

Паттерн проектирования "Фасад" (Facade)

Паттерн проектирования "Фасад" предоставляет унифицированный интерфейс к набору интерфейсов в подсистеме. Он позволяет скрыть сложность системы и предоставить более простой способ взаимодействия с ней.

Пример кода с пояснениями

Давайте рассмотрим пример применения паттерна "Фасад" для работы с комплексной системой заказа в интернет-магазине.

```
using System;

// Подсистема 1: Управление товарами
class ProductSubsystem
{
    public void AddProductToCart(int productId)
    {
        Console.WriteLine($"Товар с ID {productId} добавлен в корзину");
    }

    public void RemoveProductFromCart(int productId)
    {
        Console.WriteLine($"Товар с ID {productId} удален из корзины");
    }
}

// Подсистема 2: Оформление заказа
class OrderSubsystem
{
    public void CreateOrder()
    {
        Console.WriteLine("Заказ создан");
    }

    public void ProcessPayment()
    {
        Console.WriteLine("Платеж обработан");
    }
}
```

```
// Фасад: упрощает взаимодействие с подсистемами
class OnlineShopFacade
{
    private ProductSubsystem productSubsystem;
    private OrderSubsystem orderSubsystem;

    public OnlineShopFacade()
    {
        productSubsystem = new ProductSubsystem();
        orderSubsystem = new OrderSubsystem();
    }

    public void BuyProduct(int productId)
    {
        productSubsystem.AddProductToCart(productId);
        orderSubsystem.CreateOrder();
        orderSubsystem.ProcessPayment();
        productSubsystem.RemoveProductFromCart(productId);
        Console.WriteLine("Заказ завершен");
    }
}

class Program
{
    static void Main()
    {
        OnlineShopFacade shop = new OnlineShopFacade();
        int productId = 123;

        // Покупка товара с использованием фасада
        shop.BuyProduct(productId);
    }
}
```

Пояснения:

В данном примере есть две подсистемы: управление товарами и оформление заказа.

OnlineShopFacade предоставляет упрощенный метод BuyProduct, который выполняет последовательность действий для покупки товара:

добавление в корзину, создание заказа, обработку платежа и удаление из корзины.

Клиентский код (Main) использует только фасад для совершения покупки, не заботясь о деталях внутренней работы подсистем.

Практические задания

1. Добавьте новую подсистему (например, систему доставки) и интегрируйте ее с фасадом OnlineShopFacade. Создайте методы в фасаде для оформления доставки.

2. Реализуйте обработку ошибок и исключений в фасаде и его подсистемах. Например, обработайте случай, когда товар не может быть добавлен в корзину из-за отсутствия на складе.

3. Создайте юнит-тесты для классов фасада и его подсистем. Проверьте, что фасад правильно взаимодействует с подсистемами и обрабатывает различные сценарии.

4. Расширьте функциональность фасада, добавив методы для отслеживания истории заказов клиента. Обновите клиентский код для работы с новыми функциями.

5. Подумайте о том, как можно оптимизировать код фасада для уменьшения избыточных операций. Реализуйте оптимизированный вариант фасада.

Заключение

Паттерн "Фасад" помогает упростить сложные системы и сделать их более доступными для клиентского кода, предоставляя унифицированный интерфейс к подсистемам. Он широко используется для улучшения структуры кода и облегчения его поддержки и расширения.