



## ESTRUTURA DE DADOS II

### Revisão: Programação Orientada a Objetos com Java

Atividade (máx. três alunos)

### Objetivo

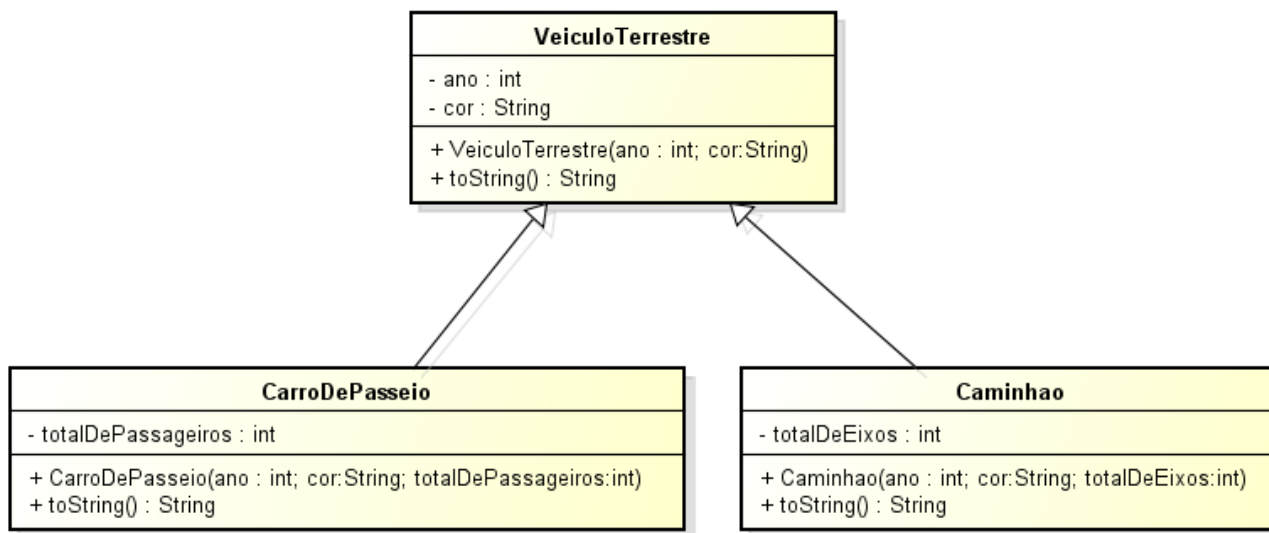
Treinar conceitos de Programação Orientada a Objetos (POO) com a linguagem Java e verificar dificuldades com a linguagem e conceitos de POO.

### Instruções

- A atividade deve ser resolvida usando a linguagem Java.
- Inclua a identificação do grupo (nome completo e RA de cada integrante) no início de cada arquivo de código, como comentário.
- Inclua todas as referências (livros, artigos, sites, vídeos, entre outros) consultadas para solucionar a atividade como comentário no arquivo `.java` que contém a `main()`.

### Enunciado

Considere o diagrama de classes abaixo com as classes: `VeiculoTerrestre`, `CarroDePasseio` (derivada de `VeiculoTerrestre`) e `Caminhao` (também derivada de `VeiculoTerrestre`).



Crie todas as classes contendo os construtores, os métodos *getters* e *setters*, além do método `toString()` em um projeto Java.

A partir disso:

a) Elabore um programa principal que cria uma lista de veículos com os seguintes dados:

`Carro de Passeio: ano = 1990, cor = Azul, Passageiros = 5;`



## ESTRUTURA DE DADOS II

```
Carro de Passeio: ano = 2004, cor = Cinza, Passageiros = 4;  
Caminhão: ano: 1999, cor = Verde, Eixos = 4;  
Carro de Passeio: ano = 2010, cor = Preta, Passageiros = 5;  
Carro de Passeio: ano = 2009, cor = Preta, Passageiros = 7;  
Caminhão: ano: 2011, cor = Verde, Eixos = 4;  
Caminhão: ano: 2000, cor = Azul, Eixos = 4;  
Caminhão: ano: 2005, cor = Preto, Eixos = 3.
```

b) Por meio da lista de veículos, procure e apresente os dados dos carros de passeio que comportem 5 passageiros e tenham ano menor do que 2010.

c) Em seguida, por meio da lista de veículos, procure e apresente a soma do total de veículos do tipo caminhão que são verdes e que tenham quatro eixos.

## Entrega

Compacte o código-fonte (somente arquivos `*.java`) no formato `zip`.

**Atenção:** O arquivo `zip` não deve conter arquivos intermediários e/ou pastas geradas pelo compilador/IDE (ex. arquivos `*.class`, etc.).

**Prazo de entrega:** via link do Moodle até 12/03/2024 23:59.



## ESTRUTURA DE DADOS II

### Cr terios de avalia  o

A nota da atividade   calculada de acordo com os crit rios da tabela a seguir.

ITEM AVALIADO	PONTUA��O M�XIMA
1. Classe <code>VeiculoTerrestre</code> .	1,5
2. Classe <code>CarroDePasseio</code> .	1,5
3. Classe <code>Caminhao</code> .	1,5
4. Classe do programa principal e lista de ve�culos.	1,5
5. Apresenta��o dos dados do item (b).	1,5
6. Apresenta��o dos dados do item (c).	1,5
7. Funcionamento geral do programa, de acordo com o enunciado.	1,0

Tabela 1 - Crit rios de avalia  o.

A tabela a seguir cont m crit rios de avalia  o que podem **reduzir** a nota final da atividade.

ITEM INDESEJ�VEL	REDU���O DE NOTA
O projeto � c�pia de outro projeto.	Projeto � zerado
H� erros de compila��o e/ou o programa trava durante a execu��o <sup>1</sup> .	-1,0
N�o h� identifica��o do grupo. N�o h� indica��o de refer�ncias. Arquivos enviados em formatos incorretos. Arquivos e/ou pastas intermedi�rias que s�o criadas no processo de compila��o ou pela IDE foram enviadas junto com o c�digo-fonte.	-1,0

Tabela 2 - Crit rios de avalia  o (redu   o de nota).

O c digo-fonte ser  compilado com o compilador `javac` (21.0.2) na plataforma Windows da seguinte forma:

```
> javac *.java -encoding utf8
```

O c digo compilado ser  executado com `java` (21.0.2) na plataforma Windows da seguinte forma:

```
> java <Classe>
```

Sendo que `<Classe>` deve ser substituído pelo nome da classe que cont m o m todo `public static void main(String[] args)`.

---

<sup>1</sup> Sobre erros de compila  o: considere apenas erros. N o h  problema se o projeto tiver *warnings* (embora *warnings* podem avisar sobre poss veis travamentos em tempo de execu  o, como loop infinito, divis o por zero, etc.).