1. The following is an example of the FQDN (fully qualified domain name) on the Internet: nwest.sales.DomainName.com It consists of different subdomains. Each such subdomain can be mapped onto a specific directory on the file system of the computer where the DomainName.com is hosted. Each such subdomain can have different HTML files, which can be accessed through a URL. Thus a subdomain and the set of HTML files can be viewed as two main components of a Web site.

a. Define a subdomain hierarchy for an example domain.

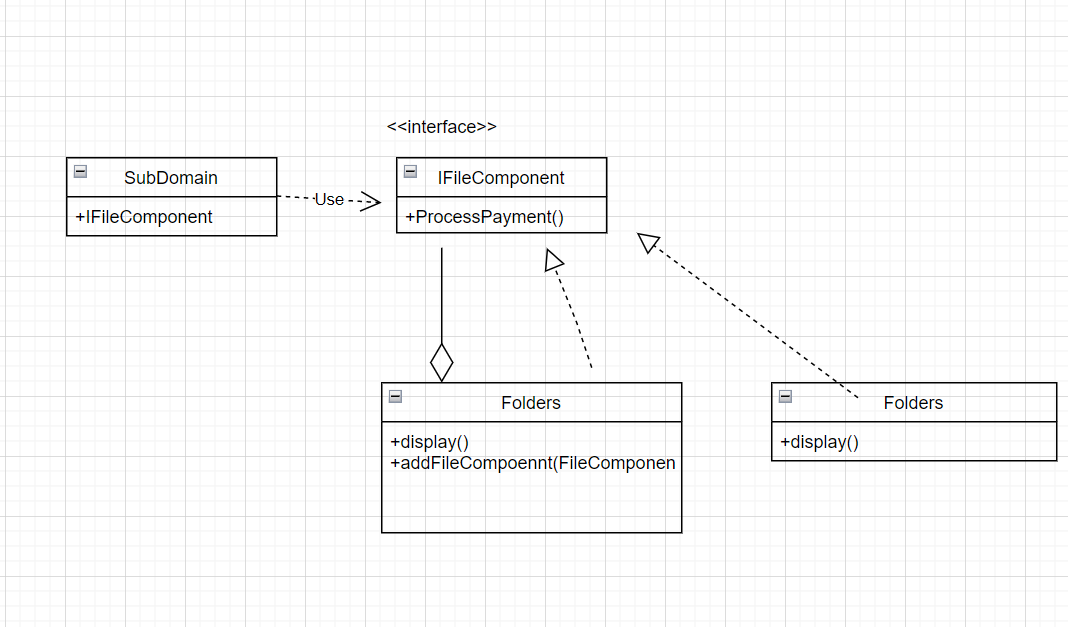
b. Create an application that uses the appropriate pattern to:

i. Display the directory a given subdomain is mapped onto

ii. Display the URLs of Web site components (either subdomains or single HTML files) in a uniform manner

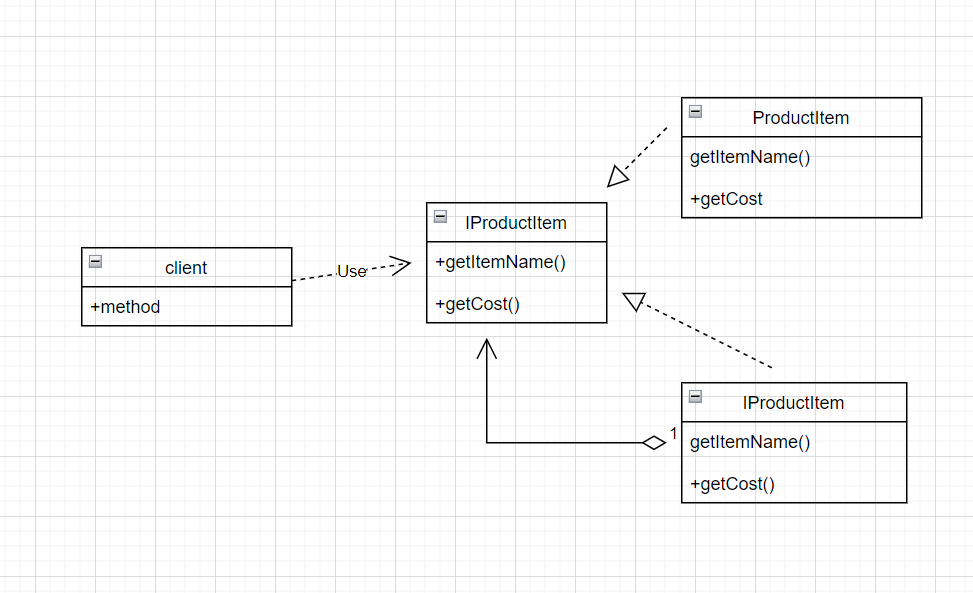
Solution :

