

# Отчет-презентация по лабораторной работе №2

## «Алгебра полиномов»

**Выполнили:** студенты группы 3821Б1ПР2

Винокуров Иван Дмитриевич

Карагодин Андрей Романович

**Проверила:** ассистент кафедры МОСТ, Усова М.А

# Постановка задачи

Разработать программную систему для выполнения алгебраических операций над полиномами от трех переменных.

Условия/требования:

1. полиномы хранятся в виде списка;
2. полиномы хранятся во всех таблицах одновременно, ключом является имя;
3. таблиц должно быть 6 видов: линейная на массиве, линейная на списке, упорядоченная на массиве, дерево (авл или красно-черное), две хэш-таблицы;
4. операции над отдельным полиномом: вычисление в точке, умножение на константу, производная.
5. операции в выражениях из полиномов: сложение, вычитание, умножение на константу, умножение полиномов, деление полиномов. операции должны выполняться, используя постфиксную форму;
6. операции над таблицами: добавление полинома (во все сразу), удаление полинома (во всех сразу), поиск (только в активной таблице, выполняется в процессе вычисления выражений, составленных из имен полиномов);
7. активная (выбранная пользователем) таблица должна выводиться на экран в формате, как минимум, двух столбцов: 1) имя полинома, 2) строковое представление полинома.

## Что имеется:

- Рабочие классы таблиц, полиномов, мономов, интерфейса таблиц.
- Свой условный синтаксис для записи выражения. К сожалению, не удалось применить обратную польскую запись к полиномам.
- Визуальный интерфейс.
- Множество бессонных ночей.

# Работа Полиномов

## Базовый принцип работы с классом

- Для полиномов требуется выражение в видео строки формата:  
*(моном) операнд (моном) операнд ...*
- Данная строка преобразуется в список мономов при помощи метода Parse(std::string). Список мономов хранится внутри объекта класса CPolynomial.
- Используя метод toString() пользователь может вывести список мономов в том виде, как он хранится в объекте класса.
- Чтобы вычислить значение полученного выражение в точке, необходимо иметь 4 аргумента: Значения (double) x, y, z и исходное выражение std::string.
- Метод findResult использует значения в точке (x,y,z) и исходное выражение вычисления значения самого монома в точке и счёта операнда для выполнения определенного действия со следующим мономом.

```
Hello! Let's calculate Polynomial strings. Format of string: (Monomial expr) +-* / (Monomial expr) +-* / ...
Your Polynomial string: (2x^2y) + (3x^2yz) + (4xy^2)
This is how the class contains the Monoms of string Polynomial: 2x^2*y^1 3x^2*y^1*z^1 4x^1*y^2
Now let's calculate the result of said Polynomial string:
Input x: 2

Input y: 3

Input z: 4

The result of (2x^2y) + (3x^2yz) + (4xy^2) In point 2 3 4 is: 240
```

```

//Example of inner workings of Polynomial. Basic HOW-TO-DO-WORK with it.
#include <iostream>
using namespace std;

int main()
{
    std::string userstring = "(2x^2y) + (3x^2yz) + (4xy^2)";
    double x, y, z;

    std::cout << "Hello! Let's calculate Polynomial strings. Format of string: (Monomial expr) +-* / (Monomial expr) +-* / ..." << std::endl;
    std::cout << "Your Polynomial string: " << userstring;
    CPolynomial usertest;
    usertest.Parse(userstring);

    std::cout << std::endl << "This is how the class contains the Monoms of string Polynomial: " << usertest.toString() << std::endl;
    std::cout << "Now let's calculate the result of said Polynomial string: " << std::endl;
    std::cout << "Input x: ";
    std::cin >> x;

    std::cout << std::endl;
    std::cout << "Input y: ";
    std::cin >> y;

    std::cout << std::endl;
    std::cout << "Input z: ";
    std::cin >> z;

    std::cout << std::endl;
    std::cout << "The result of " << userstring << " In point " << x << " " << y << " " << z << " " << "is: " << usertest.findResult(x, y, z, userstring);
    return 0;
}
```

