

USCS - UNIVERSIDADE MUNICIPAL DE SÃO CAETANO DO SUL

Curso de análise e desenvolvimento de sistemas

Rodolfo Costa Moreno

Programação Orientada a Objetos

1a Atividade

Prof. Renato Carioca Duarte

São Caetano do Sul

2022

1. Defina uma classe em Java chamada “Empregado” para armazenar as informações de um

Empregado de uma empresa. A classe deve armazenar o código do Empregado, nome, cargo e

salário. Estes atributos devem ser definidos com o operador de acesso public. O método construtor

deve receber o código do Empregado, o seu nome, seu cargo e seu salário. Definir também um

construtor sem passagem de parâmetros. Deve oferecer ainda uma função chamada “ImprimeEmpregado()” que imprime as informações do Empregado. Definir também uma função

chamada “TotalEmpregados()” que imprime a quantidade de Empregados criados no programa à

medida em que o programa for sendo executado.

2. Escrever uma classe chamada “TesteEmpregado” que possui um método main() para instanciar

Empregados.

a) Criar um objeto referenciado pela variável “e1” por meio do construtor com os parâmetros:

código=1232, nome=Paulo Silva, cargo: “Gerente Financeiro” e salário: R\$ 15.400,00.

b) Criar um segundo objeto referenciado pela variável “e2” por meio do construtor default (sem

parâmetros). Por meio de operações de atribuição inicializar o objeto com os seguintes dados:

código=4567, nome=“Ana Paula”, cargo: “Analista de Sistemas Senior” e salário: R\$ 10.700,00.

c) Para cada objeto criado, executar os métodos que imprimem informações de cada Empregado

criado.

d) Executar a função “TotalEmpregados()” que imprime a quantidade de funcionários existente na

empresa.

3. Modificar a classe “Empregado” para que esteja de acordo com os conceitos da Programação

Orientada a Objetos com relação ao Encapsulamento. Para isso, modifique os atributos do

Empregado para private.

4. Execute a Classe “TesteEmpregado” e explique o que ocorreu durante a execução.

5. Modificar a classe “Empregado” adicionando métodos de acesso à classe de modo que somente

estes métodos podem acessar as propriedades de cada Empregado.

6. Modificar a classe “TesteEmpregado” para que esta utilize os métodos getter() e setter() para

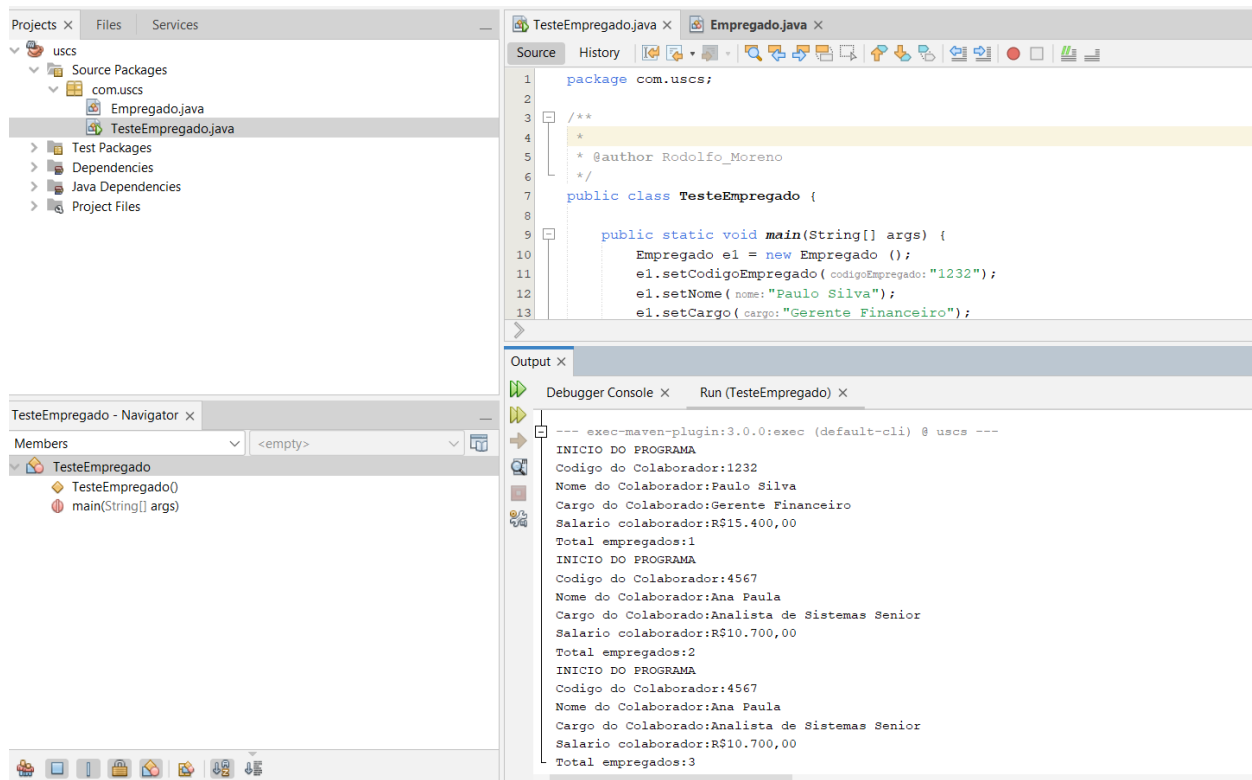
acesso aos dados do funcionário ao invés de acessá-los diretamente.

