



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação

AD1 de Programação III

2º semestre de 2012

Nome:

Matrícula:

Pólo:

Obs: *A solução para o exercício proposto deve ser entregue por escrito e em formato digital.*

O programa abaixo apresenta a definição inicial de uma classe que representa um meio de transporte (carro, ônibus, avião, etc.).

```
class MeioTransporte {  
    int capacidadePassageiros;  
    int capacidadeCarga;  
}
```

Baseado nesta classe, faça:

- Além das capacidades de passageiros e carga (expressa em litros), crie campos armazenem o consumo do meio de transporte (quilômetros / litro), o custo de manutenção anual do transporte (expresso em reais), o ano de fabricação (um inteiro) e o tipo de combustível do meio (álcool, gasolina, diesel ou gnv).
- Crie um construtor que inicialize os campos desta classe.
- Defina as classes Carro e Moto, as quais são mais específicas que a classe MeioTransporte.
- Suponha que temos interesse em calcular a eficiência de cada meio de transporte, ou seja, a relação custo-benefício destes. Defina um método eficiência em cada classe específica, a qual deve retornar a seguinte expressão:

Transporte	Eficiência
Carro	No. de passageiros vezes consumo(1) vezes 1/10 da capacidade da carga, tudo isso dividido pelo custo de manutenção vezes o no. de anos do transporte
Moto	No. de passageiros vezes consumo vezes 1/10 da capacidade da carga, tudo isso dividido pelo custo de manutenção vezes o no. de anos do transporte

(1) Para o caso específico dos carros, caso o tipo do combustível utilizado seja “gnv”, na expressão de eficiência dada, o fator consumo é multiplicado por 1.5

- e) Altere a classe MeioTransporte de forma a obrigar que toda a classe que a estenda tenha que definir este método para cálculo de eficiência.
- f) Defina um método main(), numa classe chamada Principal, e instancie 2 objetos: 1 carro com capacidade para 5 passageiros, 300 litros de carga, consumo de 10 km/l, custo de manutenção anual de R\$1000,00, 2001 como ano de fabricação e que funcione com GNV; 1 moto com capacidade para 2 passageiros, carga de 30 litros, 20 km/l de consumo, R\$500,00 de manutenção anual, ano de fabricação 2009 e movida à gasolina.
- g) Defina na classe MeioTransporte um método que compare 2 meios de transporte e retorne qual é o mais eficiente.
- h) Chame o método do item g) na main(), para os 2 objetos definidos no item f).

Dica: Para o cálculo do número de anos de existência de um meio de transporte, podemos obter a hora corrente através do construtor padrão (sem argumentos) da classe java.util.GregorianCalendar(). Obtemos o ano desta data (ano corrente) e subtraímos do ano de fabricação informado na construção do objeto MeioTransporte.

RESPOSTA:

```
import java.util.Calendar;
import java.util.GregorianCalendar;

abstract class MeioTransporte {
    int capacidadePassageiros;
    int capacidadeCarga;
    double consumo;
    double custoManutencao;
    int anoFabricacao;
    static enum tipoCombustivel {alcool, gasolina, diesel, gnv};
    tipoCombustivel combustivel;

    public MeioTransporte(int capacidadePassageiros, int capacidadeCarga,
        double consumo, double custoManutencao, int anoFabricacao,
        tipoCombustivel combustivel) {
        this.capacidadePassageiros = capacidadePassageiros;
        this.capacidadeCarga = capacidadeCarga;
        this.consumo = consumo;
        this.custoManutencao = custoManutencao;
        this.anoFabricacao = anoFabricacao;
        this.combustivel = combustivel;
    }

    abstract double eficiencia();

    int anosExistencia () {
        GregorianCalendar data = new GregorianCalendar();
        return data.get(Calendar.YEAR) - this.anoFabricacao;
    }

    boolean maisEficiente (MeioTransporte m) {
```

```

        return this.eficiencia() > m.eficiencia();
    }
}

class Carro extends MeioTransporte {
    public Carro(int capacidadePassageiros, int capacidadeCarga,
        double consumo, double custoManutencao, int anoFabricacao,
        tipoCombustivel combustivel) {
        super(capacidadePassageiros, capacidadeCarga, consumo, custoManutencao,
            anoFabricacao, combustivel);
    }

    double eficiencia() {
        return (capacidadePassageiros * (capacidadeCarga * 0.1) * (((combustivel
== tipoCombustivel.gnv) ? 1.5 : 1) * consumo)) / (custoManutencao *
this.anosExistencia());
    }
}

class Moto extends MeioTransporte {
    public Moto(int capacidadePassageiros, int capacidadeCarga, double consumo,
        double custoManutencao, int anoFabricacao,
        tipoCombustivel combustivel) {
        super(capacidadePassageiros, capacidadeCarga, consumo, custoManutencao,
            anoFabricacao, combustivel);
    }

    double eficiencia() {
        return (capacidadePassageiros * (capacidadeCarga * 0.1) * consumo) /
(custoManutencao * this.anosExistencia());
    }
}

public class AD1_2012_2 {
    public static void main(String[] args) {
        MeioTransporte escort = new Carro(5, 300, 10, 1000, 2001,
MeioTransporte.tipoCombustivel.gnv);
        MeioTransporte kawasaki = new Moto(2, 30, 20, 500, 2009,
MeioTransporte.tipoCombustivel.gasolina);
        System.out.println("Eficiencia do Carro: " + escort.eficiencia());
        System.out.println("Eficiencia da Moto: " + kawasaki.eficiencia());
        System.out.println(escort.maisEficiente(kawasaki));
    }
}

```