

## Lista de Exercícios

|               |                                                 |
|---------------|-------------------------------------------------|
| Curso         | Ciência da Computação                           |
| Disciplina    | Linguagem de Programação III                    |
| Professor(es) | Otacílio José Pereira                           |
| Assunto       | - Capítulo 2: Introdução à Orientação a Objetos |

### **Bloco 1 – Introdução aos conceitos de Orientação a Objetos**

- Leituras recomendadas:

- Livro Deitel: Tóp. 1.5 – Introdução à Tecnologia de Objetos  
Cap 3 – Introdução à Objetos, Classes e ...
- Apostila sugerida: Cap 4 – Orientação a Objetos Básica

1.a) No paradigma orientado a objetos, diferente do paradigma estruturado que visa focar o fluxo de controle de algoritmos, o foco na OO é pensar os objetos e a troca de mensagens entre eles.

Identifique um **aplicativo comum de seu dia a dia** e aponte pelo menos três tipos de **classes com seus respectivos objetos** e alguns métodos que poderiam ser empregados.

Obs: Não use uma aplicativo de Banco e de Mobilidade pois foram explorados nos nossos exemplos.

1.b) Apresente as **semelhanças e as diferenças** entre **Estruturas** na linguagem C e as **Classes** na linguagem Java.

### **Bloco 2 – Primeira classe e atributos**

2.a) Imagine um aplicativo de mobilidade em que Motoristas podem ser chamados para atender os clientes. Exemplos ou exemplares de motoristas são João, Everaldo e Cristiano. Um motorista tem um nome, o modelo do carro, a placa do carro, o número de viagens e a sua avaliação (nota entre 0 e 5).

**Apresente o código da classe Motorista com seus atributos.**

2.b) Crie agora uma classe TestaMotorista e dentro do método main crie e instancie dois objetos da classe Motorista.

### **Bloco 3 – Métodos**

3.a) Quais as semelhanças e as diferenças entre os métodos no Java e as funções na linguagem C.

3.b) Escolha um dos atributos e crie os métodos get e set para ele? Qual o papel destes métodos ao se aplicar a ideia de encapsulamento.

3.b) Realize uma pesquisa sobre o que é um estado de um objeto? Em geral, qual o efeito de um método sobre o estado de um objeto?

3.c) No bloco 2 foram tratados métodos get e set que basicamente manipulam atributos em um encapsulamento.

Crie agora o método RegistraNovaViagem que recebe uma avaliação da viagem e daí aumenta o número de viagens e atualiza a avaliação do motorista considerando a nota da nova viagem.

3.d) Para operar com o sistema é necessário prever operações para entrada e saída de dados. Implemente o método ImprimirMotorista que apresenta todos os dados do motorista.

3.e) Implemente o método LerMotorista que a partir de uma classe Scanner solicita para um usuário os dados de um Motorista.

### **Bloco 4 – Visibilidade, encapsulamento, getters e setters**

4.a) Realize uma pesquisa sobre o que é um modificador de classe ou de atributo e quais tipos de modificadores de classe existem?

4.b) Pensando agora especificamente sobre modificador de acesso ou de visibilidade. Qual papel deste modificador?

4.c) Desconsiderando as questões de pacotes, focando apenas para o acesso aos elementos da classe, para que serve os modificadores public, protected e private?

4.b) Quais os problemas ter o acesso público a atributos de uma classe?

4.c) O que você entende por encapsulamento? Como uma estratégia análoga ao encapsulamento ocorre em uma linguagem estruturada como a linguagem C?

4.d) Quais as ações são necessárias para encapsular um atributo em uma classe em Java?

4.e) Para que servem os métodos getteres e setters? Qual a vantagem de usá-los?

4.f) Apresente uma situação em que um método get é usado para recuperar uma informação que não necessariamente seja um atributo.

4.g) Qual teste poderia ser feito no método setAvaliacao da classe motorista para garantir a consistência desta informação?

## Bloco 5 – Construtores

5.a) Quando uma classe não possui explicitamente um construtor, como um atributo pode ser inicializado com um determinado valor em específico?

Por exemplo, na classe *Motorista*, sem a criação de um construtor, faça com que todo novo objeto da classe *Motorista* comece com o número de viagens 0 (zero) e avaliação 0 (zero).

5.b) Revise o código da inicialização dos atributos de *Motorista* através de um construtor sem parâmetros?

5.c) Crie um construtor que inicializa um objeto recebendo todos os seus atributos por parâmetro.

5.d) Considere o trecho de código a seguir:

```
public class Conta {
    public String NomeTitular;
    public Integer Agencia;
    public Integer Numero;
    public Double Valor;

    Conta()
    {
        //...
        System.out.print(" Construtor 1");
    }

    Conta(String NomeTitular, Integer Agencia)
    {
        //...
        System.out.print(" Construtor 2");
    }
}
```

O que será impresso a partir do código a seguir:

```
Conta c1, c2, c3;
c1 = new Conta("Fabiana", 1230);
c2 = new Conta("Jose", 1230);
c3 = new Conta();
```

5.e) Explique o que você entende por declaração, instanciação e inicialização de um objeto.

5.f) No código a seguir, quais os valores dos atributos *Agencia* dos objetos tratados por *c1*, *c2* e *c3*.

```
Conta c1, c2, c3;
c1 = new Conta("Fabiana", 10);
c2 = new Conta("Jose", 20);
c1 = c2;
c3 = c1;
c2.setAgencia(30);
```