



МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО”

Факультет прикладної математики
Кафедра програмного забезпечення комп'ютерних систем

Лабораторна робота № 5
з дисципліни “Програмування. Об'єктно-орієнтоване
програмування та шаблони проєктування”
тема “Поведінкові шаблони.”

Виконала
студентка II курсу
групи КП-02
Красношاپка Анастасія Андріївна

Перевірила
“ ____ ” “ ____ ” 20__ р.
викладач
Заболотня Тетяна Миколаївна

Варіант 5

Мета роботи

Ознайомитися зі поведінковими шаблонами, навчитися визначати, де саме потрібно використовувати кожен з розглянутих варіантів, а також розробляти програмне забезпечення на мові C# з їх використанням.

Постановка задачі

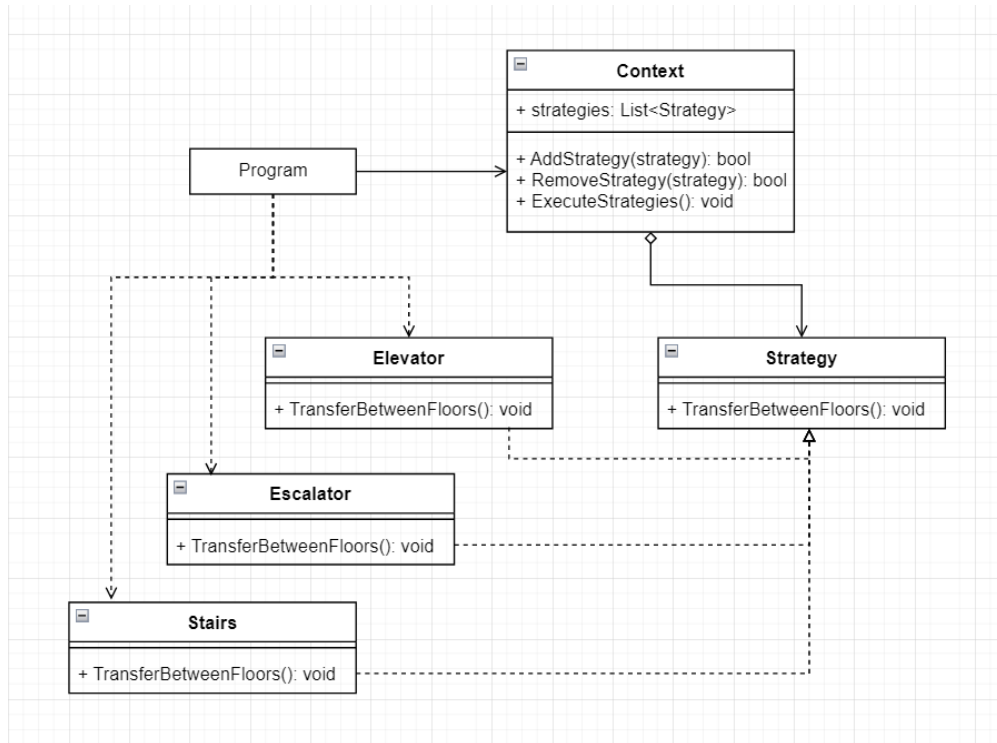
1. За допомогою шаблону проєктування реалізувати декілька способів переміщення нагору та вниз у торговому центрі. Наприклад, можна ходити сходами, їздити на ескалаторі та їздити на ліфті. В залежності від стану устаткування або від часу доби активувати той чи інший спосіб пересування відвідувачів. Наприклад, з 22.00 до 10.00 кожного дня ескалатори та ліфти перестають працювати. А під час профілактичних робіт з ескалатором має працювати ліфт та навпаки.
2. На певному заводі цілодобово працюють 3 зміни. I зміна – з 6.00 до 15.00, II зміна – з 14.00 до 23.00 та III зміна – з 22.00 до 7.00. Якщо завдання робітникам певної зміни надходить менш, ніж за годину до закінчення зміни, вони передають його на виконання наступній зміні. За допомогою шаблону проєктування реалізувати механізм обробки завдань робітниками описаного заводу.

