Федеральное агентство по образованию Российской Федерации

Государственное образовательное учреждение

Высшего профессионального образования

Нижегородский государственный университет им. Н.И. Лобачевского

Институт информационных технологий математики механики

Выполнил:

студент ИИТММ гр. 381706-2

Антипин А.С.\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Проверил:

ассистент каф. МОСТ, ИИТММ

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

Полиномы

Отчет по лабораторной работе

Нижний Новгород

2018 г.

**Содержание**

[1.Введение 3](#_Toc1076410)

[2.Постановка целей и задач 4](#_Toc1076411)

[3.Руководство пользователя 5](#_Toc1076412)

[4.Руководство программиста 6](#_Toc1076413)

[4.1.Описание структуры программы 6](#_Toc1076414)

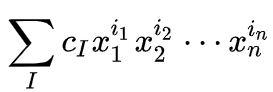
[4.2.Описание структур данных 8](#_Toc1076415)

[4.3.Описание алгоритмов 10](#_Toc1076416)

[5.Заключение 11](#_Toc1076417)

[6.Литература 12](#_Toc1076418)

# 1.Введение

В математике огромное значение имеет удобное для визуального представления и правильное с технической точки зрения представление выражений и функций для их дальнейшего решения. Полиномы является одним из таких способов. Полиномом от n переменных называется мономов или, строго конечная формальная сумма вида, где - набор из целых неотрицательных чисел, являющимися степенями x-ов, а - число, именуемое коэффициент монома. Изучение полиномиальных уравнений и их решений составляло едва ли не главный объект «классической алгебры». Многочлены также играют ключевую роль в алгебраической геометрии, объектом которой являются множества, определённые как решения систем многочленов. 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.

# 2.Постановка целей и задач

Целью лабораторной работы является создание структуры хранения типа «Полином» и методов работы с ним, таких как:

* Математические операции с полиномами;
* Доступ к отдельному моному в полиноме;
* Добавление к полиному мономов.

Для реализации алгоритмов будет использоваться 2 класса:

* Monom;
* Polinom.

Для проверки правильности работы этих классов будут написаны тесты с использованием фреймворка Google Test, а также тестовый образец программы, которая будет использует класс список.

