МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение

высшего образования

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

**Национальный исследовательский университет**

**Институт информационных технологий, математики и механики**

**Кафедра математического обеспечения и суперкомпьютерных технологий**

**Отчет по учебной практике**

**«Система для проведения арифметических действий над многочленами нескольких переменных»**

**Выполнил:** студент группы 381706-1

Кольтюшкина Янина Вадимовна

\_\_\_\_\_\_\_\_\_\_\_ Подпись

**Научный руководитель:**

ассистент каф. МОСТ ИИТММ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Лебедев И.Г

Нижний Новгород

2018.

Содержание

[1. Введение 3](#_Toc1153162)

[2. Постановка задачи 4](#_Toc1153163)

[3. Руководство пользователя 5](#_Toc1153164)

[4. Руководство программиста 6](#_Toc1153165)

[4.1. Описание структуры программы 6](#_Toc1153166)

[4.2. Описание структур данных 6](#_Toc1153167)

[4.3. Описание алгоритмов 8](#_Toc1153168)

[5. Заключение 11](#_Toc1153169)

# Введение

**Целью** данной лабораторной работы является практическая реализация системы для выполнения арифметических операций над полиномами.

Ознакомимся со следующими базовыми понятиями.

**Mоном** – это произведение, состоящее из двух условных частей: постоянного числового множителя и одной или нескольких переменных, каждая из которых имеет неотрицательную целую степень.

Записать моном математически можно следующим образом:

|  |  |
| --- | --- |
|  |  |

, где – константа, – переменная,

**Полином** - это сумма мономов (конечная).

Записать полином математически можно:

|  |  |
| --- | --- |
|  |  |

, где – набор всевозможных целых неотрицательных чисел (так называемый мультииндекс), - число, (именуемое коэффициент многочлена) зависящее только от мультииндекса *I*.

В ходе реализации методов работы с полиномами будет использоваться такая структура данных, как линейный список. Это самая удобная структура данных для хранения и дальнейшей работы с полиномами. Более подробно про линейный список уже говорилось ранее, подчеркнем здесь наиболее важные для нас моменты. По своей сути, это структура данных, состоящая из элементов одного типа, связанных между собой последовательно посредством указателей. Каждый элемент списка имеет указатель на следующий элемент. Элемент, на который нет указателя, является первым элементом списка. Последний элемент списка указывает на NULL. Здесь ссылка в каждом узле указывает на следующий узел в списке.

Теперь о том, как все это относится к полиномам. Узлами нашего списка являются мономы, все из которых упорядочены лексикографически и связаны между собой по мультииндексу: моном с большим мультииндексом указывает на следующий за ним моном с меньшим индексом. Все это гарантирует единственность представления полинома в памяти компьютера.

# Постановка задачи

Одной из главных задач данной лабораторной работы является разработка эффективной системы для выполнения арифметических операций над полиномами.

Программное решение будет выглядеть следующим образом:

1. Разработка и реализация вспомогательного класса монома TMonom (по аналогии с узлом списка), а так же класса полинома TPolynom.
2. Пример программы, демонстрирующая работу класса TPolynom.
3. Написание набора автоматических тестов с использованием Google C++ Testing Framework и проверка работоспособности методов классов.
4. Класс для обработки исключений – MyException.

# Руководство пользователя

При запуске программы задаются трехпеременные мономы, в количестве шести штук. В дальнейшем на их основе будет сформированы два полинома. Все мономы выводятся на экран для проверки правильности дальнейших действий.

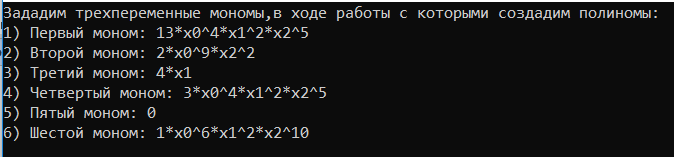


Рисунок 1 Создание шести мономов и вывод их на экран

Данные мономы формируются в два полинома и выводятся на экран. После чего, с полученными полиномами проводят базовые арифметические операции, а их результат также выводится на экран.

