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**

**«Односвязный список с использованием массивов»**

**Выполнил:** студент группы 381706-1

Суслов Егор Игоревич

\_\_\_\_\_\_\_\_\_\_\_ Подпись

**Научный руководитель:**

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

Нижний Новгород

2018.

Содержание

[1. Введение 3](#_Toc533083472)

[2. Постановка задачи 4](#_Toc533083473)

[3. Руководство пользователя 5](#_Toc533083474)

[4. Руководство программиста 6](#_Toc533083475)

[4.1. Описание структуры программы 6](#_Toc533083476)

[4.2. Описание структур данных 6](#_Toc533083477)

[4.3. Описание алгоритмов 7](#_Toc533083478)

[5. Заключение 8](#_Toc533083479)

[6. Литература 9](#_Toc533083480)

# Введение

Изучение полиномиальных уравнений и их решений составляло едва ли не главный объект «классической алгебры».

С изучением многочленов связан целый ряд преобразований в математике: введение в рассмотрение [нуля](https://ru.wikipedia.org/wiki/%D0%9D%D0%BE%D0%BB%D1%8C_(%D1%87%D0%B8%D1%81%D0%BB%D0%BE)), [отрицательных](https://ru.wikipedia.org/wiki/%D0%9E%D1%82%D1%80%D0%B8%D1%86%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D0%BE%D0%B5_%D1%87%D0%B8%D1%81%D0%BB%D0%BE), а затем и [комплексных чисел](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D0%BB%D0%B5%D0%BA%D1%81%D0%BD%D0%BE%D0%B5_%D1%87%D0%B8%D1%81%D0%BB%D0%BE), а также появление [теории групп](https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BE%D1%80%D0%B8%D1%8F_%D0%B3%D1%80%D1%83%D0%BF%D0%BF) как раздела математики и выделение классов [специальных функций](https://ru.wikipedia.org/wiki/%D0%A1%D0%BF%D0%B5%D1%86%D0%B8%D0%B0%D0%BB%D1%8C%D0%BD%D1%8B%D0%B5_%D1%84%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D0%B8) в анализе.

Техническая простота вычислений, связанных с многочленами, по сравнению с более сложными классами функций, а также тот факт, что множество многочленов [плотно](https://ru.wikipedia.org/wiki/%D0%A1%D0%BB%D0%BE%D0%B2%D0%B0%D1%80%D1%8C_%D1%82%D0%B5%D1%80%D0%BC%D0%B8%D0%BD%D0%BE%D0%B2_%D0%BE%D0%B1%D1%89%D0%B5%D0%B9_%D1%82%D0%BE%D0%BF%D0%BE%D0%BB%D0%BE%D0%B3%D0%B8%D0%B8#%D0%9F) в пространстве [непрерывных функций](https://ru.wikipedia.org/wiki/%D0%9D%D0%B5%D0%BF%D1%80%D0%B5%D1%80%D1%8B%D0%B2%D0%BD%D0%BE%D0%B5_%D0%BE%D1%82%D0%BE%D0%B1%D1%80%D0%B0%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5) на [компактных подмножествах](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D0%B0%D0%BA%D1%82%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D1%81%D1%82%D1%80%D0%B0%D0%BD%D1%81%D1%82%D0%B2%D0%BE) [евклидова пространства](https://ru.wikipedia.org/wiki/%D0%95%D0%B2%D0%BA%D0%BB%D0%B8%D0%B4%D0%BE%D0%B2%D0%BE_%D0%BF%D1%80%D0%BE%D1%81%D1%82%D1%80%D0%B0%D0%BD%D1%81%D1%82%D0%B2%D0%BE) (см. [аппроксимационная теорема Вейерштрасса](https://ru.wikipedia.org/wiki/%D0%90%D0%BF%D0%BF%D1%80%D0%BE%D0%BA%D1%81%D0%B8%D0%BC%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%82%D0%B5%D0%BE%D1%80%D0%B5%D0%BC%D0%B0_%D0%92%D0%B5%D0%B9%D0%B5%D1%80%D1%88%D1%82%D1%80%D0%B0%D1%81%D1%81%D0%B0)), способствовали развитию методов разложения в ряды и полиномиальной интерполяции в математичесском анализе.

Многочлены также играют ключевую роль в [алгебраической геометрии](https://ru.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%B5%D0%B1%D1%80%D0%B0%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B0%D1%8F_%D0%B3%D0%B5%D0%BE%D0%BC%D0%B5%D1%82%D1%80%D0%B8%D1%8F), объектом которой являются множества, определённые как решения систем многочленов.

