



Escola Estadual Professor João Anastácio
CURSO TÉCNICO EM INFORMÁTICA
AVALIAÇÃO DO 4º BIMESTRE DA DISCIPLINA
POO/JAVA – LOG.PROGRAMAÇÃO

ALUNO (a): _____

PROFESSOR(a): _____

DATA: / /

VALOR: 07 pontos TURMA: 2.2 Nota: _____

"Educação não transforma o mundo. Educação muda as pessoas. Pessoas transformam o mundo."

(Paulo Freire)

1. Implementação de Herança (Classes e Atributos)

Crie a seguinte hierarquia de classes, utilizando herança para reutilizar código e modelar o relacionamento "é um":

- **Veiculo (Classe Base/Superclasse):**
 - **Tipo:** Declare esta classe como **abstrata**.
 - **Atributos:** marca (String), modelo (String), ano (int).
 - **Construtor:** Defina um construtor que inicialize todos os atributos.
 - **Métodos:** Implemente métodos *getter* para todos os atributos.
 - **Método Abstrato:** Declare um método abstrato chamado `exibirDetalhes()` que retorne uma String.
- **CarroPasseio (Subclasse):**
 - Deve herdar de Veiculo.
 - **Atributo Específico:** numeroPortas (int).
- **Caminhao (Subclasse):**
 - Deve herdar de Veiculo.
 - **Atributo Específico:** capacidadeCargaTon (double).

2. Aplicação de Polimorfismo

O Polimorfismo será demonstrado de duas formas: sobrescrita de método e polimorfismo dinâmico.

Sobrescrita de Método (@Override):

Em ambas as subclasses (CarroPasseio e Caminhao), você deve **sobrescrever** o método `exibirDetalhes()` da superclasse Veiculo.

- A implementação de `exibirDetalhes()` em cada subclasse deve:
 - Chamar `super.getMarca()`, `super.getModelo()` e `super.getAno()` (ou uma chamada a um método auxiliar) para obter as informações da superclasse.

- Adicionar a informação específica da subclasse (o número de portas ou a capacidade de carga) na `String` de retorno.

Polimorfismo Dinâmico (Classe de Teste):

- Crie uma classe chamada `FrotaTest` com o método `main`.
- **Instanciação:** Crie pelo menos uma instância de `CarroPasseio` e uma de `Caminhao`.
- **Lista Polimórfica:** Crie uma lista (por exemplo, `ArrayList`) que armazene objetos do tipo `Veiculo` (`List<Veiculo>`).
- Adicione as instâncias de `CarroPasseio` e `Caminhao` a esta lista.
- **Iteração e Chamada:** Itere sobre a lista (`List<Veiculo>`) e chame o método `exibirDetalhes()` para cada objeto.
- Imprima o resultado da chamada no console. O compilador deve aceitar a chamada a `exibirDetalhes()`, e em tempo de execução, o método correto (o do `CarroPasseio` ou o do `Caminhao`) deve ser executado, demonstrando o polimorfismo dinâmico.