

Instituto Federal do Norte de Minas Gerais - Campus Pirapora	
Disciplina: Linguagem de POO I	Prof.: Luciano Soares de Souza
Atividade: Exercício de Fixação	Turma: 4º Período Sistemas de Informação
Discente: Heitor Oliveira Campos.	

1. O objetivo aqui é criar um sistema para gerenciar os funcionários do Banco.
 - a) Crie uma classe `FuncionarioV1`. Ela deve ter o nome do funcionário (`String`), o departamento onde trabalha (`String`), seu salário (`double`), a data de entrada no banco (`String`), seu RG (`String`) e um atributo chamado `ativo` que é um valor booleano que indique se o funcionário está na empresa no momento ou se já foi embora.
 - b) Crie um método *bonifica* que aumenta o salario do funcionário de acordo com o parâmetro passado como argumento. Esse argumento representa a porcentagem de aumento que o funcionário irá receber.
 - c) Crie, também, um método *demite*, que não recebe parâmetro algum, só modifica o valor booleano indicando que o funcionário não trabalha mais na empresa.
 - d) Crie uma outra classe que contenha o *main* para poder testar a classe `FuncionarioV1`. (Sugestão de nome para classe de teste é `TestaFuncionarioV1`). Atribua valor a todos os atributos, chame os métodos e depois imprima os atributos que foram modificados nos métodos para verificar se os valores estão corretos..

```
public class FuncionarioV1 {

    double salario;
    //seus outros atributos e metodos

    void bonifica(double aumento){
        //desenvolver o metodo aqui
    }

    void demite(){
        //desenvolver o metodo aqui
    }

}
```

2. Copiando a classe `FuncionarioV1`, crie uma classe `FuncionarioV2` adicionando um método chamado *mostrar()* que não recebe nem devolve nenhum valor e somente imprime todos os atributos do funcionário.
 - a) Crie uma classe `TestaFuncionarioV2` onde, ao invés de imprimir os atributos diretamente (como feito anteriormente), você chamará o método *mostrar* para isso.

```
public class TestaFuncionarioV2 {

    public static void main(String[] args) {

        FuncionarioV2 f = new FuncionarioV2();

        //colocar os valores nos atributos aqui

        f.mostra(); //imprime os atributos

    }

}
```

3. Em qual dos casos abaixo o programa vai imprimir "iguais"?

a) Situação 1

```
public static void main(String[] args) {  
  
    FuncionarioV2 f1 = new FuncionarioV2();  
    f1.nome = "Pedro";  
    f1.salarario = 100.0;  
  
    FuncionarioV2 f2 = new FuncionarioV2();  
    f2.nome = "Predro";  
    f2.salarario = 100.0;  
  
    if(f1 == f2){  
        System.out.println("Iguais");  
    }  
  
}
```

b) Situação 2

```
public static void main(String[] args) {  
  
    FuncionarioV2 f1 = new FuncionarioV2();  
    f1.nome = "Pedro";  
    f1.salarario = 100.0;  
  
    FuncionarioV2 f2 = new FuncionarioV2();  
    f2 = f1;  
  
    if(f1 == f2){  
        System.out.println("Iguais");  
    }  
  
}
```

4. Cria uma classe FuncionarioV3, copiando tudo que tem em FuncionarioV2. Crie uma classe chamada Data que possui 3 campos int, para dia, mês e ano.

Modifique a classe FuncionarioV3 para que, ao invés de utilizar uma String para representar a data, utilize a classe Data criada. (É parecido como quando fizemos a classe Conta ter referência para um Cliente).

a) Crie uma classe TestaFuncionarioV3 para testar essa parte da atribuição da data. Atribua valor a todos os atributos do funcionario e chame todos os métodos (mostrar por último) (Obs: o método mostrar vai ter que ser modificado)

5. Crie uma classe FuncionarioV4 (baseada na FuncionarioV3) que tenha um método chamado equals e receba como parâmetro um outro FuncionarioV4 e compare se os dois são iguais baseados nos seus valores de atributos. Crie uma classe TestaFuncionarioV4 para testar o método equals.

