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**

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

**Выполнил:** студент группы 381706-1

Остапович Денис Евгеньевич

\_\_\_\_\_\_\_\_\_\_\_ Подпись

**Научный руководитель:**

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

Нижний Новгород

2018.

Содержание

[1. Введение 3](#_Toc2287544)

[2. Постановка задачи 4](#_Toc2287545)

[3. Руководство пользователя 5](#_Toc2287546)

[4. Руководство программиста 6](#_Toc2287547)

[4.1. Описание структуры программы 6](#_Toc2287548)

[4.2. Описание структур данных 6](#_Toc2287549)

[4.3. Описание алгоритмов 6](#_Toc2287550)

[5. Заключение 9](#_Toc2287551)

[6. Литература 10](#_Toc2287552)

# Введение

**Mоном** - произведение, состоящее из [числового](https://ru.wikipedia.org/wiki/%D0%A7%D0%B8%D1%81%D0%BB%D0%BE) [множителя](https://ru.wikipedia.org/wiki/%D0%9C%D0%BD%D0%BE%D0%B6%D0%B8%D1%82%D0%B5%D0%BB%D1%8C) и одной или нескольких переменных, взятых каждая в [неотрицательной](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D0%BB%D0%BE%D0%B6%D0%B8%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D0%BE%D0%B5_%D1%87%D0%B8%D1%81%D0%BB%D0%BE) [целой](https://ru.wikipedia.org/wiki/%D0%A6%D0%B5%D0%BB%D0%BE%D0%B5_%D1%87%D0%B8%D1%81%D0%BB%D0%BE) степени. Математическая запись монома:

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

, где – константа, – переменная,

**Полином** - конечная сумма мономов, вида

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

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

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

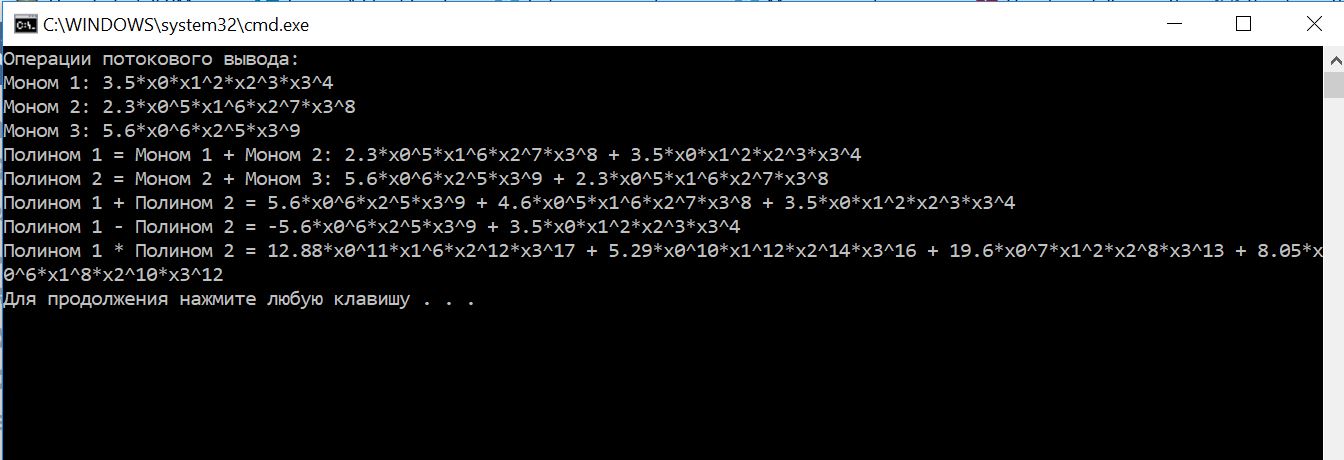
# Постановка задачи

В данной лабораторной работе нужно разработать эффективную библиотеку для работы с полиномами.

Для этого необходимо:

* Описать класс мономов **TMonomial**.
* Описать класс полиномов **TPolynomial**.
* Описать и реализовать функцию для вычисления выражения по ОПЗ.
* Протестировать методы классов **TMonomial** и **TPolynomial** с помощью Google Test.
* Продемонстрировать работу класса **TPolynomial**.

# Руководство пользователя

Программа работает следующим образом.

# Руководство программиста

## Описание структуры программы

Программа состоит из модулей:

1. **Polynomials** – содержит в себе файл **main.cpp** с реализацией примера использования класса **TPolynom**.
2. **PolynomialsLib –** содержит в себе файлы **monomial.h** и **polynomial.h**, в которых описаны и реализованы классы **TMonomial** и **TPolynomial** соответственно.
3. **PolynomialsTest** – содержит в себе файл **PolynomialTest.cpp**, в котором находится набор тестов, для проверки работоспособности класса **TPolynomial**.

## Описание структур данных

**Класс TMonomial.**

Класс **TMonomial** содержит четыре поля со спецификатором **protected**:

* **TMonomial** **\* next** – указатель на следующий моном.
* **int\* degree** – массив степеней монома.
* **int count** – количество переменных в мономе.
* **double coefficient** – коэффициент монома.

