

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет інформаційних технологій

Кафедра комп'ютерної інженерії та інформаційних систем

## Лабораторна робота № 3

з дисципліни «Об'єктно-орієнтовані технології програмування»

на тему:

**«Поведінкові патерни»**

**Виконав:**

студент 1 курсу, групи КІ2м-23-2

\_\_\_\_\_ Мариняк В.М.

**Перевірив:**

\_\_\_\_\_ Лисенко С.М.

## Хід роботи

Розглянемо варіант реалізації патернів Стратегія та Спостерігач на прикладі застосунку для системи моніторингу погоди, де патерн Спостерігач використовуватиметься для відображення оновлених даних про погоду на декількох дисплеях, а патерн Стратегія дозволить впровадити різні типи відображення даних.

### Технічне завдання:

1. Визначити інтерфейси ISubject та IObservable.
2. Створити клас WeatherData, що реалізовує інтерфейс ISubject.
3. Створити конкретний клас, що реалізовує інтерфейс IObservable.
4. Визначити IDisplayStrategy інтерфейс.
5. Створити конкретні класи, що реалізовуватимуть різні типи відображення даних.
6. Інтегрувати компоненти патерну Стратегія в класи-спостерігачі аби вони могли обирати тип відображення даних.

### Реалізація:

1. Створення інтерфейсів для патерну Спостерігач

```
1 namespace StrategyAndObserverExample.Maryniak.Models;
2
3 public interface IObservable
4 {
5     void Update(float temperature, float humidity, float pressure);
6 }
```

```
1 namespace StrategyAndObserverExample.Maryniak.Models;
2
3 public interface ISubject
4 {
5     void RegisterObserver(IObservable observer);
6     void RemoveObserver(IObservable observer);
7     void NotifyObservers();
8 }
```

## 2. Створення конкретного Subject класу, що реалізовує інтерфейс ISubject

```
1 namespace StrategyAndObserverExample.Maryniak.Models;
2
3 public class WeatherData : ISubject
4 {
5     private List<IObserver> observers;
6     private float temperature;
7     private float humidity;
8     private float pressure;
9
10    public WeatherData() => observers = [];
11
12    public void RegisterObserver(IObserver observer) => observers.Add(observer);
13
14    public void RemoveObserver(IObserver observer) => observers.Remove(observer);
15
16    public void NotifyObservers()
17    {
18        foreach (var observer in observers)
19        {
20            observer.Update(temperature, humidity, pressure);
21        }
22    }
23
24    public void MeasurementsChanged() => NotifyObservers();
25
26    public void SetMeasurements(float temperature, float humidity, float pressure)
27    {
28        this.temperature = temperature;
29        this.humidity = humidity;
30        this.pressure = pressure;
31        MeasurementsChanged();
32    }
33 }
```

## 3. Створення інтерфейсу для патерну Стратегія

```
1 namespace StrategyAndObserverExample.Maryniak.Models;
2
3 public interface IDisplayStrategy
4 {
5     void Display(float temperature, float humidity, float pressure);
6 }
```

## 4. Створення конкретних класів, що реалізують інтерфейс IDisplayStrategy

```
1 namespace StrategyAndObserverExample.Maryniak.Models;
2
3 public class SimpleDisplayStrategy : IDisplayStrategy
4 {
5     public void Display(float temperature, float humidity, float pressure)
6     {
7         Console.WriteLine($"Simple Display: " +
8             $"Temp={temperature}°C, " +
9             $"Humidity={humidity}%");
10    }
11 }
```

```

1 namespace StrategyAndObserverExample.Maryniak.Models;
2
3 public class DetailedDisplayStrategy : IDisplayStrategy
4 {
5     public void Display(float temperature, float humidity, float pressure)
6     {
7         Console.WriteLine($"Detailed Display: " +
8             $"Temp={temperature}°C, " +
9             $"Humidity={humidity}%, " +
10            $"Pressure={pressure} hPa");
11     }
12 }

