

Exercícios para a prática de métodos, herança, polimorfismo e atributos estáticos

P00 – Valor: 0.5pts

A atividade deverá ser apresentada ao professor na aula do dia 20/06/2017

Pode ser feita em dupla.

1 - Métodos e Herança;

Este é um suposto sistema que organizará veículos. Criar, em um projeto Java, as classes Veiculo, Caminhao, Carro e Moto. As três últimas (Carro, Caminhao e Moto) serão subclasses de Veiculo, ou seja, como exemplo, "public class Carro extends Veiculo".

As classes deverão conter os seguintes atributos e métodos;

Veículo:

CONSTRUTOR

+ Veiculo()

ATRIBUTOS

- marca : String

- numeroDeRodas : int

- numeroDePortas : int

- velocidadeAtual : int //Em Km/h

MÉTODOS:

+ getMarca() : String //Getter padrão

+ setMarca(String marca) : void //Setter padrão

+ getNumeroDeRodas() : int //Getter padrão

+ setNumeroDeRodas(int numeroDeRodas) : void //Setter padrão

+ getNumeroDePortas() : int //Getter padrão

+ setNumeroDePortas(int numeroDePortas) : void //Setter padrão

+ getVelocidadeAtual() : int //Getter padrão

+ acelerar(int velocidade) : void //Deverá colocar a velocidadeAtual = velocidade passada.

+ frear() : void //Deverá colocar a velocidadeAtual = 0

Carro:

CONSTRUTOR

+ Carro(int numeroDePortas) //Deve settar o numeroDePortas como 2 ou 4, além de o numeroDeRodas como 4, utilizando a chamada "super"

ATRIBUTOS

- tamanhoPortaMalas : int //Em litros

MÉTODOS:

+ getTamanhoPortaMalas() : int //Getter padrão

+ setTamanhoPortaMalas(int tamanhoPortaMalas) : void //Setter padrão

Moto:

CONSTRUTOR

+ Moto() //Deve settar o numeroDePortas como 0 e o numeroDeRodas como 2, usando a chamada "super", além de bauleto como false

ATRIBUTOS

- bauleto : boolean //Se a moto tem ou não um bauleto

MÉTODOS

+ colocarBauleto() : void //Deve settar bauleto como true

+ tirarBauleto() : void //Deve settar bauleto como false

+ temBauleto() : boolean //Retorna se tem ou não bauleto, como true ou false

Caminhao:

CONSTRUTOR

+ Caminhao(int numeroDeRodas) //Deve settar o numeroDeRodas, além de o numeroDePortas como 2, ambos com a chamada "super"

ATRIBUTOS

- cargaMaxima : int //Em toneladas

MÉTODOS

+ getCargaMaxima() : int //Getter padrão

+ setCargaMaxima(int cargaMaxima) : void //Setter padrão

Após a implementação das classes, recomenda-se a criação de uma classe de Teste, que instancie pelo menos um Carro, uma Moto e um Caminhão, para a realização de testes, para responder as seguintes questões relacionadas a "métodos" e "herança";

a) Quando instanciado, quais serão os atributos que um Carro terá ? E uma Moto ? E um Caminhão ?

b) Uma Moto pode, por exemplo, chamar "getCargaMaxima", já que tanto Caminhão quanto Moto extends Veiculo ? Justifique

2 - Polimorfismo

Crie agora, na classe Veiculo, um método comparaVelocidade(Veiculo veiculo), que retorna void. Dentro deste método, compare as velocidades de veículos, exibindo no console qual dos veículos está mais rápido. Teste este método com alguns carros, motos e caminhões.

Após isso, responda:

a) Porque um Carro, mesmo tendo sido instanciado como Carro, pode ser passado para este método que recebe um Veiculo ?

b) Um método que recebe uma Moto, poderia receber um Carro ? Porque ?

3 - Atributos Estáticos

Agora, na classe Veiculo, crie um "static int num", que será = 0, além de um "private int id", juntamente com o getter para o id. Faça com que, no construtor dessa classe, o id seja = ++num; Refatore o código conforme necessário para adequar ao novo atributo que acabou de ser criado, tal como, indicar o id dos veículos comparados em comparaVelocidade(), por exemplo.

Baseado nisso, responda:

- a) Caso seja feito um getNum(), qual informação ele nos dará ?
- b) A variável "num" pode ser chamada de uma classe que não é a própria Veiculo ?
- c) A variável num pode ser chamada, por exemplo, através de uma instância de um carro, como no exemplo abaixo ?

```
Carro c = new Carro(2);  
  
System.out.println(c.num);
```

E, se sim, a variável num seria a mesma para ambas as instâncias no caso abaixo ?
Porque ?

```
Carro c = new Carro(2);  
  
Carro c2 = new Carro(4);  
  
System.out.println(c.num + " " + c2.num);
```

- d) Caso não haja nenhuma instância de veículo, é possível acessar o atributo estático num através de Veiculo.num ou, até mesmo, Carro.num, Moto.num ou Caminhao.num ? O que ele retorna ?