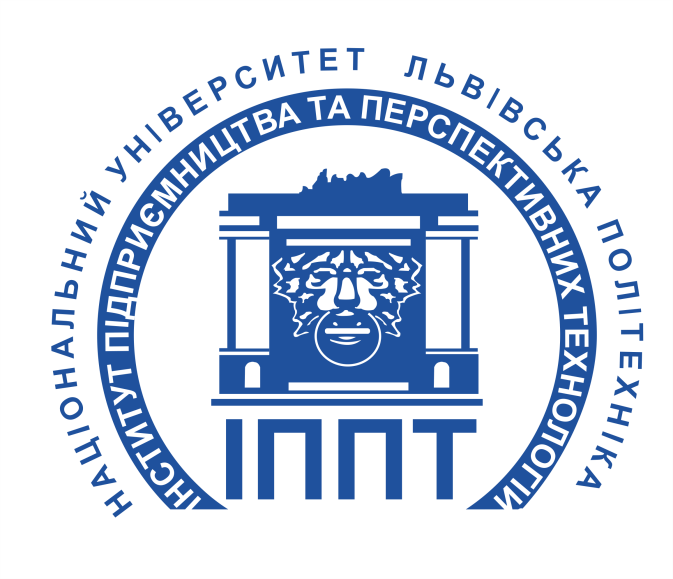
**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”**

**Н-Н ІНСТИТУТ ПІДПРИЄМНИЦТВА ТА ПЕРСПЕКТИВНИХ ТЕХНОЛОГІЙ**



**ЗВІТ**

до виконаної лабораторної роботи №6

на тему

***«*НАСЛІДУВАННЯ КЛАСІВ*»***

Виконав:

студент гр. КН-214

**Дубницький Ю.І.**

Прийняла:

**Машевська М.В.**

**Львів–2018**

**Лаболаторна робота №6**

**Завдання**

Реалізувати наслідування класів на прикладі задач опрацювання числових

послідовностей.

Базовим класом є числова послідовність. Похідними класами є: клас, що описує

арифметичну прогресію, або клас, що описує геометричну прогресію.

В базовому класі:

• оголосити захищений цілочисельний одновимірний масив;

• оголосити закриту цілочисельну змінну для збереження першого елемента

послідовності;

• оголосити закриту цілочисельну змінну для збереження змінної, що описує крок

послідовності чисел;

• розробити захищений конструктор, що приймає одне цілочисельне значення та

створює масив з відповідною кількістю елементів;

• розробити відкриту властивість для читання та запису першого елемента

послідовності;

• розробити відкриту віртуальну властивість для читання та запису змінної, що

описує крок послідовності чисел (задає змінній кроку значення 2);

• розробити метод, що повертає елементи масиву:

public int[] **return\_elements** ()

{…}

• оголосити абстрактні методи:

• метод для пошуку суми ( public abstract int **sum** (int k); )

• метод для пошуку *k*-го елемента ( public abstract int **k\_elem** (int k); ).

В похідному класі:

• розробити відкритий пустий конструктор з одним параметром, що посилається на

конструктор базового класу для створення масиву чисел;

• перевизначити (**override**) віртуальну властивість базового класу для задавання

різниці (арифметична прогресія) та знаменника (геометрична прогресія);

• реалізувати (використати **override**) абстракті методи базового класу: метод для

пошуку суми, метод для пошуку *k*-го елемента;

• розробити закритий метод для пошуку наступного елемента послідовності;

int **next**(int prev, int step)

{ … }

• розробити метод, що генерує масив *N* елементів послідовності (для формування

масиву використати метод **next**)

public void **array\_value**()

{ … }

В класі для роботи з формою створити екземпляр вашого похідного класу. Викликати

всі методи, результати вивести на елементи форми та в MessageBox.

