



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 “Основи програмування”  
тема “Поведінкові шаблони проектування”

Виконав  
студент II курсу  
групи КП-01

Беліцький Олександр Сергійович  
*(прізвище, ім'я, по батькові)*

варіант № 3

Перевірів  
“ \_\_\_\_ ” “ \_\_\_\_ ” 20\_\_ р.  
викладач

Заболотня Тетяна Миколаївна  
*(прізвище, ім'я, по батькові)*

Київ 2021

## **Постановка завдання**

### **Завдання 1.**

Різні люди по-різному створюють свої фінансові накопичення. Одні люди купують валюту, інші - кладуть гроші на депозит в банку, треті – купують акції чи облігації. Забезпечити виконання функції «Відкласти гроші» відповідно до обраної людиною стратегії накопичення.

### **Завдання 2.**

допомогою шаблону проєктування змодельовати роботу викладачів, які перевіряють екзаменаційні роботи студентів з певного предмету. Кожна робота перевіряється тільки одним викладачем. Якщо перший викладач зайнятий перевіркою роботи, вона передається для перевірки другому і т.д.

## Аналіз вимог і проектування

### Завдання 1.

У даній задачі доцільно використати шаблон Стратегія.

Вибір впав саме на цей шаблон, адже він дозволяє визначати набір алгоритмів, інкапсулює кожен з них та забезпечує їх взаємозамінність. Залежно від ситуації ми можемо легко замінити один алгоритм, що використовується, іншим. Це якраз і потрібно при виборі вкладу грошей.

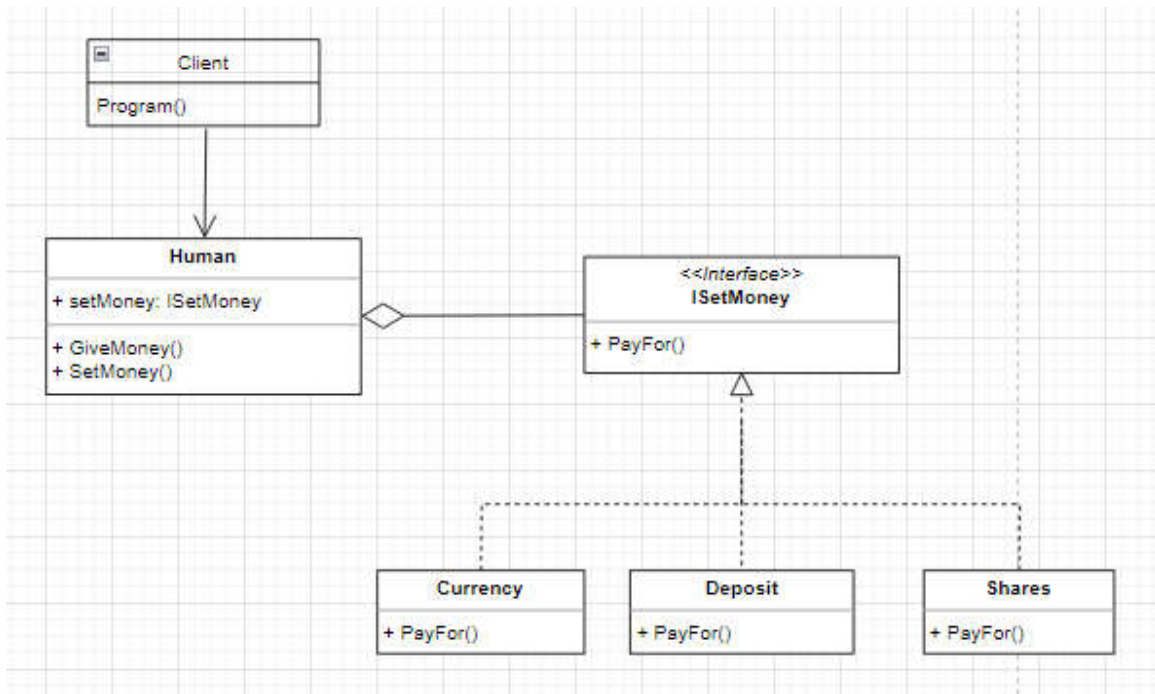


рис.1 UML-діаграма класів

## Завдання 2.

У даній задачі доцільно використати шаблон Ланцюг Обов'язків.