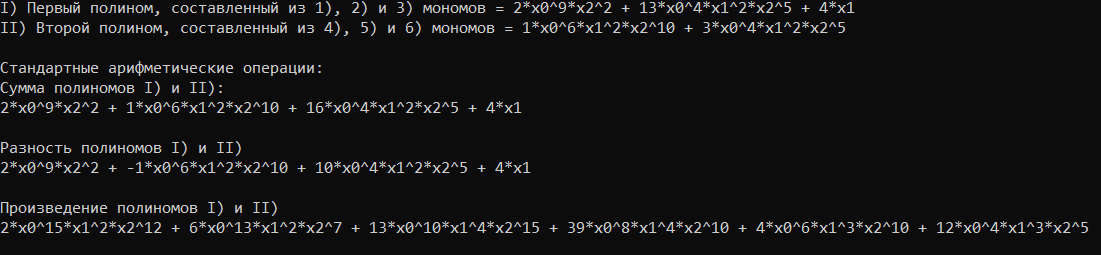


Рисунок 2 Создание полиномов и результат арифметических операций

Далее, полином, полученный умножением изначальных двух подвергают ряду дополнительных манипуляций. На экран выводится число мономов, входящих в его состав. Кроме того, выводится и первый из этих мономов для проверки корректности вывода. Но и на этом еще не конец. Этот полином сравнивается с одним из данных, присваивается к другому и снова выводится на экран.

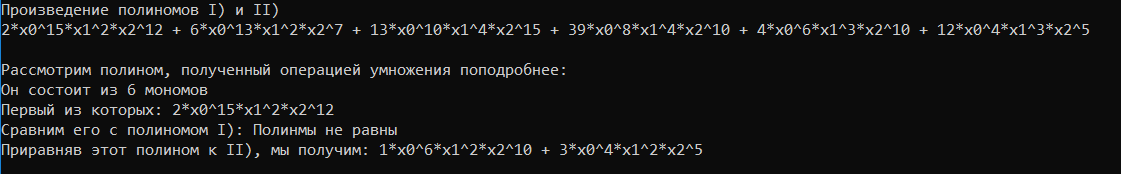


Рисунок 3 Длина полинома, его первый моном и операции сравнения, присваивания

# Руководство программиста

## Описание структуры программы

Программа состоит из следующих модулей:

* Модуль Polynomial. Содержит пример использования полинома. Реализация в файле Polynomial*\_main.cpp*
* Модуль PolynomialLib - статическая библиотека. Содержит файл Monom.h, в котором описан интерфейс и реализация вспомогательного класса *TMonom.* А так же файл Polynom.h, в котором описан интерфейс и реализация класса *TPolynom*.
* Модуль PolynomialTest. Содержит 41 тест, описанные в файле Polynomial*Test.cpp* и разработанные с помощью использования Google C++ Testing Framework.
* Модуль ExceptionLib – библиотека, позволяющая создавать собственные исключения.

## Описание структур данных

*Класс TMonom:*

Поля:

TMonom\* next; - указатель на следующий моном

int \*deg; - массив степеней

double coef; - коэффициент перед мономом

int countMon; - количество переменных в мономе

Конструкторы и деструктор:

TMonom();

TMonom(int \_count, int\* \_deg, double \_coef);

TMonom(TMonom& mon);

virtual ~TMonom();

Методы:

void SetNext(TMonom\* \_next); - ввести адрес следующего монома

void SetDeg(int\* \_deg); - ввести массив степеней монома

void SetCoef(double \_coef); - ввести коэффициент при мономе

void SetCountMon(int \_countMon); - ввести количество переменных в мономе

TMonom\* GetNext(); - получить адрес следующего монома

int\* GetDeg(); - получить массив степеней монома

double GetCoef(); - получить коэффициент при мономе

int GetCountMon(); - получить количество переменных в мономе

Перегрузки:

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 наследуется от класса TMonom:*

Поля:

TMonom \*start; - указатель на начальный моном в полиноме

int countVar; - количество переменных в каждом мономе

int leng; - количество мономов в полиноме

Конструкторы и деструктор:

TPolinom(int \_countVar = 3);

TPolynom(TPolynom& pol);

~TPolynom();

Методы:

int GetLeng(); - получить число мономов в полиноме

TMonom\* GetStart(); - получить адрес первого монома в полиноме

Перегрузки:

