

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 № 3 з дисципліни  
«Основи програмування 2. Модульне програмування»

Перевантаження операторів

Варіант 9

Виконав студент ІП-14 Демченко Філіпп Ігорович  
(шифр, прізвище, ім'я, по батькові)

Перевірів Вітковська Ірина Іванівна  
(прізвище, ім'я, по батькові)

Київ 2022

# Лабораторна робота 3

## Перевантаження операторів

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

### Завдання

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.

### Код програми

Polynomial.h

```
C Polynomial.h Polynomial.h/...
#ifndef POLYNOMIAL_H
#define POLYNOMIAL_H

#include <cmath>

//ax^3 + bx^2 + cx + d - polynomial

class Polynomial {
    double *coefficients;
public:
    Polynomial(double a = 0, double b = 0, double c = 0, double d = 0);
    Polynomial(const Polynomial &ref);
    ~Polynomial();

    double getA() const { return coefficients[0]; }
    double getB() const { return coefficients[1]; }
    double getC() const { return coefficients[2]; }
    double getD() const { return coefficients[3]; }

    double getValueInPoint(double x) const
    { return coefficients[0] * pow(x, 3) + coefficients[1] * pow(x, 2) + coefficients[2] * x + coefficients[3]; }

    Polynomial operator+(const Polynomial &ref);
    Polynomial operator*(const Polynomial &ref);
    Polynomial operator=(const Polynomial &ref);

    void PrintPolynomial() const;
};

#endif
```

## Реалізація методів класу Polynomial.h

```
Polynomial.cpp Polynomial.cpp/...
#include "Polynomial.h"
#include <cstdio>

Polynomial::Polynomial(double a, double b, double c, double d)
{
    coefficients = new double[4];

    coefficients[0] = a;
    coefficients[1] = b;
    coefficients[2] = c;
    coefficients[3] = d;
}

Polynomial::Polynomial(const Polynomial &ref)
{
    coefficients = new double[4];

    for (int i = 0; i < 4; i++){
        coefficients[i] = ref.coefficients[i];
    }
}

Polynomial::~Polynomial()
{
    if (coefficients)
        delete [] coefficients;
}

Polynomial Polynomial::operator+(const Polynomial &ref)
{
    return Polynomial(coefficients[0] + ref.coefficients[0],
                      coefficients[1] + ref.coefficients[1],
                      coefficients[2] + ref.coefficients[2],
                      coefficients[3] + ref.coefficients[3]);
}
```

```
Polynomial Polynomial::operator*(const Polynomial &ref)
{
    return Polynomial(coefficients[0] * ref.coefficients[0],
                      coefficients[1] * ref.coefficients[1],
                      coefficients[2] * ref.coefficients[2],
                      coefficients[3] * ref.coefficients[3]);
}

Polynomial Polynomial::operator=(const Polynomial &ref)
{
    for (int i = 0; i < 4; i++){
        coefficients[i] = ref.coefficients[i];
    }

    return *this;
}

void Polynomial::PrintPolynomial() const
{
    printf("%.2fx^3 + %.2fx^2 + %.2fx + %.2f\n", this->getA(), this->getB(), this->getC(), this->getD());
}
```

## Головна програма. MainProgram.cpp

```

MainProgram.cpp MainProgram.cpp/...
#include "Polynomial.h"
#include <stdio>

int main()
{
    double a, b, c, d;
    double x;

    // Introduction
    printf("Polynomial has 4 coefficients:\n");
    printf("\tax^3 + bx^2 + cx + d\n\n");

    // Create first P1
    printf("Enter a for P1: "); scanf("%lf", &a);
    printf("Enter b for P1: "); scanf("%lf", &b);
    printf("Enter c for P1: "); scanf("%lf", &c);
    printf("Enter d for P1: "); scanf("%lf", &d);
    putchar('\n');

    Polynomial P1 = Polynomial(a, b, c, d);

    // Create P2
    printf("Enter a for P2: "); scanf("%lf", &a);
    printf("Enter b for P2: "); scanf("%lf", &b);
    printf("Enter c for P2: "); scanf("%lf", &c);
    putchar('\n');

    Polynomial P2 = Polynomial(a, b, c);

    // Create P3
    printf("Enter a for P3: "); scanf("%lf", &a);
    printf("Enter b for P3: "); scanf("%lf", &b);
    putchar('\n');

    Polynomial P3 = Polynomial(a, b);

    // Create P4
    Polynomial P4 = P1 + P2;
    Polynomial P5 = P2 * P3;

    printf("P4 = P1 + P2\n");
    printf("P4: "); P4.PrintPolynomial();
    putchar('\n');

    printf("P5 = P2 * P3\n");
    printf("P5: "); P5.PrintPolynomial();
    putchar('\n');

    printf("Enter some point on the x-axis: "); scanf("%lf", &x);
    putchar('\n');

    printf("Value P4 in point %.2f: %.2f\n\n", x, P4.getValueInPoint(x));
    printf("Value P5 in point %.2f: %.2f\n\n", x, P5.getValueInPoint(x));

    return 0;
}
```

## Виконання програми

```
Polynomial has 4 coefficients:  
    ax^3 + bx^2 + cx + d  
  
Enter a for P1: 1  
Enter b for P1: 2  
Enter c for P1: 3  
Enter d for P1: 4  
  
Enter a for P2: 5  
Enter b for P2: 6  
Enter c for P2: 7  
  
Enter a for P3: -4  
Enter b for P3: -3  
  
P4 = P1 + P2  
P4: 6.00x^3 + 8.00x^2 + 10.00x + 4.00  
  
P5 = P2 * P3  
P5: -20.00x^3 + -18.00x^2 + 0.00x + 0.00  
  
Enter some point on the x-axis: 2  
  
Value P4 in point 2.00: 104.00  
Value P5 in point 2.00: -232.00
```

**Висновок.** Під час виконання лабораторної роботи було вивчено механізми створення перевантажених операторів для класів користувача. Розглянуто різні варіанти створення: у вигляді методів класів, та у вигляді окремих функцій.