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 № 4 з дисципліни

«Основи програмування – 2. Об’єктно-орієнтоване програмування»

«Перевантаження операторів»

Варіант 21

Виконав студент ІП-11, Ляля Іван Олександрович

Перевірила Вітковська Ірина Іванівна\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Київ 2022

**Лабораторна робота 4**

**Перевантаження операторів**

**Мета** – вивчити механізми створення класів з використанням перевантажених операторів (операцій).

**Задача:**

Визначити клас “Поліном” розмірності 4, членами якого є коефіцієнти полінома. Реалізуватии для нього декілька конструкторів, геттери, метод обчислення значення поліному в заданій точці. Перевантажити оператори “+=”/”-=” – для збільшення/зменшення усіх коефіцієнтів полінома на вказану величину, “==” – для визначення рівності відповідних коефіцієнтів двох поліномів. Створити три полінома (Р1,Р2,Р3), використовуючи різні конструктори. Коефіцієнти полінома Р1 збільшити на вказану величину, а полінома Р2 – зменшити на цю ж саму величину. З’ясувати, чи рівні між собою відповідні коефіцієнти поліномів Р1 і Р2. Обчислити значення поліному Р3 в заданій точці.

**Постановка задачі:**

Опишемо клас Polynom, private атрибутами якого є size – ціле число, розмір полінома, та quots – покажчик на динамічний масив коефіцієнтів відповідного полінома. Опишемо public методи, а саме: конструктор за замовчуванням – створює поліном розмірності 4 з коефіцієнтами 0, конструктор з параметрами – створює поліном заданої розмірності з заданими коефіцієнтами та конструктор копіювання, що створює поліном-копію на основі уже існуючого. Опишемо деструктор класу, оскільки під масив коефіцієнтів в нас виділяється динамічна пам’ять; геттери для наших атрибутів; метод value() обчислення значення полінома в заданій точці та перевантажені відповідно до завдання оператори +=,-=,==.

В основній функції ми заповняємо масив коефіцієнтів полінома Р1 вводом від користувача, Р2 ініціалізуємо конструктором за замовчуванням нулями, а Р3 ініціалізуємо конструктором копіювання, передаючи йому атрибути Р1. Виводимо наші поліноми описаною нами функцією Polynom\_Output. Вводимо k – величину на яку будемо збільшувати коефіцієнти Р1 та зменшувати коефіцієнти Р2; проводимо ці операції,використовуючи перевантажені оператори +=/-=. Тепер перевіряємо, чи рівні відповідні коефіцієнти поліномів Р1 і Р2 перевантаженим оператором ==, виводимо результат перевірки на екран. Вводимо аргумент х, в якому будемо обчислювати значення полінома Р3 – обчислюємо описаним нами методом value() та виводимо на екран.

**Текст і результат роботи програми на С++:**

**cpp lab 4.cpp:**

#include "Header.h"

int main()

{

setlocale(LC\_ALL, "ukr");

const int n = 4;

cout << "Полiном вигляду C0 + C1\*(x^1) + C2\*(x^2) + C3\*(x^3)" << endl;

float quots[n]; //масив коефіцієнтів поліному 1

cout << "Коефiцiєнти для P1:"<<endl;

for (int i = 0; i < n; i++) //ініціалізація масиву коефіцієнтів для Р1

{

cout << "C" << i << " = ";

cin >> quots[i];

}

Polynom P1(quots,n); //виклик конструктора з параметрами для ініціалізації Р1

Polynom P2; //виклик конструктора за замовчуванням для ініціалізації Р2

Polynom P3(P1); //виклик конструктора копіювання для Р3

cout << "P1 = ";

Polynom\_Output(P1);

cout << "P2 = ";

Polynom\_Output(P2);

cout << "P3 = ";

Polynom\_Output(P3);

cout << "На яку величину будемо збiльшувати Р1 i зменшувати Р2? k = ";

float k;

cin >> k;

P1 += k;

P2 -= k;

cout << "P1 = ";

Polynom\_Output(P1);

cout << "P2 = ";

Polynom\_Output(P2);

if (P1 == P2) cout << "Полiноми Р1 i Р2 рiвнi\n";

else cout << "Полiноми Р1 i Р2 НЕ рiвнi\n";

cout << "Значення полiнома Р3 будемо обчислювати в точцi х = ";

float x; cin >> x;

cout << "P3(" << x << ") = " << P3.value(x) << endl;;

return 0;

