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**

**«Структура данных: полином»**

**Выполнил:** студент группы 381706-1

Соболева Ю.А.

\_\_\_\_\_\_\_\_\_\_\_ Подпись

**Научный руководитель:**

аспирант каф. МОСТ ИИТММ

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Лебедев И.Г.

Нижний Новгород

2018.

Содержание

[1. Введение 3](#_Toc533027637)

[2. Постановка задачи 4](#_Toc533027638)

[3. Руководство пользователя 5](#_Toc533027639)

[4. Руководство программиста 6](#_Toc533027640)

4.[1. Описание структуры программы 6](#_Toc533027641)

4.[2. Описание структур данных 6](#_Toc533027642)

4.[3. Описание алгоритмов 8](#_Toc533027643)

[5. Заключение 9](#_Toc533027644)

[6. Литература 10](#_Toc533027645)

# **Введение**

**Структура данных** - программная единица, которая определяет метод хранения и обработки различных логически связанных данных в вычислительной технике. Знание структур данных позволяет наиболее компактно и практично расположить данные в памяти компьютера. В данной работе мы рассмотрим такую структуру данных, как **полином**.

В [математике](https://dic.academic.ru/dic.nsf/ruwiki/750), **многочлены или полиномы** от одной переменной - функции вида

F(x) = c_0 + c_1 x + \cdots + c_n x^n,

где *ci* фиксированные [коэффициенты](https://dic.academic.ru/dic.nsf/ruwiki/71939), а *x*  переменная. **Полиномы** составляют один из важнейших классов [элементарных функций](https://dic.academic.ru/dic.nsf/ruwiki/154043).

Каждое слагаемое полинома называется **мономом.**

**Целью данной лабораторной работы** является разработка структуры данных для хранения полиномов с использованием класса мономов, а также освоение таких инструментов разработки программного обеспечения, как система контроля версий Git и фрэймворк для разработки автоматических тестов Google Test.

# **Постановка задачи**

Выполнение работы предполагает решение следующих задач:

1. Реализация класса полинома TPolynom, унаследованного от класса монома TMonom.
2. Реализация класса монома TMonom.
3. Разработка интерфейса для данных классов.
4. Обеспечение работоспособности примера использования.
5. Реализация нескольких тестов на базе Google Test.

# **Руководство пользователя**

При запуске программы на экран выводится сообщение, что все тесты пройдены успешно. Примеры вывода сообщений для класса TPolynom показаны на рис.1.

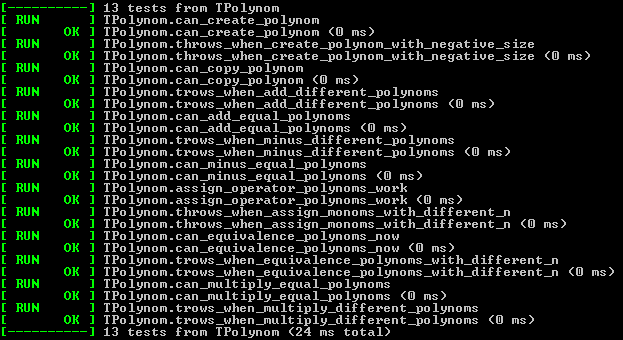


Рис. 1

# **Руководство программиста**

## **Описание структуры программы**

**Проект “Poly” состоит из следующих файлов:**

1. mainPoly.cpp (в нём находится main)

**Проект “ PolyLib” состоит из следующих файлов:**

1. PolyLib.h (описание класса “TPolynom”)
2. MonoLib.h (описание класса “TMonom”)
3. PolyLib.cpp (реализация методов класса “TPolynom” и перегрузка операций)
4. MonoLib.cpp (реализация методов класса “TMonom” и перегрузка операций )

**Проект “ Test” состоит из следующих файлов:**

1. test\_polylib.cpp (реализация тестов для класса TPolynom)

## **Описание структур данных**

***Класс TMonom:***

*Поля:*

* TMonom\* next; - указатель на следующий моном;
* int \*power; - степень множителей;
* double c; - коэфициент;
* int n; - количество множителей;

*Конструкторы и деструктор:*

* TMonom(); - конструктор по умолчанию;
* TMonom(int \_n, int\* \_power, double \_c); - конструктор инициализации;
* TMonom(const TMonom& A); - конструктор копирования;
* virtual ~TMonom(); - деструктор;

*Методы:*

* TMonom\* GetNext(); - получить следующий моном;
* int\* GetPower(); - получить степень множителей;
* double GetC(); - получить коэффициент;
* int GetN(); - получить количество множителей;
* void SetNext(TMonom\* \_next); - установить следующий моном;
* void SetPower(int\* \_power); - установить степени;
* void SetC(double \_c); - установить коэфициенты;
* void SetN(int \_n); - установить количество множителей;