Composite pattern :



1. A typical product database consists of two types of product components — product categories and product items. A product category is generally composite in nature. It can contain product items and also other product categories as its subcategories. Example Product Categories: a. Computers b. Desktops c. Laptops d. Peripherals e. Printers f. Cables the Computers product category contains both the Desktops and the Laptops product categories as its subcategories. The Desktop category can contain a product item such as Compaq Presario 5050. Product items are usually individual, in the sense that they do not contain any product component within. Design and implement an application to list the dollar value of a product component. Use the appropriate pattern to allow the client application to refer to both the product categories and the product items in a uniform manner

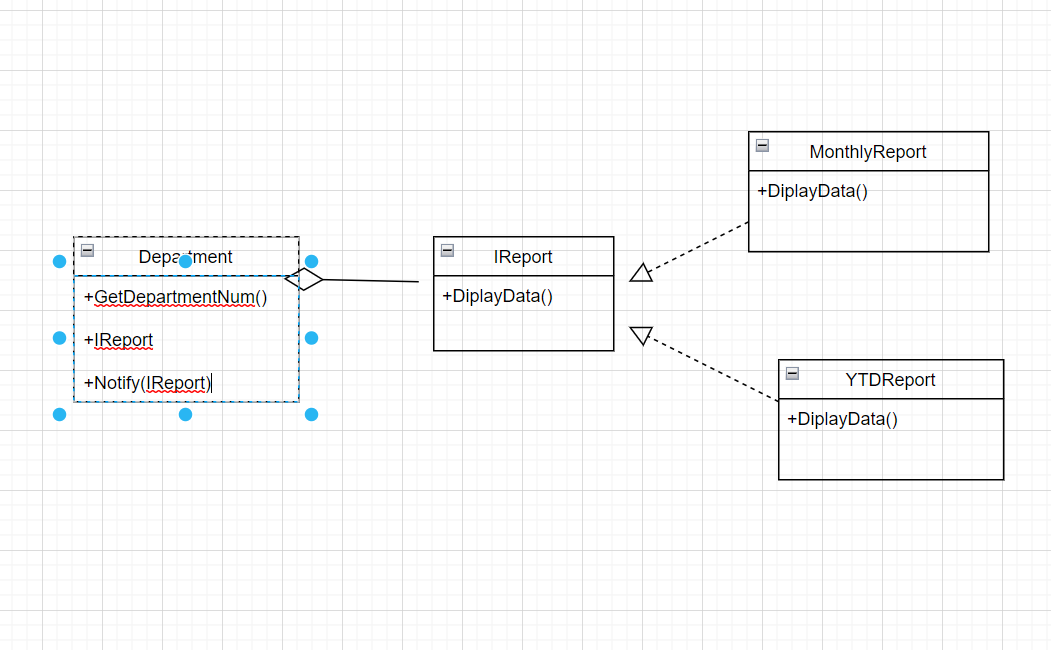
Sol : cpmposite pattern (product category is composite of product item)



1. Let us build a sales reporting application for the management of a store with multiple departments. The features of the application include:
   1. Users should be able to select a specific department they are interested in.
   2. Upon selecting a department, two types of reports are to be displayed:
      1. Monthly report — A list of all transactions for the current month for the selected department.
      2. YTD sales chart — A chart showing the year-to-date sales for the selected department by month.

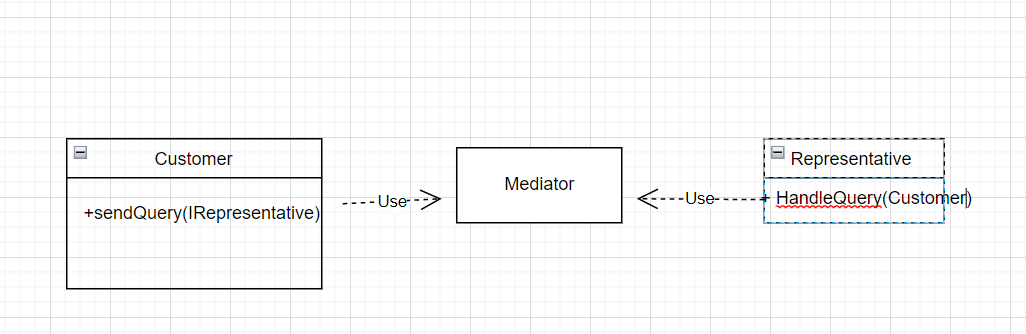
Whenever a different department is selected, both of the reports should be refreshed with the data for the currently selected department.

