



# Solução da Lista de Exercícios nº 3

## Usando Classes em JAVA

Prof. Marcos Esteves



```
class Exercicio1
{
    public static void main(String args[])
    {
        String s1 = new String("abc");
        String s2 = new String("abc"); // O operador new força a criação de um novo objeto
        String s3 = "abc";
        String s4 = "abc"; // Referencia o objeto s3 já criado
        String s5 = s1.intern(); // Que corresponde a "abc" e portanto também referencia s3
        String s6 = s2.intern(); // Que corresponde a "abc" e portanto também referencia s3
        if (s1 == s2) System.out.println("Linha1");
        if (s3 == s4) System.out.println("Linha2"); // Linha 2
        if (s5 == s6) System.out.println("Linha3"); // Linha 3
        if (s5 == s4) System.out.println("Linha4"); // Linha 4
        if (s5 == s2) System.out.println("Linha5");
    }
}

class StaticTest // Exercicio 2
{
    private static int x = 100;
    private int y = 101;
    public static void main(String args[])
    {
        StaticTest st1 = new StaticTest();
        st1.x++; // 100 + 1 = 101
        st1.y++; // 101 + 1 = 102
        StaticTest st2 = new StaticTest();
        st2.x++; // 101 + 1 = 102
        st2.y++; // 101 + 1 = 102
        StaticTest.x++; // 102 + 1 = 103
        System.out.println("st1.x= "+st1.x+" st1.y= "+st1.y); // Solução: st1.x = 103 e st1.y = 102
    }
}

class ABC // Exercicio 23
{
    public static void main(String args[])
    {
        int x = 10;
        String s;
        if (x == 10)
            s = new String("cinco");
        System.out.println(s); // Não compila pois s como variável local deveria ter sido inicializada
    }
}

class Exercicio4
{
    public static void main(String args[])
    {
        Holder h = new Holder();
        h.held = 100;
        h.bump(h); // h corresponde ao endereço do objeto
    }
}
```

```

        System.out.println(h.held); // Imprime 101
    }
}
class Holder // Exercício4 continuação
{
    public int held;
    public void bump(Holder theHolder) // o parâmetro é o endereço passado e portanto o mesmo de h
    {
        theHolder.held++; // equivale a h.held++;
    }
}

class Q7 // Exercício5
{
    public static void main(String args[])
    {
        double d = 123;
        Decrementer dec = new Decrementer();
        dec.decrement(d); // Tipos primitivos são passados por valor, ou seja, o conteúdo de d que
                        // no caso é um valor e não um endereço
        System.out.println(d); // Imprimiria 123 pois d nada tem a haver com decMe que já nem
                        // existe mais pois era uma variável local
    }
}

class Decrementer // Exercício5 continuação
{
    public void decrement(double decMe) // decMe = 123
    {
        decMe = decMe - 1.0;
        // System.out.println(decMe); Imprimiria 122
    }
}

public class Funcionario // Exercício6
{
    static int mat;
    String nome, funcao;
    double salario;
}

import java.util.*;
public class TestaFuncionario // Exercício7
{
    public static void main(String args[])
    {
        Scanner in = new Scanner(System.in);
        Funcionario f1, f2, f3;
        f1 = new Funcionario();
        f2 = new Funcionario();
        f3 = new Funcionario();
        try
        {
            System.out.println("Entre com os dados para o Funcionario 1 ");