*Перегруженные операторы:*

* TMonom& operator=(const TMonom& A);
* TMonom operator+(TMonom& A);
* TMonom operator+=(TMonom& A);
* TMonom operator\*(const TMonom& A) const;
* TMonom operator\*=(TMonom& A);
* TMonom operator-(TMonom& A);
* TMonom operator-=(TMonom& A);
* bool operator==(TMonom& A);
* bool operator>(TMonom& A);
* bool operator<(TMonom& A);

***Класс TPolynom:***

*Поля:*

* TMonom \*start; - начальный моном;
* int n; - количество мономов;
* int size; - размер полинома;

*Конструкторы:*

* TPolynom(int \_n = 3); - конструктор инициализации;
* TPolynom(TPolynom &p); - конструктор копирования;
* ~TPolynom(); - деструктор;

*Методы:*

* int GetSize(); - получить размер полинома;
* TMonom\* GetStart(); - получить начальный моном;

*Перегруженные операторы:*

* TPolynom operator-(TPolynom &p);
* TPolynom operator+(TPolynom &p);
* TPolynom& operator=(const TPolynom &p);
* bool operator==(const TPolynom &p);
* TPolynom operator\*(TPolynom &p);
* TPolynom& operator+=(TMonom &m);
* TPolynom& operator-=(TMonom &m);

## **Описание алгоритмов**

**Сложение полиномов A и B.**

Алгоритм сложения двух полиномов представлен на блок-схеме (рис. 1).

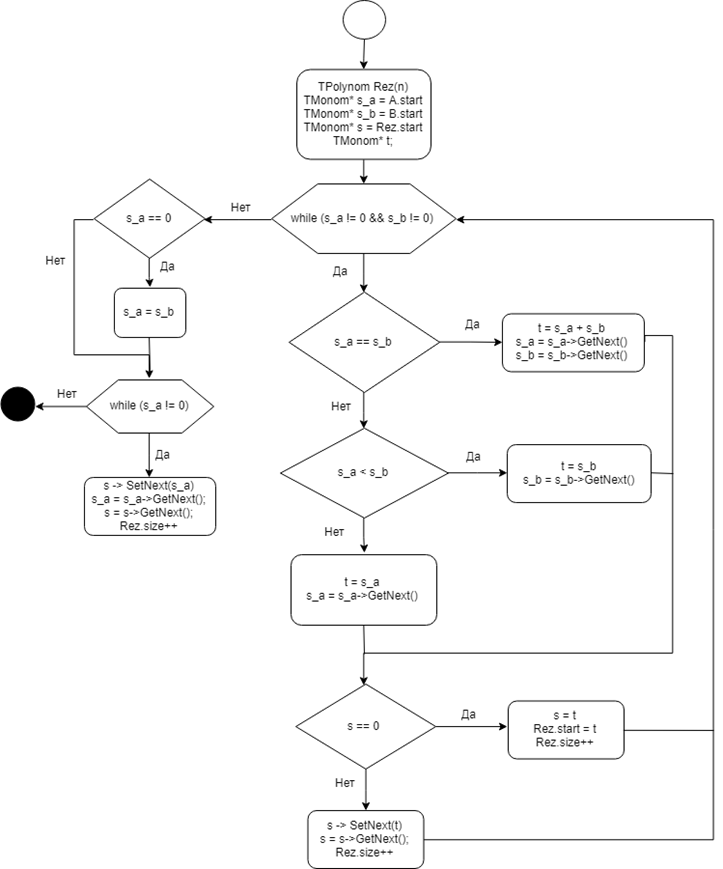
.

Рис. 1

# **Заключение**

В ходе выполнения лабораторной работы я смогла реализовать такую структуру данных, как мультистэк. Вместе с ней, согласно заданному интерфейсу, я реализовала классы TMonom и TPolynom. Написание нескольких своих тестов, помогло мне разобраться с системой автоматических тестов Google Test.

В результате проделанной работы у меня получилось

1. Реализовать класс полинома TPolynom, унаследованного от класса монома TMonom.

2. Реализовать класс монома TMonom.

3. Разработать интерфейс для данных классов.

4. Обеспечить работоспособность примера использования.

5. Реализовать некоторые тесты на базе Google Test.

Таким образом, данная лабораторная работа отвечает всем поставленным задачам. Это означает, что цель работы была достигнута.

# **Литература**

1. Лабораторный практикум. Составители: Барышева И.В., Мееров И.Б., Сысоев А.В., Шестакова Н.В. Под редакцией Гергеля В.П. Учебно-методическое пособие. – Нижний Новгород: Нижегородский госуниверситет, 2017. – 105с.

URL: <http://www.unn.ru/books/met_files/Pract_ADS.pdf>

1. Национальный открытый университет «Интуит». Курс «Основы программирования». Лекция 11: Структуры данных: общее понятие, реализация. Простейшие структуры данных: очередь, стек. Использование стека и обратная польская запись.

URL: <https://www.intuit.ru/studies/courses/2193/67/lecture/1980?page=3>

1. URL: https://dic.academic.ru/dic.nsf/ruwiki/1099935