7. Modificar a classe “Empregado” para que o código do Empregado seja AUTO-INCREMENTADO. Ou

seja, o código do Empregado deverá ser automaticamente fornecido a partir do valor 1 para cada

Empregado criado. Modifique os construtores da classe “Empregado” e a classe “TesteEmpregado” para que esta manutenção seja efetivada.

Obs. As duas classes devem ser criadas dentro de um package chamado uscs.



```
package com.uscs;
```

```
/**
```

```
*
```

```
* @author Rodolfo_Moreno
```

```
*/
```

```
public class TesteEmpregado {
```

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

```
        Empregado e1 = new Empregado ();
```

```
        e1.setCodigoEmpregado("1232");
```

```
        e1.setNome("Paulo Silva");
```

```

        e1.setCargo("Gerente Financeiro");

        e1.setSalario("R$15.400,00");

        e1.imprimeEmpregado();

        Empregado e2 = new Empregado ();

        e2.setCodigoEmpregado("4567");

        e2.setNome("Ana Paula");

        e2.setCargo("Analista de Sistemas Senior");

        e2.setSalario("R$10.700,00");

        e2.imprimeEmpregado();

        Empregado e3 = new Empregado ();

        e3.setCodigoEmpregado("4567");

        e3.setNome("Ana Paula");

        e3.setCargo("Analista de Sistemas Senior");

        e3.setSalario("R$10.700,00");

        e3.imprimeEmpregado();

    }

}

=====

package com.uscs;

/**
 *
 * @author Rodolfo_Moreno
 */

```

```
public class Empregado {  
  
    private String codigoEmpregado;  
  
    private String nome;  
  
    private String cargo;  
  
    private String salario;  
  
    private static int contador;  
  
    /**  
     * @return the codigoEmpregado  
     */  
    public String getCodigoEmpregado() {  
        return codigoEmpregado;  
    }  
  
    /**  
     * @param codigoEmpregado the codigoEmpregado to set  
     */  
    public void setCodigoEmpregado(String codigoEmpregado) {  
        contador++;  
        this.codigoEmpregado = codigoEmpregado;  
    }  
  
    /**
```

```
* @return the nome

*/

public String getNome() {

    return nome;

}

/**

* @param nome the nome to set

*/

public void setNome(String nome) {

    this.nome = nome;

}

/**

* @return the cargo

*/

public String getCargo() {

    return cargo;

}

/**

* @param cargo the cargo to set

*/
```

```
public void setCargo(String cargo) {  
    this.cargo = cargo;  
}
```

```
/**
```

```
 * @return the salario
```

```
 */
```

```
public String getSalario() {  
    return salario;  
}
```

```
/**
```

```
 * @param salario the salario to set
```

```
 */
```

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

```
/**
```

```
 * @return the contador
```

```
 */
```

```
public int getContador() {  
    return contador;  
}
```



```
/**
 * @param contador the contador to set
 */
public void setContador(int contador) {
    this.contador = contador;
}

public void totalEmpregados(){
    setContador(getContador()+1);
}

public void imprimeEmpregado(){
    System.out.println("INICIO DO PROGRAMA");
    System.out.println("Codigo do Colaborador:"+getCodigoEmpregado());
    System.out.println("Nome do Colaborador:"+getNome());
    System.out.println("Cargo do Colaborado:"+getCargo());
    System.out.println("Salario colaborador:"+getSalario());
    System.out.println("Total empregados:"+getContador());
}
}
```