```

```

System.out.print("Entre com a matricula: ");
f1.mat = in.nextInt();
in.nextLine(); // Esta leitura é para consertar um bug do readLine()
System.out.print("Entre com o nome: ");
f1.nome = in.nextLine();
System.out.print("Entre com a funcao: ");
f1.funcao = in.next();
System.out.print("Entre com o salario: ");
f1.salario = in.nextDouble();
System.out.println("Entre com os dados para o Funcionario 2 ");
System.out.print("Entre com a matricula: ");
f2.mat = in.nextInt();
in.nextLine(); // Esta leitura é para consertar um bug do readLine()
System.out.print("Entre com o nome: ");
f2.nome = in.nextLine();
System.out.print("Entre com a funcao: ");
f2.funcao = in.next();
System.out.print("Entre com o salario: ");
f2.salario = in.nextDouble();
System.out.println("Entre com os dados para o Funcionario 3 ");
System.out.print("Entre com a matricula: ");
f3.mat = in.nextInt();
in.nextLine(); // Esta leitura é para consertar um bug do readLine()
System.out.print("Entre com o nome: ");
f3.nome = in.nextLine();
System.out.print("Entre com a funcao: ");
f3.funcao = in.next();
System.out.print("Entre com o salario: ");
f3.salario = in.nextDouble();

System.out.println("Os dados do Funcionario 1 sao: ");
System.out.println("matricula: "+f1.mat);
System.out.println("nome: "+f1.nome);
System.out.println("funcao: "+f1.funcao);
System.out.printf("salario: %.2f",f1.salario);
System.out.println("Os dados do Funcionario 2 sao: ");
System.out.println("matricula: "+f2.mat);
System.out.println("nome: "+f2.nome);
System.out.println("funcao: "+f2.funcao);
System.out.printf("salario: %.2f",f2.salario);
System.out.println("Os dados do Funcionario 3 sao: ");
System.out.println("matricula: "+f3.mat);
System.out.println("nome: "+f3.nome);
System.out.println("funcao: "+f3.funcao);
System.out.printf("salario: %.2f",f3.salario);
}
catch(InputMismatchException e)
// Não obrigatório e não necessário se usasse o método hasNext().. antes de cada leitura // com next...
{
    System.out.println("Valor digitado nao corresponde ao esperado !");
}
}

public class Funcionario_1 // Exercício8
{
    private static int mat;

```

```

        private String nome, funcao;
        private double salario;
    }

    public class Funcionario_2 // Exercício9
    {
        private static int mat;
        private String nome, funcao;
        private double salario;

        public static void setMat(int mat_aux)
        {
            mat = mat_aux;
        }

        public static int getMat()
        {
            return mat;
        }

        public void setNome(String nome)
        {
            this.nome = nome;
        }

        public String getNome()
        {
            return nome;
        }

        public void setFuncao(String funcao)
        {
            this.funcao = funcao;
        }

        public String getFuncao()
        {
            return funcao;
        }

        public void setSalario(double salario)
        {
            this.salario = salario;
        }

        public double getSalario()
        {
            return salario;
        }
    }

    import java.util.*;
    public class TestaFuncionario_2 // Exercício10
    {
        public static void main(String args[])
        {

```

```

Scanner in = new Scanner(System.in);
Funcionario_2 f1, f2, f3;
f1 = new Funcionario_2();
f2 = new Funcionario_2();
f3 = new Funcionario_2();
try
{
    System.out.println("Entre com os dados para o Funcionario 1 ");
    System.out.print("Entre com a matricula: ");
    f1.setMat(in.nextInt());
    in.nextLine(); // Esta leitura é para consertar um bug do readLine()
    System.out.print("Entre com o nome: ");
    f1.setNome(in.nextLine());
    System.out.print("Entre com a funcao: ");
    f1.setFuncao(in.next());
    System.out.print("Entre com o salario: ");
    f1.setSalario(in.nextDouble());
    System.out.println("Entre com os dados para o Funcionario 2 ");
    System.out.print("Entre com a matricula: ");
    f2.setMat(in.nextInt());
    in.nextLine(); // Esta leitura é para consertar um bug do readLine()
    System.out.print("Entre com o nome: ");
    f2.setNome(in.nextLine());
    System.out.print("Entre com a funcao: ");
    f2.setFuncao(in.next());
    System.out.print("Entre com o salario: ");
    f2.setSalario(in.nextDouble());
    System.out.println("Entre com os dados para o Funcionario 3 ");
    System.out.print("Entre com a matricula: ");
    f3.setMat(in.nextInt());
    in.nextLine(); // Esta leitura é para consertar um bug do readLine()
    System.out.print("Entre com o nome: ");
    f3.setNome(in.nextLine());
    System.out.print("Entre com a funcao: ");
    f3.setFuncao(in.next());
    System.out.print("Entre com o salario: ");
    f3.setSalario(in.nextDouble());

    System.out.println("Os dados do Funcionario 1 sao: ");
    System.out.println("matricula: "+f1.getMat());
    System.out.println("nome: "+f1.getNome());
    System.out.println("funcao: "+f1.getFuncao());
    System.out.printf("salario: %.2f",f1.getSalario());
    System.out.println("Os dados do Funcionario 2 sao: ");
    System.out.println("matricula: "+f2.getMat());
    System.out.println("nome: "+f2.getNome());
    System.out.println("funcao: "+f2.getFuncao());
    System.out.printf("salario: %.2f",f2.getSalario());
    System.out.println("Os dados do Funcionario 3 sao: ");
    System.out.println("matricula: "+f3.getMat());
    System.out.println("nome: "+f3.getNome());
    System.out.println("funcao: "+f3.getFuncao());
    System.out.printf("salario: %.2f",f3.getSalario());
}
catch(InputMismatchException e)
{