**Код програми**

**Form1.cs**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace lab6c

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void button2\_Click(object sender, EventArgs e)

{

Class\_arifm arr = new Class\_arifm(Convert.ToInt32(numericUpDown3.Value))

{

First = Convert.ToInt32(numericUpDown1.Value),

Step = Convert.ToInt32(numericUpDown2.Value)

};

string str = "\t";

arr.array\_value();

foreach (int x in arr.return\_elements())

str += "" + x + "\t";

textBox1.Text = "елементи геометричної прогресії : " + str;

}

private void button1\_Click\_1(object sender, EventArgs e)

{

Class\_arifm arr = new Class\_arifm(Convert.ToInt32(numericUpDown3.Value))

{

First = Convert.ToInt32(numericUpDown1.Value),

Step = Convert.ToInt32(numericUpDown2.Value)

};

arr.array\_value();

int n = Convert.ToInt32(numericUpDown4.Value);

MessageBox.Show("значення " + n + "-го елемента = " + arr.n\_elem(n));

}

private void button3\_Click\_1(object sender, EventArgs e)

{

Class\_arifm arr = new Class\_arifm(Convert.ToInt32(numericUpDown3.Value))

{

First = Convert.ToInt32(numericUpDown1.Value),

Step = Convert.ToInt32(numericUpDown2.Value)

};

arr.array\_value();

MessageBox.Show("сума елементів прогресії = " + arr.summa());

}

}

abstract class Class\_base

{

protected int[] arr;

private int first;

private int step;

protected Class\_base(int size)

{

arr = new int[size];

}

public int First { get; set; }

public virtual int Step { get { return step; } set { step = value; } }

public int[] return\_elements()

{

return arr;

}

public abstract int summa();

public abstract int n\_elem(int n);

}

class Class\_arifm : Class\_base

{

public Class\_arifm(int size) : base(size) { }

public override int Step { get => base.Step; set => base.Step = value; }

public override int summa()

{

int sum = 0;

foreach (int x in arr)

sum += x;

return sum;

}

public override int n\_elem(int n)

{

n = First + Step \* (n - 1);

return n;

}

public int next(int prev, int step)

{

return prev + step;

}

public void array\_value()

{

arr[0] = First;

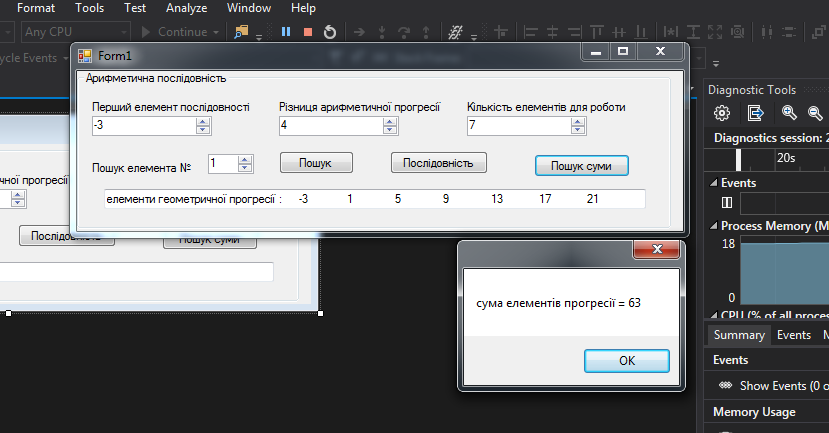
for (int i = 1; i < arr.Length; i++)

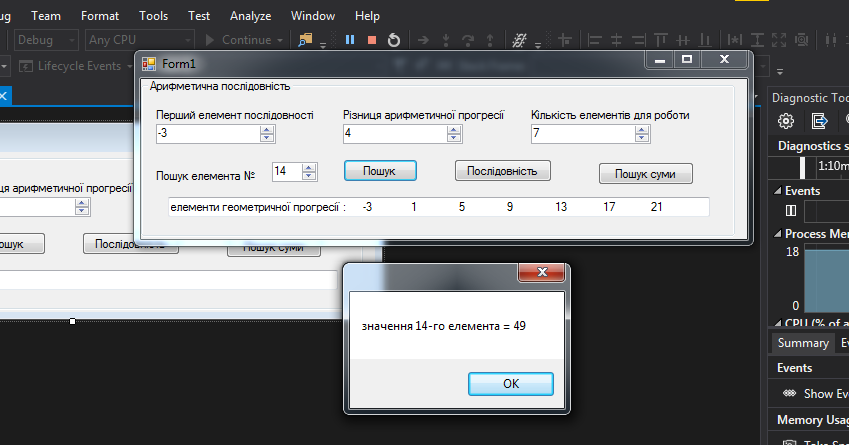
arr[i] = next(arr[i - 1], Step);

}

}

}



****

**Висновок:** У цій лаболаторній роботі я розробив програму в Form Application для перевантаження операторів для роботи з наслідуванням класів.