***Лабараторна робота №7***

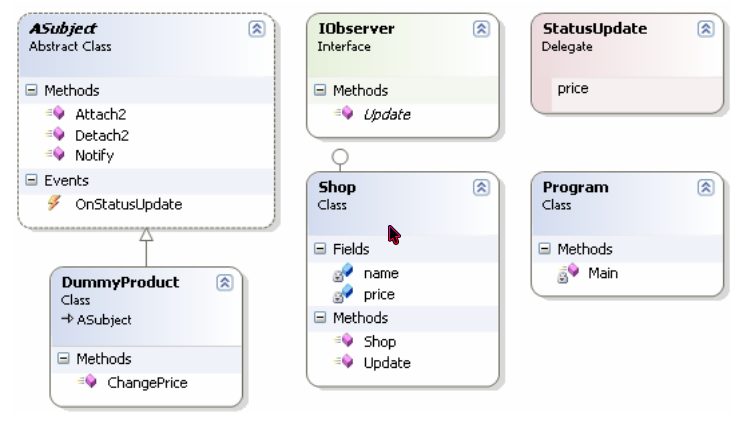
***Варіант №14***

**Завдання №1**

Нехай необхідно створити програмне забезпечення, що дає змогу стежити за зміною ціни на товар у магазинах. У рамках поставленої задачі найдоцільніше використовувати паттерн Спостерігач.

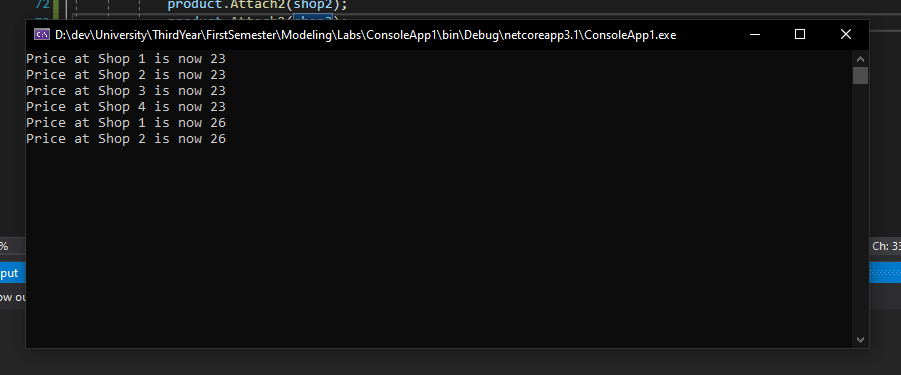
***Результати виконання завдання 1:***

*Діаграма класів додатку (рис.1):*



**Рис. 1 Діаграма класів додатку**

*Результати роботи додатку (рис.2)*



**Рис. 2 Результат виконаня додатку**

*Лістинг програмного коду:*

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApp1

{

public delegate void StatusUpdate(float price);

abstract class ASubject

{

public event StatusUpdate OnStatusUpdate;

public void Attach2(Shop product)

{

// lets assign attach the observers with subjects

OnStatusUpdate += new StatusUpdate(product.Update);

}

public void Detach2(Shop product)

{

// lets assign detach the observers with subjects

OnStatusUpdate -= new StatusUpdate(product.Update);

}

public void Notify(float price)

{

// lets notify the observers with change

if (OnStatusUpdate != null)

{

OnStatusUpdate(price);

}

}

}

class DummyProduct : ASubject

{

public void ChangePrice(float price)

{

Notify(price);

}

}

interface IObserver

{

void Update(float price);

}

class Shop : IObserver

{

//Name if the product

string name;

float price = 0.0f; //default

public Shop(string name)

{

this.name = name;

}

public void Update(float price)

{

this.price = price;

//Lets print on console just to test the working

Console.WriteLine(@"Price at {0} is now {1}", name, price);

}

}

class Program

{

static void Main(string[] args)

{

DummyProduct product = new DummyProduct();

// We have four shops wanting to keep updated price set by product owner

Shop shop1 = new Shop("Shop 1");

Shop shop2 = new Shop("Shop 2");

Shop shop3 = new Shop("Shop 3");

Shop shop4 = new Shop("Shop 4");

//Lets use WAY\_2 for other two shops

product.Attach2(shop1);

product.Attach2(shop2);

product.Attach2(shop3);

product.Attach2(shop4);

//Now lets try chaging the products price, this should update the shops automatically

product.ChangePrice(23.0f);

product.Detach2(shop3);

product.Detach2(shop4);

//Now lets try changing the products price again

product.ChangePrice(26.0f);

Console.Read();

}

}

}