

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 1. Об'єктно-орієнтоване програмування та шаблони проєктування”

тема “ C# .Net. Шаблони поведінки у мові C#”

|  |  |  |
| --- | --- | --- |
| Виконала  студентка IІ курсу  групи КП-02  Гудзіцька Дарина Сергіївна |  | Перевірив  “\_\_\_\_” “\_\_\_\_\_\_\_\_\_\_\_\_” 20\_\_\_ р.  викладач  Заболотня Тетяна Миколаївна  (*прізвище, ім’я, по батькові*) |

Київ 2021

**Мета роботи**

Ознайомитися з такими можливостями мови програмування C# як структурні шаблони поведінки, набути розуміння як і коли їх використовувати.

**Постановка 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.
2. За допомогою шаблону проєктування змоделювати 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. У разі, якщо були зайняті всі викладачі і екзаменаційна робота залишилась неперевіреною, вона надходить на перевірку повторно до тих пір, поки її не перевірить якийсь викладач.

**Обгрунтування вибору шаблону**

**Завдання 1. Strategy**

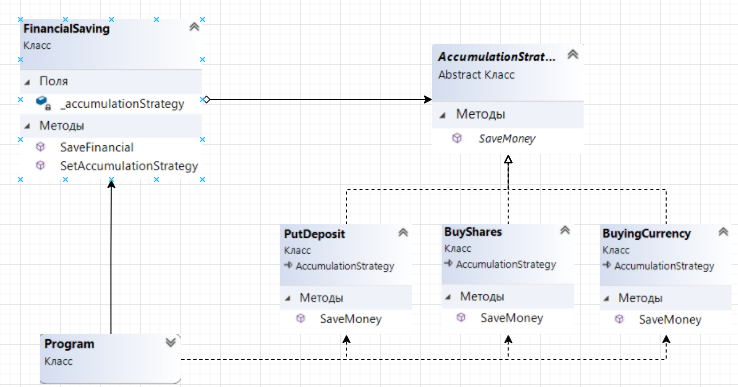
Для створення накопичень люди обирають різні методи(стратегії), але у них єдина ціль.Щоб основний клас не виконував основну функцію, яка обиралась за допомогою умов, можна створити декілька стратегій, на які основний клас буде посилатись. І щоб змінити алгоритм не потрібно сильно змінювати код, а лише підставити іншу стратегію.

**Завдання 2. Chain of responsibility**

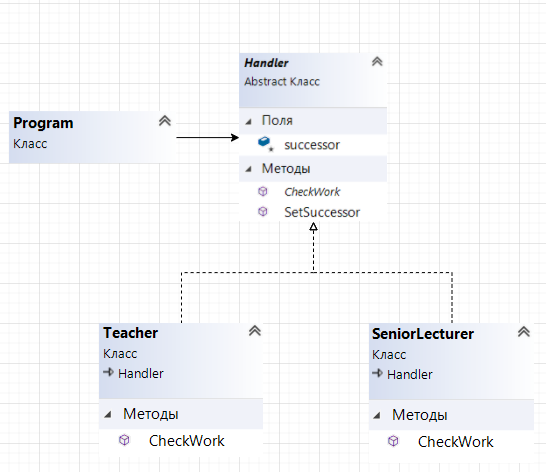
Потрібно зв’язати всіх викладачів у єдиний ланцюг, за посиланням на наступного викладача. Поточний викладач зможе, або перевірити роботи, або передати це наступному викладачеві.

**UML діаграма класів**

**Завдання 1.**



**Завдання 2.**



**Фрагменти коду**

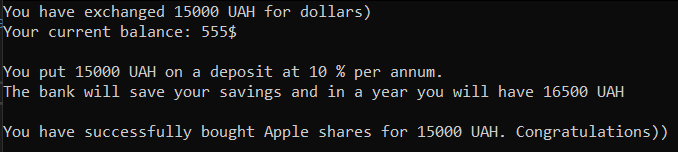
**Завдання 1.**

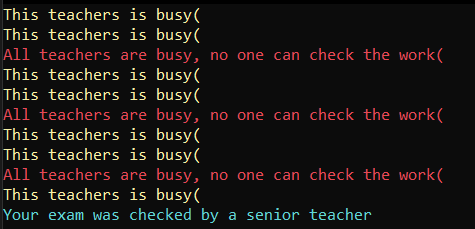
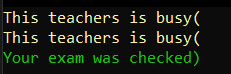
|  |
| --- |
| using static System.Console;  namespace task\_1  {  abstract class AccumulationStrategy  {  public abstract void SaveMoney(int amound);  }  class BuyingCurrency : AccumulationStrategy  {  public override void SaveMoney(int amound)  {  WriteLine($"You have exchanged {amound} UAH for dollars)\nYour current balance: {amound / 27 }$");  }  }  class PutDeposit : AccumulationStrategy  {  public override void SaveMoney(int amound)  {  WriteLine($"\nYou put {amound} UAH on a deposit at 10 % per annum.\n" +  $"The bank will save your savings and in a year you will have {amound + amound \* 0.1} UAH");  }  }  class BuyShares : AccumulationStrategy  {  public override void SaveMoney(int amound)  {  WriteLine($"\nYou have successfully bought Apple shares for {amound} UAH. Congratulations))");  }  }  class FinancialSaving  {  private AccumulationStrategy \_accumulationStrategy;  public void SetAccumulationStrategy(AccumulationStrategy accumulationStrategy)  {  this.\_accumulationStrategy = accumulationStrategy;  }  public void SaveFinancial(int num)  {  \_accumulationStrategy.SaveMoney(num);  }  }  } |

**Завдання 2.**

|  |
| --- |
| using System;  namespace task\_2  {  abstract class Handler  {  protected Handler successor;  public void SetSuccessor(Handler successor)  {  this.successor = successor;  }  public abstract void CheckWork();  }  class Teacher : Handler  {  public override void CheckWork()  {  if (new Random().Next(1, 10) % 3 == 0)  {  Console.ForegroundColor = ConsoleColor.Green;  Console.WriteLine("Your exam was checked)");  Console.ResetColor();  Environment.Exit(1);  }  else if (successor == null)  {  Console.ForegroundColor = ConsoleColor.Red;  Console.WriteLine("All teachers are busy, no one can check the work(");  return;  }  else if (successor != null)  {  Console.ForegroundColor = ConsoleColor.Yellow;  Console.WriteLine("This teachers is busy(");  successor.CheckWork();  }  }  }  class SeniorLecturer : Handler  {  public override void CheckWork()  {  Console.ForegroundColor = ConsoleColor.Cyan;  Console.WriteLine("Your exam was checked by a senior teacher");  Console.ResetColor();  }  }  } |

**Приклади результатів**

**Завдання 1.**

**Завдання 2.**

**Висновки**

Виконавши дану лабораторну роботу відбулося ознайомлення з такими можливостями мови програмування C# як шаблони поведінки, було набуте розуміння як і коли їх використовувати.