

АНАЛІЗ ВИКОРИСТАННЯ ПАТТЕРНА «ЦЕПОЧКА ОБОВ'ЯЗКІВ» В ПРОГРАМУВАННІ

Гасюк М.О.

*Національний технічний університет
«Харківський політехнічний інститут», м. Харків*

У програмуванні паттерн «Цепочка обов'язків» (Chain of Responsibility) – це поведінковий паттерн, який дозволяє передавати запити вздовж ланцюжка обробників. Кожен наступний обробник вирішує, чи може він самостійно обробити запит або передати його далі по ланцюжку. Давайте розглянемо простий приклад використання цього патерну на C#:

```
public abstract class Handler
{
    protected Handler successor;

    public void SetSuccessor(Handler successor)
    {
        this.successor = successor;
    }

    public abstract void HandleRequest(int request);
}

public class ConcreteHandler1 : Handler
{
    public override void HandleRequest(int request)
    {
        if (request >= 0 && request < 10)
        {
            Console.WriteLine($"{this.GetType().Name} handled request {request}");
        }
        else if (successor != null)
        {
            successor.HandleRequest(request);
        }
    }
}

public class ConcreteHandler2 : Handler
{
    public override void HandleRequest(int request)
    {
        if (request >= 10 && request < 20)
        {
            Console.WriteLine($"{this.GetType().Name} handled request {request}");
        }
        else if (successor != null)
        {
            successor.HandleRequest(request);
        }
    }
}
```

Рисунок 1 – Приклад коду

Перевагами є зниження залежності, тобто відправник не знає конкретного обробника запиту, що спрощує архітектуру, а також гнучкість, наприклад можливість змінювати обробники у ланцюжку без модифікації відправника. Цей паттерн зручно використовувати, коли є кілька різних, але пов'язаних операцій, які можуть бути виконані незалежно одна від одної, і коли важливо забезпечити гнучкість обробки різних запитів.

Література:

1. Gamma E., Helm R., Johnson R., Vlissides J. Design Patterns: Elements of Reusable Object-Oriented Software / Gamma E., Helm R., Johnson R., Vlissides J. // Addison-Wesley Professional, 2017 – P. 223-230.
2. Freeman E., Freeman E., Bates B., Sierra K. Head First Design Patterns / Freeman E., Freeman E., Bates B., Sierra K. // O'Reilly Media, Inc., 2019 – P. 181-190.