# Краткие детали работы парсера

- Парсер выполняет работу по разбору строки в список мономов следующим образом:
  1. Подается строка и проверяется первый же символ ‘(’. Если найден, значит, перед нами начинается моном и записываем скобку в стек.
  2. Если комбинация последующих символов является числом — записываем в коэффициент.
  3. Если следующий символ является алфавитным символом, то в зависимости от значения (x, y или z) вычисляется индекс степени с которым будем далее работать (0, 1, 2 соответственно) и увеличивается на 1.
  4. Если следующий символ - знак степени ‘^’, то считываются следующие символы (число) и записываются в степень соответствующей переменной, с которой происходила работа.
  5. Если символ ‘)’, убираем из стека скобку ‘(’ и добавляем составленный моном в список хранения мономов в классе. Очищаем рабочий объект монома для записи следующих.
  6. Повторение с шага 1, пока не достигнем последнего символа строки.

# Краткие детали работы метода вычисления

- Простой метод вычисления:
  1. Берем из списка первый моном и вычисляем его значение в точке.
  2. Идем по строке и считываем операнд.
  3. Достаем из списка следующий моном и вычисляем значение в точке.
  4. В зависимости от операнда результаты мономов складываются/вычитаются/перемножаются/делятся.
  5. Сохраняем результат и идем дальше с шага 2, пока не достигнем конца строки.

## Некоторые замечания по классу.

- Во всех классах перегружены операторы `==`, `!=`, `=`, `+`, `-`, `*` monom, `*` coef, `/`.
- Во всех классах имеются методы `toString()`.
- В классе `Spolynomial` имеются служебные методы, позволяющие преобразовать строку в обратную польскую запись. А также служебный метод вычисления монома в точке.
- В классе `Cmonomial` имеется метод `getDerivative`, позволяющий вычислить производную по 3-м переменным.

# Таблицы

- Все таблицы одновременно хранят значение `std::string` (ключ) и сам объект Полинома.
  - Все таблицы имеют как минимум 4 метода:
    1. Найти по ключу.
    2. Удалить объект.
    3. Вставить объект.
    4. Находится ли объект в таблице.



# 7 Видов таблиц

## • Itable

- Фактически интерфейс для таблиц.
- Содержит виртуальные методы для всех таблиц.

## • CHashTableMix

- Представляет собой таблицу на векторе.
- Имеет хеш-функцию, позволяющую открыто записывать или находить элемент в векторе по хеш-ключу.

## • CHashTableList

- Представляет собой таблицу на списке.
- Имеет хеш-функцию, позволяющую записывать или находить элемент в списке по хэш-ключу.

## • COrderedTableArray

- Представляет собой упорядоченную таблицу на массиве.
- Есть фактический размер памяти и реальный размер таблицы (сколько ячеек массива заполнено). Порядок следования элементов в таблице определяется упорядоченностью ключей.

## • CLinearTable List

- Представляет собой неупорядоченную таблицу на списке.
- Поиск осуществляется проходом по списку и сравнением ключей.

## • CLinearTable Array

- Представляет собой линейную таблицу на массиве.
- Поиск осуществляется проходом по массиву и сравнением ключей.

## • CTreeTable

- Представляет из себя таблицу на поисковом дереве.
- Дерево поиска является АВЛ-деревом, если для каждой его вершины высота левого и правого поддеревьев отличается не более, чем на единицу.

# Ответственности в проекте

1. Винокуров Иван - Интерфейс. Линейная таблица на массиве. Таблица на поисковом дереве (AVL).  
Упорядоченная таблица на массиве. Хеш-таблица со списками (метод цепочек).
2. Андрей Карагодин - Полином (доработанная до 3 переменных версия). Линейная таблица на списке. Хеш-таблица с открытым перемешиванием. Общий интерфейс таблиц. Постфикс для полиномов (доработанная обратная польская запись).

Спасибо за внимание