

## Exercício: Classe Abstrata ou Interface ???

### Estrutura de Dados e Armazenamento

Imagine o seguinte cenário:

Uma escola tem vários funcionários.

Todos os funcionários da escola devem ter o método `calculaSalario()`.

Dentro desse cenário, considere os funcionários: Secretaria, Professor e Coordenador.

Como fazer para que as classes que representam esses 3 funcionários sejam obrigadas a implementar o método `calculaSalario()`? Seria melhor usar classe abstrata ou interface? Justifique.

Além disso, no final de cada ano, a escola paga um bônus sobre o salário apenas dos coordenadores e dos professores. Os demais funcionários da escola não ganham esse bônus.

Há uma regra na escola de que o coordenador não deve dar aula, e de que o professor não deve coordenar.

Se quisermos obrigar que a classe que representa o Professor e a classe que representa o Coordenador implementem o método: `getValorBonus()`, qual seria a melhor forma: classe abstrata ou interface? Justifique.

**Atenção:** As outras classes que representam funcionários que não recebem bônus não devem ter o método `getValorBonus()`. Além da classe Professor e da classe Coordenador, você também deverá considerar a classe Secretaria, que não recebe esse bônus. **Não** é necessário representar no diagrama de classes nem implementar as outras classes de funcionários.

**Também não** há necessidade de representar nem implementar a classe Escola.

Desenhe o diagrama de classes, de acordo com a sua resposta.

Atributos de Secretaria: `cpf`, `nome`, `salario`, `extra` (supondo que esse extra é um valor adicional por trabalhar horas a mais no mês)

Atributos de Professor: `cpf`, `nome`, `quantidade de aulas por semana`, `valor da hora aula`

Atributos de Coordenador: `cpf`, `nome`, `quantidade de horas de coordenação por semana`, `valor da hora de coordenação`, `quantidade de cursos que coordena`

Cálculo do salário da Secretaria: `salario + extra`

Cálculo do salário do Professor: `quantidade de aulas por semana * valor da hora aula * 4.5`

Cálculo do salário do Coordenador: `quantidade de horas de coordenação por semana * valor da hora de coordenação * 4.5 + quantidade de cursos que coordena * 500`

Bônus do Professor: 15% do seu salário

Bônus do Coordenador: 20% do seu salário

Considere também a classe `ControleBonus`, que terá como atributo um ou mais `List` para conter todos os objetos que recebem bônus. Pergunta: é necessário ter 2 `List` ou apenas um?

Implemente na classe `ControleBonus` o método que adiciona o objeto ao(s) `List`, o método que exibe o conteúdo do(s) `List`, e o método que calcula o total de bônus do(s) `List`.

Não se esqueça de implementar o construtor e o `toString()` nas diversas classes, parecido com o que fizemos nos exercícios.

Implemente sua solução.

Implemente a classe que contém o main, e dentro do main, crie objetos Professor, Coordenador e Secretaria, crie objeto da classe ControleBonus, e chame seus métodos.

O polimorfismo está presente nesse sistema? Justifique.

A resposta das perguntas pode ser entregue num arquivo txt zipado junto com o projeto, ou como comentário dentro do código.