

Чумин Владимир Владимирович

КЭ-142

Лабораторная № 6

Задание:

Создать класс `Polynomial`, который представляет полином с использованием массива коэффициентов. Класс будет реализовывать сложение двух полиномов с помощью перегрузки оператора (+), а также метод для вычисления значения полинома при заданном значении  $x$ .

Реализовать класс полинома на C++ со следующими возможностями:

- ✓ Определить конструктор для инициализации полинома с заданной степенью.
- ✓ Реализовать метод для установки коэффициентов для каждого члена полинома.
- ✓ Перегрузить оператор + для выполнения сложения двух полиномов.
- ✓ Создать метод для вычисления значения полинома для заданного значения  $x$ .
- ✓ Отобразить полином в удобочитаемом формате.

Студентам необходимо написать полную реализацию класса `Polynomial`, включая конструктор, деструктор, методы для установки коэффициентов, перегрузку оператора +, вычисление полинома и отображение полинома.

Код:

```
#include <iostream>
#include <cmath>
using namespace std;

class Polynomial {
protected:
    int* coefficients = nullptr;
    int index;
```

```

    bool isInteractive = false;

public:
    Polynomial(bool isInteractive = false) :
isInteractive(isInteractive) {
        if (!isInteractive) {
            this->index = 1;
            this->coefficients = new int[1]{0};
        } else {
            int max_index;
            cout << "How many coefficients do you
want to enter? ";
            cin >> max_index;

            if (max_index <= 0) max_index = 1;

            int* newCoefficients = new
int[max_index];

            for (int i = 0; i < max_index; i++) {
                cout << "Enter coefficient #" << i
+ 1 << ": ";
                cin >> newCoefficients[i];
            }

            this->index = max_index;
            this->coefficients = newCoefficients;
        }
    }

    Polynomial(int* new_coefficients, int index) :
index(index) {
        this->coefficients = new int[index];
        for (int i = 0; i < index; i++) {

```

```

        this->coefficients[i] =
new_coefficients[i];
    }
}

Polynomial(int index) {
    this->index = max(1, index);
    this->coefficients = new
int[this->index]{0};
}

void printPolinomial() {
    std::string result = "";

    for (int i = 0; i < index; i++) {
        int coeff = coefficients[i];
        int power = index - i - 1;

        if (coeff == 0) continue;

        if (!result.empty()) {
            result += (coeff > 0 ? " + " : " -
");
        } else if (coeff < 0) {
            result += "-";
        }

        if (abs(coeff) != 1 || power == 0) {
            result +=
std::to_string(abs(coeff));
        }

        if (power > 0) {
            result += "x";
            if (power > 1) {

```

```

        result += "^" +
std::to_string(power);
    }
}

cout << (result.empty() ? "0" : result) <<
endl;
}

double getPolinomialSum() {
    int x;
    double sum = 0;

    cout << "Enter x: ";
    cin >> x;

    for (int i = 0; i < index; i++) {
        sum += coefficients[i] * pow(x, (index
- i - 1));
    }
    return sum;
}

Polynomial operator+(const Polynomial& other)
const {
    int maxIndex = max(1, max(this->index,
other.index));
    Polynomial Result(maxIndex);

    for (int i = 0; i < this->index; i++) {
        Result.coefficients[maxIndex -
this->index + i] += this->coefficients[i];
    }
    for (int i = 0; i < other.index; i++) {

```

```

        Result.coefficients[maxIndex -
other.index + i] += other.coefficients[i];
    }

    return Result;
}

Polynomial operator= (const Polynomial& other)
{
    if (this == &other) return *this;

    delete [] this -> coefficients;
    this -> index = other.index;
    this -> coefficients = new int[this ->
index];

    for (int i = 0; i < this -> index; i++) {
        this -> coefficients[i] =
other.coefficients[i];
    }
}

~Polynomial() {
    delete[] coefficients;
}

};

int main() {
    Polynomial firstPolynomial(true),
secondPolynomial(true);
    Polynomial sum = firstPolynomial +
secondPolynomial;
    sum.printPolinomial();
    return 0;
}

```

```
}
```

Результат:

```
vol@vol-All-Series:~/CLionProjects/C++_Labs$ g++ Lab6/lab_6.cpp && ./a.out
How many coefficients do you want to enter? 4
Enter coefficient #1: 3
Enter coefficient #2: 2
Enter coefficient #3: 1
Enter coefficient #4: 4
How many coefficients do you want to enter? 3
Enter coefficient #1: 4
Enter coefficient #2: 2
Enter coefficient #3: 3
3x^3 + 6x^2 + 3x + 7
```