# 3.Руководство пользователя

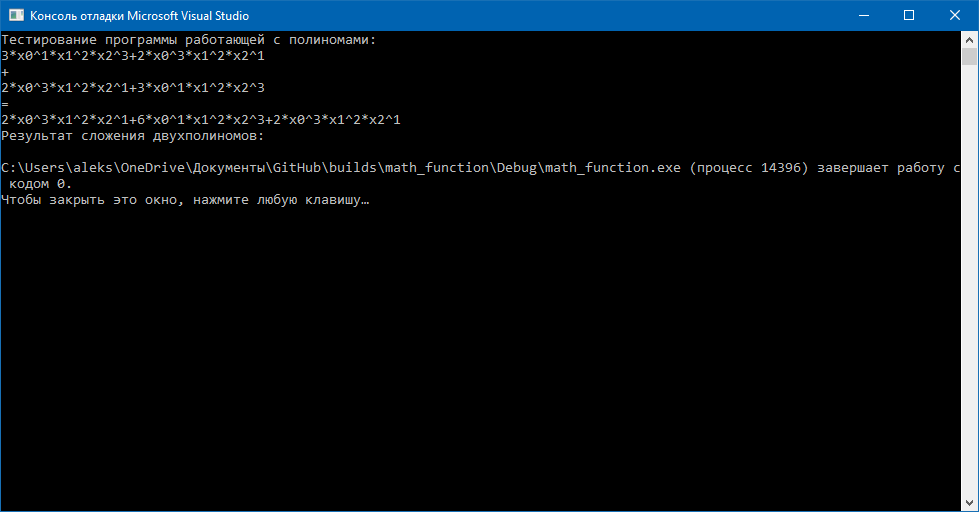
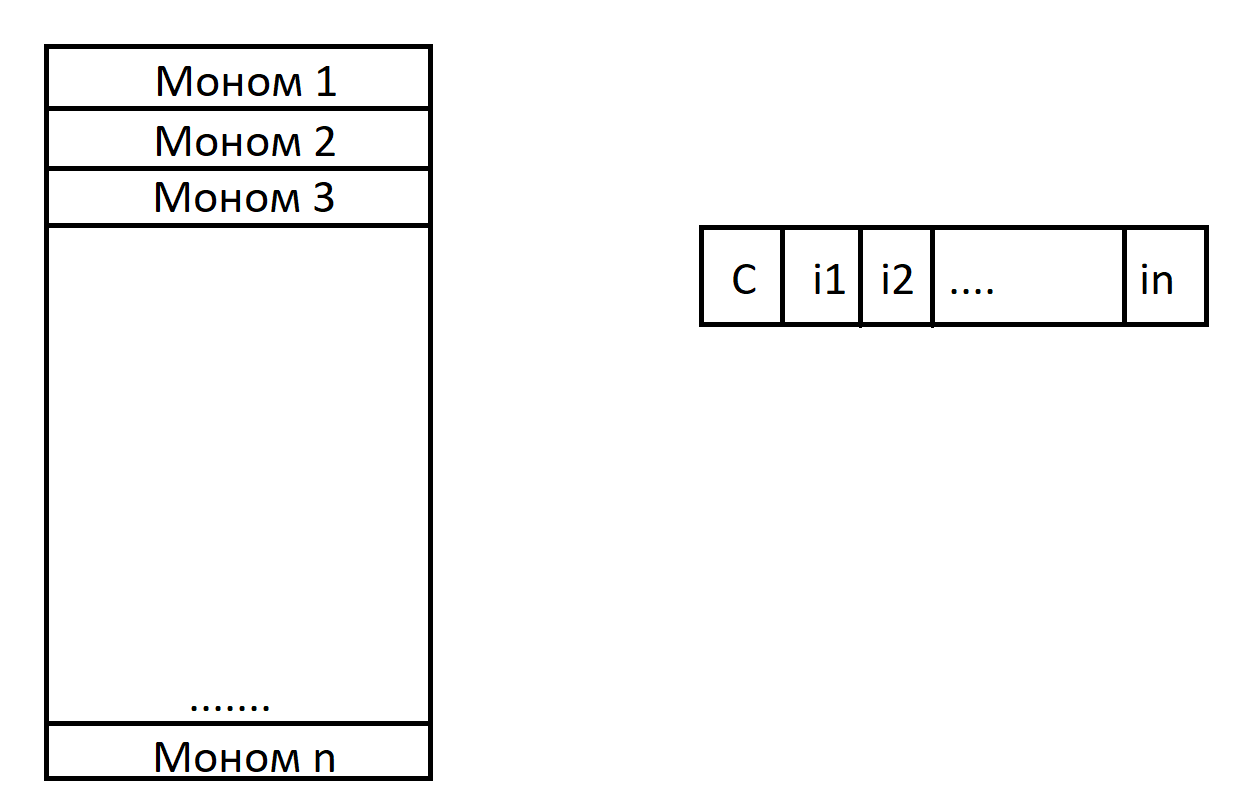
После запуска программы пользователя встречает консольное окно (рис. 1):

рис. 1 (вывод программы тестирования полиномов для пользователя)

в которой сначала приводятся два полинома, а затем выводится сумма этих полиномов.

# 4.Руководство программиста

## 4.1.Описание структуры программы

Класс «полином» будет реализован с помощью набора элементов «моном», соответственно каждый моном будет состоять из вещественного коэффициента и массива степеней x-ов:

Где с – коэффициент монома;

i1, i2, …., in – степени монома.

То есть для реализации алгоритмов будет использовано 2 класса:

* Класс «Моном» (Monom).
* Класс «Полином» (Polinom), который будет наследоваться от класса List с шаблоном в виде Monom.

А также проект использующий фреймворк Google Test, для проверки правильности работы этих классов и тесовый проект, который будет показываться пользователю.

Древо классов

**Класс gtest**

**test\_main.cpp**

**Monom\_test.cpp**

**Polinom\_test.cpp**

**Проект для пользователя**

**main.cpp**

**Polinom.h**

**Класс List**

**Monom.h**

**Класс TDatLink**

**Polinom.cpp**

**Monom.cpp**

**Класс Monom:**

Класс «Моном» содержит представление отдельных мономов полинома. Также в этом классе перегружены математические операции присущие мономам, такие как сложение, вычитание и др., а также операции сравнения.

**Класс Polinom:**

В классе «Полином» организована структура полинома – списка мономов. Был взят класс List и в качестве шаблона использовался моном. Здесь перегружены математические операции с многочленами и вывод этих многочленов на консоль.

**Класс gtest:**

Класс gtest реализует тестирование классов Monom и Polinom, по средствам фреймворка Google Test. Тесты пишутся для каждого метода классов, каждого ветвления этих методов и для всех возможных исключений этих методов.

**Проект List:**

В данном проекте реализован примет использования многочленов, в котором складываются два монома и результат выводится на консоль.

