Міністерство освіти і науки України

Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

Звіт

з лабораторної роботи №5

з дисципліни

«Основи розробки програмного забезпечення на платформі

Microsoft.NET»

«Шаблони проектування»

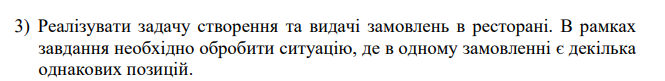
Виконала ІП-21 Голованьов Г.О.

Київ 2024

**Комп‘ютерний практикум № 5**

**Шаблони проектування. Структурні шаблони**

**Мета:** ознайомитися з основними шаблонами проектування, навчитися застосовувати їх при проектуванні і розробці ПЗ.



**Обґрунтування обраного патерну: Composite**

Для реалізації задачі створення та видачі замовлень в ресторані був обраний структурний патерн **Компонувальник (Composite)**. Цей патерн дозволяє будувати дерево об'єктів, де кожен об'єкт може бути як простим, так і складеним з інших об'єктів. В нашому випадку, це дозволяє легко управляти меню, яке може містити окремі елементи або підменю, та замовленням, яке може складатися з декількох однакових елементів.

Чому саме Компонувальник:

1. Гнучкість в управлінні елементами меню: Можливість створення меню, яке містить як окремі елементи (страви), так і підменю.
2. Проста реалізація ієрархій: Легко реалізувати ієрархічну структуру меню та замовлень.
3. Полегшене управління замовленнями: Можливість роботи з кожним елементом замовлення як з єдиним об'єктом незалежно від того, є він окремим елементом або підменю.
4. Легке масштабування: Додавання нових елементів меню або нових типів замовлень без значних змін в існуючому коді.

**Опис архітектури проекту:**

1. Інтерфейс IMenuItem:
   * Методи:
     + decimal GetPrice(): Отримання ціни елемента.
     + void Display(): Відображення інформації про елемент.
2. Клас MenuItem (реалізує IMenuItem):
   * Поля:
     + string name: Назва елемента.
     + decimal price: Ціна елемента.
   * Конструктор:
     + MenuItem(string name, decimal price): Ініціалізація назви та ціни елемента.
   * Методи:
     + decimal GetPrice(): Повертає ціну елемента.
     + void Display(): Відображає інформацію про елемент.
     + string GetName(): Повертає назву елемента.
3. Клас Menu (реалізує IMenuItem):
   * Поля:
     + List<IMenuItem> menuItems: Список елементів меню.
     + string name: Назва меню.
   * Конструктор:
     + Menu(string name): Ініціалізація назви меню.
   * Методи:
     + void Add(IMenuItem menuItem): Додавання елемента в меню.
     + void Remove(IMenuItem menuItem): Видалення елемента з меню.
     + decimal GetPrice(): Повертає загальну ціну всіх елементів меню.
     + void Display(): Відображає інформацію про всі елементи меню.
     + List<IMenuItem> GetMenuItems(): Повертає список елементів меню.
4. Клас Order:
   * Поля:
     + List<(IMenuItem item, int quantity)> orderItems: Список елементів замовлення та їх кількість.
   * Методи:
     + void AddItem(IMenuItem menuItem, int quantity): Додає елемент до замовлення.
     + decimal GetTotalPrice(): Повертає загальну ціну всього замовлення.
     + void DisplayOrder(): Відображає інформацію про замовлення.

**UML-діаграма**

A computer screen shot of a server

Description automatically generated

**Особливості реалізації обраного патерну:**

1. Інтерфейс IMenuItem: Всі елементи меню реалізують цей інтерфейс, що дозволяє управляти як окремими стравами, так і меню однаково.
2. Клас MenuItem: Реалізує елементи меню з простою інформацією (назва та ціна).
3. Клас Menu: Реалізує складені елементи меню, які можуть містити інші елементи меню або підменю.
4. Клас Order: Управляє списком елементів замовлення, дозволяє додавати однакові елементи з різною кількістю.

# Текст програми

// FilePath: Program.cs

using System;

using System.Collections.Generic;

// Інтерфейс MenuItem

interface IMenuItem

{

decimal GetPrice();

void Display();

}

// Клас MenuItem (Leaf)

class MenuItem : IMenuItem

{

private string name;

private decimal price;

public MenuItem(string name, decimal price)

{

this.name = name;

this.price = price;

}

public decimal GetPrice()

{

return price;

}

public void Display()

{

Console.WriteLine($"{name}: {price:C}");

}

public string GetName()

{

return name;

}

}

// Клас Menu (Composite)

class Menu : IMenuItem

{

private List<IMenuItem> menuItems = new List<IMenuItem>();

private string name;

public Menu(string name)

{

this.name = name;

}

public void Add(IMenuItem menuItem)

{

menuItems.Add(menuItem);

}

public void Remove(IMenuItem menuItem)

{

menuItems.Remove(menuItem);

}

public decimal GetPrice()

{

decimal total = 0;

foreach (var item in menuItems)

{

total += item.GetPrice();

}

return total;

}

public void Display()

{

Console.WriteLine($"{name}:");

foreach (var item in menuItems)

{

item.Display();

}

}

public List<IMenuItem> GetMenuItems()

{

return menuItems;

}

}

// Клас Order

class Order

{

private List<(IMenuItem item, int quantity)> orderItems = new List<(IMenuItem item, int quantity)>();

public void AddItem(IMenuItem menuItem, int quantity)

{

orderItems.Add((menuItem, quantity));

}

public decimal GetTotalPrice()

{

decimal total = 0;

foreach (var orderItem in orderItems)

{

total += orderItem.item.GetPrice() \* orderItem.quantity;

}

return total;

}

public void DisplayOrder()

{

Console.WriteLine("Order Details:");

foreach (var orderItem in orderItems)

{

Console.WriteLine($"{orderItem.quantity} x");

orderItem.item.Display();

}

Console.WriteLine($"Total Price: {GetTotalPrice():C}");

}

}

// Головний клас

class Program

{

static void Main(string[] args)

{

// Створення елементів меню

MenuItem pizza = new MenuItem("Pizza", 8.99m);

MenuItem burger = new MenuItem("Burger", 5.49m);

MenuItem soda = new MenuItem("Soda", 1.99m);

// Створення меню

Menu mainMenu = new Menu("Main Menu");

mainMenu.Add(pizza);

mainMenu.Add(burger);

mainMenu.Add(soda);

// Відображення меню

mainMenu.Display();

// Створення замовлення

Order order = new Order();

// Взаємодія з користувачем

bool ordering = true;

while (ordering)

{

Console.WriteLine("Enter the item name to add to your order (or type 'done' to finish): ");

string itemName = Console.ReadLine();

if (itemName.ToLower() == "done")

{

ordering = false;

continue;

}

IMenuItem selectedItem = mainMenu.GetMenuItems().Find(item => item is MenuItem menuItem && menuItem.GetName().ToLower() == itemName.ToLower());

if (selectedItem != null)

{

Console.WriteLine("Enter the quantity: ");

if (int.TryParse(Console.ReadLine(), out int quantity))

{

order.AddItem(selectedItem, quantity);

Console.WriteLine($"{quantity} x {itemName} added to your order.");

}

else

{

Console.WriteLine("Invalid quantity. Please enter a number.");

}

}

else

{

Console.WriteLine("Item not found in menu. Please try again.");

}

}

// Відображення замовлення

order.DisplayOrder();

}

}

# Скріншоти виконання

A screenshot of a computer menu

Description automatically generated