Особые свойства преобразования коэффициентов при умножении многочленов используются в алгебраической геометрии, [алгебре](https://ru.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%B5%D0%B1%D1%80%D0%B0), [теории узлов](https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BE%D1%80%D0%B8%D1%8F_%D1%83%D0%B7%D0%BB%D0%BE%D0%B2) и других разделах математики для кодирования или выражения многочленами свойств различных объектов.

**Полином** - конечная сумма мономов, вида:

|  |
| --- |
| Рисунок 1. Полином общего вида  **Одночлен** **(или моном)** — простое [математическое](https://ru-wiki.ru/wiki/%D0%90%D0%BB%D0%B3%D0%B5%D0%B1%D1%80%D0%B0) [выражение](https://ru-wiki.ru/wiki/%D0%9F%D1%80%D0%B8%D0%BC%D0%B5%D1%80), прежде всего рассматриваемое и используемое в [элементарной алгебре](https://ru-wiki.ru/wiki/%D0%AD%D0%BB%D0%B5%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%80%D0%BD%D0%B0%D1%8F_%D0%B0%D0%BB%D0%B3%D0%B5%D0%B1%D1%80%D0%B0), а именно, произведение, состоящее из [числового](https://ru-wiki.ru/wiki/%D0%A7%D0%B8%D1%81%D0%BB%D0%BE) [множителя](https://ru-wiki.ru/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-wiki.ru/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-wiki.ru/wiki/%D0%A6%D0%B5%D0%BB%D0%BE%D0%B5_%D1%87%D0%B8%D1%81%D0%BB%D0%BE) степени  Рисунок 2. Моном общего вида   * I = () — набор из целых неотрицательных чисел, именуемый мультииндексом, * — число, именуемое коэффициент многочлена, зависящее только от мультииндекса I. |

# Постановка задачи

В рамках лабораторной работы ставится задача эффективной реализации структуры данных –полинома состоящего из мономов.

Для работы с полиномом необходимо реализовать операции:

* Вычитание,
* Сложение,
* Присваивание,
* Умножение,
* Сравнение,
* Вывод полинома.

Программное решение будет выглядеть следующим образом:

1. Класс полинома– TPolynom.
2. Класс монома – Tmonom.
3. Класс для обработки исключений – MyException.
4. Программа, демонстрирующая работу класса TPolynom (получается из файла кода «Polynom \_main.cpp».
5. Набор автоматических тестов с использованием Google C++ Testing Framework.

# Руководство пользователя

При запуске консольного приложения «Polynom\_main.cpp» пользователю выводятся на консоль 6 заполненных мономов. Затем они делятся по три, и из каждой тройки составляется полином путем суммирования мономов. Далее производятся действия над полиномами. Все действия сопровождаются выводом на консоль.

# Руководство программиста

## Описание структуры программы

Программа состоит из следующих модулей:

* Модуль Polynomial. Содержит пример работы с полиномами. Реализация в файле «Polinom\_main.cpp».
* Модуль PolynomLib – статическая библиотека. Содержит файл Polynomial.h, в котором описан интерфейс и реализация шаблонного класса TPolynomList и «Monom.h», в котором описан интерфейс и реализация шаблонного класса TMonom.
* Модуль ExceptionLib – библиотека, содержащая класс исключений.
* Модуль PolynomTest. Содержит тесты, описанные в файле «PolynomTest.cpp» и разработанных с помощью использования Google C++ Testing Framework.

.

## Описание структур данных

#### Класс TMonom

Класс TMonom является классом мономом. Объекты этого класса будут узлами списка полинома. В классе объявлены четыре поля со спецификатором доступа protected:

* TMonom\* next - указатель на следующий моном;
* int n - кол-во переменных в мономе;
* int \*power - массив степеней переменных монома;
* double coeff – коэффицтент при мономе;

Cо спецификатором доступа public:

