Федеральное агентство по образованию Российской Федерации

Государственное образовательное учреждение

высшего профессионального образования

Нижегородский государственный университет им. Н.И. Лобачевского

Факультет информационных технологий математики и механики

Отчёт по лабораторной работе

**Лабораторная работа №5 Аналитические преобразования полиномов от нескольких переменных (списки)**

Выполнил:

студент ф-та ИТММ гр. 381808-2

Ратов Святослав.

Проверил:

ассистент каф. МО ЭВМ, ВМК

Панов А.

Нижний Новгород

2019 г.

Содержание

[Введение 3](#_Toc27613231)

[Постановка задачи 4](#_Toc27613232)

[Руководство пользователя 5](#_Toc27613233)

[Руководство программиста 6](#_Toc27613234)

[Описание структуры программы 6](#_Toc27613235)

[Описание структур данных 6](#_Toc27613236)

[Описание алгоритмов 7](#_Toc27613237)

[Эксперименты 9](#_Toc27613238)

[Заключение 10](#_Toc27613239)

[Литература 11](#_Toc27613240)

[Приложения) 12](#_Toc27613241)

[Приложение 2 12](#_Toc27613242)

# Введение

Наряду с привычным вычислительным применением компьютеры широко используются и для аналитической обработки данных. Среди примеров таких приложений – компьютерное доказательство теорем, логический вывод, анализ текстовой информации и многое другое. Среди таких примеров и задача обработки полиномов, задаваемых в общей аналитической форме. Полиномы являются хорошо изученной областью математики (алгебра полиномов), которая широко используется в приложениях (аппроксимация экспериментальных данных, построение функциональных зависимостей и т.п.).

Лабораторная работа направлена на изучение методов компьютерной обработки полиномов. С этой целью в лабораторной работе изучаются различные варианты структуры хранения и разрабатываются программы для обработки полиномов. Основной учебной целью работы является практическое освоение методов организации структур хранения данных с помощью списков. В ходе выполнения лабораторной работы разрабатывается общая форма представления линейных списков, разрабатываются программы работы со списками, которые могут быть использованы и в других областях приложений.

# Постановка задачи

В рамках лабораторной работы ставится задача реализации программ, обеспечивающих поддержку списков, и разработки программных средств, позволяющих создавать и производить аналитические операции с полиномами.

В программе должны быть реализованы следующие возможности:

* Ввод полиномов;
* Возможность выполнения операций между двумя полиномами;
* Форматированный вывод полиномов.

# Руководство пользователя

Для тестирования созданных классов можно использовать программу ввода-вывода полиномов и арифметических операций над ними, которая выводит введенные пользователем переменные и их степени и преобразует в полином, и также вычисляет результат операций с другим полиномом.

Для подключения класса «TPolynom» необходимо в рабочий проект добавить проект библиотеки

1. В свойствах проекта указать ссылку на библиотеку
2. Создать включение заголовочного файла «TPolynom.h», указав в свойствах его путь.

# Руководство программиста

## Описание структуры программы

Программа содержит в себе следующие проекты:

1. TPolynom.h, TPolynom.cpp – модуль, реализующийполином;
2. Test.cpp – модуль программы тестирования.

## Описание структур данных

В классе TPolynom содержится структура TMonom, которая необходима для реализации монома, и сам класс TPolynom, представляющий из себя полином. Данный класс позволяет вывести, введенный пользователем полином и выполнить с ним необходимые арифметические операции.

Класс включает в себя следующие поля:

* TMonom\* pHead; // заголовок
* int ListLen; // длинна списка

Класс позволяет работать со следующими методами и перегрузками:

* TPolynom(); // конструктор
* TPolynom(const TPolynom& tp); // конструктор копирования
* ~TPolynom(); // деструктор
* int GetListLength() const { return ListLen; } // к-во звеньев
* void ClearList(void); // очистить список
* void Insert(TMonom tm); // вставка в список
* void Insert(double coeff, int power);
* void Add(const TPolynom tp); // добавление полинома к другому
* void InsFirst(TMonom tm); // вставка в начало
* void DelFirst(); //удаление первого элемента
* void InsLast(TMonom tm); // вставка в конец
* void DelList(TMonom tm); // удаление элемента из списка
* int IsInList(int power); // проверка на наличие в списке
* int Calculate(int val); // вычисление
* TPolynom operator+(const TPolynom& q) const; // сложение
* TPolynom operator-(TPolynom& q); // вычитание
* TPolynom operator\*(const TPolynom& q); // умножение
* TPolynom operator\*(const double& q) const;
* const TPolynom& operator = (const TPolynom& q); // присваивание