Далее идут методы класса со спецификатором доступа **public**:

* **TMonomial** **()** –конструктор по умолчанию.
* **TMonomial** **(int \_count, int\* \_degree, double \_coefficient)** –конструктор-инициализатор.
* **TMonomial** **(const TMonomial** **&A)** –конструктор копирования.
* **~ TMonomial** **()** –деструктор.
* **TMonomial** **\* GetNext()** –метод, возвращающий указатель на следующий моном.
* **int\* GetDegree()** - метод, возвращающий массив степеней монома.
* **double GetCoefficient()** -метод, возвращающий коэффициент монома.
* **int GetCount()** -метод, возвращающий количество переменных в мономе.
* **void SetNext(TMonom\* \_next)** -метод, устанавливающий указатель на следующий моном.
* **void SetDegree(int\* \_degree)** -метод, устанавливающий массив степеней монома.
* **void SetCoefficient(double \_cofficient)** -метод, устанавливающий коэффициент монома.
* **void SetCount(int \_count)** -метод, устанавливающий количество переменных в мономе.
* **TMonomial** **& operator=(const TMonomial** **&A)** –оператор присваивания для мономов.
* **TMonomial** **operator+( TMonomial** **&A)** –операторсложения мономов.
* **TMonomial** **operator\*(const TMonomial** **&A) const** –оператор умножения мономов.
* **TMonomial** **operator-( TMonomial** **&A)** –оператор разности мономов.
* **bool operator==( TMonomial** **&A)** –оператор проверки мономов на равенство.
* **bool operator>( TMonomial** **&A)** –оператор «больше» сравнения мономов.
* **bool operator<( TMonomial** **&A)** –оператор «меньше» сравнения мономов.
* **friend istream& operator>>(istream &istr, TMonomial** **&A)** –дружественная функция ввода монома.
* **friend ostream& operator<<(ostream &ostr, TMonomial** **&A)** -дружественная функция вывода монома.

**Класс TPolynomial.**

Класс **TPolynomial** содержит три поля со спецификатором **protected**:

* **TMonomial\* start** – указатель на стартовый моном в полиноме.
* **int size** – количество мономов в полиноме.
* **int count** – количество переменных в каждом мономе полинома.

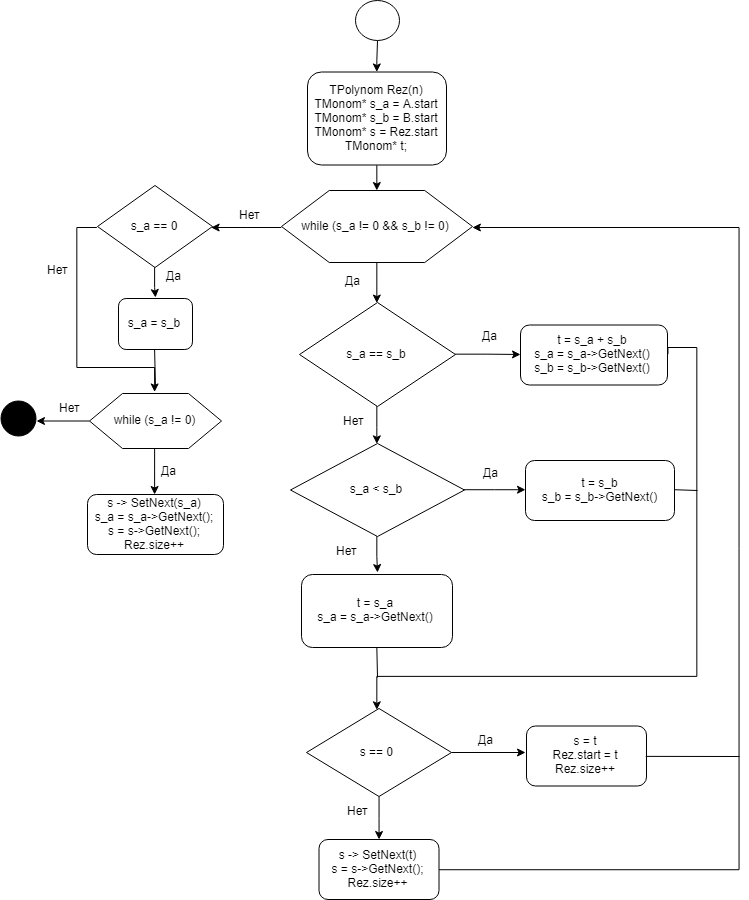
Далее идут методы класса со спецификатором доступа **public**:

* **TPolynomial** **(int \_count = 3)** – конструктор-инициализатор.
* **TPolynomial** **(TPolynom &A)** – конструктор копирования.
* **int GetSize()** – метод, возвращающий количество мономов в полиноме.
* **TMonomial\* GetStart()** – метод, возвращающий указатель на первый моном в полиноме.
* **TPolynomial** **operator-( TPolynomial** **&A)** – оператор разности полиномов.
* **TPolynomial** **operator+( TPolynomial** **&A)** – оператор сложения полиномов.
* **TPolynomial** **& operator=(const TPolynomial** **&A)** – оператор присваивания для полиномов.
* **bool operator==(const TPolynom &A)** – оператор проверки полиномов на равенство.
* **TPolynomial** **operator\*( TPolynomial** **&A)** – оператор умножения полиномов.
* **TPolynomial** **& operator+=( TMonomial &A)** – оператор добавления монома к полиному.
* **TPolynomial** **& operator-=( TMonomial &A)** – оператор удаления монома из полинома.
* **friend std::ostream& operator<<(std::ostream& ostr, TPolynomial** **& A)** – дружественная функция вывода полинома.

## Описание алгоритмов

**Сложение полиномов A и B.**

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

****

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

# Заключение

В ходе выполнения лабораторной была разработана система для арифметических действий над полиномами. Разработана статическая библиотека PolynomialsLib, реализующая классы полинома TPolynomial и монома TMonomial. Разработаны тесты для данного программного проекта с использованием Google C++ Testing Framework.

# Литература

* Книги
  + - 1. A.O. Грудзинский. Методы программирования, Издательство Нижегородского госуниверситета, 2006.
      2. Bjarne Stroustrup The C++ programming language
* Ссылки в Internet
  + - 1. 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.
      2. Википедия: https://ru.wikipedia.org/wiki/Многочлен
      3. Википедия: https://ru.wikipedia.org/wiki/Связный список