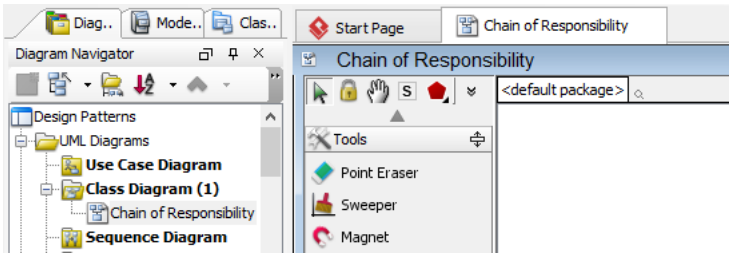
Chain of Responsibility Pattern Tutorial

Written Date : October 14, 2009

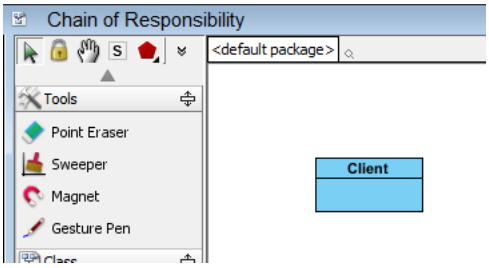
Modeling a Design Pattern with a Class Diagram

1. Create a new project named *Design Patterns*.

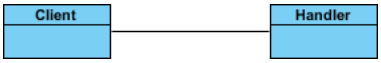
2. Create a class diagram named *Chain of Responsibility*.



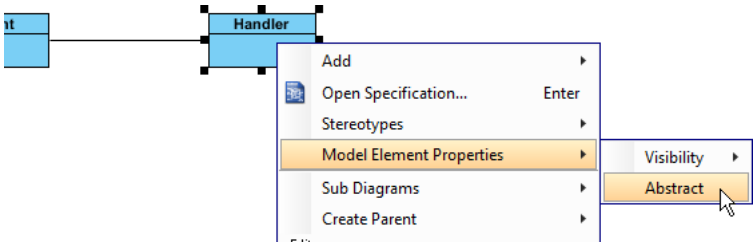
3. Select **Class** from the diagram toolbar. Click on the diagram to create a class and name it *Client*.



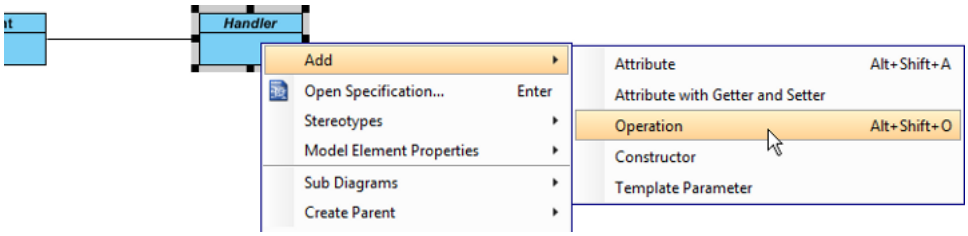
4. Move the mouse cursor over the *Client* class and drag out **Association > Class** to create an associated class named *Handler*.



