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**

**«Полиномы»**

**Выполнил:** студент группы 381706-1

Корнев Никита Алексеевич

\_\_\_\_\_\_\_\_\_\_\_ Подпись

**Научный руководитель:**

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

Нижний Новгород

2018.

Содержание

[1. Введение 3](#_Toc533799739)

[2. Постановка задачи 4](#_Toc533799740)

[3. Руководство пользователя 5](#_Toc533799741)

[4. Руководство программиста 6](#_Toc533799742)

[4.1. Описание структуры программы 6](#_Toc533799743)

[4.2. Описание структур данных 7](#_Toc533799744)

[4.3 Описание алгоритмов 8](#_Toc533799745)

[5. Заключение 9](#_Toc533799746)

[6. Литература 10](#_Toc533799747)

# Введение

Структура данных (англ. data structure) — программная единица, позволяющая хранить и обрабатывать множество однотипных и/или логически связанных данных в вычислительной технике. Для добавления, поиска, изменения и удаления данных структура данных предоставляет некоторый набор функций, составляющих её интерфейс. Одной из важных структур является список.

Многочлен (или полином от греч. πολυ- «много» + лат. nomen «имя») от n переменных — это сумма одночленов или, строго, — конечная формальная сумма вида

* I — набор из целых неотрицательных чисел, именуемый мультииндексом,
* — число, именуемое коэффициент многочлена, зависящее только от мультииндекса I.



* — фиксированные коэффициенты,
* — переменная.

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, отрицательных, а затем и комплексных чисел, а также появление теории групп как раздела математики и выделение классов специальных функций в анализе.

Особые свойства преобразования коэффициентов при умножении многочленов используются в алгебраической геометрии, алгебре, теории узлов и других разделах математики для кодирования или выражения многочленами свойств различных объектов.

В нашей лабораторной работе нам предстоит самостоятельно реализовать струтуру данных для хранения объектов типа полином.

# Постановка задачи

1. Написать класс, реализующий мономы.
2. Написать класс, реализующий полиномы.
3. Написать тесты на основе Google Tests для проверки работы классов.

# Руководство пользователя

Чтобы использовать данный класс в своем проекте, необходимо подключить библиотеку «Polinom.h». Библиотека позволяет:

* Создавать объекты типа *моном* и *полином :*

TMonom A; **конструктор по умолчанию (размер равен 1)**

TMonom A(3, a, 7); **конструктор с параметром**

TMonom(const TMonom& A); **конструктор копирования**

TPolinom(int \_n = 1); **конструктор по умолчанию (размер равен 1)**

TPolinom(const TPolinom &A); **конструктор копирования**

* Применять математические операции:

TMonom m1, m2, m3;

m3 = m1 + m2;

m3 = m2 – m1;

m3 = m1 \* m2;

TPolinom p1, p2, p3;

p3 = v1 + v2;

p3 = v1 – v2;

p3 = v1 \* v2;

* Сравнивать:

TMonom m1, m2;

bool a = m2 > m1;

bool b = m2 < m1;

bool c = m2 == m1;

# Руководство программиста

## Описание структуры программы

Проект состоит из следующих модулей:

1. Monom

Модуль cодержит файл код «main.cpp», в котором продемонстрирован пример использования структуры.

1. PolinomLib

Модуль содержит файл заголовок «Monom.h», описывающий структуру «моном», а также файл кода «main.cpp».

1. PolinomTest

Модуль содержит файлы кода «test\_main.cpp», «MonomTest.cpp» и «PolinomTest.cpp». В них реализованы тесты для проверки корректности работы методов классов.

## Описание структур данных

#### *Структура TMonom*

Структура **TMonom**. 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 c** // коэффицтент при мономе;

Поля со 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;
* **friend ostream& operator<<(ostream& ostr, TMonom& A) //** оператор вывода;

#### *Структура TPolinom*

Структура TVector. 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;

Поля со спецификатором доступа «public»:

* **TPolynom(int \_n = 1)** *//* 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); //** оператор вывода

## Описание алгоритмов

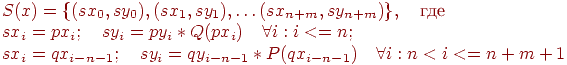
Рассмотрим операцию умножения полиномов:

http://ok-t.ru/life-prog/baza1/1559884969551.files/image166.gif

Нетрудно понять, что полином S(x) является полиномом степени n + m и имеет коэффициент n + m + 1. Как вычисляется произведение, если заданы полиномы сомножители P(x) и Q(x)? Замечу, что произведение полиномов часто встречается на практике и имеет специальное имя - свертка полиномов.

В отличие от сложения полиномов проще всего найти свертку, если заданы корни обоих полиномов. В этом случае никаких вычислений не требуется, поскольку n корней P(x) и m корней Q(x) будут n + m корнями S(x). Если у полиномов P(x) и Q(x) есть совпадающие корни, то у S(x) появятся кратные корни.

Если исходные полиномы P(x) и Q(x) заданы своими точками, то нетрудно получить набор точек для полинома произведения. Схема во многом похожа на ту, что имеет место при сложении полиномов, заданных точками:

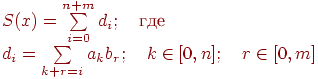


Для получения множества точек, задающих представление полинома S(x), приходится вычислять значение полинома Q(x) в n точках и значение полинома P(x) в m точках, а затем выполнять соответствующее умножение значений двух полиномов.

Если исходные полиномы P(x) и Q(x) заданы своими коэффициентами, то имеем:

http://ok-t.ru/life-prog/baza1/1559884969551.files/image174.gif

Каждый член первой суммы приходится умножать на все члены второй суммы и затем приводить подобные члены при одинаковых степенях x. Нетрудно заметить, что в результате коэффициенты полинома S(x) определяются следующими соотношениями:



Суммирование идет по всем наборам k и r, дающим в сумме значение i. Понятно, что для крайних значений (i=0 и i=n+m) сумма состоит из одного члена, поскольку подобные члены для x в нулевой степени и степени n+m отсутствуют. Число членов суммирования увеличивается при приближении к середине интервала [0, n+m].

# Заключение

В данном лабораторной работе мне удалось:

* Успешно реализовать класс для монома
* Успешно реализовать класс для полинома
* Продемонстрировать пример использования данного класса
* Написать тесты на основе Google Tests для проверки корректной работы данного класса.

# Литература

Интернет-источники:

1. <https://ru.wikipedia.org/wiki/Многочлен>
2. <https://ru.wikipedia.org/wiki/Одночлен>
3. <https://habr.com/ru/post/303342/>

Книги:

1. Гергель В.П. Методические материалы по курсу «Методы программирования 2», Нижний Новгород, 2015.
2. A.O. Грудзинский. Методы программирования, Издательство Нижегородского госуниверситета, 2006.