

## Практическая работа по паттерну "Наблюдатель" (Observer) в C#

### Цель

Цель этой практической работы - изучить и применить паттерн "Наблюдатель" для создания системы, которая позволяет объектам автоматически оповещать своих наблюдателей о изменениях своего состояния.

### Задание

Разработать простую систему учета погоды с использованием паттерна "Наблюдатель" на языке C#. Эта система будет включать в себя следующие компоненты:

Погодный отчет (WeatherReport): Этот компонент будет представлять собой объект, который наблюдается. Он будет содержать текущую температуру.

Наблюдатель погоды (WeatherObserver): Этот компонент будет представлять собой интерфейс для всех наблюдателей. У него будет метод Update(WeatherReport report), который будет вызываться при изменениях погоды.

Конкретный наблюдатель (ConcreteWeatherObserver): Этот компонент будет представлять собой конкретный наблюдатель, который будет реагировать на изменения погоды.

Конкретный погодный отчет (ConcreteWeatherReport): Этот компонент будет реализовывать интерфейс WeatherReport и предоставлять метод для установки новой температуры и оповещения наблюдателей о изменениях.

### Примеры кода

Вот примеры кода для каждого из компонентов:

```
// Интерфейс наблюдателя погоды
public interface IWeatherObserver
{
    void Update(WeatherReport report);
}

// Конкретный наблюдатель погоды
```

```

public class ConcreteWeatherObserver : IWeatherObserver
{
    private string name;

    public ConcreteWeatherObserver(string name)
    {
        this.name = name;
    }

    public void Update(IWeatherReport report)
    {
        Console.WriteLine($"{name} получил обновленный отчет
о погоде. Температура: {report.Temperature}°C");
    }
}

// Интерфейс погодного отчета
public interface IWeatherReport
{
    int Temperature { get; }
}

// Конкретный погодный отчет
public class ConcreteWeatherReport : IWeatherReport
{
    private int temperature;
    private List<IWeatherObserver> observers = new
List<IWeatherObserver>();

    public int Temperature
    {
        get { return temperature; }
    }

    public void SetTemperature(int temperature)
    {
        this.temperature = temperature;
        NotifyObservers();
    }

    public void AddObserver(IWeatherObserver observer)
    {
        observers.Add(observer);
    }
}

```

```
public void RemoveObserver(IWeatherObserver observer)
{
    observers.Remove(observer);
}

private void NotifyObservers()
{
    foreach (var observer in observers)
    {
        observer.Update(this);
    }
}
```

### Инструкции

Создайте объекты `ConcreteWeatherReport` и несколько объектов `ConcreteWeatherObserver`.

Зарегистрируйте наблюдателей в объекте `ConcreteWeatherReport` с помощью метода `AddObserver`.

Измените температуру в объекте `ConcreteWeatherReport` с помощью метода `SetTemperature`, и убедитесь, что наблюдатели получают уведомления об изменениях.

Проведите тестирование вашей системы и убедитесь, что наблюдатели правильно реагируют на изменения погоды.