**Do Factory Adapter Design Pattern**

**Notes:-**

**1-Convert the interface of a class into another interface clients expect. Adapter lets classes work together that couldn't otherwise because of incompatible interfaces.**

**2-by implement adapter design by make general class and then generate class inherit from the base class and hold instance of the adapter class that use to adapt result as we want**

**Example:-**

**using System;**

**namespace DoFactoryAdapter{**

**class Compound{**

**protected string \_chemical;**

**protected float \_boilingPoint;**

**protected float \_meltingPoint;**

**protected double \_molecularWeight;**

**protected string \_molecularFormula;**

**// Constructor**

**public Compound(string chemical){this.\_chemical = chemical;}**

**public virtual void Display(){**

**Console.WriteLine("\nCompound: {0} ------ ", \_chemical);}}}**

**//make RickCompound inherit from the Compound and pass the name**

**using System;**

**namespace DoFactoryAdapter{**

**class RichCompound : Compound{**

**//we inject instance of \_bank to provide support of adapting**

**private ChemicalDatabank \_bank;**

**// Constructor**

**public RichCompound(string name): base(name){}**

**public override void Display(){**

**// The Adaptee**

**\_bank = new ChemicalDatabank();**

**\_boilingPoint = \_bank.GetCriticalPoint(\_chemical, "B");**

**\_meltingPoint = \_bank.GetCriticalPoint(\_chemical, "M");**

**\_molecularWeight = \_bank.GetMolecularWeight(\_chemical);**

**\_molecularFormula = \_bank.GetMolecularStructure(\_chemical);**

**base.Display();**

**Console.WriteLine(" Formula: {0}", \_molecularFormula);**

**Console.WriteLine(" Weight : {0}", \_molecularWeight);**

**Console.WriteLine(" Melting Pt: {0}", \_meltingPoint);**

**Console.WriteLine(" Boiling Pt: {0}", \_boilingPoint);}}}**

**//Adapter Class that use to adapt the result incoming**

**namespace DoFactoryAdapter{**

**class ChemicalDatabank{**

**// The databank 'legacy API'**

**public float GetCriticalPoint(string compound, string point){**

**// Melting Point**

**if (point == "M"){**

**switch (compound.ToLower())**

**{case "water": return 0.0f;**

**case "benzene": return 5.5f;**

**case "ethanol": return -114.1f;**

**default: return 0f;}}**

**// Boiling Point**

**else{**

**switch (compound.ToLower()){**

**case "water": return 100.0f;**

**case "benzene": return 80.1f;**

**case "ethanol": return 78.3f;**

**default: return 0f;}}}**

**public string GetMolecularStructure(string compound){**

**switch (compound.ToLower()){**

**case "water": return "H20";**

**case "benzene": return "C6H6";**

**case "ethanol": return "C2H5OH";**

**default: return "";}}**

**public double GetMolecularWeight(string compound){**

**switch (compound.ToLower()){**

**case "water": return 18.015;**

**case "benzene": return 78.1134;**

**case "ethanol": return 46.0688;**

**default: return 0d;}}}}**

**using static System.Console;**

**namespace DoFactoryAdapter{**

**class Program{**

**static void Main(string[] args){**

**// Non-adapted chemical compound**

**Compound unknown = new Compound("Unknown");**

**unknown.Display();**

**// Adapted chemical compounds**

**Compound water = new RichCompound("Water");**

**water.Display();**

**Compound benzene = new RichCompound("Benzene");**

**benzene.Display();**

**Compound ethanol = new RichCompound("Ethanol");**

**ethanol.Display();**

**// Wait for user**

**ReadKey();}}}**