

## 4ª lista de exercícios Criação de Classes e Métodos Básicos

Crie ou altere as classes conforme definido nos itens abaixo e crie um programa para testar essas classes:

1. Crie a classe `Numero` que deverá ter um atributo `n` do tipo `float`. Nessa classe, implemente os métodos:
  - `float fatorial()` que deverá retornar  $n!$ .
  - `float potencia(int x)` que deverá retornar  $n^x$ .
  - `int parteInteira()` que deverá retornar a parte inteira de `n`.
  - `float parteDecimal()` que deverá retornar a parte decimal de `n`.
2. Crie a classe `Onibus` que representa um ônibus com a quantidade máxima de passageiros e o preço da passagem. Nessa classe, implemente os métodos:
  - `bool entrar(int qtdPassageiros)` que realiza a entrada de `qtdPassageiros` no ônibus. Se não couberem esses passageiros no ônibus o método deverá retornar `false`.
  - `bool sair(int qtdPassageiros)` que realiza a saída de `qtdPassageiros` no ônibus. Se não existirem esses passageiros no ônibus o método deverá retornar `false`.
  - `float faturamento()` que retorna o total faturado com as passagens.
  - `int qtdePassageiros()` que retorna a quantidade de passageiros no ônibus.
3. Crie a classe `Ponto` que representa um ponto no plano cartesiano e implemente o construtor adequado. Em seguida, implemente o método `float distancia(Ponto outroPonto)` que retorna a distancia do ponto ao outro ponto.
4. Crie uma classe `Linha` usando a classe `Ponto`. Em seguida, escreva três construtores para a classe `Linha`:
  - a) um que receba um argumento do tipo `Ponto` e que considere que a linha comece na origem e termine no ponto passado como argumento
  - b) um que receba duas instâncias da classe `Ponto` como argumentos
  - c) um que receba quatro valores de ponto flutuante, correspondentes às duas coordenadas da linha.Em seguida implemente o método `tamanho()` que retorna o tamanho da linha.
5. Crie a classe `Triangulo` usando a classe `Ponto`. Defina um construtor para `Triangulo` que recebe os três pontos que o formam. Caso, os pontos não formem um triângulo, gere uma exceção. Por último defina os métodos `float perimetro()` e `float area()`. Para calcular a área do triângulo use:

$$área = \sqrt{S \cdot (S - a) \cdot (S - b) \cdot (S - c)}$$

onde `a`, `b` e `c` são os lados do triângulo e `S` é o perímetro dividido por 2, ou seja  $(a+b+c)/2$ .

6. Implemente a classe `Poligono` que possui um vetor de pontos. Implemente também o método `float perimetro()` para calcular o perímetro do polígono.
7. Crie a classe `Equacao2Grau` que possui os valores de  $a$ ,  $b$  e  $c$  da expressão

$$ax^2 + bx + c = 0$$

e implemente os métodos

- `Float getX1()`
- `Float getX2()`

que devem retornar:

$$x1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}$$

$$x2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$$

Dicas: use as fórmulas abaixo para calcular os dois valores de  $x$  e o método estático `Math.sqrt` para calcular a raiz quadrada. Cuidado: não existe raiz quadrada de valor negativo. Nesse caso é impossível calcular os valores de  $x1$  e  $x2$  e os métodos devem retornar `null` !

8. Implemente a classe `CaixaEletronico` que deverá ter:
  - a) Atributos para armazenar quantas notas de R\$5, R\$10, R\$20, R\$50 e R\$100 existem no caixa. Implemente outros atributos que julgar necessários para implementar os demais métodos da classe.
  - b) Um construtor adequado para alimentar o caixa eletrônico com as quantidades iniciais de cada nota.
  - c) O método `int[] sacar(float valor)` que deverá retornar um vetor de 5 inteiros informando quantas notas R\$5, R\$10, R\$20, R\$50 e R\$100 deverão ser entregues ao cliente. Se não for possível montar a quantidade solicitada com as notas existentes no caixa o método deve retornar `null`.
  - d) O método `float totalSaque()` que deverá informar o montante de dinheiro sacado no caixa.
  - e) O método `int quantidadeSaque()` que deverá informar a quantidade de saques realizados no caixa.
  - f) O método `float saldo()` que deverá informar o montante de dinheiro existente no caixa.
9. Implemente a classe `Estatistica` com quatro métodos estáticos para: ler um vetor de *doubles*, calcular o somatório, a média aritmética, variância e o desvio padrão de um vetor de *doubles*. Implemente os seguintes métodos:

```
public static double[] lerVetor()
public static double soma(double[] vetor)
public static double media(double[] vetor)
public static double variancia(double[] vetor)
public static double desvioPadrao(double[] vetor)
```

Dica: para um vetor de  $n$  elementos, onde cada elemento é  $x_i$ , use as formulas:

$$somatório = \sum_{i=1}^n x_i$$

$$média = \frac{somatório}{n}$$

$$variância = \frac{\sum_{i=1}^n (x_i - média)^2}{n - 1}$$

$$desvio\ padrão = \sqrt{variância}$$