Polinom3

# Вспомогательные классы

**PolinomItem** - часть многочлена, содержащая название переменной и степень при данной переменной (x^3, y^2, z^0 и т.д.)

-get/set методы для степеней переменных

-equals

-toString

**PolinomListItem** - многочлен (элемент списка многочленов), состоящий из 3-х PolinomItem`ов, коэффициента (3 \* x^2 \* y^1, -1 \* x^3 и т.д.) и ссылки на следующий PolinomListItem; Реализует Comparable для соблюдения лексикографического порядка в полиноме.

-get/set методы для степеней переменных, коэффициента и ссылки на следующий элемент списка

-compareTo

-equals

-toString

-copy - возвращает копию объекта с теми же атрибутами и ссылкой на следующий элемент, равной null.

**PolinomList** - список многочленов, состоящий из PolinomListItem. Содержит ссылку на голову списка и размер списка

**Методы :**

-**add**(PolinomListItem), -**add**(int coef, int deg1, int deg2, int deg3)

Добавляет новый многочлен с список или заменяет уже существующий. Управляет размером списка. Для поиска подходящей позиции используются линейный поиск и метод compareTo у PolinomListItem. Сложность по времени: T = O(N), тк как используем линейный поиск. Сложность по памяти - M =O(1).

-**delete**(PolinomListItem), -**delete**(int coef, int deg1, int deg2, int deg3)

Находит и удаляет многочлен с соответствующими степенями, если такого нет - завершает работу метода. Управляет размером списка. Использует линейный поиск.

T = O(N)

M = O(1)

-**equals**(Object o)

Сравнивает 2 объекта. Использует линейный поиск для сравнения соответствующих элементов.

T = O(N)

M = O(1)

# Класс Polinom3

Содержит в себе ссылку на список многочленов (PolinomList).

**Методы:**

-**Polinom3**(String filename)

Формирует PolinomList на основе файла, лежащем на пути filename. Использует Scanner. Внутри файла данные о многочлене хранятся в следующем виде:

**коэффициент степень1 степень2 степень3**

Метод считывает очередную строку, делит ее по пробелам, парсит числа и добавляет соответствующий PolinomListItem в список. Использует массив для хранения промежуточных данных.

Сложность по времени: T = O(N), тк считывает все данные из файла последовательно; M = O(N).

-**insert**(PolinomListItem), -**insert**(int coef, int deg1, int deg2, int deg3)

Работает аналогично add в PolinomList. По аналогии,

T = O(N)

M = O(1)

-**add**(Polinom3 polinom)

Складывает 2 полинома. Использует метод двух указателей для слияния двух упорядоченных списков многочленов

Благодаря методу двух указателей имеет временную сложность T = O(K + N), где K и N - размер списков складываемых полиномов. Для работы использует список длины N + K, следовательно пространственная сложность

M = O(N)

-**delete**(int xDeg, int yDeg, int zDeg)

Находит и удаляет в списке элемент с соответствующими степенями. Работает аналогично delete из PolinomList.

T = O(N); M = O(1)

-**derivative**(int i)

Взять производную по i-ой переменной исходного полинома

Для каждого многочлена из списка высчитываем производную по заданной переменной и записываем в новый список многочленов.

Алгоритм вычисления производной:

1.При помощи тернарных операторов находим нужную нам степень переменной.

2.Если степень не равна нулю, умножаем коэффициент на текущую степень, понижаем степень и записываем получившийся многочлен в новый список. Если степень соответствующей переменной равна нулю, то считаем многочлен константой, производная которой равна нулю.

Последовательно высчитывает производные и использует метод add из PolinomList. Создает новый лист, максимальная длина которого не превышает N.

T = O(N)

M = O(N)

-**value**(int x, int y, int z)

Вычисляет значение полинома в точке (x,y,z)

Последовательно вычисляет значения многочленов в точке и складывает их. Возводим в соответствующие степени переменные и перемножаем их с коэффициентом. Использует Math.pow;

3 \* x^2 \* y^1 \* z^1 -> (1, 2, 3) -> 3 \* 1^2 \* 2^1 \* 3^1 = 18

T = O(N)

M = O(1)

# toString

Чтобы понять как работает toString Polinom3, рассмотрим toString вспомогательных классов.

**PolinomItem**

Печатает название переменной и ее степень через символ “^”(“x^2”, “y^5”, “z^1” и т.д).

**PolinomListItem**

Печатает многочлен, используя toString из PolinomItem, а так же модуль его коэффициента; Печатает знак “\*”, если PolinomItem не является первым; Если степень PolinomItem = 0, он не печатается.

(“5 \* x^2”, “1 \* x^2 \* y^1”, “4” и т.д)

**PolinomList**

Используется в Polinom3. Печатает весь полином. Последовательно распечатывает многочлены, используя toString из PolinomListItem. Печатает “ - “ или “ + ” между многочленами в зависимости от знака коэффициента после них.

T = O(N)

M = O(1)