



## Lista 02 - UML

1. Modele e escreva, em Java ou C++, uma classe capaz de definir polinômios

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

Cada termo do polinômio deve ser representado como um objeto de uma classe Termo.

A classe Polinomio deve possuir os seguintes métodos:

- um construtor que recebe o grau máximo do polinômio;
- um método Add que adiciona um termo  $a_n x^n$ , o qual deve garantir que o grau máximo do polinômio seja respeitado; caso um termo adicionado  $a_n x^n$  já exista, ele deverá ser somado ao termo já existente somando-se os valores de seus coeficientes ( $a_{n\_existente} + a_{n\_novo}$ );
- um método Mostra que exibe o polinômio
- um método Calcula que recebe um valor de X e retorna o valor calculado.

Use o ArrayList do Java, ou o std::list do C++; ou qualquer outra estrutura que julgar adequada.

2. O sistema operacional usa drivers de dispositivos de rede, de impressão, e de vídeo. Todos eles possuem funcionalidades e dados comuns como `ligaDispositivo`, `verificaStatus`, e `executaTeste`. Todos eles possuem funcionalidades específicas, como `enviaPacoteDeDados`, `imprimePaginas`, e `alteraBrilhoDeExibição`.

Faça a correspondente modelagem UML usando herança e escreva o esboço de código Java ou C++. Defina atributos necessários para representar o estado dos dispositivos após cada operação.

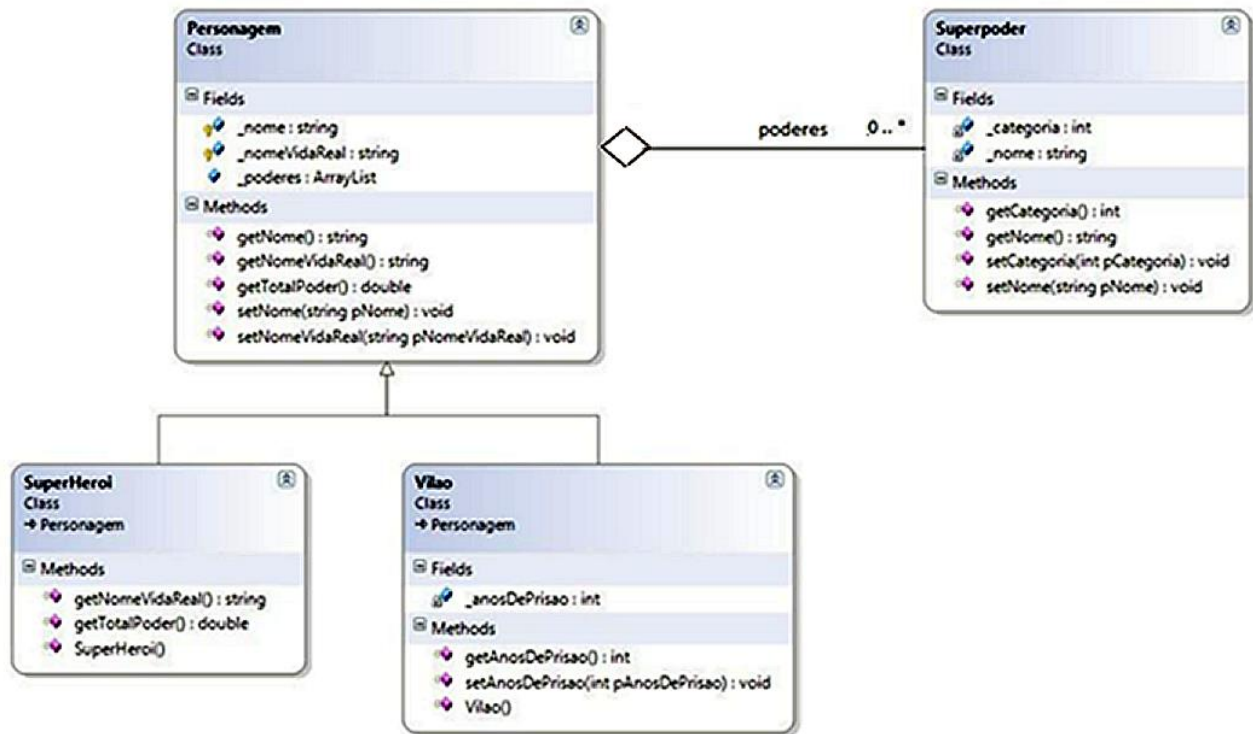
3. Modele em UML e implemente em Java a seguinte descrição.

Um software de cálculo matemático possui diversas funcionalidades. Uma delas se refere a uma operação matemática entre dois operandos do tipo float, a qual pode ser: soma, subtração, produto, ou coeficiente.

- A operação matemática possui o método “realizar cálculo” que recebe um inteiro indicando qual operação será realizada, e retornando o resultado da respectiva operação.
- O resultado da operação é descrito pelas seguintes informações: usuário que realizou a operação; o tipo da operação; além da data (3 inteiros: dia, mês, e ano) da operação.
- Um usuário é descrito por seu nome, idade, e CPF.

Em sua implementação Java, no programa principal, crie alguns objetos do seu modelo, realize algumas operações, e exiba dados dos resultados.

4. Considere o seguinte diagrama, no qual há uma hierarquia de classes Personagem->SuperHeroi, e Personagem->Vilao; além de uma agregação da classe Personagem com a classe Superpoder, isto é, um Personagem pode ter nenhum ou vários Superpderes.



Implemente este projeto em Java acrescentando as seguintes características:

- uma propriedade (atributo) "Vida" em Personagem que registre a quantidade de energia que ele possui – use os métodos de acesso (Get e Set);
- um método Atacar para os personagens, o qual deve receber como parâmetro a "intensidade do ataque", o nome do SuperPoder usado (o qual deve existir), e um objeto Personagem, o qual irá receber o ataque; este método deverá tirar (subtrair) "intensidade do ataque" da vida do personagem atacado com probabilidade 50% (use o método Math.random()).

No programa principal, escreva um loop onde dois personagens irão duelar até que um deles morra (vida == 0).