**Section13 Facade Design Pattern**

**Lesson01 what is Façade Design Pattern**

**Notes:-**

**1-facad design pattern: exposing multiple components through single interface**



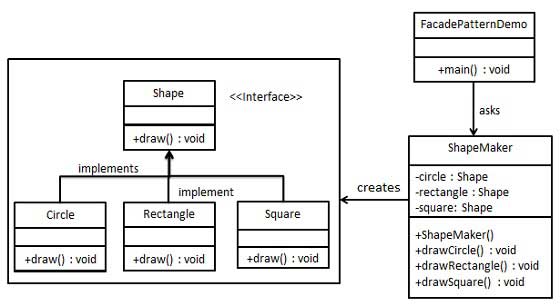
**3-as another sample of using façade design pattern when to use any method from any DLL it just call method inside interface**

**4-Façade provide simple, easy to understand user interface over large of sophisticated body of code**

**5-Facade pattern hides the complexities of the system and provides an interface to the client using which the client can access the system. This type of design pattern comes under structural pattern as this pattern adds an interface to existing system to hide its complexities.**

**Example:-**

**In the below example we see that we implement façade design pattern by make interface that implement circle , rectangle , square to shape interface and using gateway class to inject the shapes as properities not arguments as private and using public method to using it directly as below**



**Example:-**

**namespace FacadeDesignPro{**

**public interface Shape{void draw();}}**

**using System;**

**namespace FacadeDesignPro{**

**public class Rectangle : Shape{**

**public void draw(){Console.WriteLine("Rectangle::draw()");}}}**

**using System;**

**namespace FacadeDesignPro{**

**public class Circle : Shape{**

**public void draw(){Console.WriteLine("Circle Shape");}}}**

**using System;**

**namespace FacadeDesignPro{**

**public class Square : Shape{**

**public void draw(){Console.WriteLine("Square Shape");}}}**

**namespace FacadeDesignPro{**

**//we see that on the ShapeMaker class we declare the instance of each terminal class by provide the type of the interface**

**//so the facade design pattern hides the complexity system by provide you the ShapeMaker class only**

**public class ShapeMaker{**

**private Shape circle;**

**private Shape rectangle;**

**private Shape square;**

**public ShapeMaker(){**

**circle = new Circle();**

**rectangle = new Rectangle();**

**square = new Square();}**

**public void drawCircle(){circle.draw();}**

**public void drawRectangle(){rectangle.draw();}**

**public void drawSquare(){square.draw();}}}**

**using FacadeDesignPro.DotNetDesignPatternDemos.Structural.Facade;**

**using System;**

**namespace FacadeDesignPro{**

**class Program{**

**public static void Main(String[] args){**

**ShapeMaker shapeMaker = new ShapeMaker();**

**//we see that once initialize the ShapeMaker it will create instance of all the terminal class and //internally can access to the implementation to the target sub classes**

**shapeMaker.drawCircle();**

**shapeMaker.drawRectangle();**

**shapeMaker.drawSquare();}}}**