* TMonom(int \_n, int\* \_power, double \_c) – конструктор по умолчанию;
* TMonom(TMonom& A) – конструктор копирования;
* virtual ~TMonom()– деструктор;
* TMonom\* GetNext() – получить адрес следующего монома;
* int\* GetPower() – получить массив степеней монома;
* double GetC() - получить коэффициент при мономе;
* int GetN() – получить количество переменных в мономе;
* void SetNext(TMonom\* \_next) – установить адрес следующего монома;
* void SetPower(int\* \_power) – установить массив степеней монома;
* void SetC(double \_c) – установить константу монома;
* void SetN(int \_n) – установить количество переменных монома;
* TMonom& operator=(const TMonom& A);
* TMonom operator+(TMonom& A);
* TMonom operator\*(const TMonom& A);
* TMonom operator-(TMonom& A);
* bool operator==(TMonom& A);
* bool operator>(TMonom& A);
* bool operator<(TMonom& A);
* friend istream& operator>>(istream& istr, TMonom& A) – оператор ввода монома;
* friend ostream& operator<<(ostream& ostr, TMonom& A) – оператор вывода монома;

#### Класс TPolynom

Класс TPolynom является классом полиномов. В классе объявлены три поля со спецификатором доступа protected:

* TMonom \*start – указатель на первый моном в полиноме;
* int n - количество переменных в каждом мономе полинома;
* int size - количество мономов в полиноме;

Cо спецификатором доступа public:

* TPolynom(int \_n = 3) – конструктор по умолчанию;
* TPolynom(TPolynom &p) – конструктор копирования;
* ~TPolynom() – деструктор;
* int GetSize() – получить текущее количество мономов в полиноме;
* TMonom\* GetStart() – получить адрес первого монома в полиноме;:
* TPolynom operator-(TPolynom &p);
* Polynom operator+(TPolynom &p);
* TPolynom& operator=(const TPolynom &p);
* bool operator==(TPolynom &p);
* TPolynom operator\*(TPolynom &p);
* TPolynom& operator+=(TMonom &m) – сложение монома и полинома;
* TPolynom& operator-=(TMonom &m) – вычитание монома из полинома;
* friend std::ostream& operator<<(std::ostream& \_s, TPolynom& Tm);

## Описание алгоритмов

Алгоритм 1.

Умножение двух полиномов.

Заданы два полинома в виде циклических списков - первый u , второй v .

Шаг 1. Начальная установка.

Формируем новый (нулевой) полином r – результат работы программы.

Если один из сомножителей – нулевой полином, то закончить алгоритм. Берем первый, старший моном полинома v , делаем его текущим мономом t .

Шаг 2. Сложение результата с произведением полинома, первого сомножителя, на текущий моном. Для этих целей используется специальная процедура сложения AddPoly, описанная в предыдущей лекции и видоизмененная в процедуру AddPolyT с помощью введения умножения второго слагаемого на текущий моном t , который задается в виде параметра процедуры.

Шаг 3. Цикл по мономам полинома v . Если полином v не исчерпан, то переходим к следующему моному в этом полиноме и делаем его текущим мономом t , после чего передаем управление на

Шаг 4; в противном случае закончить алгоритм. Приведенный алгоритм умножения полиномов будет работать и для полиномов от нескольких переменных, если в процедуре AddPolyT реализовано умножение на одночлен для полинома, зависящего от нескольких переменных, а в процедуре InsertMonom делается вставка такого же типа одночлена. Пример программ умножения полиномов от одной переменной на разных языках программирования приводится ниже. Теорема. Для полинома А, содержащего m одночленов, и полинома В, содержащего n одночленов, трудоемкость приведенного алгоритма имеет порядок O( n ) .

# Заключение

Все поставленные задачи были выполнены это подтверждается Google C++ Testing Framework (все 38 тестов, 17 на класс TPolynom и 21 на вспомогательный класс TMonom, успешно выполняются) и работой с файлом кода «Polynom \_main.cpp» использующего наш класс.

# Литература

* Книги:

1. A.O. Грудзинский. Методы программирования, Издательство Нижегородского госуниверситета, 2006.
2. Топп У., Форд У. Структуры данных в С++. - М. Бином, 1999.
3. Мейн М., Савитч У. Структуры данных и другие объекты в С++. - М.: Издательский дом “Вильямс”, 2003.
4. Гергель В.П. Методические материалы по курсу «Методы программирования 2», Нижний Новгород, 2015.

* Ссылки в Internet:

1. Учебно-методическое пособие из электронной библиотеки ННГУ: «ЛАБОРАТОРНЫЙ ПРАКТИКУМ» : [http://www.unn.ru/books/met\_files/Pract\_ADS.pdf].
2. Справочник по алгоритму перемножения полиномов [http://www.itlab.unn.ru/Uploads/coaChapter06.pdf]
3. Определение полинома, монома и прочая справочная информация [https://ru.wikipedia.org/wiki/Многочлен]