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

Мазур Даниил Андреевич

\_\_\_\_\_\_\_\_\_\_\_ Подпись

**Научный руководитель:**

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

Нижний Новгород

2018.

Содержание

[1. Введение 3](#_Toc536523975)

[2. Постановка задачи 4](#_Toc536523976)

[3. Руководство пользователя 5](#_Toc536523977)

[4. Руководство программиста 6](#_Toc536523978)

[4.1. Описание структуры программы 6](#_Toc536523979)

[4.2. Описание структур данных 6](#_Toc536523980)

[4.3. Описание алгоритмов 8](#_Toc536523981)

[5. Заключение 11](#_Toc536523983)

[6. Литература 12](#_Toc536523984)

# 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*.

***Линейный список*** — это структура данных, состоящая из элементов одного типа, связанных между собой последовательно посредством указателей. Каждый элемент списка имеет указатель на следующий элемент. Последний элемент списка указывает на [NULL](https://ru.wikipedia.org/wiki/NULL_(%D0%A1%D0%B8)). Элемент, на который нет указателя, является первым (головным) элементом списка. Здесь ссылка в каждом узле указывает на следующий узел в списке.

Линейный список, как структура данных, очень удобен как для хранения полиномов, так и для работы с ними. Узлами списка, относительно хранения полинома, являются мономы. Все мономы лексикографически упорядочены между собой по мультииндексу: моном с большим мультииндексом указывает на следующий за ним моном с меньшим индексом. Поэтому представление полинома в памяти компьютера единственно.

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

# Постановка задачи

Выполнение работы предполагает решение следующих задач:

1. Разработка и реализация вспомогательного класса – TMonom – узел списка(моном).
2. Разработка и реализация базового класса – TPolynom.
3. Создание класса для обработки исключений – MyException, которые могут возникнуть при выполнении различных операций.
4. Разработка программы, демонстрирующей работу классов TPolynom и TMonom.
5. Реализация набор автоматических тестов с использованием Google C++ Testing Framework.

# Руководство пользователя

Рассмотрим пример использования класса TStackList.

