

Lista de Exercícios

MC302 - Programação Orientada a Objetos
Instituto de Computação
Universidade Estadual de Campinas

Classe, Herança e Polimorfismo
André Santanchè
2011

Questão 1

Considere um polinômio de grau n :

$$P(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x^1 + a_0 x^0$$

Escreva uma classe `Termo` que represente um termo deste polinômio com os seguintes métodos:

<code>construtor</code>	Recebe dois parâmetros : a_i e i , e cria um objeto em memória na forma $a_i x^i$.
<code>insere</code>	Recebe um objeto da classe <code>Termo</code> e substitui os valores $a_i x^i$ do termo corrente por aqueles do termo recebido como parâmetro.
<code>calcula</code>	Recebe um valor de x como parâmetro e retorna o valor do termo calculado.

Escreva uma classe `Polinomio` que representa polinômio completo na forma de uma sequência de objetos da classe `Termo`, com os seguintes métodos:

<code>construtor</code>	Recebe um objeto da classe <code>Termo</code> e cria um polinômio em memória na forma: $P(x) = a_i x^i$.
<code>insere</code>	Recebe um objeto da classe <code>Termo</code> e adiciona o termo $a_i x^i$ ao polinômio recebido como parâmetro. O polinômio pode ter um termo $a_q x^q$ cujo valor de q seja igual a i , neste caso a função deve unificar ambos em um único termo.
<code>calcula</code>	Recebe um valor de x como parâmetro e retorna o valor de $P(x)$.
<code>fusao</code>	Recebe como parâmetro outro objeto da classe <code>Polinomio</code> e realiza a fusão do polinômio recebido como parâmetro com o polinômio corrente.

Acrescente os métodos que achar necessários nas classes solicitadas.

Exercício inspirado em exemplo dos slides de prof. Tomasz Kowaltowski : “Estruturas de Dados e Técnicas de Programação”, 2010.

Questão 2

Dada a seguinte interface:

```
public interface Calculavel {
    public void insere(Termo termoInserido);
    public float calcula();
}
```

Modifique as classes da questão anterior para que implementem a interface `Calculavel`.

Escreva uma classe `PolinomioComSub` que represente um polinômio com sub-polinômios. Considere o seguinte exemplo:

$$P(x) = ((a_5 x^5 + a_3 x^3) + b_0 x^0) + c_1 x^1 + (d_3 x^3 + d_1 x^1)$$

Neste exemplo, o polinômio $P(x)$ contém 3 sub-polinômios. Os parênteses aparecem apenas para identificar sub-polinômios dentro de polinômios. O primeiro sub-polinômio, por sua vez, contém dois sub-

polinômios.

A classe `PolinomioComSub` deve ser capaz manter uma lista de termos (objetos da classe `Termo`) ou sub-polinômios (objetos da classe `Polinomio` ou `PolinomioComSub`). Ela possui os seguintes métodos:

<code>insere</code>	Recebe um objeto da classe <code>Termo</code> e adiciona o termo $a_i x^i$ ao polinômio recebido como parâmetro. O polinômio pode ter um termo $a_q x^q$ cujo valor de q seja igual a i , neste caso a função deve unificar ambos em um único termo.
<code>insere</code>	Recebe um objeto da classe <code>Polinomio</code> ou <code>PolinomioComSub</code> e o adiciona como sub-polinômio.
<code>calcula</code>	Recebe um valor de x como parâmetro e retorna o valor de $P(x)$.
<code>fusao</code>	Não recebe parâmetros. Realiza uma fusão de todos os sub-polinômios do objeto corrente, de tal modo que este objeto corrente passe a ter um polinômio sem sub-polinômios, sintetizando os termos de todos os sub-polinômios que sofreram fusão.
<code>fusao</code>	Recebe como parâmetro um objeto da classe <code>Polinomio</code> ou <code>PolinomioComSub</code> e se comporta como o método função anterior, acrescentando o objeto recebido como parâmetro no processo.

Acrescente os métodos que achar necessários nas classes e interfaces solicitadas. Em alguns casos pode ser interessante definir novas interfaces ou sub-interfaces. Analise a possibilidade de adaptar a classe `PolinomioComSub` para que se torne herdeira de `Polinomio` e avalie as vantagens e desvantagens. Decisões de re-projeto fazem parte desta questão.