МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение

высшего образования

**«Национальный исследовательский**

**Нижегородский государственный университет им. Н.И. Лобачевского»**

**(ННГУ)**

**Институт информационных технологий, математики и механики**

**Кафедра математического обеспечения и суперкомпьютерных технологий**

**Отчет по учебной практике**

**«Разработка системы для арифметических действий над многочленами нескольких переменных»**

**Выполнил:** студент группы 381706-1

Голованова Елена Александровна

\_\_\_\_\_\_\_\_\_\_\_ Подпись

**Научный руководитель:**

ассистент каф. МОСТ ИИТММ

Лебедев Илья Геннадьевич

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Подпись

Нижний Новгород

2018.

Оглавление

[1. Введение 4](#_Toc1864402)

[2. Постановка задачи 5](#_Toc1864403)

[3. Руководство пользователя 6](#_Toc1864404)

[4. Руководство программиста 7](#_Toc1864405)

[a. Описание структуры программы 7](#_Toc1864406)

[b. Описание структур данных 7](#_Toc1864407)

[c. Описание алгоритмов 9](#_Toc1864408)

[5. Заключение 12](#_Toc1864409)

[6. Список литературы 13](#_Toc1864410)

# Введение

**Многочлен** (или **полином**) от *n* переменных — это сумма одночленов или, строго, — конечная формальная сумма вида

\sum_I c_I x_1^{i_1}x_2^{i_2}\cdots x_n^{i_n}, где

I=(i_1,i_2,\dots,i_n) — набор из целых неотрицательных чисел, именуемый **мультииндексом**,

* — число, именуемое **коэффициент многочлена**, зависящее только от мультииндекса *I*.

В частности, многочлен от одной переменной есть конечная формальная сумма вида

c_0 + c_1x^1 + \dots + c_mx^m, где

* c_i — фиксированные коэффициенты,
* x — переменная.

С помощью многочлена выводятся понятия алгебраическое уравнение и алгебраическая функция.

Удобнее всего хранить полиномы на базе линейных списков.

***Линейный список*** — это структура данных, состоящая из элементов одного типа, связанных между собой последовательно посредством указателей. Каждый элемент списка имеет указатель на следующий элемент. Последний элемент списка указывает на [NULL](https://ru.wikipedia.org/wiki/NULL_(%D0%A1%D0%B8)). Элемент, на который нет указателя, является первым (головным) элементом списка. Здесь ссылка в каждом узле указывает на следующий узел в списке.

Линейный список, как структура данных, очень удобен как для хранения полиномов, так и для работы с ними. Узлами списка, относительно хранения полинома, являются мономы. Все мономы лексикографически упорядочены между собой по мультииндексу: моном с большим мультииндексом указывает на следующий за ним моном с меньшим индексом. Поэтому представление полинома в памяти компьютера единственно.

Цель данной лабораторной работы – разработать системы для арифметических действий над полиномами.

# Постановка задачи

Выполнение работы предполагает решение следующих задач:

1. Реализация класса TPolynom, при помощи вспомогательного класса TMonom.
2. Выполнение определенных функций, операторов и конструкторов в ранее упомянутых классах.
3. Создать класс TException для избежания ошибок.
4. Выполнение примера использования программы.
5. Реализация простых тестов на базе Google Test.

# Руководство пользователя

Работа пользователя происходит так:

1. В начале программы создаются несколько мономов.
2. Далее пользователь должен ввести количество элементов, которые будут лежать в конце списка.
3. С помощью них формируются два полинома.
4. Далее происходят различные арифметические операции.
5. На экран выводятся результат этих операций.

# 4. Руководство программиста

## Описание структуры программы

Для реализации лабораторной работы создается несколько модулей:

* Модуль PolynomialLib

Этот модуль содержит описание методов с реализаций вспомогательного класса TMonom и описание методов с реализацией класса TPolynom.

* Модуль Polynomial

Этот модуль содержит пример использования программы.

* Модуль PolynomialTest

Этот модуль содержит тесты, проверяющие работу каждого методов класса TPolynom и TMonom. Они разработаны с помощью использования Google C++ Testing Framework.

* Модуль Exception

Этот модуль содержит класс исключений.

## Описание структур данных

Класс TMonom – вспомогательный класс. Он является шаблонным.

1. В зоне protected определены четыре поля:

* TMonom\* next; - указатель на следующий моном
* int \*deg; - массив степеней;
* double coef; - коэффициент;
* int count; - количество переменных в мономе.

2. В зоне public:

* TMonom() - конструктор;
* TMonom(int \_count, int\* \_deg, double \_сoef) – конструктор с параметрами;
* TMonom(TMonom& mon) - конструктор копирования;
* virtual ~TMonom() - деструктор;
* void SetNext(TMonom\* \_next) - задать адрес следующего монома;
* void SetDeg(int\* \_deg) - задать массив степеней монома;
* void SetСoef(double \_сoef) - задать коэффициент при мономе;
* void SetCount(int \_count) - задать количество переменных в мономе;
* TMonom\* GetNext() - получить адрес следующего монома;
* int\* GetDeg() - получить массив степеней монома;
* double GetKoef() - получить коэффициент при мономе;
* int GetCount() - получить количество переменных в мономе;
* TMonom operator + (TMonom& mon);
* TMonom operator += (TMonom& mon);
* TMonom operator - (TMonom& mon);
* TMonom operator -= (TMonom& mon);
* TMonom operator \* (TMonom& mon);
* TMonom operator \*= (TMonom& mon);
* TMonom& operator = (TMonom& mon);
* bool operator == (TMonom& mon);
* bool operator > (TMonom& mon);
* bool operator<(TMonom& mon);
* friend istream& operator >> (istream& istr, TMonom& mon);
* friend ostream& operator << (ostream& ostr, TMonom& mon);

Класс TPolynom – класс шаблонный. Наследуется от класса TMonom.