При запуске создается 6 мономов: A, B, C, D, E, F

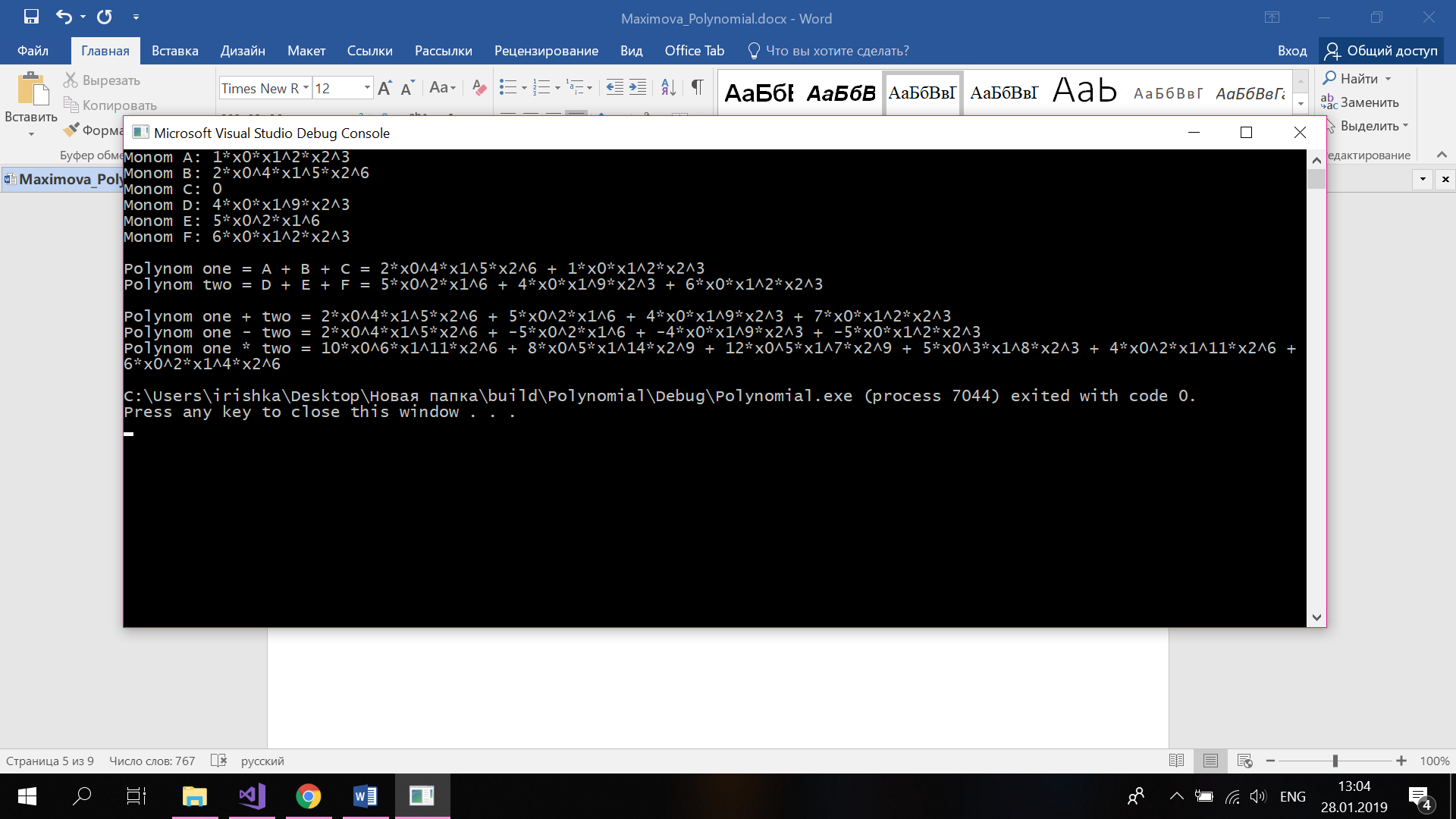


Рисунок 1. Инициализация мономов

Затем создается первый полином - one путем сложения мономов A, B и C, и второй полином – two путем сложения мономов D, E и F:

