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

Шашкин Евгений Вадимович

\_\_\_\_\_\_\_\_\_\_\_ Подпись

**Научный руководитель:**

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

Нижний Новгород

2018.

**Оглавление**

[1. Введение. 2](#_Toc1329212)

[2. Постановка задачи. 3](#_Toc1329213)

[3. Руководство пользователя. 4](#_Toc1329214)

[4. Руководство программиста. 5](#_Toc1329215)

[4.1. Описание структуры программы. 5](#_Toc1329216)

[4.2. Описание структур данных. 5](#_Toc1329217)

[4.3. Описание алгоритмов. 7](#_Toc1329218)

[5. Заключение. 8](#_Toc1329219)

[6. Литература. 9](#_Toc1329220)

# Введение.

**Моном** — простое математическое выражение, прежде всего рассматриваемое и используемое в элементарной алгебре , а именно, произведение, состоящее из числового множителя и одной или нескольких переменных, взятых каждая в неотрицательной целой степени.

Мономом также считается каждое отдельное число (без буквенных множителей), причём степень такого одночлена равняется нулю.

При сложении одночленов моном можно получить только при одинаковых степенях слагаемых. При умножении перемножаются коэффициенты и складываются показатели степеней.

**Полином** от **n** переменных — это сумма мономов или, строго, — конечная формальная сумма вида, где

— набор из целых неотрицательных чисел, именуемый **мультииндексом**,

— число ,именуемое **коэффициент полинома**, зависящее только от мультииндекса.

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

,где

— фиксированные коэффициенты,

— переменная.

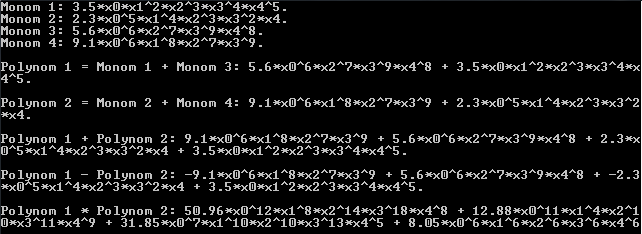
# Постановка задачи.

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

Для этого нам нужно:

* Описать и реализовать класс мономов **TMonom**.
* Описать и реализовать класс полиномов **TPolynom**.
* Описать и реализовать функцию для вычисления выражения по ОПЗ.
* Протестировать методы классов **TMonom** и **TPolynom** с помощью Google Test.
* Реализовать класс **TMyException** для обработки исключений, которые могут возникнуть в результате работы.
* Продемонстрировать работу класса **TPolynom**.

# Руководство пользователя.

****

*Рис 1. Пример работы программы*

Программа работает следующим образом:

* Создаются четыре монома и выводятся на экран.
* Из мономов формируются два полинома и выводятся на экран.
* Полиномы складываются и результат выводится на экран.
* Полиномы вычитаются и результат выводится на экран.
* Полиномы перемножаются и результат выводится на экран.

# Руководство программиста.

# Описание структуры программы.

Программа состоит из модулей:

* **Polynomial** – содержит в себе файл **polynomial\_main.cpp** с реализацией примера использования класса **TPolynom**.
* **PolynomialLib –** содержит в себе файлы **Monom.h** и **Polynom.h**, в которых описаны и реализованы классы **TMonom** и **TPolynom** соответственно.
* **Polynomial Test** – содержит в себе файл **test\_ polynomial.cpp**, в котором находится набор тестов, для проверки работоспособности класса **TPolynomial**.
* **MyExceptionLib** –содержитв себе файл **MyException.h** с реализацией класса исключений **TMyException**.

# Описание структур данных.

**Класс TMonom.**

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

* **TMonom\* 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**:

* **TMonom()** –конструктор по умолчанию.
* **TMonom(int \_count, int\* \_degree, double \_coefficient)** –конструктор-инициализатор.
* **TMonom(const TMonom &obj)** –конструктор копирования.
* **~TMonom()** –деструктор.
* **TMonom\* 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.
* **TMonom& operator=(const TMonom &obj)** –оператор присваивания для мономов.
* **TMonom operator+(TMonom &obj)** –операторсложения мономов.
* **TMonom operator\*(const TMonom &obj) const** –оператор умножения мономов.
* **TMonom operator-(TMonom &obj)** –оператор разности мономов.
* **bool operator==(TMonom &obj)** –оператор проверки мономов на равенство.
* **bool operator>(TMonom &obj)** –оператор «больше» сравнения мономов.
* **bool operator<(TMonom &obj)** –оператор «меньше» сравнения мономов.
* **friend istream& operator>>(istream &istr, TMonom &obj)** –дружественная функция ввода монома.
* **friend ostream& operator<<(ostream &ostr, TMonom &obj)** -дружественная функция вывода монома.

