

**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
«Система поддержки арифметических действий над полиномами»

Выполнил: студент группы 381706-1
Митягина Дарья Сергеевна

_____ Подпись

Научный руководитель:
ассистент каф. МОСТ ИИТММ
_____ Лебедев И.Г

Нижний Новгород
2018.

Оглавление

1. Введение	2
2. Постановка задачи	3
3. Руководство пользователя	4
4. Руководство программиста.....	5
4.1. Описание структуры программы.	5
4.2. Описание структур данных.	5
4.3. Описание алгоритмов.....	7
5. Заключение.....	10
6. Список литературы.....	11

1. Введение

Основные определения:

Моном – это произведение положительного коэффициента и переменных в целых неотрицательных степенях.

Полином – конечная сумма мономов.

Целью данной лабораторной работы является разработка системы поддержки арифметических действий над полиномами.

2. Постановка задачи

В данной работе стояла задача разработать следующие элементы:

1. Класс мономов `TMonomial`
2. Класс полиномов `TPolynomial`. Полиномы реализованы с помощью такой структуры хранения данных как линейный список, узлами которого служат объекты класса `TMonomial`.
3. Набор тестов для классов `TMonomial` и `TPolynomial` с помощью `GT`.
4. Класс исключений `TException`.

3. Руководство пользователя

При запуске программы пользователь увидит на экране простой пример использования мономов и полиномов.

Создаются три монома: Moly1, Moly2, Moly3.

Затем создаем полином Poly, к которому последовательно прибавляются мономы Moly1, Moly3.

После чего пользователю предлагается задать свой моном. Для этого необходимо ввести число, которое далее будет являться коэффициентом монома moly, и задать степени переменных.

```
<<< An example of using the List will be implemented here >>>

Моном 1: 2(x0^3)(x1^2)(x2^1)(x3^4)(x4^9)
Моном 2: 0
Моном 3: 1(x1^1)(x2^4)(x3^6)(x4^7)

Polynom = Moly1 + Moly3 = 2(x0^3)(x1^2)(x2^1)(x3^4)(x4^9) + 1(x1^1)(x2^4)(x3^6)(x4^7)
coefficient = 5
power = 2 3 4 5 6

your моном : 5(x0^2)(x1^3)(x2^4)(x3^5)(x4^6)

Для продолжения нажмите любую клавишу . . .
```

Рис. 1 – пример использования мономов и полиномов

4. Руководство программиста

4.1. Описание структуры программы.

Программа состоит из модулей:

1. MonomialLib – статическая библиотека. Содержит файлы Monomial.h и Monomial.cpp, содержащие описание и реализацию класса TMonomial.
2. PolinomialLib - статическая библиотека. Содержит файлы Polinomial.h и Polinomial.cpp, содержащие описание и реализацию класса TPolinomial.
3. Exception – класс исключений.
4. MonomialTest – содержит файл moly_test.cpp с тестами для класса TMonomial.
5. PolinomialTest - содержит файл poly_test.cpp с тестами для класса TPolinomial.
6. Polynomial – содержит файл poly_main_poly.cpp, в котором реализован простой пример использования объектов описанных выше классов.

4.2. Описание структур данных.

Класс TMonomial

Рассмотрим protected-часть:

1. int* power - массив степеней переменных;
2. int size - кол-во переменных в каждом мономе;
3. double coeff- коэффициент;
4. TMonomial *next - указатель на следующий моном полинома;

Рассмотрим public-часть:

1. TMonomial(int _Size, int *M, double _Coeff) – конструктор инициализации;
 2. TMonomial(TMonomial &A) - конструктор копирования;
 3. TMonomial() - конструктор по умолчанию;
 4. virtual ~TMonomial() - деструктор
 5. Доступ к полям - "сеттеры"
- void SetPower(int *M) – задать массив степеней переменных монома;

void SetSize(int _Size) – задать размер монома (количество переменных монома);

void SetCoeff(double _Coeff) – задать коэффициент монома;

void SetNext(TMonomial* _Next) – задать адрес следующего элемента;

6. Доступ к полям - "геттеры"

int *GetPower() – получить массив степеней переменных монома;

int GetSize() – получить размер монома (количество переменных монома);

double GetCoeff() – получить коэффициент монома;

TMonomial* GetNext() – получить адрес следующего элемента;

