



Programação Orientada a Objetos

Vetores e Matrizes

Lista 01

Professor: Edwar Saliba Júnior

1. Considere a seguinte classe:

```
5 package exercicio02;
6
7 public class Cliente {
8
9     private int codigo;
10    private String nome;
11    private boolean eClienteEspecial;
12    private double limiteCredito;
13
14    public Cliente(int c, String n) {
15        codigo = c;
16        nome = n;
17        limiteCredito = 0;
18        eClienteEspecial = false;
19    }
20
21    public String obterNome() {
22        return(nome);
23    }
24 }
25
```

Figura 1: Classe Cliente

- Escreva um método *atualizaLimite* que atualize o limite de crédito do cliente. O método deverá receber como parâmetros um valor do tipo *double*. Caso o cliente seja especial, o seu limite de crédito deverá ser atualizado com o valor recebido como parâmetro e o método deverá retornar *true*, indicando que a operação foi realizada. Caso contrário, o método deverá retornar *false*.
- Escreva um método *atualizaSituacao* que atualize a situação do cliente (o seu atributo *eClienteEspecial*). O método deverá receber como parâmetro um *boolean*. Além de atualizar o referido atributo com o valor recebido como parâmetro, se este valor for *false*, o atributo *limiteCredito* deverá ser atualizado com o valor 0 (zero).

2. Considere a classe cliente resultante do exercício anterior e o código em Java a seguir:

```

5  package exercicio02;
6
7  public class Main {
8
9      public static void main(String[] args) {
10         Cliente cliA, cliB;
11
12         cliA = new Cliente(1, "Luiza");
13         cliA.atualizaSituacao(true);
14         cliB = new Cliente(2, "Wesley");
15
16         if (!cliA.atualizaLimite(500)) {
17             System.out.println("Este não é um cliente especial.");
18         }
19
20         cliB = cliA;
21         cliB.atualizaSituacao(false);
22
23         if (!cliA.atualizaLimite(700)) {
24             System.out.println("Este não é um cliente especial.");
25         }
26     }
27 }

```

Figura 2: Classe Main

- a) Se o comando `cliA.atualizaSituacao(true)` (linha 13) fosse executado na linha 11, ocorreria um erro em tempo de execução. Por quê?
 - b) Explique o que ocorre na linha 12.
 - c) O que o programa imprime?
3. Implemente uma aplicação em Java que crie um objeto Pessoa com os seguintes atributos: Nome (*String*) e Idade (*int*). O programa deverá solicitar que o usuário entre com 10 nomes e suas respectivas idades. Em seguida o programa deverá imprimir os nomes e idades digitadas, na ordem inversa à da digitação. (Para resolver este problema faça uso de vetor.)
 4. Implemente uma aplicação em Java que crie um objeto Veículo com os seguintes atributos: Marca e Modelo (*String*) e AnoFabricacao (*int*). O programa deverá solicitar que o usuário preencha os dados de 25 veículos. Em seguida o programa deverá imprimir os atributos de cada veículo, na ordem inversa à da digitação. (Para resolver este problema faça uso de uma matriz 5 x 5.)
 5. Crie um *software* que solicite que o usuário digite 200 valores inteiros quaisquer, armazene estes valores em dois vetores de 100 posições cada (`vet1` e `vet2`). Crie um terceiro vetor de 100 posições (`vet3`) para armazenar o seguinte cálculo: `vet3[0] = vet1[0] * vet2[99]; vet3[1] = vet1[1] * vet2[98];` e assim por diante. Finalizado o cálculo, imprima os valores de `vet3` na ordem inversa à da entrada.