## Описание алгоритмов

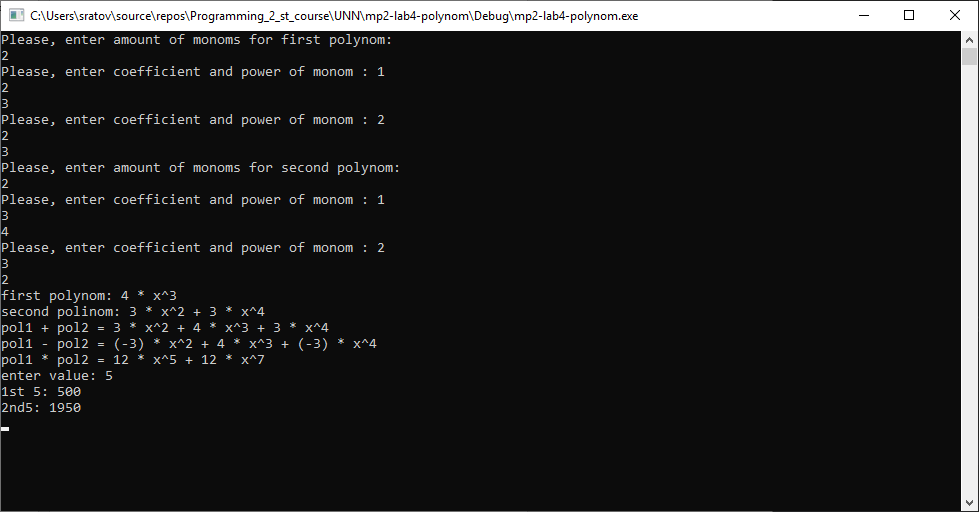
**Класс TPolynom:**

1. Конструктор  
    Выполняется инициализация pHead и ListLen нулями.
2. Конструктор копирования  
   Позволяет сделать копию полинома. Происходит поэлементная вставка указателей для коэффициэнтов и степеней и для следующих элементов. Так же копируется длина.
3. Деструктор  
   Вызывает метод ClearList()
4. Очистка списка  
   Происходит поэлементная очистка списка с помощью метода DelFirst()
5. Проверка на наличие в списке  
   Выполняется поэлементная проверка на наличие заданого power.
6. Вычисление значения  
   Позволяет вычислить значение полинома при заданом значении val.
7. Оператор сложения  
   Позволяет сложить один полином с другим. Происходит сложение мономов в полниномах.
8. Оператор вычитания  
   Позволяет вычесть один полином из другого.
9. Оператор умножения  
   Происходит умножение коэффициэнтов одного полинома на мономы второго полинома.
10. Оператор присваивания

Присваивает один полином другому.

1. Вставка в список  
   Вставляет переданый моном в список. Происходит добавление элементов. Длина списка увеличивается на 1.
2. Добавление другого полинома  
   Выполняет слияние двух полиномов. Если один элементы не совпадает, то происходит его добавление в другой полином.
3. Вставка в начало  
   Позволяет вставить элемент в начало списка. Увеличивает длину списка на 1.
4. Удаление первого элемента  
   Удаляет первый элемент в списке. Передвигает указатель и очищает память. Уменьшает длину списка на 1.
5. Вставка в конец списка  
   Позволяет вставить элемент в конец списка. Находит конец списка и добавляет указатель на новый элемент. Увеличивает длину списка на 1. При отсутствии заголовка - создает новый.
6. Удаление элемента из списка  
   Если модуль заголовка меньше малого значения, то происходит удаление первого элемента. Происходит поиск и удаление элемента из списка. Выполняется очистка памяти и уменьшение длины списка.
7. Оператор вывода  
   Осуществляет форматированный вывод.

# Эксперименты

1. Программа верно выводит, введенные пользователем полиномы и вычисляет их значение:   
   

# Заключение

В рамках лабораторной работы была реализованы программы, обеспечивающие поддержку списков, и также были разработаны программные средства, производящие обработку арифметических выражений, записи полиномов, построение полиномов и вычисление результата.

Данная программа может быть применена (с небольшими улучшениями) для решения задач аналитической алгебры и дискретной математики.

# Литература

1. Столлингс, В. Структурная организация и архитектура компьютерных систем, 5-е изд.: Пер. с англ. — М.: Издательский дом «Вильямс», 2002. — 896 с.: ил. — Парал. тит. англ.
2. Johnson M. Superscalar Microprocessor Design. — Englewood Cliff, New Jersey: Prentice Hall, 1991.
3. Касперски К. Техника оптимизации программ. Эффективное использование памяти. — СПб.: БХВ-Петербург, 2003. — 464 с.: ил.
4. Stone H. High performance Computer Architecture. — Reading, MA: Addison-Wesley, 1993.
5. Tullsen D.M., Eggers S.J. Effective Cache Prefetching on a Bus-Based Multiprocessor. — ACM Transactions on Computer Systems, pp. 57-88, Feb 1995.
6. Chandra D., Guo F., Kim S., Solihin Y. Predicting inter-thread cache contention on a chip multi-processor architecture. — Proceedings of the 11th International Symposium on High Performance Computer Architecture (HPCA), pp. 340–351, Feb 2005.
7. Press W., Teukolsky S., Vetterling W., Flannery B. Numerical Recipes in C. The Art of Scientific Computing. Second Edition. — Cambridge University Press, 1992.
8. Камаев А.М., Сиднев А.А., Сысоев А.В. Об одном подходе к анализу эффективности приложений // Труды 50-й научной конференции МФТИ «Современные проблемы фундаментальных и прикладных наук»: Часть I. Радиотехника и кибернетика. - М.: МФТИ, 2007.
9. Debugging and performance monitoring. Intel® 64 and IA-32 Architectures Software Developer’s Manual. Volume 3B: System Programming Guide, Part 2. May 2007. — [http://www.intel.com/products/processor/manuals/]
10. Юнаковский А.Д. Начала вычислительных методов для физиков. – Н. Новгород: ИПФ РАН, 2007.