Обґрунтування вибору шаблонів

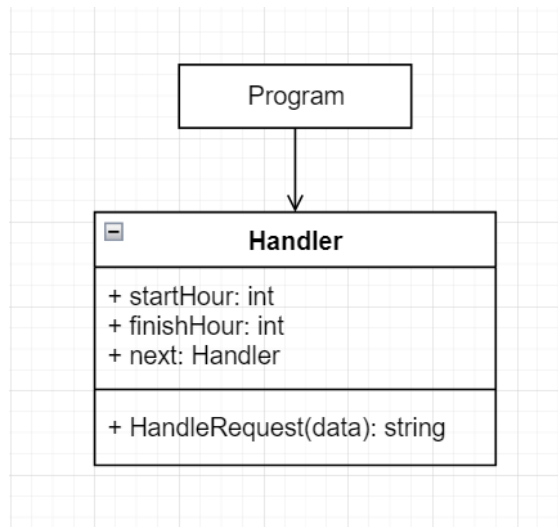
1. Для першого завдання був вибраний шаблон стратегія (Strategy), оскільки цей шаблон дозволяє виконати певну дію (переміщення у торговому центрі) декількома способами в залежності від певних умов (година дня, наявність профілактичних робіт та стан устаткування), що й було потрібно в завданні.
2. Для другого завдання був обраний шаблон ланцюжок обов'язків (Chain of Responsibilities), оскільки він дозволяє створити ланцюжок обробників задачі, які можуть як обробити певний запит самостійно, так і передати його наступному обробнику в залежності від певних умов (у завданні в залежності від години дня).

UML діаграми класів

Завдання 1



Завдання 2



Текст програми

Завдання 1

Program.cs

```
using System;
using System.Collections.Generic;

class Program
{
    static void Main()
    {
        Context context = new Context();

        if(DateTime.Now.Hour >= 22 || DateTime.Now.Hour < 10)
        {
            context.AddStrategy(new Stairs());
        }
        else if(GetProphylacticWorksPresence())
        {
            context.AddStrategy(new Stairs());
            context.AddStrategy(new Elevator());
            context.AddStrategy(new Escalator());
        }
        else
        {
            int elevatorState = GetEquipmentState("elevator");
            int escalatorState = GetEquipmentState("escalator");
            if(elevatorState > escalatorState)
            {
                context.AddStrategy(new Stairs());
                context.AddStrategy(new Elevator());
            }
            else
            {
                context.AddStrategy(new Stairs());
                context.AddStrategy(new Escalator());
            }
        }

        context.ExecuteStrategy();
    }

    static bool GetProphylacticWorksPresence()
    {
        bool prophylacticWorks;
        while(true)
        {
            Console.WriteLine("Is there a prophylactic work today? Y/N");
            ConsoleKeyInfo key = Console.ReadKey();
            if(key.Key == ConsoleKey.Y)
            {
                prophylacticWorks = true;
                break;
            }
            if(key.Key == ConsoleKey.N)
            {
                prophylacticWorks = false;
                break;
            }
            Console.WriteLine("Press either Y or N");
        }
        Console.WriteLine();
        return prophylacticWorks;
    }
}
```

```

static int GetEquipmentState(string equipment)
{
    Console.WriteLine($"What state is {equipment} in?");
    string stateStr = Console.ReadLine();
    int state;
    if(int.TryParse(stateStr, out state) && state >= 0)
    {
        return state;
    }
    Console.WriteLine($"Enter integer positive value.");
    return GetEquipmentState(equipment);
}

abstract class Strategy
{
    public abstract void TransferBetweenFloors();
}

class Stairs : Strategy
{
    public override void TransferBetweenFloors()
    {
        Console.WriteLine("Customers transfer between floors by stairs.");
    }
}

class Elevator : Strategy
{
    public override void TransferBetweenFloors()
    {
        Console.WriteLine("Customers transfer between floors by elevator.");
    }
}

class Escalator : Strategy
{
    public override void TransferBetweenFloors()
    {
        Console.WriteLine("Customers transfer between floors by escalator.");
    }
}

class Context
{
    private List<Strategy> strategies;

    public Context()
    {
        this.strategies = new List<Strategy>();
    }

    public bool AddStrategy(Strategy strategy)
    {
        foreach(Strategy str in this.strategies)
        {
            if(str == strategy)
            {
                return false;
            }
        }
        this.strategies.Add(strategy);
        return true;
    }

    public bool RemoveStrategy(Strategy strategy)
    {
        return this.strategies.Remove(strategy);
    }

    public void ExecuteStrategy()

