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

Колесова Кристина Юрьевна

\_\_\_\_\_\_\_\_\_\_\_ Подпись

**Научный руководитель:**

ассистент каф. МОСТ ИИТММ

Лебедев Илья Геннадьевич

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

Нижний Новгород

2018.

Оглавление

[1. Введение 3](#_Toc3580449)

[2. Постановка задачи 4](#_Toc3580450)

[3. Руководство пользователя 5](#_Toc3580451)

[4. Руководство программиста 6](#_Toc3580452)

[4.1. Описание структуры программы 6](#_Toc3580457)

[4.2. Описание структур данных 6](#_Toc3580458)

[4.3. Описание алгоритмов 8](#_Toc3580459)

[5. Заключение 10](#_Toc3580460)

[6. Список литературы 11](#_Toc3580461)

# Введение

Mоном - произведение, состоящее из числового множителя и одной или нескольких переменных, взятых каждая в неотрицательной целой степени. Математическая запись монома: , где – константа, – переменная,

Полином - конечная сумма мономов, имеющая следующий вид:  
, где – набор всевозможных целых неотрицательных чисел (мультииндекс), - число, (именуемое коэффициент многочлена) зависящее только от мультииндекса *I*.

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. 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 – разработка системы для арифметических действий над полиномами.

# Постановка задачи

Выполнение работы предполагает решение следующих задач:

* + 1. Разработка и реализация вспомогательного класса – TMonom – узел списка(моном).
    2. Разработка и реализация базового класса – TPolynom.
    3. Создание класса для обработки исключений – MyException, которые могут возникнуть при выполнении различных операций.
    4. Разработка программы, демонстрирующей работу классов TPolynom и TMonom.
    5. Реализация тестов на базе Google Test

# Руководство пользователя

При запуске программы пользователь сможет увидеть простейший пример работы с мономами и полиномами.

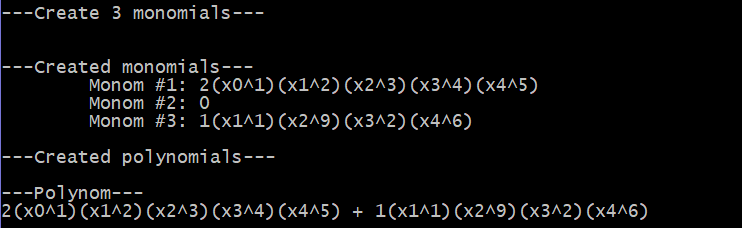
При запуске программа создает 3 монома, после чего выводит сообщение о том, что мономы созданы и выводит на консоль сами мономы. Далее из созданных мономов, путем арифметических операций, создается полином. Полученный полином выводится на консоль. На этом работа программы завершается.

Рисунок 1. Пример работы программы

# Руководство программиста



## Описание структуры программы

В программе содержатся следующие модули:

* MonomailLib – статическая библиотека. Содержит файлы Monomial.h и Monomial.cpp, содержащие описание и реализацию класса TMonomial.
* PolinomialLib - статическая библиотека. Содержит файлы Polinomial.h и Polinomial.cpp, содержащие описание и реализацию класса TPolinomial
* Модуль ExceptionLib – библиотека, позволяющая создавать собственные исключения.
* MonomialTest – содержит файл moly\_test.cpp с тестами для класса TMonomial.
* PolinomailTest - содержит файл poly\_test.cpp с тестами для класса TPolinomial.
* Polynomial – содержит файл main\_ poly.cpp, в котором реализован пример использования объектов описанных выше классов.

## Описание структур данных

* Класс TMonomial

Protected:

TMonom\* next - указатель на следующий моном;

int size - количество переменных в мономе;

int \*power - массив степеней переменных монома;

double coeff - коэффициент;

Public:

TMonomial() - конструктор по умолчанию;

TMonomial(int \_Size, int \*M, double \_Coeff) - конструктор инициализации;

TMonomial(TMonomial& A) - конструктор копирования;

virtual ~TMonomial() - деструктор;

void SetNext(TMonomial\* \_Next) - задать адрес следующего монома;

void SetPower(int \*M) - задать массив степеней переменных монома;

void SetC(double \_Coeff) - задать коэффициент монома;

void SetN(int \_Size) - задать количество переменных монома;

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

int\* GetPower() - получить массив степеней переменных монома;

double GetCoeff() - получить коэффициент монома;

int GetSize() - получить количество переменных в мономе;

Перегрузка арифметических операторов для мономов:

TMonomial &operator = (TMonomial &A)

TMonomial operator + (TMonomial &A)

TMonomial operator += (TMonomial &A)

TMonomial operator - (TMonomial &A)

TMonomial operator -= (TMonomial &A)

TMonomial operator \* (TMonomial &A)

TMonomial operator \*= (TMonomial &A)

Операторы сравнения для мономов:

bool operator == (TMonomial &A)

bool operator < (TMonomial &A)

bool operator > (TMonomial &A)

Перегрузга операторов ввода/вывода:

friend istream& operator>>(istream &in, TMonomial &m)

friend ostream& operator<<(ostream &out, TMonomial &m)

* Класс TPolynom

Protected:

TMonomial \*begin - указатель на первый моном в полиноме;

int n - количество переменных в каждом мономе;

int sizee - количество мономов в полиноме;

Public:

TPolynomial(int \_N = 10) - конструктор;

