Relatório 3

Problema 3 - Sparse Polynomials

Grupo 10

Alexandra Brito nº50075, Cyrill Brito nº52875, Gonçalo Pedras nº51702

Introdução:

Para a resolução deste problema foi necessário conhecer e saber aplicar alguns cálculos algébricos, para fazermos a soma e a multiplicação dos polinómios.

Um polinómio é constituído pela união de duas ou mais variáveis e constantes, relacionadas através de operações de multiplicação, subtração ou adição. Um termo é o conjunto de variável constate e o seu expoente. Para calcula o valor do polinómio num ponto substituímos a variável pelo valor do ponto.

Para a soma de polinómios é feita a soma dos coeficientes dos termos com expoentes iguais. Como é exemplificado abaixo:

$$(2x^3 - 9x) + (4x^3 + 5) = 6x^3 - 9x + 5$$

Em amarelo estão os termos de expoente igual, sendo estes somados.

Na multiplicação de polinómios são multiplicados todos os termos de um polinómio com os termos do outro polinómio, ou seja, a propriedade distributiva:

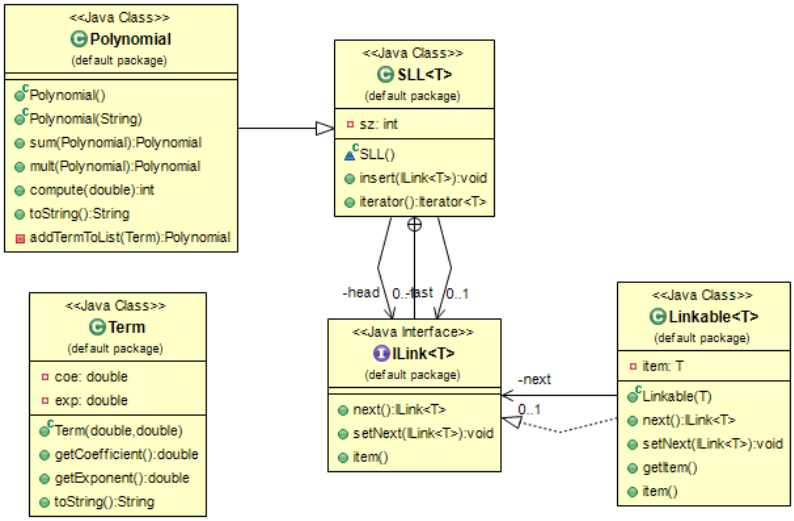
$$(a + b)(c + d) = ac + ad + bc + bd$$

Para multiplicar dois termos multiplica-se os coeficientes e soma-se os expoentes.

$$2x^3 * 9x = (2 * 9)x^{3+1} = 18x^4$$

Diagrama UML:

Este diagrama UML foi criando através do addon sugerido pelo professor, ObjectAid. Aqui está demostrado todas as classes do programa com respetivos atributos e funções.



Classes:

- Classe SLL<T>

A classe SLL, simple linked list, é uma classe que permite fazer listas de qualquer tipo de objetos, implementa iterable, o que facilita a sua utilização. Esta classe foi fornecida pelo professor nos documentos de apoio. Na classe modificamos o *insert* para que ao inserir objetos na lista estes sejam inseridos no final em vez do principio.

- Classe Linkable<T>

Esta classe é necessária para fazer a ligação entre objetos e a classe *SLL*, ou seja, em vez de inserimos o objeto na lista, criamos um *Linkable* do objeto que queremos inserir na lista e inserimos o *Linkable* na lista. Esta classe também foi nos fornecida pelo professor nos documentos de apoio.

- Classe Term

Nesta classe é criado um termo do polinómio, contem as variáveis coeficiente e expoente.

- Classe Polynomial

A classe é uma lista (*SLL*) de termos (*Term*). É utilizada para guardar um polinómio. Alem disso tem métodos para somar e multiplicar polinómios e também para calcular o valor do polinómio no ponto.

Testes unitários:

No teste 1 inserimos dois polinómios (A e B) e testamos a soma e a multiplicação dos polinómios. Calculamos também o valor do polinómio A no ponto 0 e o valor do polinómio B no ponto 10.

```
@Test
public void test1() {
    Polynomial polyA = new Polynomial("1 1024 1 0");
    Polynomial polyB = new Polynomial("1 1");

    assertEquals(polyA.sum(polyB).toString(), "1 1024 1 1 1 0"); // Testing sum
    assertEquals(polyA.mult(polyB).toString(), "1 1025 1 1");    // Testing multiply

    assertEquals(polyA.compute(0), 1);   // Testing compute polyA
    assertEquals(polyB.compute(10), 10); // Testing compute polyB
}
```

No segundo teste inserimos dois polinómios onde um deles é nulo. Testamos a soma e a multiplicação dos polinómios. Calculamos também o valor do polinómio B no ponto 1.

```
@Test
public void test2() {
    Polynomial polyA = new Polynomial();
    Polynomial polyB = new Polynomial("2 100 1 10 3 1 2 0");

    assertEquals(polyA.sum(polyB).toString(), "2 100 1 10 3 1 2 0"); // Testing sum
    assertEquals(polyA.mult(polyB).toString(), "");                  // Testing multiply

    assertEquals(polyB.compute(1), 8); // Testing compute polyB
}
```

No ultimo teste inserimos dois polinómios (A e B) em que a soma dos termos criasse um termo nulo, ou seja, de confidente zero. Foi testada a soma e a multiplicação dos polinómios.

```
@Test
public void test3() {
    Polynomial polyA = new Polynomial("1 10 -1 5 8 3");
    Polynomial polyB = new Polynomial("1 5 -8 3");

    assertEquals(polyA.sum(polyB).toString(), "1 10"); // Testing sum
}
```

Discussão:

Para a resolução deste problema foi nos proposto a utilização das classes *SLL* e *Linkable.* Apos analise do problema chegamos á conclusão que a maneira mais correta de abordar o problema seria criar uma classe *Term* e uma classe *polynomial*, em que esta seja constituída por uma lista de termos(*Term*).

Para fazermos a soma e multiplicação tentamos vários métodos que acabaram por ser demasiado confusos e/ou complexos, tornando-se muito deles em falhanços. Apos muito deliberação encontramos um método mais simples de resolver o problema. Tirando inspiração de funções como lastsort, criamos o método *addTermToList,* que soma um termo a um polinómio que esteja ordenado por expoentes. A soma de polinómios reduz-se a utilizar o este método para todos os termos do polinómio A no polinómio B. A multiplicação baseia-se em aplicar a propriedade distributiva, multiplicar os termos e usar a função *addTermToList* para adicionar o termo a uma nova lista.

No calculo do valor do polinómio num ponto, criamos um ciclo em que cada termo é calculado e somado para o resultado final.

Referencias:

http://ganuff.weebly.com/uploads/1/9/2/5/19255685/reviso_de_polinomios.pdf

https://en.wikipedia.org/wiki/Horner%27s_method#Citations

http://www.infoescola.com/matematica/adicao-subtracao-e-multiplicacao-de-polinomios/