```

```

    {
        foreach (Strategy strategy in this.strategies)
        {
            strategy.TransferBetweenFloors();
        }
    }
}

```

Завдання 2

Program.cs

```

using System;

class Program
{
    static void Main()
    {
        Handler thirdShift = new Handler(22, 7, null);
        Handler secondShift = new Handler(14, 23, thirdShift);
        Handler firstShift = new Handler(6, 15, secondShift);
        thirdShift.NextHandler = firstShift;

        string data = "request";
        string response;

        if (DateTime.Now.Hour >= 7 || DateTime.Now.Hour < 15)
        {
            response = firstShift.HandleRequest(data);
        }
        else if (DateTime.Now.Hour >= 15 || DateTime.Now.Hour < 23)
        {
            response = secondShift.HandleRequest(data);
        }
        else
        {
            response = thirdShift.HandleRequest(data);
        }

        Console.WriteLine(response);
    }
}

class Handler
{
    private Handler next;
    private int startHour;
    private int finishHour;

    public Handler NextHandler
    {
        set
        {
            next = value;
        }
    }

    public Handler(int startHour, int finishHour, Handler nextHandler)
    {
        this.startHour = startHour;
        if (startHour < 0 || startHour > 23)
        {

```

```

        this.startHour = 6;
    }
    this.finishHour = finishHour;
    if(finishHour < 0 || finishHour > 23)
    {
        this.finishHour = 15;
    }
    next = nextHandler;
}

public string HandleRequest(string data)
{
    if (DateTime.Now.Hour >= this.startHour && DateTime.Now.Hour < this.finishHour - 1)
        return $"Request handled by the shift that works from {this.startHour} till
{this.finishHour}.";
    else if (next != null)
    {
        return "Request was handed to the next shift.\r\n" + next.HandleRequest(data);
    }
    return "Next handler unspecified. Request not handled.";
}
}

```


Приклади результатів

Завдання 1

```
PS D:\KPI\SEMESTER 3\Progbase\lab5\task5> dotnet run
Is there a prophylactic work today? Y/N
y
Customers transfer between floors by stairs.
Customers transfer between floors by elevator.
Customers transfer between floors by escalator.
PS D:\KPI\SEMESTER 3\Progbase\lab5\task5> dotnet run
Is there a prophylactic work today? Y/N
n
What state is elevator in?
2
What state is escalator in?
5
Customers transfer between floors by stairs.
Customers transfer between floors by escalator.
```

```
PS D:\KPI\SEMESTER 3\Progbase\lab5\task5> dotnet run
Customers transfer between floors by stairs.
```

Завдання 2

```
PS D:\KPI\SEMESTER 3\Progbase\lab5\task6> dotnet run
Request handled by the shift that works from 6 till 15.
```

```
PS D:\KPI\SEMESTER 3\Progbase\lab5\task6> dotnet run
Request was handed to the next shift.
Request handled by the shift that works from 6 till 15.
```

Висновки

Під час лабораторної роботи я ознайомилася зі поведінковими шаблонами, дізналась про особливості їх використання. Навчилася відрізняти, де саме який шаблон потрібно використовувати. А також попрактикувалась у розробленні програмного забезпечення на мові С# з використанням шаблонів Стратегія та Ланцюжок обов'язків.