Паттерн Ланцюг Обов'язків дозволяє уникнути жорсткої прив'язки відправника запиту до одержувача. Всі можливі обробники запиту утворюють ланцюжок, а сам запит переміщається по цьому ланцюжку. Кожен об'єкт у цьому ланцюжку при отриманні запиту вибирає або закінчити обробку запиту або передати запит на обробку наступному по ланцюжку об'єкту.

В даній задачі нам потрібно мати ланцюг із вчителів, які будуть перевіряти роботи студентів по принципу "Вільний - перевіряю".

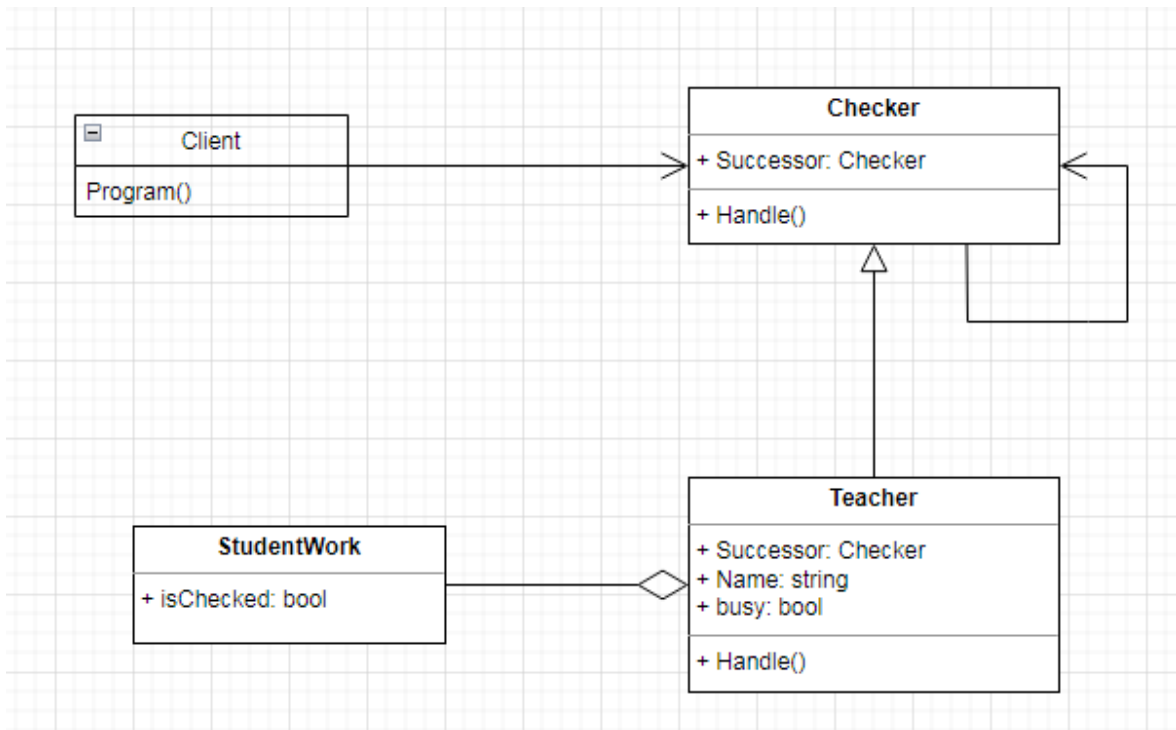


рис.2 UML-діаграма класів

## Тексти коду програм

### Завдання 1.

#### Program.cs

```
using System;

class Program
{
    static void Main(string[] args)
    {
        Human human = new Human(new Currency());
        human.GiveMoney();
        human.setMoney = new Bank();
        human.GiveMoney();
        human.setMoney = new Shares();
        human.GiveMoney();
    }
}

interface ISetMoney
{
    void PayFor();
}

class Currency : ISetMoney
{
    public void PayFor()
    {
        Console.WriteLine("Currency was purchased");
    }
}

class Bank : ISetMoney
{
    public void PayFor()
    {
        Console.WriteLine("Money was put on deposit");
    }
}

class Shares : ISetMoney
{
    public void PayFor()
    {
        Console.WriteLine("Shares was purchased");
    }
}

class Human
{
    public Human(ISetMoney setMoney)
    {
        this.setMoney = setMoney;
    }
    public ISetMoney setMoney { private get; set; }

    public void GiveMoney()
    {
        setMoney.PayFor();
    }
}
```