**Класс TPolynom.**

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

* **TMonom\* 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**:

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

**Класс TMyException.**

Класс **TMyException** содержит одно поле со спецификатором доступа **private**:

* **string str** – строка, хранящая сообщение об ошибке.

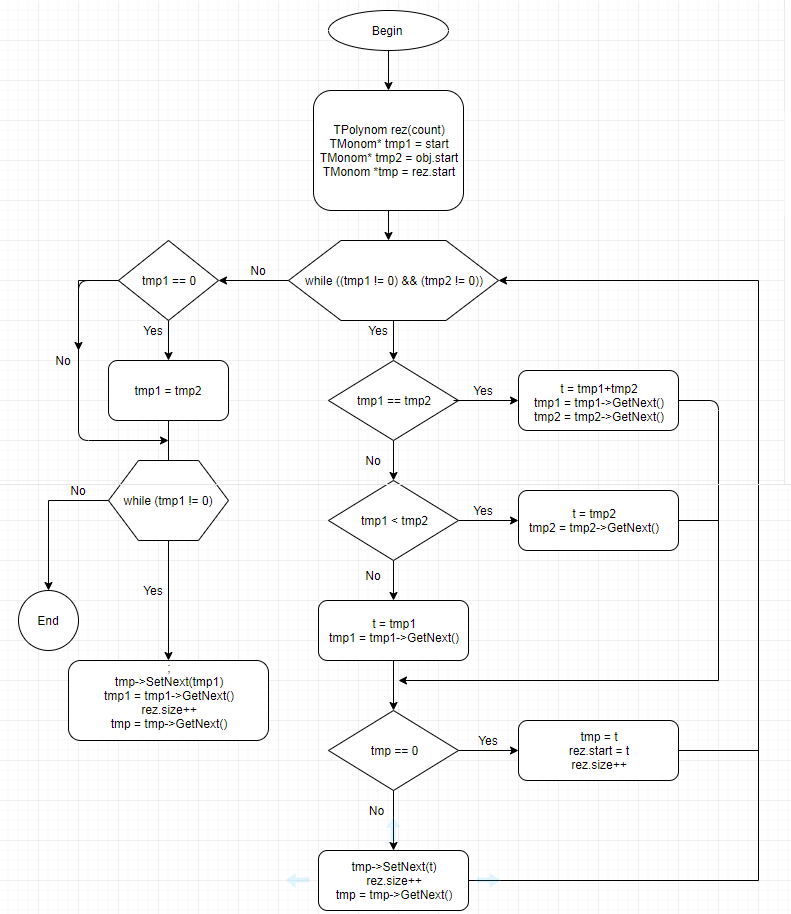
Далее идут методы класса со спецификатором доступа **public**:

* **TMyException(std::string \_str)** – конструктор инициализатор.
* **void What()** – метод вывода ошибки на экран.

# Описание алгоритмов.

**Сложение двух полиномов.**

Блок-схема сложения двух полиномов:



*Рис. 2. Алгоритм сложения полиномов.*

# Заключение.

В данной лабораторной работе мне удалось реализовать библиотеку для работы с полиномами, а именно:

* Удалось реализовать вспомогательный класс монома **TMonom**.
* Удалось реализовать класс полинома **TPolynom**.
* Удалось протестировать методы класса **TPolynom**, а также обеспечить их работоспособность.
* Удалось реализовать класс для обработки исключений **TMyException**.

Таким образом, я смог реализовать инструмент для работы с полиномиальными выражениями.

# Литература.

1. Ссылка из Википедии про моном:

<https://ru.wikipedia.org/wiki/%D0%9E%D0%B4%D0%BD%D0%BE%D1%87%D0%BB%D0%B5%D0%BD>

1. Ссылка из Википедии про полином:

<https://ru.wikipedia.org/wiki/%D0%9C%D0%BD%D0%BE%D0%B3%D0%BE%D1%87%D0%BB%D0%B5%D0%BD>

1. Гергель В.П. Методические материалы по курсу «Методы программирования 2», 2015.