HOMEWORK: Design pattems

NAME: Hoàng Trung Hiếu

CLASS: 20Se3

Bài Tập

using System;

using System.Collections.Generic;

// Subject (Chủ đề)

public class Subject

{

private List<IObserver> observers = new List<IObserver>();

private int state;

public int State

{

get { return state; }

set

{

state = value;

NotifyObservers();

}

}

public void Attach(IObserver observer)

{

observers.Add(observer);

}

public void Detach(IObserver observer)

{

observers.Remove(observer);

}

public void NotifyObservers()

{

foreach (IObserver observer in observers)

{

observer.Update();

}

}

}

// Observer

public interface IObserver

{

void Update();

}

public class ConcreteObserver : IObserver

{

private string name;

private Subject subject;

public ConcreteObserver(string name, Subject subject)

{

this.name = name;

this.subject = subject;

}

public void Update()

{

Console.WriteLine($"Observer {name} received update. New state: {subject.State}");

}

}

class Program

{

static void Main(string[] args)

{

Subject subject = new Subject();

ConcreteObserver observer1 = new ConcreteObserver("Observer 1", subject);

ConcreteObserver observer2 = new ConcreteObserver("Observer 2", subject);

subject.Attach(observer1);

subject.Attach(observer2);

subject.State = 5; // Thay đổi trạng thái và thông báo cho các observer.

}

}

1. Khai báo interface IObserver:

Interface này định nghĩa một phương thức Update(). Tất cả các lớp Observer phải triển khai phương thức này để nhận thông báo từ Subject khi trạng thái của nó thay đổi.

2. Khai báo lớp Subject:

Lớp này là "Subject" trong mô hình Observer. Nó chứa dữ liệu trạng thái (ở đây là state) và danh sách các đối tượng "Observer" đang theo dõi nó.

Có các phương thức sau:

Attach(IObserver observer): Được gọi để thêm một đối tượng Observer vào danh sách theo dõi của Subject.

Detach(IObserver observer): Được gọi để xóa một đối tượng Observer khỏi danh sách theo dõi của Subject.

NotifyObservers(): Được gọi để thông báo cho tất cả các Observer trong danh sách khi trạng thái thay đổi.

State: Thuộc tính này là trạng thái mà Subject theo dõi. Khi nó thay đổi, phương thức NotifyObservers() được gọi để thông báo cho các Observer.

3. Khai báo lớp ConcreteObserver:

Lớp này triển khai interface IObserver và đại diện cho các đối tượng Observer cụ thể.

Nó có hai thuộc tính: name để định danh đối tượng Observer và subject để theo dõi Subject.

Phương thức Update() được triển khai để xử lý thông báo khi Subject thay đổi trạng thái.

4. Trong phần Main của chương trình:

Tạo một đối tượng Subject và hai đối tượng ConcreteObserver.

Gọi phương thức Attach() để đăng ký các Observer vào Subject để theo dõi trạng thái của nó.

Sau đó, thay đổi trạng thái của Subject bằng cách gọi subject.State = 5;. Khi trạng thái thay đổi, NotifyObservers() sẽ được gọi và thông báo sẽ được gửi đến các Observer. Mỗi Observer sẽ nhận thông báo và in ra màn hình trạng thái mới của Subject.

Kết quả : A screenshot of a computer

Description automatically generated