7. перегрузка арифметических операторов

TMonomial &operator = (TMonomial &A);

TMonomial operator + (TMonomial &A);

TMonomial operator += (TMonomial &A);

TMonomial operator - (TMonomial &A);

TMonomial operator -= (TMonomial &A);

TMonomial operator * (TMonomial &A);

TMonomial operator *= (TMonomial &A);

8. операторы сравнения

bool operator == (TMonomial &A);

bool operator < (TMonomial &A);

bool operator > (TMonomial &A);

9. Дружественные функции ввода-вывода

friend istream& operator>>(istream &in, TMonomial &m);

friend ostream& operator<<(ostream &out, TMonomial &m);

Класс TPolynomial

Рассмотрим protected-часть:

1. TMonomial *begin – указатель на первый элемент полинома;

2. int n - количество переменных каждого монома;

3. int sizee – размер полинома (количество мономов);

Рассмотрим public-часть:

1. TPolynomial(int _N = 10) – конструктор;

2. TPolynomial(TPolynomial &A) - конструктор копирования;

3. Доступ к полям -"геттеры":

int GetSize() – получить размер полинома (количество мономов);

int GetN() – получить количество переменных каждого монома;

TMonomial* GetBegin() – получить адрес первого элемента полинома;

4. Перегрузки операторов

TPolynomial operator + (TPolynomial &A);

TPolynomial operator - (TPolynomial &A);

TPolynomial operator * (TPolynomial &A);

TPolynomial &operator = (TPolynomial &A);

TPolynomial &operator += (TPolynomial &A);

TPolynomial &operator += (TMonomial &M);

TPolynomial &operator -= (TMonomial &M);

bool operator==(TPolynomial &A);

5. Дружественная функция вывода

friend ostream& operator<<(ostream& out, TPolynomial& M);

4.3. Описание алгоритмов

В данной части не будут рассматриваться тривиальные методы, внимание уделим лишь некоторым.

1. Приравнивание для полиномов

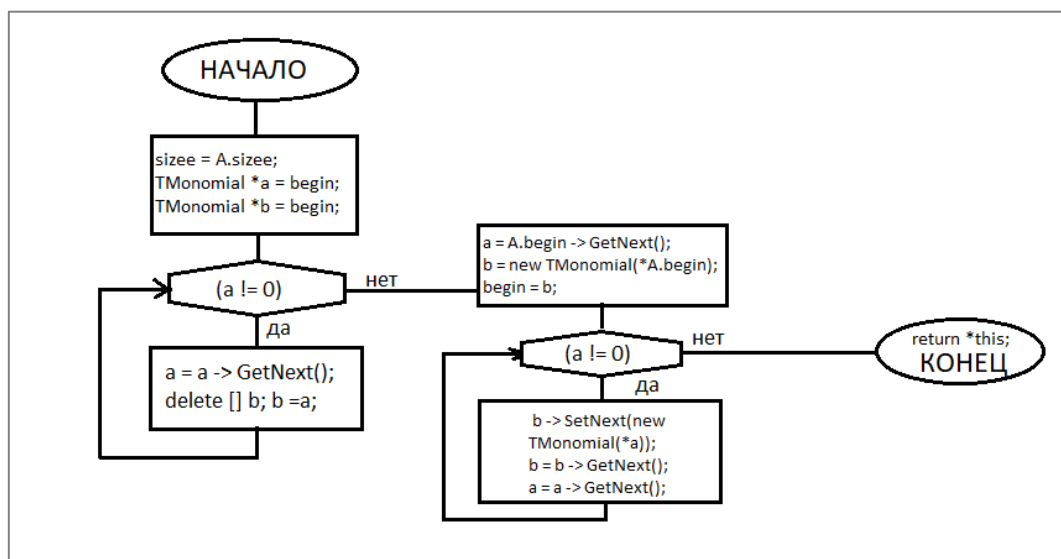


Рис. 2 - приравнивание

2. Сумма полинома и монома (operator +=).

