INSTITUTO FEDERAL
Paraíba

BACHARELADO EM ENGENHARIA DE CONTROLE E AUTOMAÇÃO
PROGRAMAÇÃO ORIENTADA A OBJETOS

Turma: Terceiro Período (P3) **Semestre**: 2022.1 **Campus**: Cajazeiras - PB
Professor: Leandro Luttiane da S. Linhares
Nome: _____ **Data**: ___/___/_____

POO - Lista de Exercícios 04: Herança de Classes

1. Crie uma classe denominada `Pessoa` que possa como atributo protegido o nome da pessoa. Logo após, desenvolva duas outras classes, denominadas `PessoaFisica` e `PessoaJuridica` que herdam da classe `Pessoa`. A classe `PessoaFisica` deve ter como atributos privados o CPF e o nome da pessoa, enquanto a classe `PessoaJuridica` deve possuir como atributos privados o CNPJ, a razão social e o nome fantasia. Crie métodos `get` e `set` para todos os atributos das três classes.

2. Desenvolva uma classe chamada `Funcionario` derivda da classe `PessoaFisica` da questão 1. Essa classe deve ter como atributos privados a matrícula, o salário base do funcionário, a carga horária mensal (quantidade de horas mensais) e a quantidade de horas trabalhadas no mês. A classe deve ainda ter um método protegido chamado `calculaBonus` que deverá simplesmente retornar o valor 0.0 (zero). Esse método terá um propósito maior adiante. Além disso, a classe deve possuir um método público chamado `calculaSalarioBruto`, que não terá nenhum parâmetro e deverá ser capaz de calcular e retornar o salario bruto através da seguinte equação: $salarioBase\ x\ quantidadeHorasTrabalhadas \div caragaHorariaMensal + calculaBonus()$. Por fim, crie métodos `get` e `set` para os atributos. Note que a quantidade de horas trabalhadas não poderá superar a carga horária mensal e nem ser inferior a 0. Garanta isso dentro da classe.

3. Crie uma classe chamada `Gerente` que deve herdar da classe `Funcionario` da questão 2. Essa classe deve possuir três atributos privados adicionais: quantidade de funcionários gerenciados, quantidade de horas excedentes e senha no sistema. Considere que somente o gerente terá acesso ao sistema de uma determinada empresa, de modo que a autenticação do gerente se dará por um método chamado `autenticar` que deverá ser criado na classe `Gerente`. Esse método vai receber uma senha qualquer por parâmetro e deverá retornar `true` se a senha informada for igual à senha armazenada no objeto, ou `false` caso contrário.

   Deve haver métodos `get` e `set` para a quantidade de funcionários gerenciados e quantidade de horas excedentes. Todavia, a senha só poderá ser alterada mediante a chamada de um método (`alterarSenha`) que receba por parâmetro a senha antiga e a nova senha. O método altera a senha do gerente para a nova senha quando a senha armazenada no atributo do objeto é igual à senha antiga informada por parâmetro. Nesse caso, o método retorna `true`. Caso a senha antiga não seja igual à senha do gerente, o método retorna `false`. Se não houver senha informada para o gerente, o método não fará a verificação de igualdade da senha antiga e permitirá a alteração. O atributo senha é privado e nenhum método retorna seu valor. Sobrescreva o método que altera o a quantidade de horas trabalhadas de modo que, quando ultrapassar a carga horária mensal, o excedente seja atribuído ao atributo de horas excedentes. Por fim, sobrescreva o método `calculaBonus`. Nessa classe, o bônus será calculado de modo que, para cada funcionário gerenciado, o gerente terá 0.5% sobre o seu próprio salário base. Esse bônus não poderá ultrapassar 30% do salário bruto, independente da quantidade de funcionários gerenciados. Além disso, para cada hora adicional trabalhada, o bônus ainda terá a adição de $salarioBase\ x\ 1.5 \div cargaHorariaMensal$ (ou seja, para cada hora extra, será pago 150% de uma hora de trabalho do funcionário).

4. Crie uma classe chamada `Estagiario` que deve herdar da classe `Funcionario` da questão 2. Nessa classe, adicione um atributo para armazenar o tempo de contrato restante (em meses) e outro para controlar o tipo de estagiário: jovem aprendiz ou universitário. Sobrescreva o método `calculaBonus` de modo que o bônus do estagiário se dê da seguinte forma: se for jovem aprendiz, o bônus será 5% do salário base dividido pela

quantidade de meses restantes para o fim do contrato; se for universitário, o bônus será de 10% do salário base dividido pela quantidade de meses restantes para o fim do contrato.

5. Crie uma classe chamada `Cliente` que herde da classe `PessoaFisica` da questão 1. Essa classe deverá ter atributos privados que armazenem um telefone e um endereço. Crie métodos `get` e `set` para esses atributos.
6. Crie uma classe chamada `Empresa` que herde da classe `PessoaJuridica` da questão 1. Essa classe deverá ter uma lista de funcionários e um outra lista de clientes (pode ser vetor). Crie métodos para adicionar funcionários e clientes. Crie um método para imprimir matrícula, nome e salário bruto dos funcionários e outro para imprimir nome, telefone e endereço dos clientes. Crie também um método chamado `calcularFolhaDePagamento` que deverá calcular o salário bruto de todos os funcionários e retornar o total a ser gasto com os funcionários.
7. Crie um único programa que teste todas essas classes. Considere que haverá uma empresa e que o usuário poderá adicionar funcionários e clientes nessa empresa, além de poder solicitar listagem de funcionários e de clientes, bem como verificar qual será o valor da folha de pagamento total.