

## Objetivo 2: Herança Simples, Herança Múltipla e Polimorfismo

### Herança Múltipla - Lista 1

---

#### Objetivos

1. Assimilar os conceitos de herança múltipla;
2. Entender a aplicação das Interfaces;
3. Entender as regras das Interfaces, e sua evolução ao longo do tempo;
4. Interpretar diagramas de classe da UML para transformá-los em código na tecnologia Java;
5. Instanciar objetos destas classes;
6. Manipular estes objetos;
7. Criar pequenos *apps* em Java.

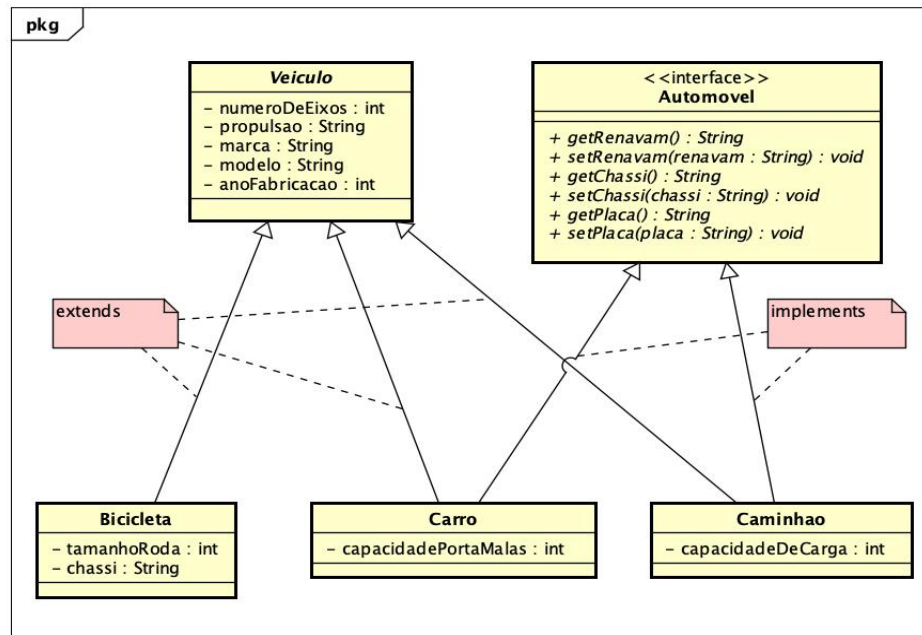
#### EXERCÍCIOS

1. Faça o que se pede:
  - a) Interprete o diagrama de classes abaixo na tecnologia Java;
  - b) Organize o projeto em camadas MVC;
  - c) Crie uma instância da classe Bicicleta e adicione valores válidos para os atributos dessa instância, considerando, também, os atributos da sua superclasse, a classe Veículo, conforme expresso no diagrama de classes abaixo. Imprima este objeto e verifique o resultado;
  - d) Crie uma instância da classe Carro e adicione valores válidos para os atributos dessa instância, considerando, também, os atributos da sua superclasse, bem como, os métodos da interface de quem ele herda, conforme especificado no diagrama de classes abaixo. Imprima este objeto e verifique o resultado;

- e) Agora repita as mesmas tarefas do item d para uma instância da classe Caminhao. Imprima este objeto e verifique o resultado.

2. Faça o que se pede:

- a) Suponha que estes veículos fazem parte do cadastro do Dentrán-RS, então, crie uma coleção de objetos para representar a estrutura do cadastro de veículos do Dentrán-RS;
- b) Acrescente neste cadastro, pelo menos, 5 bicicletas, 5 carros e 5 caminhões. Tenha o cuidado de inserir nos atributos dados válidos para este cadastro;
- c) Ordene e imprima os veículos cadastrados no sistema, do maior para o menor ano de fabricação;
- d) Ordene e imprima os automóveis cadastrados no sistema, do maior para o menor ano de fabricação;
- e) Ordene e imprima as bicicletas cadastradas no sistema, do maior para o menor ano de fabricação;
- f) Ordene e imprima a lista de veículos que possuem placas que começam pela letra “I” ou “i” (organize o código para que dois ou mais veículos tenham placas que comecem por “I” ou “i”), em ordem crescente pelo ano de fabricação.



Bom trabalho.

## Gabarito

Você encontra o gabarito deste exercício no Github, em:

[https://github.com/vagnersilvaifsul/lpoo/tree/main/objetivo2\\_lista1\\_heranca\\_multipla/src](https://github.com/vagnersilvaifsul/lpoo/tree/main/objetivo2_lista1_heranca_multipla/src)

## Referências

JAVA 17. **JDK 17 Documentation**. Disponível em: <https://docs.oracle.com/en/java/javase/17/>. Acesso fevereiro, 2024.

JAVA 17 API. **Java Platform, Standard Edition & Java Development Kit Version 17 API Specification**. Disponível em: <https://docs.oracle.com/en/java/javase/17/docs/api/index.html>. Acesso fevereiro, 2024.

DEITEL, Paul J.; DEITEL, Harvey M. **Java: como programar**. 8. ed. São Paulo: Pearson, 2010.

GUEDES, Gilleanes T. A. **UML 2: uma abordagem prática**. 2. ed. São Paulo: Novatec, 2011.