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

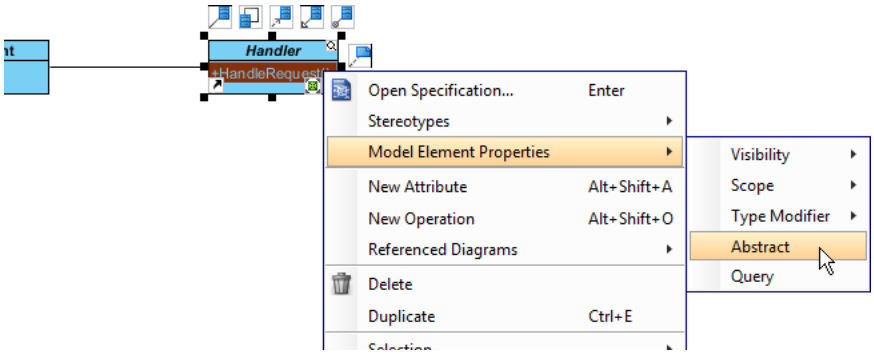


6. Right-click on the *Handler* class and select **Add > Operation** from the popup menu.

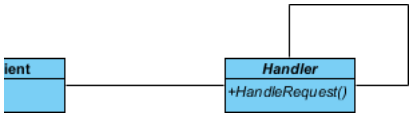


7. Name the operation *HandleRequest()*.

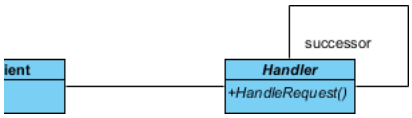
8. Right-click on *HandleRequest* and select **Model Element Properties > Abstract** to set it as abstract.



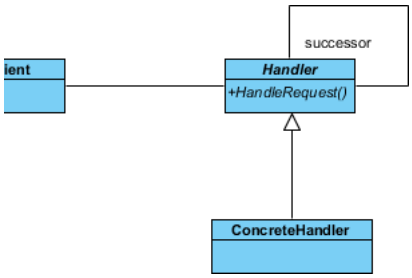
9. Move the mouse cursor over the *Handler* class and click on the **Self Association** resource icon to create a self-association.



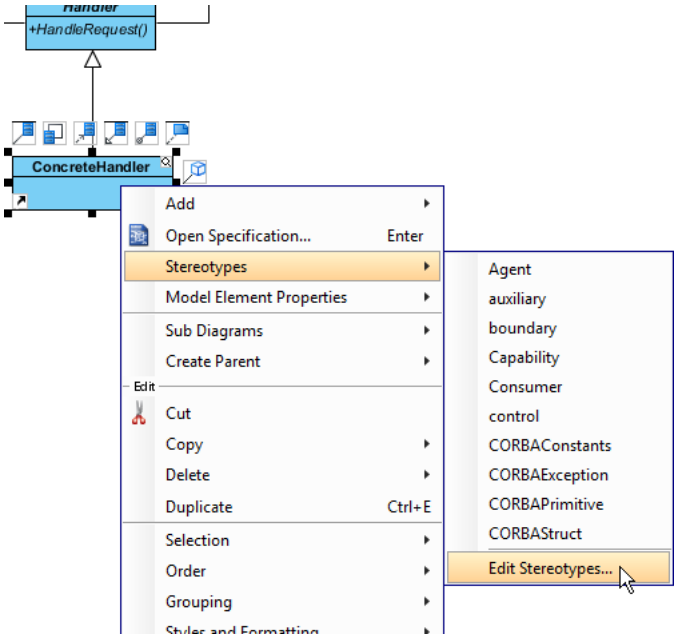
10. Name the association end "successor."



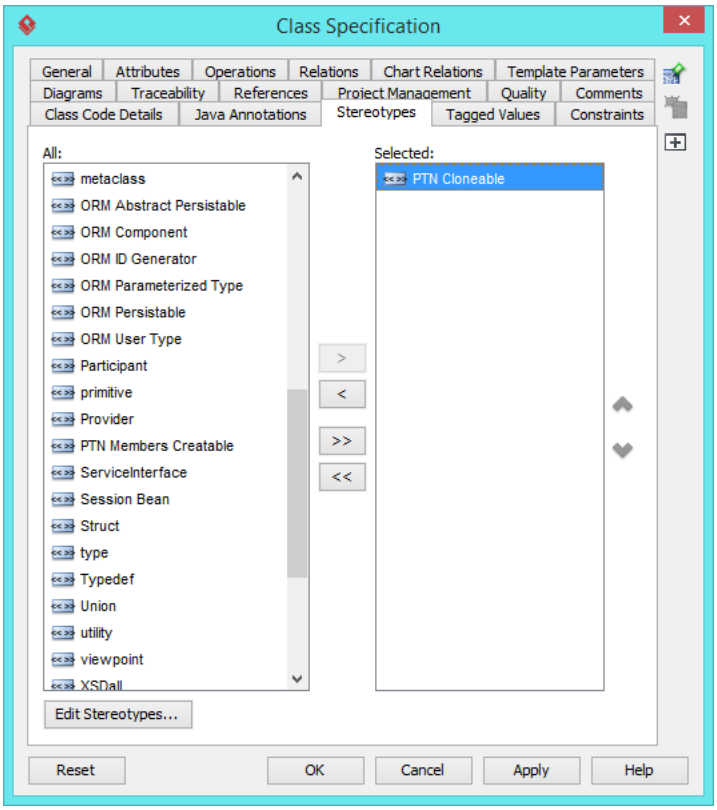
11. Move the mouse cursor over the *Handler* class and drag out **Generalization > Class** to create a subclass named *ConcreteHandler*.



