

## **Programação Orientada a Objetos**

### **2ª Lista de Exercícios**

1. Desenvolva, usando a UML ( diagrama de classe), uma classe que modele um objeto **conta poupança** em conformidade com o paradigma orientado a objeto. Posteriormente implemente esta classe. A classe deverá ter as seguintes características: nome do cliente, número da conta, agencia, saldo, sacar dinheiro e depositar dinheiro.
2. Desenvolva, usando a UML (diagrama de classe), uma classe que modele um objeto **triângulo** em conformidade com o paradigma orientado a objeto. Posteriormente implemente esta classe. A classe deverá ter as seguintes características: três lados, cálculo da área e encontrar os tipos do triângulo. Obs.: os valores dos lados não podem ser negativos e tem que formar um triângulo - cada lado individualmente tem que ser menor que a adição dos outros dois lados. Observações gerais de um triângulo:
  - quanto aos lados, pode ser equilátero ou isósceles ou escaleno;
  - quanto aos ângulos, pode ser acutângulo( $A^2 < B^2 + C^2$ ) ou obtusângulo( $A^2 > B^2 + C^2$ ) ou retângulo ( $A^2 = B^2 + C^2$ );
  - área =  $\text{raizQuadrada}(T * (T - A) * (T - B) * (T - C))$  onde  $T = (A + B + C) / 2$ .
3. Desenvolva, usando a UML (diagrama de classe), uma classe que modele um objeto **equação do segundo grau**  $ax^2 + bx + c$  em conformidade com o paradigma orientado a objeto. Posteriormente implemente esta classe.
4. Desenvolva, usando a UML ( diagrama de classe), uma classe que modele um objeto **cpf** em conformidade com o paradigma orientado a objeto. Posteriormente implemente esta classe.
5. Desenvolva, usando a UML ( diagrama de classe ), uma classe que modele um objeto **calculadora** em conformidade com o paradigma orientado a objeto. Posteriormente implemente esta classe. obs. O objeto calculadora tem as seguintes funcionalidades: adição, subtração, multiplicação, divisão, raiz e potência.
6. Desenvolva, usando a UML ( diagrama de classe ), uma classe que modele um objeto **número complexo** em conformidade com o paradigma orientado a objeto. Posteriormente implemente esta classe. O objeto número complexo tem as seguintes funcionalidades: inicialização do número, adição, subtração, multiplicação, divisão e igualdade.
  - i. um número complexo possui a seguinte forma: parte real + parte imaginária \* i ( $a+bi$ ) onde i é a raiz quadrada de -1;
  - ii. inicializar o número, recebe dois valores como argumentos para inicializar os campos da classe (parte real e imaginária);
  - iii. obter o número, devolve o número complexo encapsulado usando a notação  $a + bi$ ;
  - iv. verificar a igualdade, recebe uma outra instância da classe Número Complexo e retorna true se os valores dos campos encapsulados forem iguais aos da instância passada como argumento;
  - v. adicionar, recebe uma outra instância da classe Número Complexo e adiciona este número complexo com o encapsulado usando a fórmula  $(a+bi)+(c+di) = (a+c)+(b+d)i$  e devolve um novo Número Complexo como resposta;
  - vi. subtrair, recebe uma outra instância da classe Número Complexo e subtrai este número complexo com o encapsulado usando a fórmula  $(a+bi)-(c+di) = (a-c)+(b-d)i$  e devolve um novo Número Complexo como resposta;
  - vii. multiplicar, recebe uma outra instância da classe Número Complexo e multiplica este número complexo com o encapsulado usando a fórmula  $(a+bi)*(c+di) = (ac-bd)+(ad+bc)i$  e devolve um novo Número Complexo como resposta;

**Programação Orientada a Objetos**  
**2ª Lista de Exercícios**

- viii. dividir, recebe uma outra instância da classe Número Complexo e divide este número complexo com o encapsulado usando a fórmula  $(a+bi)/(c+di) = (ac+bd)/(c^2+d^2) + ((bc-ad)/(c^2+d^2))i$  e devolve um novo Número Complexo como resposta.