1. В зоне protected определены три поля:

* TMonom \*begin - указатель на первый моном в полиноме;
* int count - количество переменных в каждом мономе;
* int size - количество мономов в полиноме;

1. В зоне public:

* TPolinom(int \_count = 3) - конструктор;
* TPolynom(TPolynom& pol) - конструктор копирования;
* ~TPolynom() – деструктор;
* int GetSize() - получить число мономов в полиноме;
* TMonom\* GetBegin() - получить адрес первого монома в полиноме;
* TPolynom operator + (TPolynom& pol);
* TPolynom& operator += (TMonom& mon);
* TPolynom operator - (TPolynom& pol);
* TPolynom& operator -= (TMonom& mon);
* TPolynom operator \* (TPolynom& pol);
* TPolynom& operator = (TPolynom& pol);
* bool operator == (TPolynom& pol);
* friend ostream& operator << (ostream& ostr, TPolynom& pol);

## Описание алгоритмов

**Сложение двух полиномов.**

Алгоритм сложения двух полиномов А и В изображен на блок-схеме расположенной ниже.

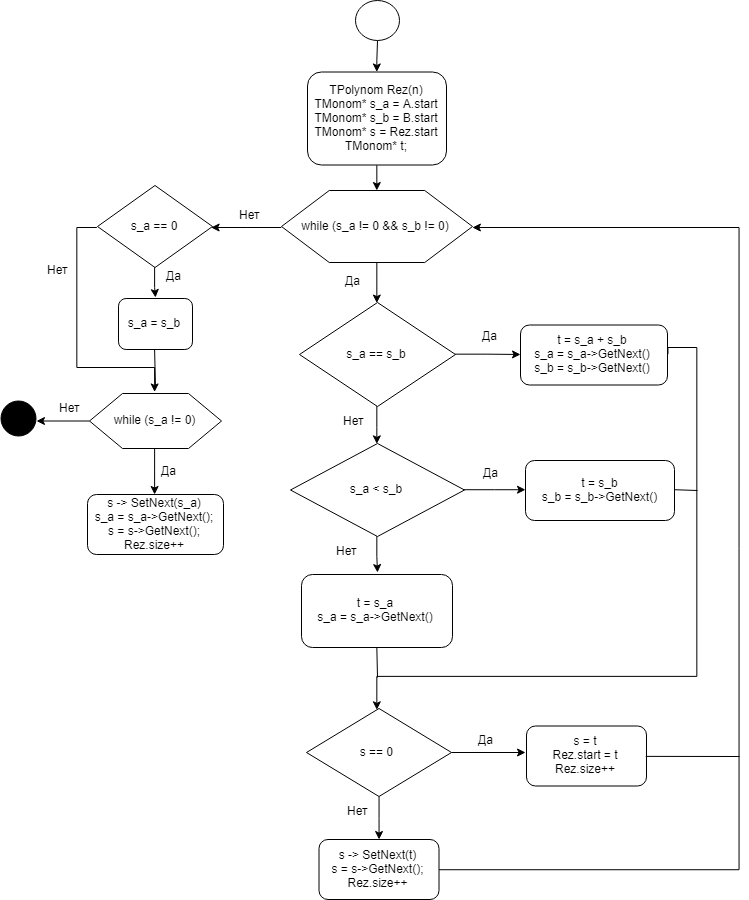


Рисунок 1. Алгоритм сложения двух полиномов

Rez – полином, полученный в результате сложения полиномов А и В, A.start и B.start – адреса первых мономов соответствующих полиномов.

**Сложение полинома и монома.**

Алгоритм сложение полинома и монома:

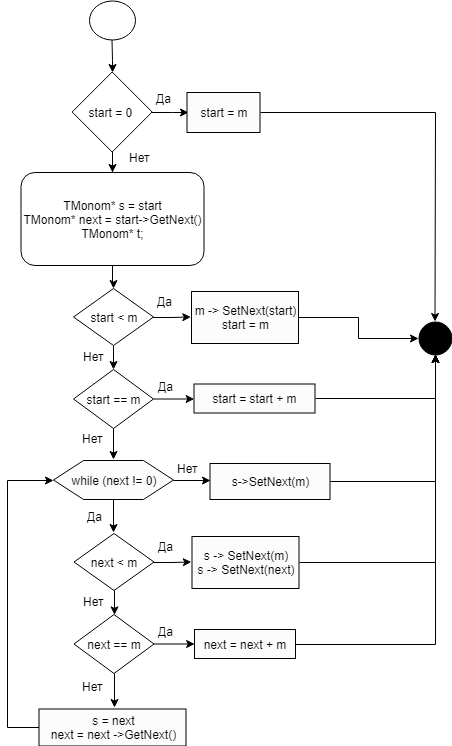


Рисунок 2. Алгоритм сложения полинома и монома

Start – адрес первого монома в полиноме, m – моном.

Алгоритмы нахождения разности и произведения схожи с данными.

# 5. Заключение

В ходе выполнения лабораторной была разработана система для арифметических действий над полиномами. Разработана библиотека PolynomialLib, реализующая классы полинома TPolynom и монома TMonom.

Получены новые знания и навыки.

Были реализованы тесты для проверки работоспособности вышеперечисленных классов на базе GoogleTest.

# 6. Список литературы

1. Васильев А.Н. Самоучитель С++ с примерами и задачами. -СПб.: Наука и Техника, 2016. -480с.
2. Гергель В.П. Методические материалы по курсу «Методы программирования 2», Нижний Новгород, 2015.
3. Википедия: свободная электронная энциклопедия: на русском языке: [https://ru.wikipedia.org/wiki/Многочлен]