



## Avaliação Java - Semana 4

### Exercício 1 (40 pts):

**Descrição:** Crie um sistema simples para gerenciar uma locadora de veículos.

**Passo 1:** Crie uma interface `Aluguel` com os seguintes métodos:

- `void alugarVeiculo(String NomeCliente)`
- `void retornarVeiculo()`

**Passo 2:** Crie uma classe abstrata `Veiculo` com os seguintes atributos e métodos:

- Atributos: `String placa`, `String modelo`, `boolean estaAlugado`
- Métodos:
  - `abstract void mostrarDetalhes()`
  - `void checarStatus()`: método concreto que imprime se o carro está emprestado ou disponível.

**Passo 3:** Crie duas subclasses `Carro` e `Moto` que estendem `Veiculo` e implementam `Aluguel`. Ambas as classes devem fornecer implementações para `mostrarDetalhes()`, `alugarVeiculo()` e `retornarVeiculo()`. \*Os métodos `alugar` e `retorar` devem cumprir seu papel lógico de alterar o status do atribuo `estaAlugado`.

**Passo 4:** No método `main`, crie instâncias de `Carro` e `Moto`, então simule o aluguel e devolução de veículos, exibindo os detalhes e status dos veículos.

## Exercício 2 (30 pts):

**Objetivo:** Combinar herança e polimorfismo para criar um sistema de pagamento.

1. Crie uma classe base chamada `Funcionario` com os seguintes atributos:

- `nome (String)`
- `salarioBase (double)`

E os seguintes métodos:

- `calcularSalario() (double)` - Retorna o salário base.

2. Crie três subclasses que herdam de `Funcionario`:

- `Gerente` - Adicione um atributo `bonus (double)` e sobreponha o método `calcularSalario()` para incluir o bônus no cálculo.
- `Vendedor` - Adicione um atributo `comissao (double)` e sobreponha o método `calcularSalario()` para incluir a comissão no cálculo.
- `Estagiario` - Adicione um atributo `desconto (double)` e sobreponha o método `calcularSalario()` para subtrair o desconto do salário base.

3. Na classe principal (`Main`):

- Crie um array de `Funcionario` e adicione pelo menos uma instância de `Gerente`, `Vendedor` e `Estagiario`.
- Use um loop para iterar sobre o array e imprimir o nome e o salário calculado para cada funcionário, demonstrando o polimorfismo.

## Exercício 3 (30 pts):

**Descrição:** Crie uma função para gerar uma matriz (array bidimensional). Na classe `main` gere uma matriz qualquer e preencha seus elementos manualmente. Por fim imprima a matriz.