TPolynomial(TPolynomial &P) - конструктор копирования;

~TPolynomial() - деструктор;

int GetSize() - получить количество мономов в полиноме;

int GetN() - получить количество переменных каждого монома;

TMonomial\* GetStart() - получить адрес первого монома полинома;

Перегрузка операторов для работы с полиномами:

TPolinomial operator + (TPolinomial &A)

TPolinomial operator - (TPolinomial &A)

TPolinomial operator \* (TPolinomial &A)

TPolinomial &operator = (TPolinomial &A)

TPolinomial &operator += (TPolinomial &A)

TPolinomial &operator += (TMonomial &M)

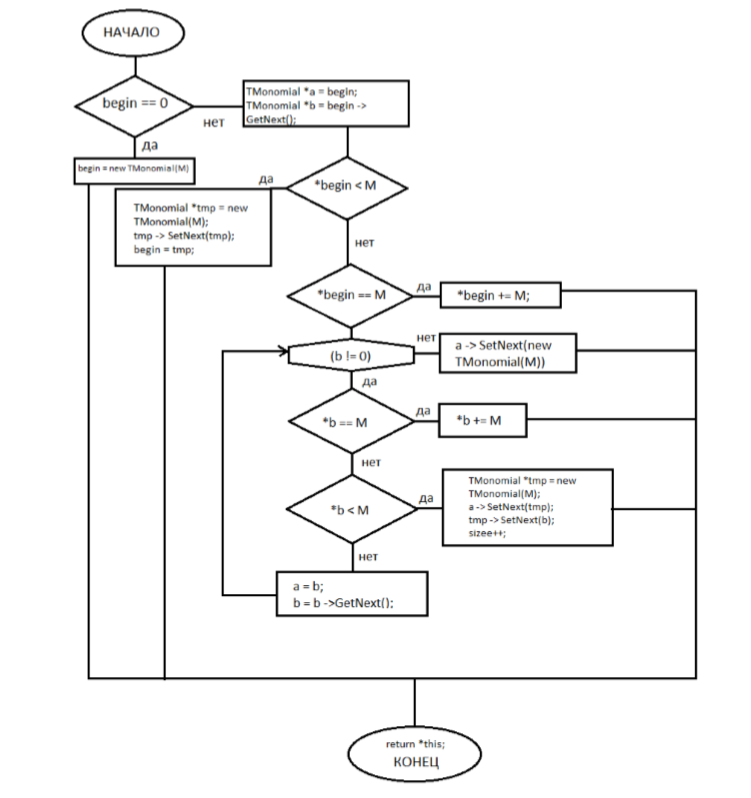
TPolinomial &operator -= (TMonomial &M)

bool operator==(TPolinomial &A)

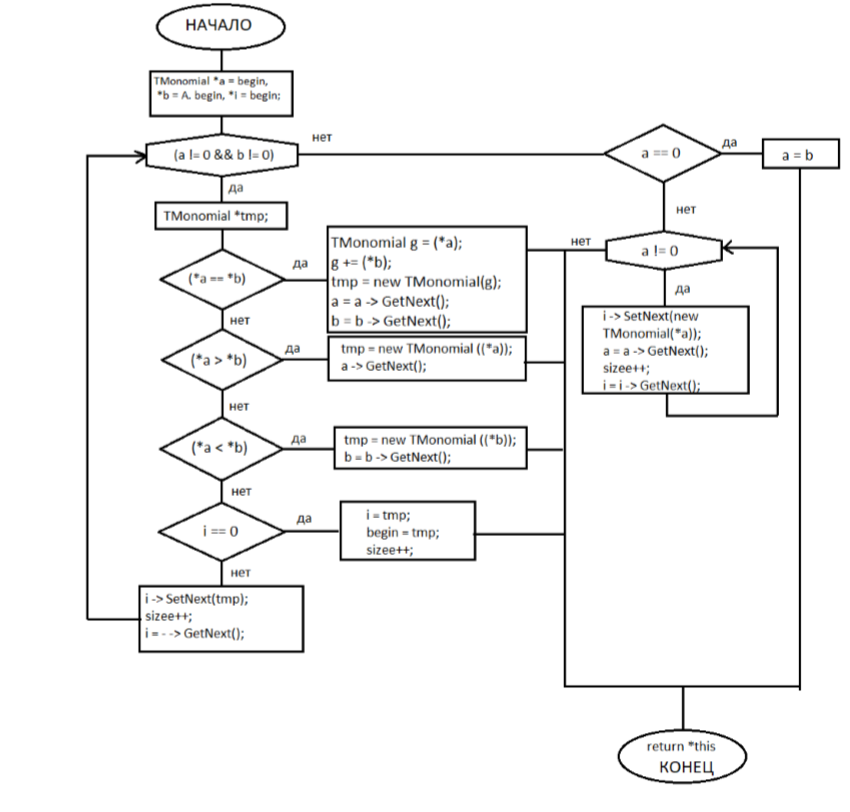
friend ostream& operator<<(ostream& out, TPolinomial& M) - вывод полинома.

## Описание алгоритмов

* Сумма полинома и монома



* Сумма двух полиномов



# Заключение

В результате выполнения лабораторной работы была разработана система поддержки арифметических действий над полиномами.

Успешно выполнены тесты, проверяющие работоспособность методов классов TMonomial и TPolynomial. Приведен пример работы программы.

# Список литературы

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