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

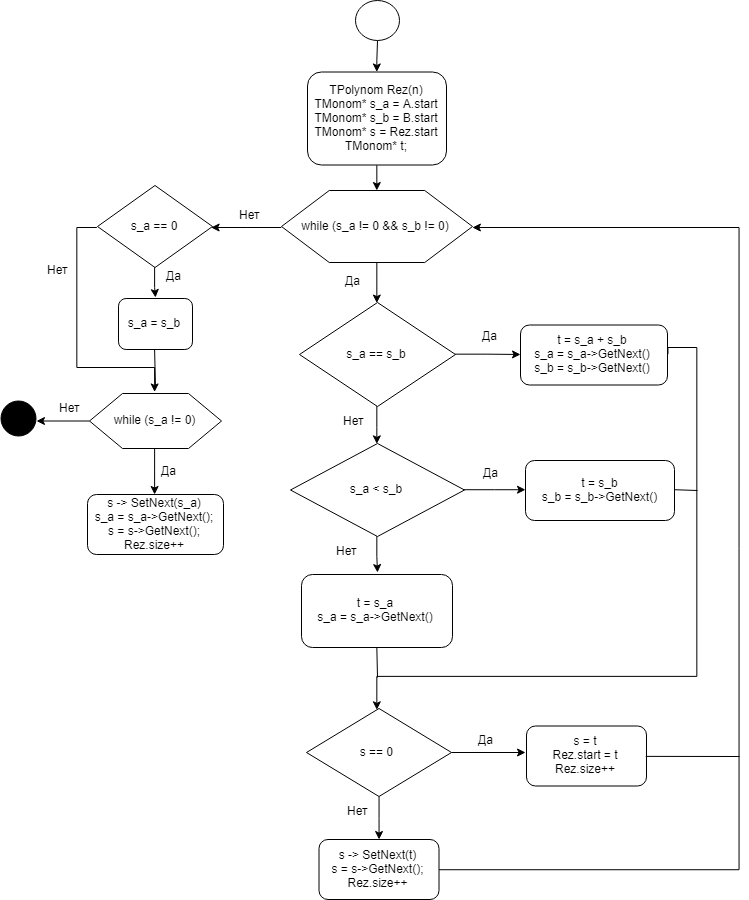


Рисунок 4 Алгоритм сложения двух полиномов

**Сложение полинома и монома.**

Алгоритм сложение полинома и монома:  
 Здесь, Start – адрес первого монома в полиноме, m – моном.

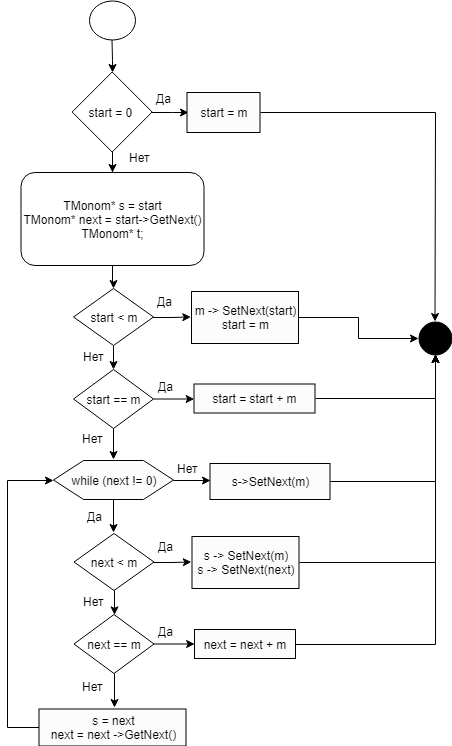


Рисунок 5 Алгоритм сложения полинома и монома

Алгоритмы для нахождения произведения и разности во многом схожи с данными.

# Заключение

В этой лабораторной работе я смогла справиться со всеми поставленными задачами. Разобравшись в том, что такое полином в общем и моном в частности и как проводить с ними арифметические операции, я реализовала класс TPolynom. Для этого была реализована библиотека PolynomialLib, в которой полностью описан класс полиномов. Также там описан и реализован и вспомогательный класс TMonom. Написанный мной набор автоматических тестов, проверяет работоспособность всех реализованных методов и делает это весьма успешно. Пример использования данных методов класса полином для пользователя полностью функционирует и готов к использованию.

1. **Литература**
2. Гергель В.П. Методические материалы по курсу «Методы программирования 2»: [http://www.itmm.unn.ru/files/2018/11/Primer-1.7.-Razrabotka-obshhego-predstavleniya-linejnogo-spiska.pdf], 2015.
3. Википедия: свободная электронная энциклопедия: на русском языке: https://ru.wikipedia.org/wiki/Многочлен