



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ _____ «Информатика и системы управления»

КАФЕДРА _____ «Теоретическая информатика и компьютерные технологии»

Лабораторная работа № 7
по курсу «Языки и методы программирования»
«Разработка простейшего класса на C++»

Студент группы ИУ9-22Б Тараканов В. Д.

Преподаватель Посевин Д. П.

Москва 2024

1 Задание

Полином степени n с целочисленными коэффициентами и операциями: 1. вычисление значения для заданного x ; 2. получение степени полинома; 3. получение ссылки на указанный коэффициент полинома; 4. сложение с полиномом (для суммы формируется новый полином, степень которого может стать меньше степеней слагаемых).

2 Результаты

Листинг 1 — Объявление класса Polynomial

```
1
2
3     #ifndef POLYNOMIAL_H
4     #define POLYNOMIAL_H
5
6     class Polynomial {
7     public:
8
9         Polynomial();
10        Polynomial(int degree);
11        Polynomial(const Polynomial& other);
12
13
14        virtual ~Polynomial();
15
16
17        Polynomial& operator=(const Polynomial& other);
18
19        int evaluate(int x) const;
20        int getDegree() const;
21        int& coefficientAt(int power);
22
23
24        Polynomial operator+(const Polynomial& other) const;
25
26    protected:
27        int degree_;
28        int* coefficients_;
29    };
30
31 #endif
32
```

Листинг 2 — Реализация методов класса Polynomial

```
1
2     #include "declaration.h"
3     #include <cmath>
4
5
6     Polynomial::Polynomial() : degree_(0), coefficients_(nullptr) {}
7
8
9     Polynomial::Polynomial(int degree) : degree_(degree),
10    coefficients_(new int[degree + 1]) {
11        for (int i = 0; i <= degree_; ++i) {
12            coefficients_[i] = 0;
13        }
14    }
15
16    Polynomial::Polynomial(const Polynomial& other) : degree_(other.
17    degree_), coefficients_(new int[other.degree_ + 1]) {
18        for (int i = 0; i <= degree_; ++i) {
19            coefficients_[i] = other.coefficients_[i];
20        }
21    }
22
23    Polynomial::~~Polynomial() {
24        delete [] coefficients_;
25    }
26
27
28
```

Листинг 3 — Реализация методов класса Polynomial(продолжение)

```
1
2     Polynomial& Polynomial::operator=(const Polynomial& other) {
3         if (this != &other) {
4             delete[] coefficients_;
5             degree_ = other.degree_;
6             coefficients_ = new int[degree_ + 1];
7             for (int i = 0; i <= degree_; ++i) {
8                 coefficients_[i] = other.coefficients_[i];
9             }
10        }
11        return *this;
12    }
13
14    int Polynomial::evaluate(int x) const {
15        int result = 0;
16        for (int i = 0; i <= degree_; ++i) {
17            result += coefficients_[i] * pow(x, i);
18        }
19        return result;
20    }
21
22
23    int Polynomial::getDegree() const {
24        return degree_;
25    }
26
27
28    int& Polynomial::coefficientAt(int power) {
29        return coefficients_[power];
30    }
31
32    Polynomial Polynomial::operator+(const Polynomial& other) const {
33        int maxDegree = std::max(degree_, other.degree_);
34        Polynomial result(maxDegree);
35        for (int i = 0; i <= maxDegree; ++i) {
36            int coeff1 = (i <= degree_) ? coefficients_[i] : 0;
37            int coeff2 = (i <= other.degree_) ? other.coefficients_[i] :
0;
38            result.coefficients_[i] = coeff1 + coeff2;
39        }
40        return result;
41    }
42
43
```

Листинг 4 — Функция main для проверки работы класса Polynomial

```
1      #include <iostream>
2      #include "declaration.h"
3
4      void testPolynomial(const Polynomial& poly) {
5          std::cout << "Testing Polynomial:" << std::endl;
6
7          int x = 2;
8          std::cout << "Value of polynomial at x = " << x << ": " << poly.
evaluate(x) << std::endl;
9
10
11         std::cout << "Degree of polynomial: " << poly.getDegree() << std
::endl;
12
13
14         Polynomial poly2(3);
15         poly2.coefficientAt(2) = 3;
16         poly2.coefficientAt(1) = -2;
17         poly2.coefficientAt(0) = 1;
18         Polynomial sum = poly + poly2;
19         std::cout << "Sum of two polynomials at x = " << x << ": " <<
sum.evaluate(x) << std::endl;
20     }
21
22     int main() {
23
24         Polynomial poly(2);
25         poly.coefficientAt(2) = 1;
26         poly.coefficientAt(1) = -3;
27         poly.coefficientAt(0) = 2;
28
29         testPolynomial(poly);
30         poly.coefficientAt(0) = 100;
31         testPolynomial(poly);
32
33         Polynomial poly3 = poly;
34         poly3.coefficientAt(1) = 20;
35         testPolynomial(poly3);
36
37         return 0;
38     }
39
```

Результат запуска представлен на рисунках 1.

```
Testing Polynomial:
Value of polynomial at x = 2: 0
Degree of polynomial: 2
Sum of two polynomials at x = 2: 9
Testing Polynomial:
Value of polynomial at x = 2: 98
Degree of polynomial: 2
Sum of two polynomials at x = 2: 107
Testing Polynomial:
Value of polynomial at x = 2: 144
Degree of polynomial: 2
Sum of two polynomials at x = 2: 153
```

Рис. 1 — Результат