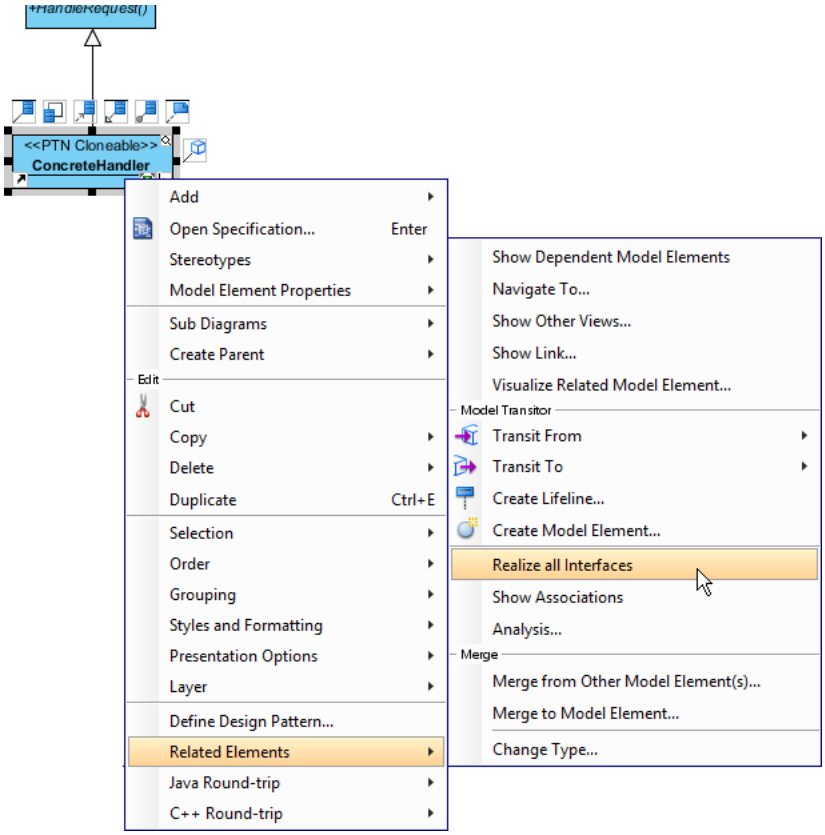
12. In practice, there may be multiple concrete handlers. To represent this, stereotype the *ConcreteHandler* class as **PTN Cloneable**. Right-click on *ConcreteHandler* and select **Stereotypes > Stereotypes...** from the popup menu.



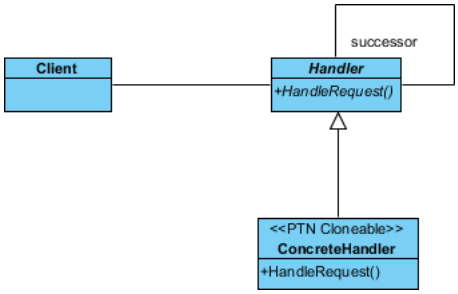
13. In the **Stereotypes** tab of the **Class Specification** dialog box, select **PTN Cloneable** and click > to assign it to the *ConcreteHandler* class. Click **OK** to confirm.



