

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. Поведінкові

шаблони »

Варіант 14

Виконав студент ІП-13 Григоренко Родіон Ярославович
(шифр, прізвище, ім'я, по батькові)

Перевірила Ліщук Катерина Ігорівна
(прізвище, ім'я, по батькові)

Київ 202 3

Лабораторна робота 5

Шаблони проектування. Поведінкові шаблони

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

Постановка задачі:

При виконанні комп'ютерного практикуму необхідно виконати наступні дії:

- 1) Вивчити поведінкові патерни. Знати загальну характеристику та призначення кожного з них, особливості реалізації кожного з породжуючих патернів та випадки їх застосування.
- 2) Реалізувати задачу згідно варіанту, запропонованого нижче у вигляді консольного застосування на мові C#. Розробити інтерфейси та класи з застосування одного або декількох патернів. Повністю реалізувати методи, пов'язані з реалізацією обраного патерну.
- 3) Повністю описати архітектуру проекту (призначення методів та класів), особливості реалізації обраного патерну. Для кожного патерну необхідно вказати основні класи та їх призначення,
- 4) Навести UML-діаграму класів
- 5) Звіт повинен містити:
 - a. обґрунтування обраного патерну (чому саме він);
 - b. опис архітектури проекту (призначення методів та класів);
 - c. UML-діаграму класів
 - d. особливості реалізації обраного патерну
 - e. текст програмного коду
 - f. скріншоти результатів виконання.

Варіант 14

Проходження платежу через банківську систему супроводжується цілою низкою дій: фіксує, контролює, такі що знімають відсоток банку та інші вирахування і дії. Побудувати ланцюжки для різного виду платежів (звичайних, пільгових, державних, внутрішньобанківських) відповідно до предметної області та розробити модель системи.

1. Архітектура Проекту.

Виконуючи п'яту лабораторну роботу, було створено консольне застосування на мові C#, котре реалізує завдання. Було створено 8 класів, а саме:

Program - власне клас програми з виконуваною функцією Main

Operation - абстрактний клас операції

FixationOperation - клас, який відображає операцію фіксації платежу

ControllingOperation - клас, який відображає операцію контролю платежу

ChargeOperation - клас, який відображає операцію стягнення комісії з платежу

TransferOperation - клас, який відображає операцію переказу коштів

Payment - клас, який відображає платіж

PaymentHandler - клас, який відображає обробник платежу

При виконанні лабораторної роботи була проаналізована задача і виходячи з цього було прийняте рішення використати паттерн Ланцюжок обов'язків

3.Код класів:

Operation.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Lab5DotNet
{
    public abstract class Operation
    {
        protected Operation _nextOperation;

        public void SetNext(Operation operation)
        {
            _nextOperation = operation;
        }

        public abstract void ProcessPayment(Payment payment);
    }
}
```

```

public class FixationOperation : Operation
{
    public override void ProcessPayment(Payment payment)
    {
        Console.WriteLine($"\\nFixing {payment.Type} payment");

        _nextOperation.ProcessPayment(payment);

    }
}

```

```

public class ControllingOperation : Operation
{
    public override void ProcessPayment(Payment payment)
    {
        if(payment.Type == "government" || payment.Type == "intrabank")
        {
            Console.WriteLine("Government and IntraBank transactions are not
subject to control");
            _nextOperation.ProcessPayment(payment);
        }
        else
        {
            if(payment.Amount > 100000)
            {
                Console.WriteLine("The transaction limit has been exceeded!");
            }
            else

```

```

    {
        Console.WriteLine("The transaction has passed control");
        _nextOperation.ProcessPayment(payment);
    }
}

}
}

```

```

public class ChargeOperation : Operation
{
    public override void ProcessPayment(Payment payment)
    {
        if (payment.Type == "government" || payment.Type == "intrabank")
        {
            Console.WriteLine("No fees are charged for government and
IntraBank transactions");
            _nextOperation.ProcessPayment(payment);
        }
        else if(payment.Type == "preferential")
        {
            payment.Amount *= 0.99f;
            Console.WriteLine("1% fee is charged from Preferential
transaction");
            _nextOperation.ProcessPayment(payment);
        }
        else
        {

```

```

        payment.Amount *= 0.95f;
        Console.WriteLine("5% fee is charged from Regular transaction");
        _nextOperation.ProcessPayment(payment);
    }
}
}

```

```

public class TransferOperation : Operation
{
    public override void ProcessPayment(Payment payment)
    {
        Console.WriteLine("Transferring funds");
    }
}
}

```

Payment.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Lab5DotNet
{
    public class Payment
    {

