



Segunda lista de exercícios de Programação Orientada a Objetos

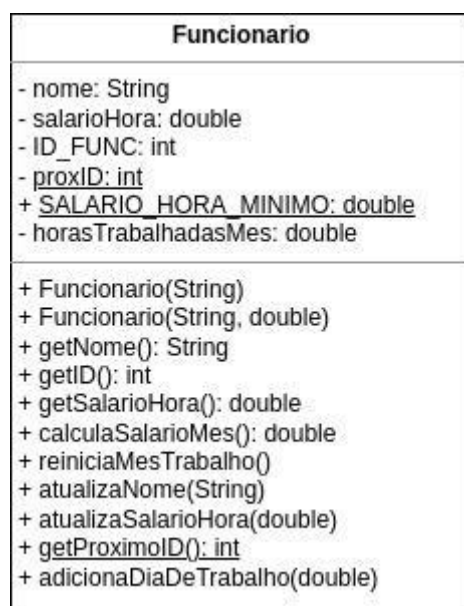
Questões teóricas

1. Explique as principais diferenças entre programação estruturada/modular e programação orientada a objetos.
2. Diferencie uma classe de um objeto.
3. Defina atributo de objeto, atributo de classe, método de objeto e método de classe.

Questões práticas

Ao criar os métodos modificadores dos atributos privados, validar conforme explicitado em cada caso. Se o valor passado não passe na validação, não modifique o atributo.

4. Implemente uma classe Funcionário conforme o diagrama a seguir:



Um objeto da classe Funcionario tem os seguintes atributos:



- nome, que armazena o nome completo do funcionário;
- salarioHora, que armazena o valor base da hora do funcionário;
- ID_FUNC, que armazena o ID do funcionário gerado uma única vez. Este ID não pode ser alterado;
- horasTrabalhadasMes, que armazena a quantidade de horas trabalhadas no mês corrente;

A classe Funcionario tem dois atributos de classe:

- proxID, que armazena o próximo ID de funcionário disponível. Inicia com 1.
- SALARIO_HORA_MINIMO, que armazena o valor mínimo pago a um funcionário por hora trabalhada, que no nosso caso é R\$12,50 por hora

Além desses atributos, a classe tem dois métodos construtores possíveis, um que só recebe o nome do funcionário e atribui a ele o valor mínimo de hora trabalhada e o segundo que recebe tanto o nome do funcionário quanto o valor a ser pago pela hora trabalhada deste funcionário.

Agora vamos a explicação dos outros métodos:

- getNome retorna o nome do funcionário;
- getIDFuncionario retorna o ID atribuído ao funcionário;
- getSalarioHora retorna o valor base da hora do funcionário;
- calcularSalarioMes computa o salário do mês do funcionário através das horasTrabalhadasMes e do salarioHora;
- reiniciarMesTrabalho reinicia o acumulador em horasTrabalhadasMes;
- atualizarNome recebe uma string como parâmetro e atualiza o nome da pessoa. É necessário ter pelo menos um sobrenome;
- atualizarSalarioHora recebe um double como parâmetro e atualiza o salário base da hora do funcionário. O novo valor de hora trabalhada tem que ser maior que o valor atual;
- getProximoID retorna o valor atual do atributo de classe proxID;
- adicionarDiaDeTrabalho recebe a quantidade de horas trabalhadas e atualizar o valor de horasTrabalhadasMes. O dia de trabalho tem que ter pelo menos 4 horas completas para ser válido.