

---

### 3ª 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 `AlunoAcademia` que representa o aluno de uma academia. Para esse aluno devem ser registrados, obrigatoriamente, o peso e a altura. Nessa classe, implemente os métodos:
  - `float imc()` que calcula o IMC baseado na fórmula  $\text{peso} / \text{altura}^2$
  - `float pesoMinimo()` que calcula o peso mínimo baseado na fórmula  $18.5 * \text{altura}^2$
  - `float pesoMaximo()` que calcula o peso máximo baseado na fórmula  $24.9 * \text{altura}^2$
  - `float pesoMedio()` que calcula o peso médio baseado na fórmula  $(\text{peso mínimo} + \text{peso máximo}) / 2$
3. Crie uma classe `Funcionario` com matrícula, nome e salário. Implemente os métodos:
  - `float ganhoAnual()` que calcula quanto o funcionário ganha no ano. O ganho anual é de 12x o salário, mais décimo terceiro e mais 1/3 de abono de férias.
  - `float descontoIR()` que calcula o desconto no imposto de renda que é de ZERO para quem ganha até R\$ 1500,00/mês, 15% para quem ganha até R\$ 5000,00/mês e de 25% para quem ganha acima de R\$ 5000,00/mês.
4. Crie uma classe `Produto` com código, descrição, preço de custo, preço de venda e margem de lucro. Implemente os métodos:
  - Construtor para código, descrição e preço de custo.
  - `void setMargemLucro(float percentual)` que deverá alterar a margem de lucro e calcular o preço de venda correspondente.
  - `void setPrecoVenda(float valor)` que deverá alterar o preço de venda e calcular a margem de lucro correspondente.
5. Crie a classe `Onibus` que representa um ônibus com a quantidade máxima de passageiros e o preço da passagem. Nessa classe, implemente os métodos:

- `boolean entrar(int qtdPassageiros)` que realiza a entrada de `qtdPassageiros` no ônibus. Se não couberem esses passageiros no ônibus o método deverá retornar `false`.
  - `boolean 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 qtdPassageiros()` que retorna a quantidade de passageiros no ônibus.
6. Crie uma classe `Funcionario` para modelar um funcionário com matrícula, nome, salário e um valor booleano que indica se o funcionário está ativo ou não. Defina um construtor adequado para criar instâncias dessa classe. Implemente também os métodos:
- `void aumenta(float percentual)` que aumenta o salário do funcionário conforme o percentual definido.
  - `void demite()` que muda o status do funcionário para inativo.
7. Crie a classe `Hora` que deverá ter, obrigatoriamente, os atributos `hora`, `minuto` e `segundo`. Em seguida implemente o método:
- `boolean ehValida()` que retorna `true` ou `false` se a hora for válida ou inválida, respectivamente.
  - `void imprime()` para imprimir a hora no formato `HH:MM:SS`
  - `Hora diferenca(Hora outraHora)`
- que deverá retornar um novo objeto da classe `Hora` que é a diferença entre a hora e `outraHora`.