Sol : event driven .. observer



1. Customer service representatives at some commercial banks handle queries from their existing and potential customers using an online chat application. At peak times, each representative may need to work with more than one customer simultaneously. Design this communication mechanism between different User objects and Representative objects

Sol : if there is n\*m communication requirement , its observer or mediator

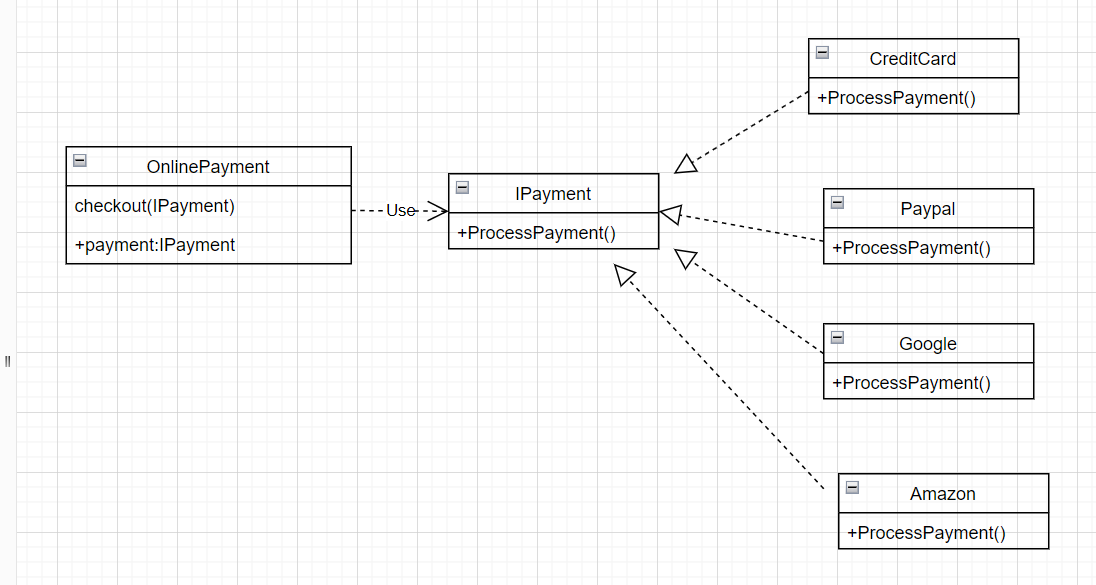


public class OnlineCart  
{  
    public void CheckOut(PaymentType paymentType)  
    {  
        switch(paymentType)  
        {  
            case PaymentType.CreditCard:  
                ProcessCreditCardPayment();  
                break;  
            case PaymentType.Paypal:  
                ProcessPaypalPayment();  
                break;  
            case PaymentType.GoogleCheckout:  
                ProcessGooglePayment();  
                break;  
            case PaymentType.AmazonPayments:  
                ProcessAmazonPayment();  
                break;  
        }  
    }  
  
    private void ProcessCreditCardPayment()  
    {  
        Print("Credit card payment chosen");  
    }  
  
    private void ProcessPaypalPayment()  
    {  
        Print("Paypal payment chosen");  
    }  
  
    private void ProcessGooglePayment()  
    {  
        Print("Google payment chosen");  
    }  
  
    private void ProcessAmazonPayment()  
    {  
        Print("Amazon payment chosen");  
    }  
}

Solution :

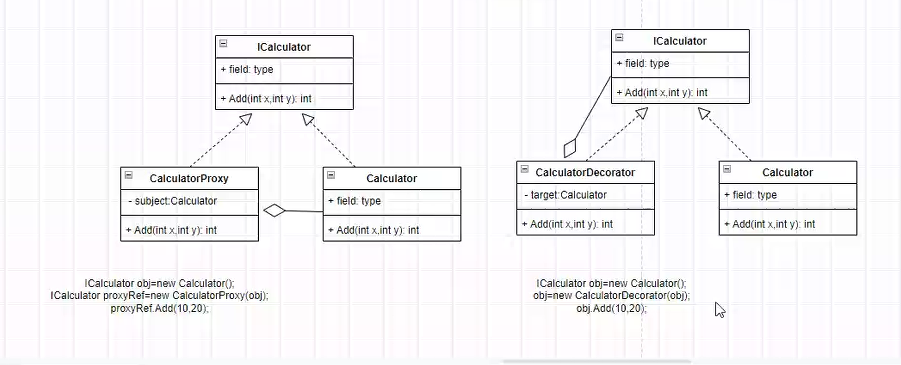
I thought initially :Violates LSP : use of visitor design pattern??

Actual solution after discussiob : violates SRP : so use strategy and that should solve the problem



1. public class ConcreteCalculator : ICalculator  
   {  
       public int Add(int x, int y)  
       {  
           Print("Add(x={0}, y={1})", x, y);  
     
           var addition = x + y;  
     
           Print("result={0}", addition);  
     
           return addition;  
       }  
   }

Solution : aspect oriented programming (can be done through proxy or decorator)



However Decorator code will have clean code while proxy class will be GOD class if there are other functionalites apart from logger added .Below is the decorator pattern

