

# PRE1 – Programação Orientada a Objetos

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

Crie um sistema de gestão de funcionários em uma empresa. O sistema deve incluir o controle de funcionários e a associação deles a departamentos. Abaixo estão os requisitos detalhados para implementar o sistema utilizando conceitos de Programação Orientada a Objetos (POO):

## Requisitos:

### 1. Classe `Pessoa`:

- Deve ser a classe base para os funcionários da empresa.
- A classe deve ter dois atributos:
  - `nome`: público, representando o nome da pessoa.
  - `idade`: público, representando a idade da pessoa.
- Ao instanciar um objeto da classe `Pessoa`, o sistema deve verificar se a pessoa tem pelo menos 18 anos. Caso contrário, o sistema deve solicitar a idade novamente até que um valor válido seja inserido.
- A verificação da idade deve ser realizada por um método `set_idade`.

### 2. Classe `Funcionario` (herda de `Pessoa`):

- Deve representar um funcionário e herdar as propriedades de `Pessoa`.
- A classe `Funcionario` deve ter um atributo privado:
  - `__salario`: representando o salário mensal do funcionário.
- Deve conter métodos acessores (getter e setter) para o salário:
  - `get_salario()`: retorna o valor do salário.
  - `set_salario(salario)`: permite modificar o valor do salário.
- A classe também deve ter um método para calcular o salário anual:
  - `calcular_salario_anual()`: multiplica o salário mensal por 12 e retorna o valor.

### 3. Classe `Departamento`:

- Representa um departamento da empresa que possui uma lista de funcionários.
- Deve ter os seguintes métodos:
  - `adicionar_funcionario(funcionario)`: adiciona um funcionário à lista de funcionários.
  - `calcular_total_salarios()`: calcula o total de salários anuais de todos os funcionários do departamento.
  - `listar_funcionarios()`: exibe o nome de cada funcionário e o seu respectivo salário anual.

### 4. Exemplo de Uso:

- Crie três objetos da classe `Funcionario` com diferentes salários e idades. Observe que um dos funcionários deve ter idade inferior a 18 anos para demonstrar a funcionalidade de verificação.
- Adicione esses funcionários a um departamento e liste todos os funcionários junto com seus salários anuais.
- Calcule e exiba o total gasto com salários anuais no departamento.

## Regras:

- O atributo `__salario` deve ser privado e acessado apenas por meio dos métodos getter e setter.
- O sistema deve garantir que todos os funcionários adicionados ao departamento tenham idade válida (18 anos ou mais).

## Dicas:

- Utilize herança para reutilizar código entre as classes.
- Trabalhe com encapsulamento, mantendo o salário como um atributo privado e utilizando métodos para acessá-lo ou modificá-lo.
- Pense no uso adequado de listas para armazenar os funcionários no departamento.