6. Observe o programa a seguir e responda as questões (tente fazer só olhando). Obs: Leve em consideração que a classe Funcionário do código abaixo é igual ao FuncionárioV4.

a) Vai acontecer algum erro? Se sim, diga em qual linha e o motivo.

b) Se houver erro(s), imagine que a(s) linha(s) que causa(m) erro(s) não existe(m) mais. Diga o que seriam impresso pelo programa e a(s) respectiva(s) linha(s).

```

5 public static void main(String[] args) {
6
7     Funcionario f1 = new Funcionario();
8     f1.setSalario(100);
9
10    System.out.println(f1.getDataEntrada());
11
12    f1.getDataEntrada().setDia(5);
13
14    System.out.println(f1.getDataEntrada().getDia());
15
16    System.out.println(f1.getNome());
17
18 }

```



7. Crie uma classe Pessoa com Atributos: nome, idade. Método: void fazAniversario(). Crie uma classe TestaPessoa onde você irá criar uma pessoa, colocar seu nome e idade iniciais, fazer alguns aniversários (aumentando a idade) e imprimir no fim seu nome e sua idade.
8. Crie uma Classe: Porta Atributos: aberta, cor, dimensaoX, dimensaoY, dimensaoZ Métodos: void abre(), void fecha(), void pinta(String s), boolean estaAberta(). Crie uma classe TestaPorta e crie uma porta, abra e feche a mesma, pinte-a de diversas cores, altere suas dimensões e use o método estaAberta para verificar se ela está aberta.
9. Crie uma Classe: Casa Atributos: cor, porta1, porta2, porta3 Método: void pinta(String s), int quantasPortasEstaoAbertas(). Numa classe TestaCasa crie uma casa e pinte-a. Crie três portas e coloque-as na casa; abra e feche as mesmas como desejar. Utilize o método quantasPortasEstaoAbertas para imprimir o número de portas abertas.
10. Crie uma classe chamada Temperatura que não tem nenhum atributo e que tenha 2 métodos um chamado converterParaCelsius e outro chamado converterParaFahrenheit.
 - a) Esses métodos recebem um valor double como parâmetro e retornam um double com a temperatura convertida.
 1. Obs: Olhe na internet a fórmula de conversão. Crie uma classe MainTemperatura e teste os métodos.
11. Crie uma classe chamada Produto com os atributos preço e nome.
 - a) Nessa classe crie o método diminuir10 que diminui o preço em 10% e o método aumenta25 que aumenta o preço em 25%.
 - b) Crie uma classe MainProduto e teste a classe Produto.
12. Crie uma classe chama Esfera que tenha um atributo que represente seu raio. Além disso nessa classe deve ter um método que calcule o volume dessa esfera. Obs: O volume de uma esfera é dado como: $(4.0/3.0) * PI * raio^3$
13. Qual a saída do programa (o que vai ser impresso) nas seguintes situações? (Obs) leve em alguns casos pode ser erro, se for explique o que está errado.

a)

```
public static void main(String[] args) {  
    int t = 7;  
    int w = 8;  
    boolean y = false;  
    boolean x = true;  
    if(x && (x && y)){  
        System.out.println(w);  
    }else if ((t < w) && x){  
        if(x){  
            System.out.println(t * t);  
        }  
    }else if(y || (t < w)){  
        System.out.println(t * w);  
    }else{  
        if((t > 9) && x){  
            System.out.println(w * 2);  
        }  
    }  
}
```

Imprime 56, pois t(7) é menor que w(8), logo executa-se

b)

```
public static void main(String[] args) {  
    double c = 5.8;  
    int x = 1;  
    int y = 4;  
    int a = (c + x) / 2 * y;  
    System.out.println(a * a);  
}
```

Vai dar erro, neste caso deveria ser do tipo double para suportar o atributo c

c)

```
public static void main(String[] args) {

    int p = 7 / 3;
    char t = 'y';
    int x = 2;

    switch(p) {

        case 1:
            t = 'n';
            break;
        case 2:
            x = 3;
            t = 'w';
            break;
        case 3:
            x = 0;
            t = 't';
            break;
        default:
            x = 7;
            break;
    }

    System.out.println(x);
}
```

Handwritten annotations for code c):

- Next to `int p = 7 / 3;`, the value `2.33333` is written, with an arrow pointing to the `case 2:` block.
- The `case 2:` block is circled in green.
- An arrow points from the `case 2:` block to the `System.out.println(x);` line.
- The value `2` is written and underlined in green, representing the integer result of `7 / 3`.

d)

```
public static void main(String[] args) {
    int i = 0;
    int b = 1;
    int x = 7;
    double x = (i * i / b) + 5;
    System.out.println(x);
}
```