}

**Header.h:**

#pragma once

#include <iostream>

#include "Polynom.h"

using namespace std;

void Polynom\_Output(Polynom); //функція виведення поліному

**Source.cpp:**

#include "Header.h"

void Polynom\_Output(Polynom P) //функція виведення поліному

{

float\* C = P.get\_quots();

int size = P.get\_size();

cout << C[0];

for (int i = 1; i < size; i++)

cout << " + " << C[i] << "\*(x^" << i << ')';

cout << '\n';

}

**Polynom.h:**

#pragma once

#include <iostream>

#include <math.h>

using namespace std;

class Polynom

{

int size; //розмірність полінома

float\* quots; //покажчик на масив коефіцієнтів полінома

public:

Polynom(); //конструктор за замовчуванням

Polynom(const Polynom& P); //конструктор копіювання

Polynom(float\* C, int n); //конструктор з параметрами

~Polynom(); //деструктор класу

float\* get\_quots() { return quots; } //геттер масиву коефіцієнтів полінома

int get\_size() { return size; } //геттер розмірності полінома

float value(float); //метод обчислення значення поліному в точці

const Polynom operator+=(const float);

const Polynom operator-=(const float);

bool operator==(Polynom);

};

**Polynom.cpp:**

#include "Polynom.h"

Polynom::Polynom() //конструктор за замовчуванням

{

size = 4;

quots = new float[size];

for (int i = 0; i < size; i++)

quots[i] = 0;

}

Polynom::Polynom(const Polynom& P) //конструктор копіювання

{

size = P.size;

quots = new float[P.size];

for (int i = 0; i < size; i++)

quots[i] = P.quots[i];

}

Polynom::Polynom(float\* C, int n) //конструктор з параметрами

{

size = n;

quots = new float[size];

for (int i = 0; i < size; i++)

quots[i] = C[i];

}

Polynom::~Polynom() //деструктор класу

{

delete[] quots;

}

float Polynom::value(float x) //метод обчислення значення поліному в точці

{

float val = quots[0];

for (int i = 1; i < size; i++)

{

val += quots[i] \* pow(x, i);

}

return val;

}

const Polynom Polynom::operator+=(const float k)

{

for (int i = 0; i < size; i++)

quots[i] += k;

return \*this;

}

const Polynom Polynom::operator-=(const float k)

{

for (int i = 0; i < size; i++)

quots[i] -= k;

return \*this;

}

bool Polynom::operator==(Polynom P)

{

bool flag = 1;

if (size != P.size) flag = 0;