```

```

        System.out.println("Valor digitado nao corresponde ao esperado !");
    }
}

```

```

public class Funcionario_3 // Exercício11

```

```

{
    private static int mat_aux=0; // mat_aux passa a ser um contador de objetos da classe
    private int mat; // mat passa a ser variável de instancia
    private String nome, funcao;
    private double salario;

    public Funcionario_3()
    {
        mat=++mat_aux;
    }

    /* public void setMat(int mat_aux)
    {
        mat = mat_aux; // Este método sai pois mat agora é de instância e autoincrementado
    }
    */

    public int getMat() // Não e mais estático devido a mat deixar de ser
    {
        return mat;
    }

    public void setNome(String nome)
    {
        this.nome = nome;
    }

    public String getNome()
    {
        return nome;
    }

    public void setFuncao(String funcao)
    {
        this.funcao = funcao;
    }

    public String getFuncao()
    {
        return funcao;
    }

    public void setSalario(double salario)
    {
        this.salario = salario;
    }

    public double getSalario()
    {
        return salario;
    }
}

```

```
}
```

```
public class Funcionario_4 // Exercício12
```

```
{
```

```
    private static int mat_aux=0; // mat_aux passa a ser um contador de objetos da classe
```

```
    private int mat; // mat passa a ser variável de instância
```

```
    private String nome, funcao;
```

```
    private double salario;
```

```
    public Funcionario_4()
```

```
    {
```

```
        mat=++mat_aux;
```

```
    }
```

```
    public Funcionario_4(String nome_aux, String funcao_aux, double sal_aux)
```

```
    {
```

```
        this(); // Isto faz com que o construtor acima seja chamado incrementando mat
```

```
        nome = nome_aux;
```

```
        funcao = funcao_aux;
```

```
        salario = sal_aux;
```

```
    }
```

```
    public int getMat()
```

```
    {
```

```
        return mat;
```

```
    }
```

```
    public void setNome(String nome)
```

```
    {
```

```
        this.nome = nome;
```

```
    }
```

```
    public String getNome()
```

```
    {
```

```
        return nome;
```

```
    }
```

```
    public void setFuncao(String funcao)
```

```
    {
```

```
        this.funcao = funcao;
```

```
    }
```

```
    public String getFuncao()
```

```
    {
```

```
        return funcao;
```

```
    }
```

```
    public void setSalario(double salario)
```

```
    {
```

```
        this.salario = salario;
```

```
    }
```

```
    public double getSalario()
```

```

        {
            return salario;
        }
    }
}