## Завдання 2.

### Program.cs

```
using System;

class Program
{
    static void Main(string[] args)
    {
        StudentWork studentWork = new StudentWork(false);

        Checker teacher1 = new Teacher("Gerd", true);
        Checker teacher2 = new Teacher("Asya", false);
        teacher1.Successor = teacher2;

        teacher1.Handle(studentWork);
    }
}

abstract class Checker
{
    public Checker Successor { get; set; }
    public abstract void Handle(StudentWork studentWork);
}

class Teacher : Checker
{
    private string _name;
    public bool busy { get; private set; }

    public Teacher(string name, bool busy)
    {
        this._name = name;
        this.busy = busy;
    }

    public override void Handle(StudentWork studentWork)
    {
        if (studentWork.isChecked == false && this.busy == false)
        {
            Console.WriteLine("Student work has been checked by " + this._name);
            studentWork.isChecked = true;
        }
        else if (Successor != null)
        {
            Console.WriteLine($"{this._name} is busy. Work has gone to next teacher");
            Successor.Handle(studentWork);
        }
    }
}

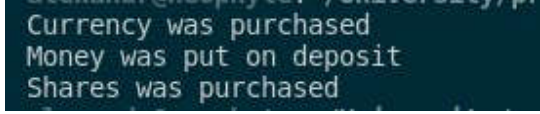
class StudentWork
{
    public bool isChecked;

    public StudentWork(bool isChecked)
    {
        this.isChecked = isChecked;
    }
}
```

## Приклади результатів

### Завдання 1.

Програма дозволяє людині обирати тип витрачання грошей:

<pre>Human human = new Human(new Currency()); human.GiveMoney(); human.setMoney = new Bank(); human.GiveMoney(); human.setMoney = new Shares(); human.GiveMoney();</pre>	
--	--

Людина, маючі гроші, хоче їх примножити і тому має функцію "віддати гроші", а програма уже обраним алгоритмом(купівля валюти, вклад в банк, купівля акцій) передає отримані гроші обраним варіантом.

## Завдання 2.

Приклад роботи програми:

```
Gerd is busy. Work has gone to next teacher  
Student work has been checked by Asya
```

У нас є два вчителя, які перевіряють роботи. Наразі вчитель Герд занятий перевіркою роботи, тому "наступний" за ланцюгом вчитель Ася бере перевірку роботи на себе.

Програмно ми повинні організувати цей ланцюжок вчителів вказуючи "наслідника".

```
Checker teacher1 = new Teacher("Gerd", true);  
Checker teacher2 = new Teacher("Asya", false);  
teacher1.Successor = teacher2;
```

Це забезпечить нам можливість передачі роботи між вчителями.



## **Висновки**

Виконавши дану лабораторну роботу було проведено вивчення поведінкових шаблонів, їх особливостей, основних задач, принципів роботи та варіанти для вдалого використання.

Я визначив для себе, що поведінкові шаблони вирішують завдання ефективної та безпечної взаємодії між об'єктами програми.

Окремо детально було розглянуто патерни Стратегія та Ланцюг Обов'язків. Визначив, що їх можна охарактеризувати як:

- Стратегія - автомобілі з різними типами двигуна. Електромобіль, авто на бензині, дизелі - різні за паливом, але всі автомобілі.
- Ланцюг Обов'язків - похід в поліклініку. Ми йдемо до поліклініки, але при цьому точно не знаємо характеру захворювання. У цьому випадку ми йдемо до терапевта, а він, залежно від захворювання, вже може або сам лікувати, або відправити на лікування іншим спеціалізованим лікарем.

Компіляція всього коду відбувалася за допомогою утиліти dotnet.