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

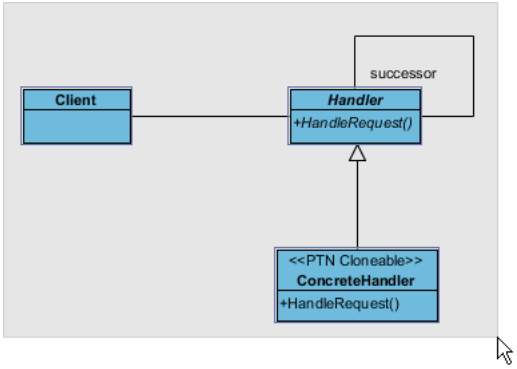


Up to now, the diagram should look like this:

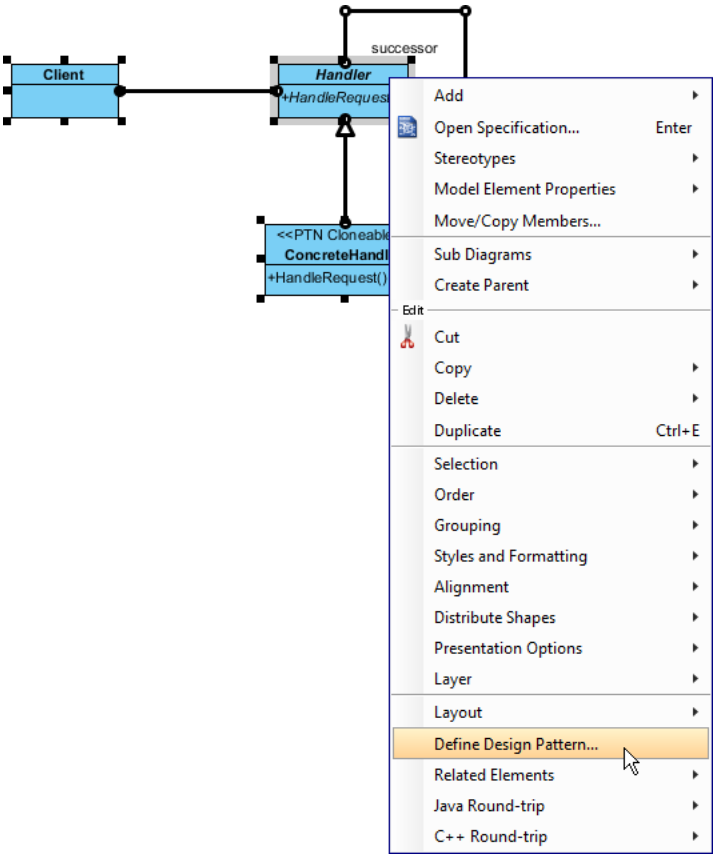


Defining a Pattern

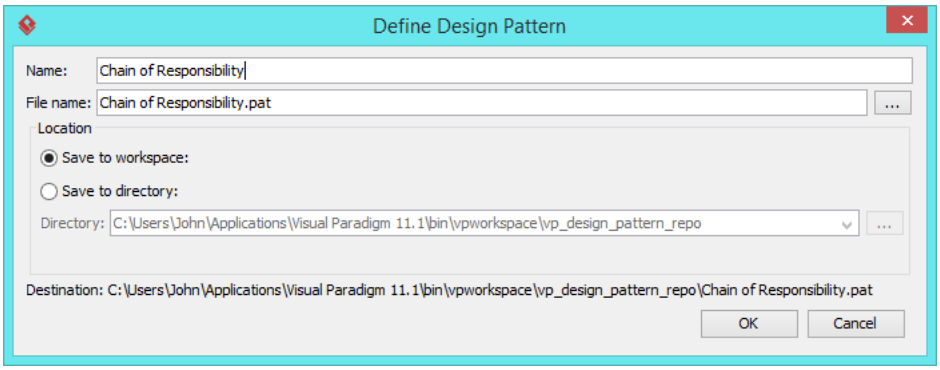
1. Select all classes on the class diagram.



