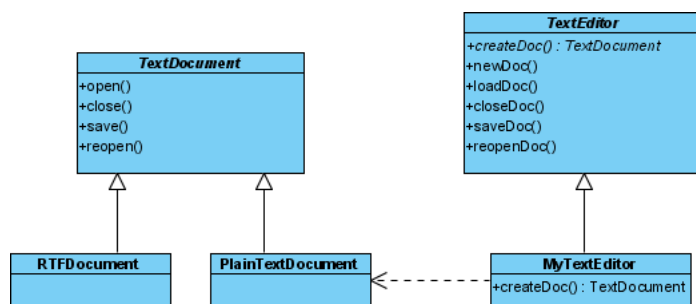
18. In the **Operation Specification** dialog box, enter *loadDoc* as the name.

19. Repeat the previous steps to create operations *closeDoc*, *saveDoc*, and *reopenDoc*.

20. Select *ConcreteCreator*. Rename *ConcreteCreator* to *MyTextEditor* and the operation *FactoryMethod* to *createDoc*.

21. Click **OK** to confirm editing and apply the pattern to the diagram.

22. Tidy up the diagram. It should look like this:



Resources

1. [Design Patterns.vpp](#)
2. [Factory Method.pat](#)

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/>)