## 4.2.Описание структур данных

**Класс Monom:**

int n – количество переменных х;

double c – коэффициент монома;

unsigned int\* power – массив степеней монома;

Exceptions\_from\_polinom\_and\_calculator exception – переменная для вызова исключений;

**Класс Polinom:**

Exceptions\_from\_polinom\_and\_calculator exception – переменная для вызова исключений;

**Описание методов:**

|  |  |
| --- | --- |
| Метод: | Описание: |
| Monom::Monom() | Конструктор по умолчанию для класса Monom. |
| Monom::Monom(int \_n, unsigned int\* \_power, double \_c) | Конструктор с параметром для класса Monom. Первый параметр – количество переменных х, второй – степени этих переменных, третий – коэффициент монома. |
| Monom::Monom(const Monom& monom) | Конструктор копирования класса Monom. |
| Monom::~Monom() | Деструктор монома. |
| int Monom::GetN() | Метод, который возвращает количество переменных х. |
| double Monom::GetC() | Метод, который возвращает коэффициент монома. |
| unsigned int\* Monom::GetPower() | Метод, который возвращает указатель на массив степеней монома. |
| void Monom::SetN(int \_n) | Метод, который позволяет изменить количество переменных в мономе. Если их становится больше, то добавляются новые с единичными степенями, если меньше, то удаляются последние. |
| void Monom::SetC(double \_c) | Метод, который позволяет изменить коэффициент монома. |
| void Monom::SetPower(unsigned int\* \_power) | Метод, позволяющий изменить степени монома. |
| Monom& Monom::operator=(const Monom& monom) | Перегрузка оператора присваивания для монома. |
| Monom Monom::operator+(const Monom& monom) | Перегрузка операции сложения мономов. |
| Monom Monom::operator-(const Monom& monom) | Перегрузка оператора вычитание для мономов. |
| Monom Monom::operator\*(const Monom& monom) | Перегрузка оператора умножения для мономов. |
| bool Monom::operator==(const Monom& monom) | Перегрузка операции сравнения для мономов. |
| bool Monom::operator>(const Monom& monom) | Перегрузка операции больше для мономов. |
| bool Monom::operator<(const Monom& monom) | Перегрузка операции меньше для мономов. |
| std::ostream& operator<<(std::ostream& os, Monom& monom) | Перегрузка вывода монома на консоль. |
| Polinom::Polinom() : List<Monom>() | Конструктор по умолчанию для полинома, который наследуется от конструктора по умолчанию списка с шаблоном монома. |
| Polinom::Polinom(const Polinom& polinom) | Конструктор копирования для полинома. |
| Polinom::Polinom(const List<Monom>& ls) | Конструктор преобразования типа из списка мономов в полином. |
| Polinom& Polinom::operator=(Polinom& polinom) | Перегрузка оператора присваивания полиномов. |
| Polinom Polinom::operator+(Polinom& polinom) | Перегрузка оператора сложения для полиномов. |
| Polinom Polinom::operator-(Polinom& polinom) | Перегрузка оператора вычитания для полиномов. |
| Polinom Polinom::operator\*(Polinom& polinom) | Перегрузка оператора умножение для полиномов. |
| Polinom& Polinom::operator+=(Monom& monom) | Прибавление к текущему полиному монома. |
| Monom Polinom::operator[](const int nomber) | Доступ к отдельному моному в полиноме. |
| std::ostream& operator<<(std::ostream& os, Polinom& polinom) | Перегрузка вывода полинома на консоль. |

## 4.3.Описание алгоритмов

**Подробное описание некоторых методов**

Изменение количества переменных в мономе:

* Проверка переданного значения на не отрицательность;
* Копирование предыдущих значений степеней во временную переменную и очистка основного поля. Выделение нового количества памяти;
* Выяснение какая из переменных (старая или новая) больше;
* Копирование в заново выделенную память значений из временного хранилища.
* При необходимости добавление единичных степеней для новых переменных х.

Сложение полиномов:

* Проверка на совпадение длин полиномов, если длины не совпадают, то вызывается исключение;
* Создаются указатели на первые элементы полиномов, и результирующий полином;
* В цикле происходит проверка на равенство мономов в полиноме и в случае успеха, они складываются. Если моном в полиноме больше соответствующего монома в другом полиноме, то сначала записывается больший, а затем дописываются меньшие;
* Возвращается результирующий полином.

# 5.Заключение

В заключении можно сказать, что все поставленные цели и задачи были выполнены, а именно: созданы классы «Monom» и «Polinom» с реализованными математическими операциями над ними, а также написаны к ним тесты, и они успешно пройдены.

# 6.Литература

* Учебные материалы к учебному курсу «Методы программирования» - Гергель В.П.
* https://ru.wikipedia.org/wiki/Многочлен