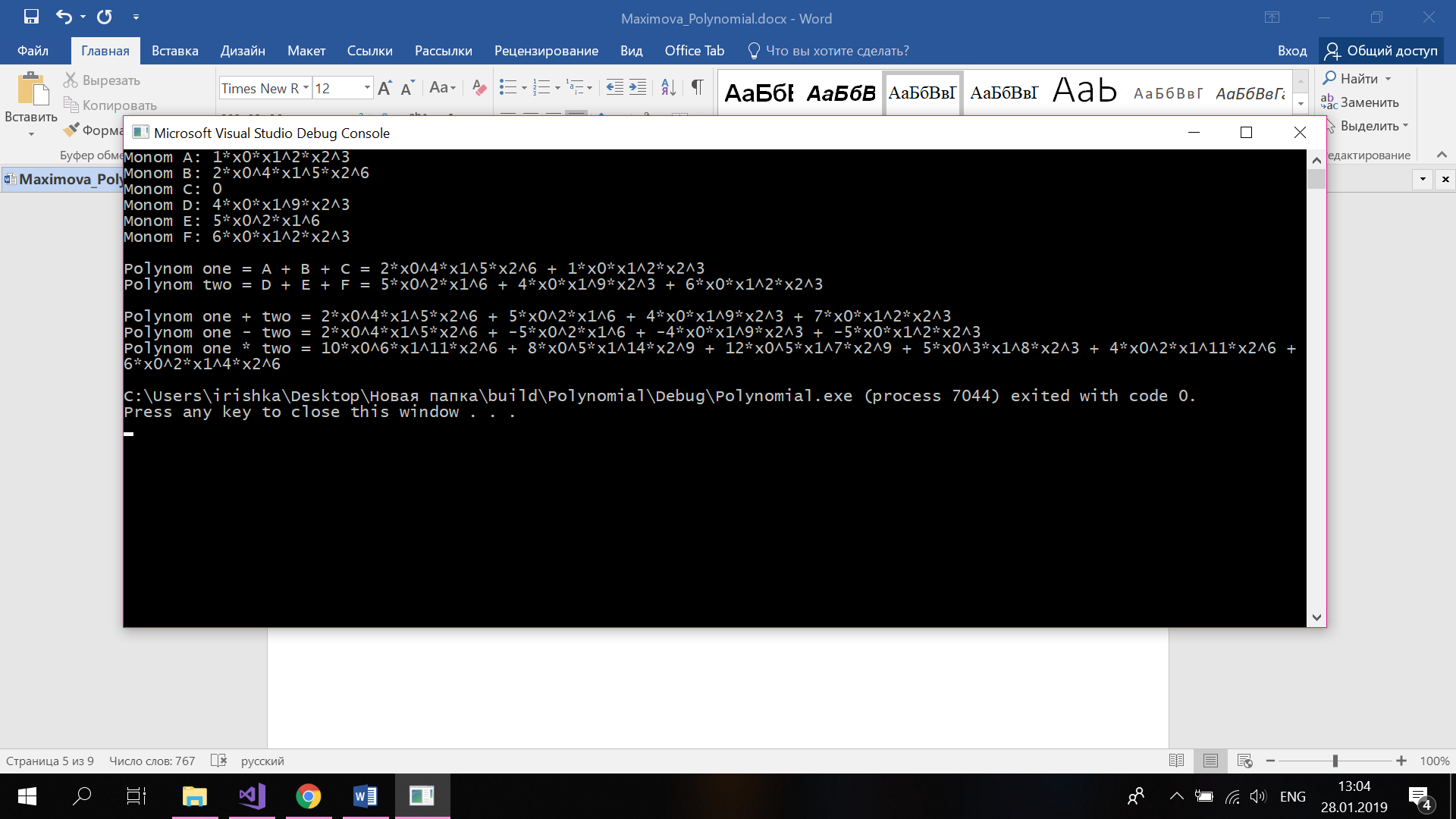


Рисунок 2. Инициализация полиномов

Далее выполняются операции сложения, вычитания и умножения первого и второго полиномов. Результаты операций выводятся на консоль. На этом работа программы прекращается.

# Руководство программиста

## Описание структуры программы

Программа состоит из следующих модулей:

* Модуль Polynomial. Содержит пример работы с полиномами. Реализация в файле *main\_Polinomial.cpp.*
* Модуль PolynomialLib – статическая библиотека. Содержит файл Polynomial.h, в котором описан интерфейс и реализация шаблонного класса *TPolynomList* (2 конструктора, деструктор и 10 методов) и и Monom.h, в котором описан интерфейс и реализация шаблонного класса *TMonom* (2 конструктора, деструктор и 17 методов)
* Модуль PolynomTest. Содержит 41 тест, описанные в файле *PolynomTest.cpp* и разработанных с помощью использования Google C++ Testing Framework.
* Модуль ExceptionLib – библиотека, содержащая класс исключений.

## 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;

Далее в публичной зоне (public):

* + - 1. Прописаны 2 конструктора и деструктор:
* *TMonom(int \_n, int\* \_power, double \_c)* – конструктор по умолчанию;
* *TMonom(TMonom& A)* – конструктор копирования;
* *virtual ~TMonom()*– деструктор;
  + - 1. Описаны методы доступа к полям класса:
* *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;
  + - 1. Перегружены операторы для работы с мономами:
* *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);*
  + - 1. Определены дружественные методы вывода/ввода мономов:
* *friend istream& operator>>(istream& istr, TMonom& A) –* оператор ввода монома*;*
* *friend ostream& operator<<(ostream& ostr, TMonom& A) –* оператор вывода монома на консоль;

#### 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;

Далее в публичной зоне (public):

Прописаны 2 конструктора и деструктор:

* *TPolynom(int \_n = 3) –* конструктор по умолчанию;
* *TPolynom(TPolynom &p) –* конструктор копирования;
* *~TPolynom() –* деструктор;

Описаны методы доступа к полям класса:

* *int GetSize() –* получить текущее количество мономов в полиноме;
* *TMonom\* GetStart() –* получить адрес первого монома в полиноме;

Перегружены операторы для работы с мономами:

* *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*;*

Определены дружественные методы вывода/ввода мономов:

* *friend std::ostream& operator<<(std::ostream& \_s, TPolynom& Tm);*

## Описание алгоритмов

**Сложение полиномов A и B.**

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

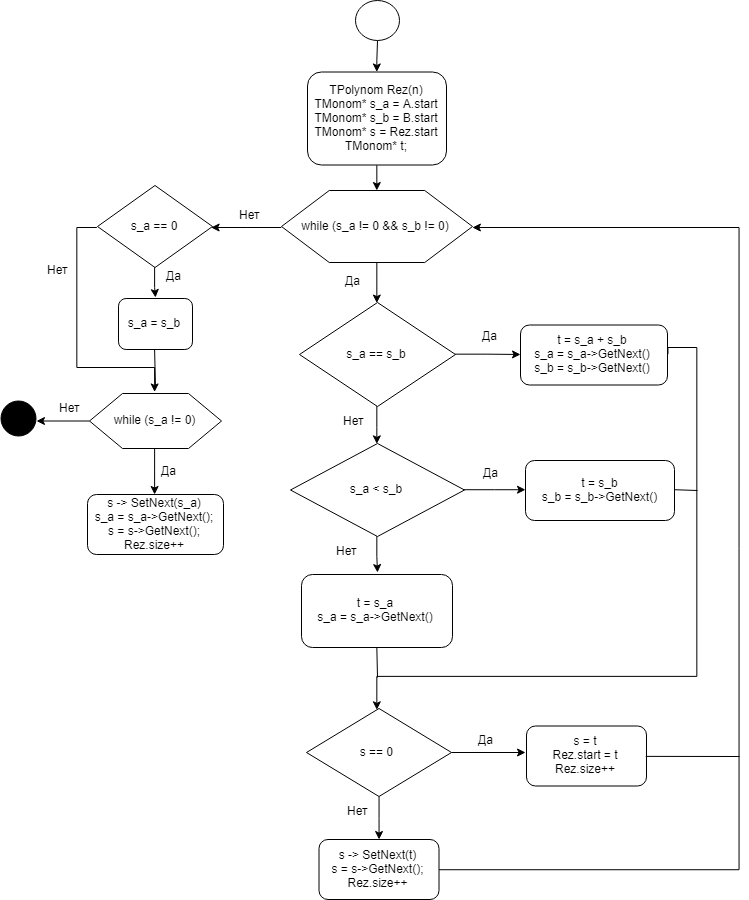
****

Рисунок 3. Алгоритм сложения двух полиномов.

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

**Сложение полинома и монома.**

Алгоритм сложение полинома и монома:

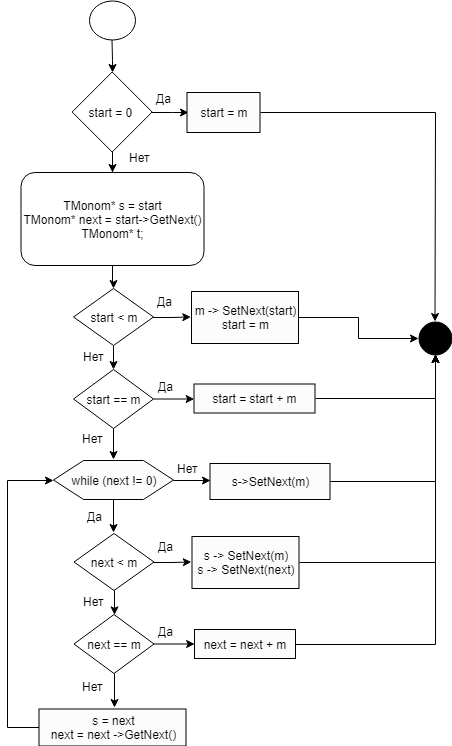
****

Рисунок 4. Алгоритм сложения полинома и монома

Start – адрес первого монома в полиноме, m – моном.

# Заключение

В ходе выполнения лабораторной была разработана система для арифметических действий над полиномами. Разработана библиотека PolynomialLib, реализующая классы полинома TPolynom и монома TMonom. В ней определены методы работы с полиномами описанные в разделе «Структуры данных».

Работоспособность методов библиотеки PolynomialLib была продемонстрирована на примере.

Разработаны и доведены до успешного выполнения тесты, разработанные для данного программного проекта с использованием Google C++ Testing Framework.

# Литература

* Книги
  + - 1. A.O. Грудзинский. Методы программирования, Издательство Нижегородского госуниверситета, 2006.
      2. Васильев А.Н. Самоучитель С++ с примерами и задачами. -СПб.: Наука и Техника, 2016. -480с.
* 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: свободная электронная энциклопедия: на русском языке: https://ru.wikipedia.org/wiki/Многочлен
      3. Википедия: свободная электронная энциклопедия: на русском языке: <https://ru.wikipedia.org/wiki/Связный> список