```

```
public string Type;
public float Amount;

public Payment(string type, float amount)
{
    Type = type;
    Amount = amount;
}

}

public class PaymentHandler
{
    private Operation _fixationOperation = new FixationOperation();
    private Operation _controllingOperation = new ControllingOperation();
    private Operation _chargeOperation = new ChargeOperation();
    private Operation _transferOperation = new TransferOperation();
    public PaymentHandler()
    {
        _fixationOperation.SetNext(_controllingOperation);
        _controllingOperation.SetNext(_chargeOperation);
        _chargeOperation.SetNext(_transferOperation);
    }

    public void Handle(Payment payment)
    {
        _fixationOperation.ProcessPayment(payment);
    }
}
```



```
}
```

Program.cs

```
using System;
```

```
namespace Lab5DotNet
```

```
{
```

```
    class Program
```

```
    {
```

```
        static void Main(string[] args)
```

```
        {
```

```
            PaymentHandler paymentHandler = new PaymentHandler();
```

```
            paymentHandler.Handle(new Payment("regular", 10000));
```

```
            paymentHandler.Handle(new Payment("regular", 150000));
```

```
            paymentHandler.Handle(new Payment("government", 150000));
```

```
            paymentHandler.Handle(new Payment("intrabank", 70000));
```

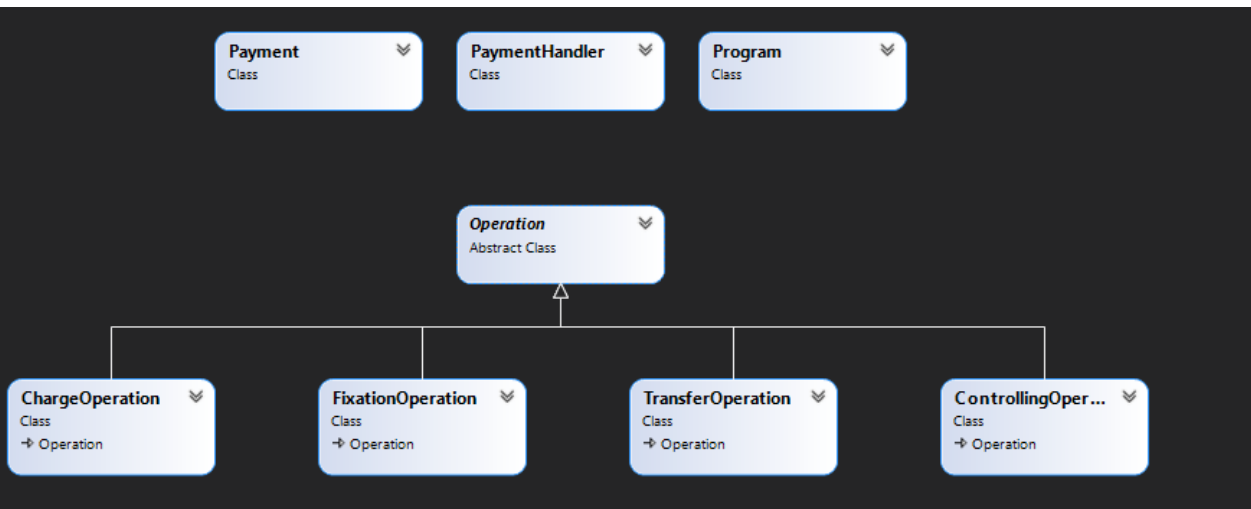
```
            paymentHandler.Handle(new Payment("preferential", 80000));
```

```
        }
```

```
    }
```

```
}
```

4.UML-діаграма класів



6.Результати виконання

В Program.cs був створений об'єкт класу PaymentHandler, після чого демонструється обробка різних платежів.

```
PaymentHandler paymentHandler = new PaymentHandler();
paymentHandler.Handle(new Payment("regular", 10000));
paymentHandler.Handle(new Payment("regular", 150000));
paymentHandler.Handle(new Payment("government", 150000));
paymentHandler.Handle(new Payment("intrabank", 70000));
paymentHandler.Handle(new Payment("preferential", 80000));
```

```
Fixing regular payment
The transaction has passed control
5% fee is charged from Regular transaction
Transferring funds

Fixing regular payment
The transaction limit has been exceeded!

Fixing government payment
Government and IntraBank transactions are not subject to control
No fees are charged for government and IntraBank transactions
Transferring funds

Fixing intrabank payment
Government and IntraBank transactions are not subject to control
No fees are charged for government and IntraBank transactions
Transferring funds

Fixing preferential payment
The transaction has passed control
1% fee is charged from Preferential transaction
Transferring funds
```

Висновок

Виконуючи дану лабораторну роботу, я покращив свої навички створення програмного забезпечення на мові C#, створив проект, використавши поведінковий паттерн Ланцюжок обов'язків, вивчив усі паттерни проектування, зрозумів як їх застосовувати в реальних проектах, та закріпив теоретичні знання на практиці.