```

```

import java.util.*;
public class TestaFuncionario_3 // Exercício13
{
    public static void main(String args[])
    {
        Scanner in = new Scanner(System.in);
        Funcionario_3 f1, f2, f3;
        String nomeAux, funcaoAux;
        double salarioAux;
        try
        {
            System.out.println("Entre com os dados para o Funcionario 1 ");
            System.out.print("Entre com o nome: ");
            nomeAux = in.nextLine();
            System.out.print("Entre com a funcao: ");
            funcaoAux = in.next();
            System.out.print("Entre com o salario: ");
            salarioAux = in.nextDouble();
            f1 = new Funcionario(nomeAux, funcaoAux, salarioAux);
            System.out.println("Entre com os dados para o Funcionario 2 ");
            System.out.print("Entre com o nome: ");
            nomeAux = in.nextLine();
            System.out.print("Entre com a funcao: ");
            funcaoAux = in.next();
            System.out.print("Entre com o salario: ");
            salarioAux = in.nextDouble();
            f2 = new Funcionario(nomeAux, funcaoAux, salarioAux);
            System.out.println("Entre com os dados para o Funcionario 3 ");
            System.out.print("Entre com o nome: ");
            nomeAux = in.nextLine();
            System.out.print("Entre com a funcao: ");
            funcaoAux = in.next();
            System.out.print("Entre com o salario: ");
            salarioAux = in.nextDouble();
            f3 = new Funcionario(nomeAux, funcaoAux, salarioAux);

            System.out.println("Os dados do Funcionario 1 sao: ");
            System.out.println("matricula: "+f1.getMat());
            System.out.println("nome: "+f1.getNome());
            System.out.println("funcao: "+f1.getFuncao());
            System.out.printf("salario: %.2f",f1.getSalario());
            System.out.println();
            System.out.println("Os dados do Funcionario 2 sao: ");
            System.out.println("matricula: "+f2.getMat());
            System.out.println("nome: "+f2.getNome());
            System.out.println("funcao: "+f2.getFuncao());
            System.out.printf("salario: %.2f",f2.getSalario());
            System.out.println();
            System.out.println("Os dados do Funcionario 3 sao: ");
            System.out.println("matricula: "+f3.getMat());

```



```

        System.out.println("nome: "+f3.getNome());
        System.out.println("funcao: "+f3.getFuncao());
        System.out.printf("salario: %.2f",f3.getSalario());
    }
    catch(InputMismatchException e)
    {
        System.out.println("Valor digitado nao corresponde ao esperado !");
    }
}
}

public class Funcionario_4    // Exercício14
{
    private static int mat_aux=0; // mat_aux passa a ser um contador de objetos da classe
    private int mat; // mat passa a ser variável de instancia
    private String nome, funcao;
    private double salario;

    public Funcionario_4()
    {
        ma t = ++mat_aux;
    }

    public Funcionario_4(String nome_aux, String funcao_aux, double sal_aux)
    {
        this(); // Isto faz com que o construtor acima seja chamado incrementando mat
        nome = nome_aux;
        funcao = funcao_aux;
        salario = sal_aux;
    }
    // Solução da questão 14
    public Funcionario_4(String nome_aux, String funcao_aux)
    {
        this(nome_aux, funcao_aux, 0); // Isto faz com que o construtor acima seja chamado com salario = 0
    }
    // Fim da solução da questão 14

    public int getMat() // Não é mais estático devido a mat deixar de ser
    {
        return mat;
    }

    public void setNome(String nome)
    {
        this.nome = nome;
    }

    public String getNome()
    {
        return nome;
    }

    public void setFuncao(String funcao)
    {
        this.funcao = funcao;
    }
}

```

```

    public String getFuncao()
    {
        return funcao;
    }

    public void setSalario(double salario)
    {
        this.salario = salario;
    }

    public double getSalario()
    {
        return salario;
    }
}

```

// Exercício15 – Compilar a classe abaixo no mesmo diretório de Funcionario\_4

```

public class CargosSalarios
{
    public static double buscaSalario(String funcao)
    {
        if (funcao.equals("diretor"))
            return 5000.00;
        else
            if (funcao.equals("vendedor"))
                return 3000.00;
            else
                return 1000.00;
    }
}

```

// Substituir o construtor da questão 14 por este

```

public Funcionario_4(String nome_aux, String funcao_aux)
{
    this(nome_aux, funcao_aux, CargosSalarios.buscaSalario(funcao_aux));
}

```