Алгоритм довольно сложен для того, чтобы расписывать его словесное описание, поэтому рассмотрим блок-схему

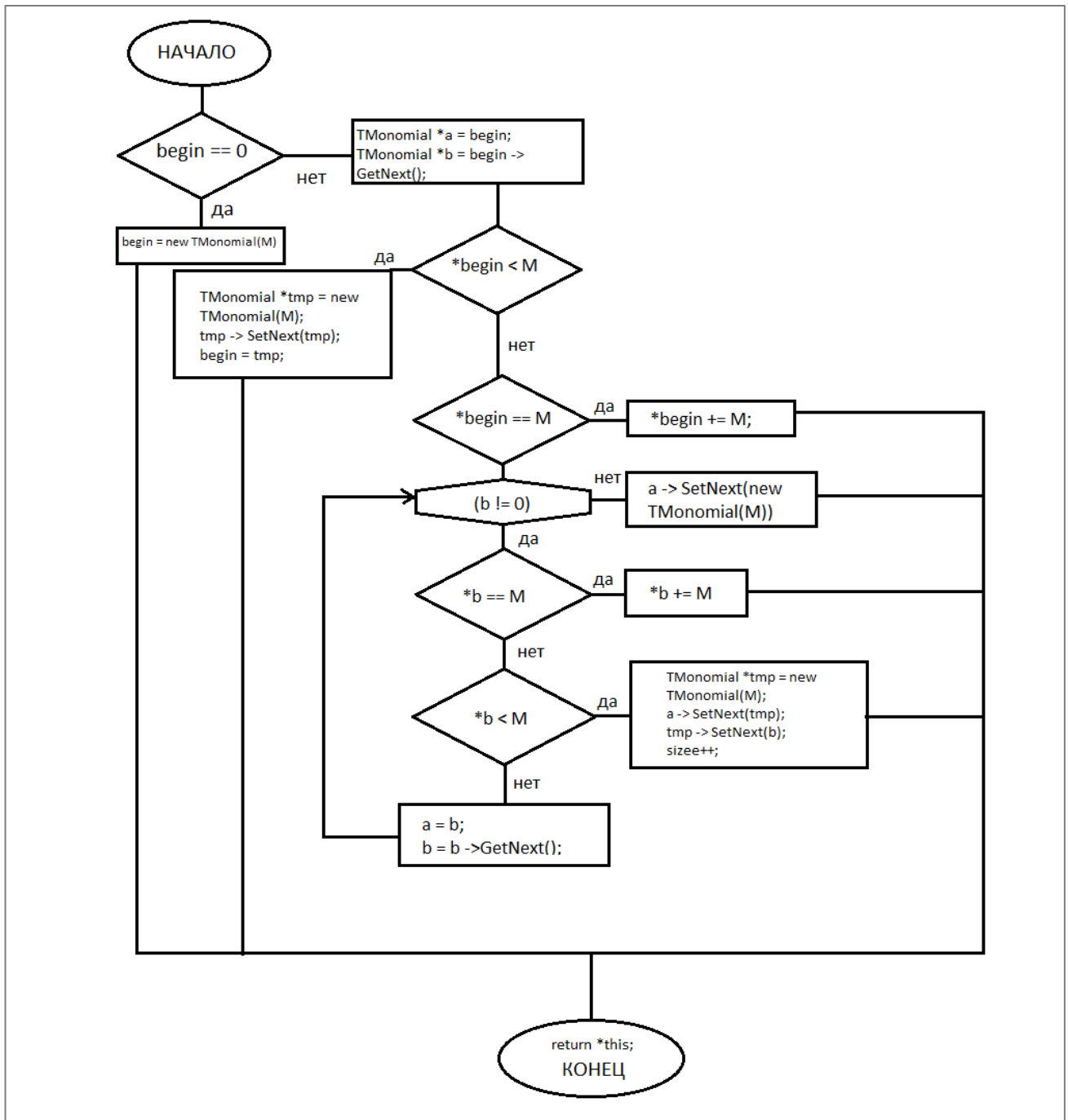


Рис. 3 – алгоритм сложения полинома и монома

Разность ищется аналогичным образом.

3. Сумма двух полиномов (operator +=).