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



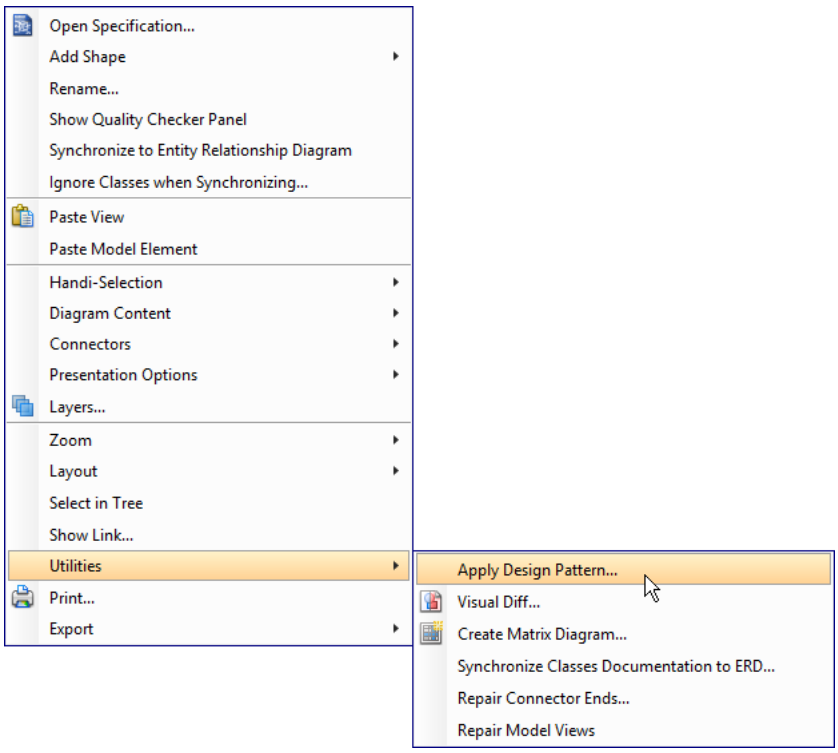
3. In the **Define Design Pattern** dialog box, specify the pattern name as *Chain of Responsibility*. Keep the file name as is and click **OK** to proceed.



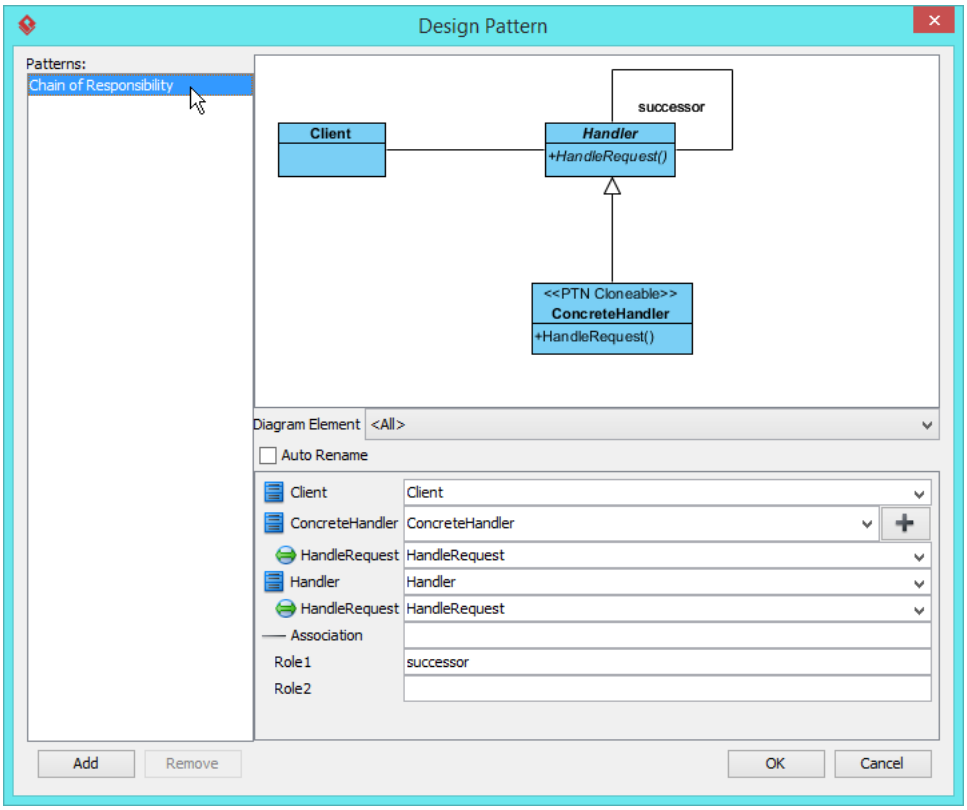
Applying a Design Pattern to a Class Diagram

In this section, we will apply the chain of responsibility pattern to model a coin dispenser.