```

5. Поєднання реалізацій інтерфейсів патернів Спостерігач та Стратегія за допомогою конкретного класу, що реалізовує інтерфейс IObserver

```

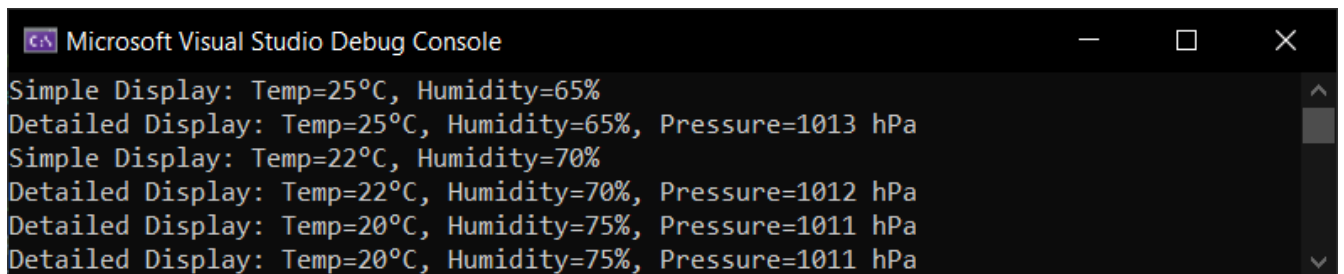
1 namespace StrategyAndObserverExample.Maryniak.Models;
2
3 public class WeatherDisplay(IDisplayStrategy displayStrategy) : IObserver
4 {
5     private IDisplayStrategy displayStrategy = displayStrategy;
6
7     private float temperature;
8     private float humidity;
9     private float pressure;
10
11     public void Update(float temperature, float humidity, float pressure)
12     {
13         this.temperature = temperature;
14         this.humidity = humidity;
15         this.pressure = pressure;
16         Display();
17     }
18
19     public void SetDisplayStrategy(IDisplayStrategy displayStrategy)
20         => this.displayStrategy = displayStrategy;
21
22     private void Display()
23         => displayStrategy.Display(temperature, humidity, pressure);
24 }

```

## 6. Клієнтський код

```
1  using StrategyAndObserverExample.Maryniak.Models;
2
3  namespace StrategyAndObserverExample.Maryniak;
4
5  internal class Program
6  {
7      static void Main()
8      {
9          // Create WeatherData (subject)
10         var weatherData = new WeatherData();
11
12         // Create display strategies
13         IDisplayStrategy simpleDisplay = new SimpleDisplayStrategy();
14         IDisplayStrategy detailedDisplay = new DetailedDisplayStrategy();
15
16         // Create observers with different display strategies
17         var simpleWeatherDisplay = new WeatherDisplay(simpleDisplay);
18         var detailedWeatherDisplay = new WeatherDisplay(detailedDisplay);
19
20         // Register observers with the subject
21         weatherData.RegisterObserver(simpleWeatherDisplay);
22         weatherData.RegisterObserver(detailedWeatherDisplay);
23
24         // Simulate new weather measurements
25         weatherData.SetMeasurements(25.0f, 65.0f, 1013.0f);
26         weatherData.SetMeasurements(22.0f, 70.0f, 1012.0f);
27
28         // Change strategy at runtime
29         simpleWeatherDisplay.SetDisplayStrategy(detailedDisplay);
30         weatherData.SetMeasurements(20.0f, 75.0f, 1011.0f);
31     }
32 }
```

## 7. Результат виконання програми



```
Microsoft Visual Studio Debug Console
Simple Display: Temp=25°C, Humidity=65%
Detailed Display: Temp=25°C, Humidity=65%, Pressure=1013 hPa
Simple Display: Temp=22°C, Humidity=70%
Detailed Display: Temp=22°C, Humidity=70%, Pressure=1012 hPa
Detailed Display: Temp=20°C, Humidity=75%, Pressure=1011 hPa
Detailed Display: Temp=20°C, Humidity=75%, Pressure=1011 hPa
```

**Висновок.** Під час лабораторної роботи я навчився реалізовувати патерни Спостерігач і Стратегія та створив консольний додаток, який демонструє принципи їхньої роботи.