

	<p>Instituto Federal de Educação, Ciência e Tecnologia da Bahia –  <b>Campus Irecê</b>  <b>Disciplina: Linguagem Técnica I</b>  <b>Profº Jonatas Bastos</b></p>
-----------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Nome: \_\_\_\_\_

### LISTA DE EXERCÍCIO 5 – OO e Herança

- 1) Adicione o modificador de visibilidade (`private`, se necessário) para cada atributo e método da classe `Funcionario`. Tente criar um `Funcionario` no `main` e modificar ou ler um de seus atributos privados. O que acontece?
- 2) Crie os `getters` e `setters` necessários da sua classe `Funcionario` .
- 3) Modifique suas classes que acessam e modificam atributos de um `Funcionario` para utilizar os `getters` e `setters` recém criados.

Por exemplo, onde você encontra:

```
f.salario = 100;
System.out.println(f.salario);
```

passa para:

```
f.setSalario(100);
System.out.println(f.getSalario());
```

- 4) Faça com que sua classe `Funcionario` possa receber, opcionalmente, o nome do `Funcionario` durante a criação do objeto. Utilize construtores para obter esse resultado.

Dica: utilize um construtor sem argumentos também, para o caso de a pessoa não querer passar o nome do `Funcionario`.

- 5) Adicione um atributo na classe `Funcionario` de tipo `int` que se chama `identificador`. Esse `identifi- cador` deve ter um valor único para cada instância do tipo `Funcionario`. O primeiro `Funcionario` instanciado tem `identificador` 1, o segundo 2, e assim por diante. Você deve utilizar os recursos aprendidos aqui para resolver esse problema. Crie um `getter` para o `identificador`. Devemos ter um `setter`?

6) Crie os getters e setters da sua classe Empresa e coloque seus atributos como private. Lembre-se de que não necessariamente todos os atributos devem ter getters e setters. Por exemplo, na classe Empresa, seria interessante ter um setter e getter para a sua array de funcionários? Não seria mais interessante ter um método como este?

```
class Empresa {  
  
    // ...  
  
    public Funcionario getFuncionario (int posicao) {  
        return this.empregados[posicao];  
    }  
}
```

7) Na classe Empresa, em vez de criar um array de tamanho fixo, receba como parâmetro no construtor o tamanho do array de Funcionario. Com esse construtor, o que acontece se tentarmos dar `new Empresa()` sem passar argumento algum? Por quê?

8) Como garantir que datas como 31/2/2005 não sejam aceitas pela sua classe Data?

9) Crie a classe PessoaFisica. Queremos ter a garantia de que pessoa física alguma tenha CPF invalido, nem seja criada PessoaFisica sem cpf inicial. (você não precisa escrever o algoritmo de validação de cpf, basta passar o cpf por um método valida(String x)....)