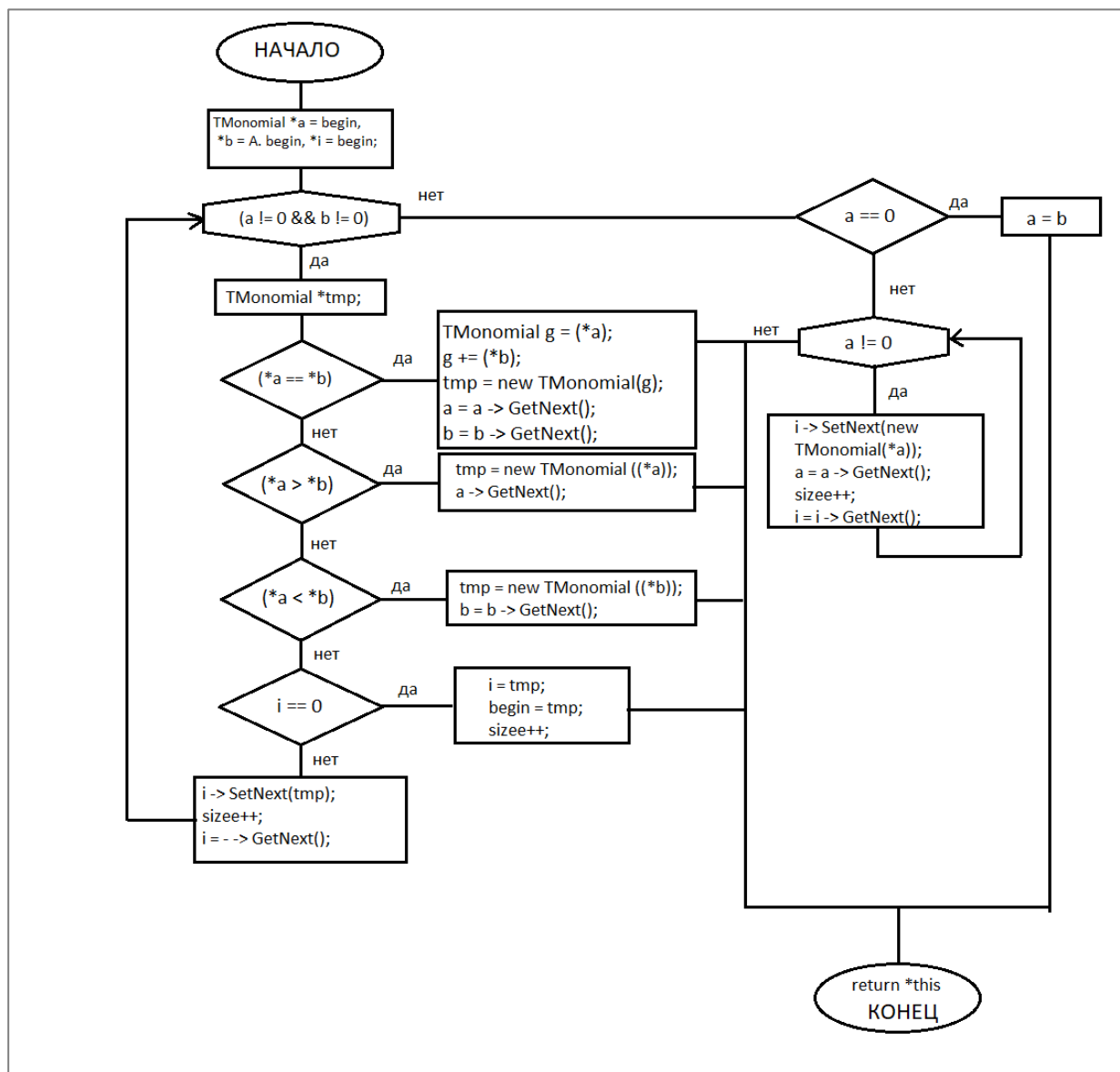


Рис. 4 – алгоритм сложения двух полиномов

5. Заключение

В процессе работы над данной лабораторной работой мне удалось разработать систему поддержки арифметических действий над полиномами.

Кроме того, было достигнуто более глубокое понимание принципов работы с данной структурой, принципов тестирования программы с помощью GT.

6. Список литературы

1. Гергель В.П. Методические материалы по курсу «Методы программирования 2», 2015. 2. Статья, посвященная теме списков [<https://prog-cpp.ru/data-ols/>]
2. Статья, посвященная теме Одночленов, в Википедии [<https://ru.wikipedia.org/wiki/Одночлен>]
3. Статья, посвященная теме списков [<https://tim4ous.com/realizatsiya-odnosvyaznogo-lineynogo-spiska-v-si/>]
4. Статья, посвященная теме списков [<https://prog-cpp.ru/data-ols/>]