Handwritten annotations for code d):

- The variable `x` in the `double x` declaration is boxed in green.
- A green arrow points from the boxed `x` to the text on the right.

vai dar erro, o atributo já havia sido declarado anteriormente, se trocar o segundo x por y rodaria normalmente sendo impresso 5.0

e)

```
public static void main(String[] args) {
    int t = 7;
    int y = 2;
    double w = (t / y) + 5;
    t = t + t + y;
    y = y + y + t;
    w = w + (t / y) + 5;
    System.out.println(w);
}
```

Handwritten annotations for code e):

- Next to `double w = (t / y) + 5;`, the value `5.8` is written.
- Next to `t = t + t + y;`, the value `16` is written.
- Next to `y = y + y + t;`, the value `20` is written.
- Next to `w = w + (t / y) + 5;`, the value `11.6` is written.

f)

```
public static void main(String[] args) {
    int x = 1;
    int y = 8;
    do{
        x = x + y;
        y = y + x - 1;
        x++;
        y--;
    }while( (x + y) < 100 );

    System.out.println(x - y);
}
```

$x = 65$
 $y = 100$

$\rightarrow -35$

g)

```
public static void main(String[] args) {
    int t = 5;
    int h = 2;
    int w = 7;
    do{
        t = t + 3 + w * h;
        w--;
        h++;
    }while(t % 3 != 0 );
    System.out.println(t / 2);
}
```

$w = 4$
 $h = 5$
 $t = 46$

$\rightarrow 33$

h)

```
public static void main(String[] args) {
    int w = 2 * 3 + 7;
    int z = w * w + 5;

    while(w < (z + 3)){
        w = w - 1;
        z = w - 3;
        z = z * z;
    }

    System.out.println(w);
}
```

$z = 1$

$\rightarrow 4$

i)

```
public static void main(String[] args) {
    int i = 0;
    int b = 1;
    double x = (i * i / b) + 5;
    System.out.println(b);
}
```

$\rightarrow 1$

j)

```
public static void main(String[] args) {
    int w = 7;
    int s = 2;
    double n = (w / s) + 5;
    System.out.println(n);
}
```

$\rightarrow \underline{\underline{5}}$

k)

```
public static void main(String[] args) {
    int a = 7;
    int b = 8;
    boolean c = false;
    boolean d = true;
    if(c && d){
        System.out.println(a);
    }else if(c || (a > b) ){
        System.out.println(b);
    }else if(d && (a < a)){
        if(d){
            System.out.println(a * a);
        }
    }else{
        if(d || (a > 9 )){
            System.out.println(b * 2);
        }
    }
}
```

imprime 16 ←

l)

```
public static void main(String[] args) {
    // TODO Auto-generated method stub
    int a = 0;
    int b = 5;
    do{
        a = a + b;
        a++;
    }while( (a + b) < 10 );

    System.out.println(a); → 6
}
```

m)

```
public static void main(String[] args) {
    int x = 2;
    double c = 5.8;
    int a = (c + x) / 2;
    System.out.println(a * a); → 15.209999999999999
}
```

vai dar erro, para solucionar o atributo a deve ser do tipo double, sendo impresso 15.209999999999999

n)

```
public static void main(String[] args) {
    int x = 2 * 3;
    int y = x * x;

    while(x < y){
        x = x - 1;
        y = x - 3;
        y = y * y;
    }
    System.out.println(x); → 5
}
```

y=4

o)

```
public static void main(String[] args) {  
  
    char c = 'y';  
    int y = 2;  
    int z = 7 % 2;  $\rightarrow 1$   
  
    switch(z) {  
        case 1:   
            c = 'n';  
            break;  
        case 2:  
            y = 3;  
            c = 'w';  
            break;  
        case 3:  
            y = 0;  
            c = 't';  
            break;  
        default:  
            y = 7;  
            break;  
    }  
  
    System.out.println(c);  $\rightarrow \underline{\underline{n}}$   
}
```

p)

```
public static void main(String[] args) {  
    int x = 5;  
    int b = 2;  
    do{  
        x = x + 3 * b;  
        if(x % x == 0) {  $\checkmark$   
            break;  
        }  
    }while(x % 2 != 0 );  $\checkmark$   
    System.out.println(x);  $\rightarrow 11$   
}
```