



### Lista de Exercícios nº 3

#### Usando Classes em JAVA

Prof. Marcos Esteves



- 1) Quais linhas seriam produzidas na saída do programa. (O método intern() é um método que devolve o string contido em um String.)

```
public class StringTeste {
    public static void main(String args[])
    {
        String s1 = new String("abc");
        String s2 = new String("abc");
        String s3 = "abc";
        String s4 = "abc";
        String s5 = s1.intern();
        String s6 = s2.intern();
        if (s1 == s2) System.out.println("linha 1");
        if (s3 == s4) System.out.println("linha 2");
        if (s5 == s6) System.out.println("linha 3");
        if (s5 == s4) System.out.println("linha 4");
        if (s5 == s2) System.out.println("linha 5");
    }
}
```

- 2) Quais os valores finais de st1.x e st1.y no programa abaixo?

```
class StaticTest {
    private static int x = 100;
    private int y = 101;
    public static void main(String args[]) {
        StaticTest st1 = new StaticTest();
        st1.x++;
        st1.y++;
        StaticTest st2 = new StaticTest();
        st2.x++;
        st2.y++;
        StaticTest.x++;
        System.out.println("st1.x = "+ st1.x + " st1.y = "+ st1.y); }
}
```

- 3) O programa a seguir compila com erros (sim, não ou por que não?)

```
class ABC {
    public static void main(String args[]) {
        int x = 10;
        String s;
        if (x==10)
            s = new String ("cinco");
        System.out.println(s); } }
```

NOTA: As questões a seguir estão em inglês por fazerem parte do teste para certificação Java que é em inglês.

4) (Heller/Roberts 97) Consider the following application:

```
1. class Q6 {
2.     public static void main(String args[]) {
3.         Holder h = new Holder();
4.         h.held = 100;
5.         h.bump(h);
6.         System.out.println(h.held); }
7. }
8. class Holder {
9.     public int held;
10.    public void bump(Holder theHolder)
11.    {
12.        theHolder.held++;
13.    } }
```

What value is printed out at line 6 ?

5) (Heller/Roberts 97) Consider the following application:

```
1. class Q7
2. {
3.     public static void main(String args[]) {
4.         double d = 12.3;
5.         Decrementer dec = new Decrementer();
6.         dec.decrement(d);
7.         System.out.println(d); }
8. }
9. class Decrementer
10. {
11.     public void decrement(double decMe)
12.     {
13.         decMe = decMe - 1.0; } }
```

What value is printed out at line 7 ?

6) Crie uma classe em Java com as seguintes características:

Classe Funcionário com mat, nome, função e salário. Todas as propriedades, com exceção de mat, devem ser de instância.

7) Faça um programa para criar 3 objetos do tipo Funcionário e solicitar ao usuário que entre com dados para os três funcionários criados. Ao final imprima os valores fornecidos para estes funcionários.

8) Altere a classe Funcionário para que esteja de acordo com as definições da Orientação a Objetos com relação ao encapsulamento, ou seja, torne as propriedades *private* e recompile a classe. Tente executar o programa do exercício 7.

9) Altere a classe Funcionário adicionando métodos de acesso à classe de modo que somente será possível ter acesso as variáveis mediante os métodos criados.

10) Altere o exercício 7 para que utilize os métodos criados para acesso as propriedades ao invés de acessá-las diretamente.

11) Altere as classes Funcionário e a classe criada no exercício 10 para que a propriedade **mat** seja auto-incrementada, ou seja, a matrícula do funcionário seja automaticamente fornecida a partir do valor 1 a cada funcionário criado. Sendo assim, a matrícula não será mais solicitada na criação de um funcionário e sim gerada na construção do objeto da classe.

12) Crie um novo método construtor para a classe Funcionário de modo que os valores das propriedades sejam fornecidos na criação de um funcionário.

13) Altere a classe criada no exercício 11 para que utilize este novo método construtor.

14) Crie um novo método construtor para a classe Funcionário que permita a construção de um Funcionário apenas com o nome e função. Este construtor deverá chamar o outro construtor já existente passando o salário igual a zero como parte dos argumentos.

15) Crie uma classe chamada CargosSalarios que possua um método de classe chamado buscaSalario que receba como parâmetro a função do funcionário e retorne o salário correspondente. Este método deverá ser chamado pelo construtor do exercício 14 ao invés de se passar o salário igual a zero. Utilize a tabela de funções a seguir como exemplo.

Função: diretor-salário: 5000,00

Função: vendedor-salário: 3000,00

Função: secretaria-salário: 1000,00