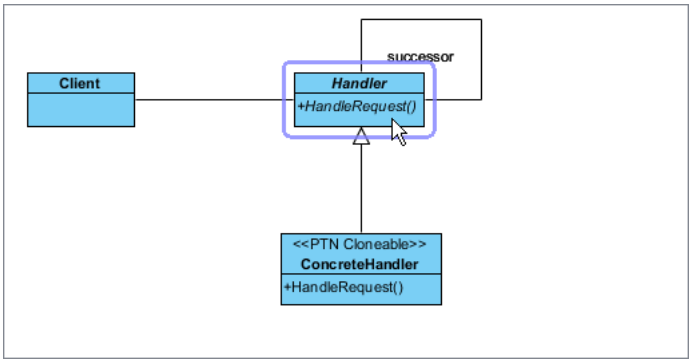
1. Create a new project named *Coin Dispenser*.
2. Create a class diagram named *Domain Model*.
3. Right-click on the class diagram and select **Utilities > Apply Design Pattern...** from the popup menu.



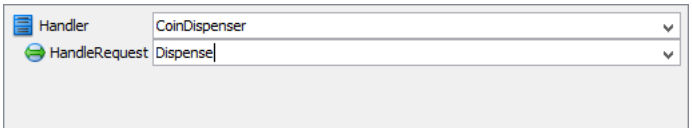
4. In the **Design Pattern** dialog box, select *Chain of Responsibility* from the list of patterns.



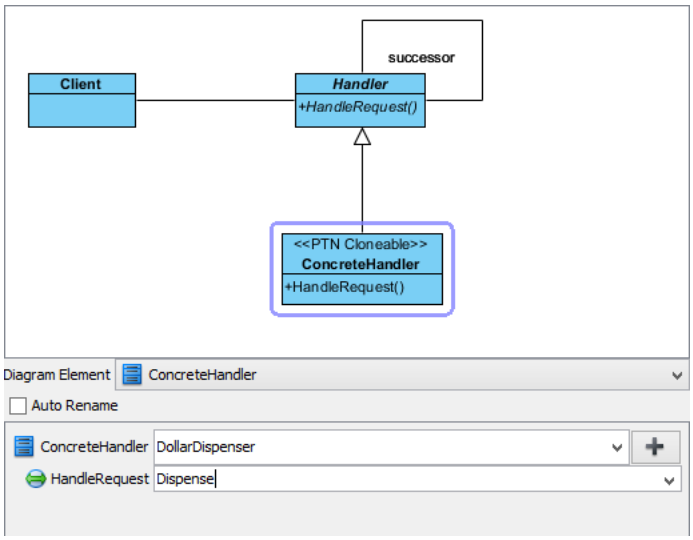
5. Click on *Handler* in the overview.



6. Rename *Handler* to *CoinDispenser* and the *HandleRequest* operation to *Dispense* in the bottom pane.

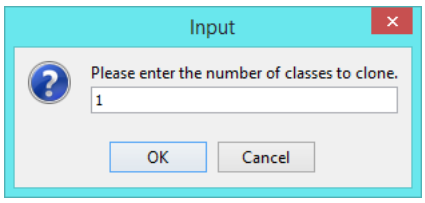


7. Click on *ConcreteHandler* in the overview and rename it to *DollarDispenser* and the *HandleRequest* operation to *Dispense*.



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

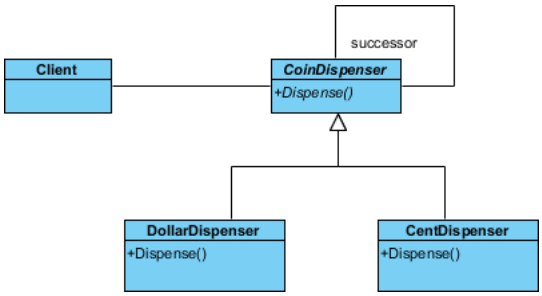
9. Enter *1* as the number of classes to clone and click **OK** to confirm.



10. Rename *ConcreteHandler2* to *CentDispenser* and the *HandleRequest* operation to *Dispense*.

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

12. Tidy up the diagram. The result should look like this:



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/)