

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.
Методології програмування»

«Перевантаження операторів»

Варіант 9

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

Перевірів Вечерковська Анастасія Сергіївна
(прізвище, ім'я, по батькові)

Лабораторна робота

4 Перевантаження операторів

9. Визначити клас "Багаточлен" ступеня 3, членами якого є коефіцієнти полінома. Реалізувати для нього декілька конструкторів, геттери, метод обчислення значення поліному в заданій точці. Перевантажити оператори додавання "+" і множення "*" поліномів. Створити три поліноми (P1, P2, P3), використовуючи різні конструктори. Визначити новий поліном P4 як суму поліномів P1 та P2 і новий поліном P5 як добуток поліномів P2 та P3. Обчислити значення поліномів P4 і P5 в заданій точці.

Варіант 9

Код програми

C++

Lab4_second_semestr.cpp

```
#include "Header.h"

int main()
{
    srand(time(NULL));
    vector<int> coefs = input_coefs();
    Polynom polynom1(coefs[0],coefs[1],coefs[2],coefs[3]);
    cout << "\nPolynom 1:\n";
    polynom1.show_polynom();

    int equal_coef;
    cout << "Enter 1 equal coef: ";
    cin >> equal_coef;
    Polynom polynom2(equal_coef);
    cout << "\nPolynom 2:\n";
    polynom2.show_polynom();

    Polynom polynom3;
    cout << "\nPolynom 3:\n";
    polynom3.show_polynom();

    Polynom polynom4 = polynom1 + polynom2;
    cout << "\nPolynom 4:\n";
    polynom4.show_polynom();

    Polynom polynom5 = polynom2 * polynom3;
    cout << "\nPolynom 5:\n";
    polynom5.show_polynom();

    int x;
    cout << "Enter x: ";
    cin >> x;
    cout << "Polynom 4 in point " << x << " equals: " << polynom4.find_result(x) << "\n";
    cout << "Polynom 5 in point " << x << " equals: " << polynom5.find_result(x);
}
```

Header.h

```
#pragma once
#include <iostream>
#include <stdlib.h>
#include <vector>
```

```

#include <string>
using namespace std;

class Polynom {
private:
int first_coef;
int second_coef;
int third_coef;
int fourth_coef;
public:
Polynom();
Polynom(int first, int second, int third, int fourth);
Polynom(int coef);
int get_first_coef();
int get_second_coef();
int get_third_coef();
int get_fourth_coef();
Polynom operator+(Polynom second_polynom);
Polynom operator*(Polynom second_polynom);
int find_result(int x);
void show_polynom();
};

vector<int> input_coefs();

vector<string> split(string str, char separator);

int pow(int value, int degree);

```

Sourse.cpp

#include "Header.h"

```
Polynom::Polynom() {  
    first_coef = rand() % 100;  
    second_coef = rand() % 100;  
    third_coef = rand() % 100;  
    fourth_coef = rand() % 100;  
}
```

```
Polynom::Polynom(int first, int second, int third, int fourth) {  
    first_coef = first;  
    second_coef = second;  
    third_coef = third;  
    fourth_coef = fourth;  
}
```

```
Polynom::Polynom(int coef) {  
    first_coef = coef;  
    second_coef = coef;  
    third_coef = coef;  
    fourth_coef = coef;  
}
```

```
int Polynom::get_first_coef() { return first_coef; }  
int Polynom::get_second_coef() { return second_coef; }  
int Polynom::get_third_coef() { return third_coef; }  
int Polynom::get_fourth_coef() { return fourth_coef; }
```

```
Polynom Polynom::operator+(Polynom second_polynom) {  
    first_coef += second_polynom.first_coef;  
    second_coef += second_polynom.second_coef;  
    third_coef += second_polynom.third_coef;  
    fourth_coef += second_polynom.fourth_coef;  
    return *this;  
}
```

```
Polynom Polynom::operator*(Polynom second_polynom) {  
    first_coef *= second_polynom.first_coef;
```

```

second_coef *= second_polynom.second_coef;
third_coef *= second_polynom.third_coef;
fourth_coef *= second_polynom.fourth_coef;
return *this;
}

```

```

int Polynom::find_result(int x) {
int res = pow(x, 3) * first_coef + pow(x, 2) * second_coef + pow(x, 1) * third_coef +
pow(x, 0) * fourth_coef;
return res;
}

```

```

int pow(int value, int degree) {
int res = 1;
for (size_t i = 0; i < degree; i++)
{
res *= value;
}
return res;
}

```

```

vector<int> input_coefs() {
cout << "Enter 4 coefitients separated wuth gaps\n";
string line;
getline(cin, line);
vector<int> coefs;
for (size_t i = 0; i < 4; i++)
{
coefs.push_back(stoi(split(line, ' ')[i]));
}

return coefs;
}

```

```

vector<string> split(string str, char separator) {
vector<string> res;
string slice = "";
str += " ";
for (int i = 0; i < str.length(); i++) {

```

```

if (str[i] == separator)
{
if (slice.length() > 0) {
res.push_back(slice);
slice = "";
}
}
else {
slice += str[i];
}
}
res.push_back(slice);
return res;
}

```

```

void Polynom::show_polynom() {
cout << "\n" << first_coef << "X^3 + " << second_coef << "X^2 + " << third_coef
<< "X + " << fourth_coef << "\n";
}

```

Тестування:

```
Enter 4 coefitients separated wuth gaps  
2 3 5 7
```

```
Polynom 1:
```

```
2X^3 + 3X^2 + 5X + 7
```

```
Enter 1 equal coef: 5
```

```
Polynom 2:
```

```
5X^3 + 5X^2 + 5X + 5
```

```
Polynom 3:
```

```
73X^3 + 37X^2 + 97X + 65
```

```
Polynom 4:
```

```
7X^3 + 8X^2 + 10X + 12
```

```
Polynom 5:
```

```
365X^3 + 185X^2 + 485X + 325
```

```
Enter x: 10
```

```
Polynom 4 in point 10 equals: 7912
```

```
Polynom 5 in point 10 equals: 388675
```

Висновки:

Я вивчив особливості перевантаження операторів та роботи з класами. Застосував ці навички на практиці.