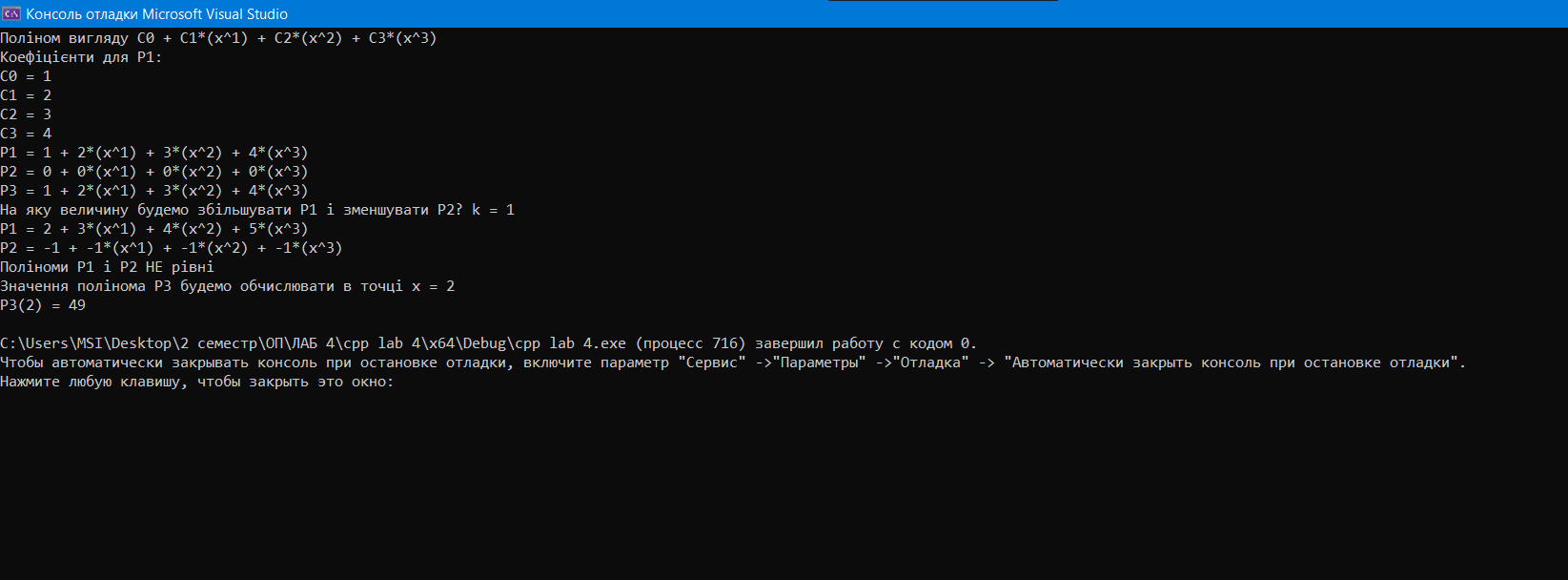
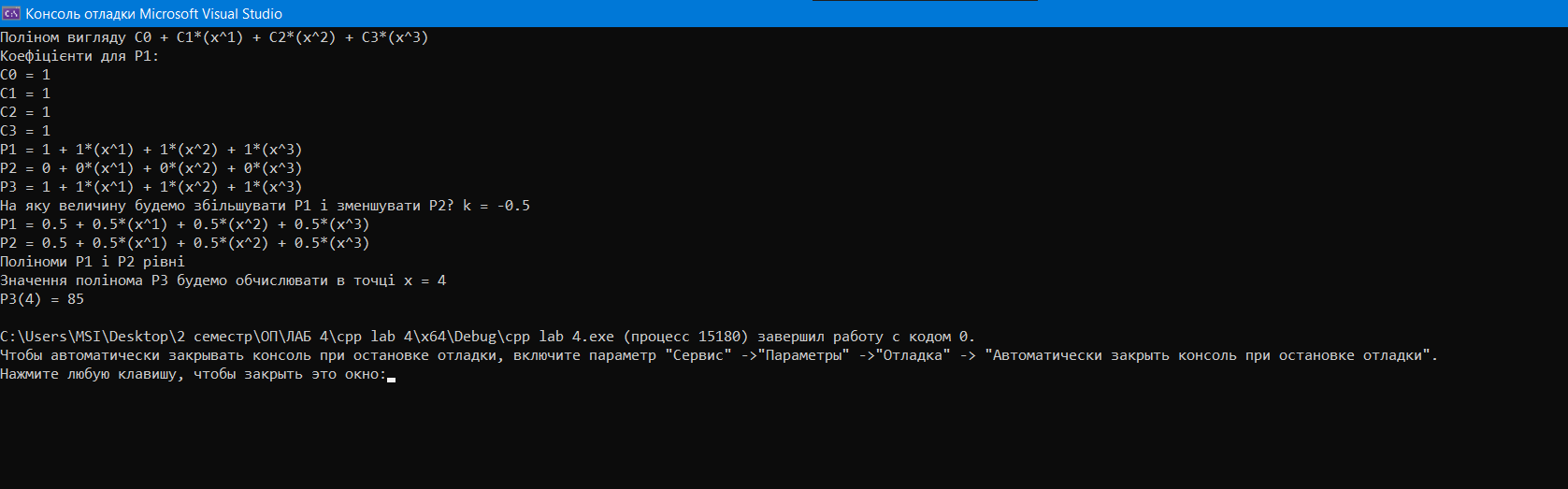
else

for (int i = 0; i < size; i++)

if (quots[i] != P.quots[i]) flag = 0;

return flag;

}

**** ****

**Висновок:**

Під час виконання лабораторної роботи я навчився створювати класи з використанням перевантажених операторів, а саме операторів +=,-=,==. Я закріпив навички роботи з різними видами конструкторів класу, деструкторами класу, гетерами і різноманітними методами. Особливістю роботи стало також використання покажчика на динамічний масив як атрибута класу.