Universidade Federal da Bahia
Instituto de Matemática
Departamento de Ciência da Computação
MAT A 55 – Linguagem de Programação Orientada a Objeto.
Professora Rita Suzana Pitangueira Maciel

12/12/2018
SEGUNDA AVALIAÇÃO

OBSERVAÇÕES
Respostas a lápis não serão consideradas para revisão.
Use apenas os conceitos ministrados até o momento.
Será analisado o uso correto do paradigma e do estilo de programação JAVA.
A legibilidade faz parte da pontuação das questões.
Consultas de qualquer natureza não são permitidas e nem o uso de celular.
As avaliações serão recolhidas às 14h50m.

Cenário

Uma determinada indústria possui empregados terceirizados e assalariados. Para realizar suas atividades a indústria depende também de serviços de outras empresas fornecedoras (terceirizadas). Todo final de mês a indústria realiza o pagamento dos fornecedores e também de seus empregados. A indústria deseja automatizar uma aplicação para o pagamento dos empregados e fornecedores.

Todos os empregados possuem atributos como nome (String), CPF (String), matrícula (String), salário base (double), cargo (String) e tempo de serviço (int), além dos detalhes da sua conta bancária, que são código banco (int), agência (int) e número da conta (String). A indústria possui dois tipos de empregados que possuem informações específicas: operador e engenheiro. Os empregados do tipo operador possuem informações adicionais como especialidade e planta locada. Os empregados do tipo engenheiro possuem informações adicionais como tipo de engenheiro (mecânico, químico, civil, etc.) e uma indicação se ocupam cargo de chefia.

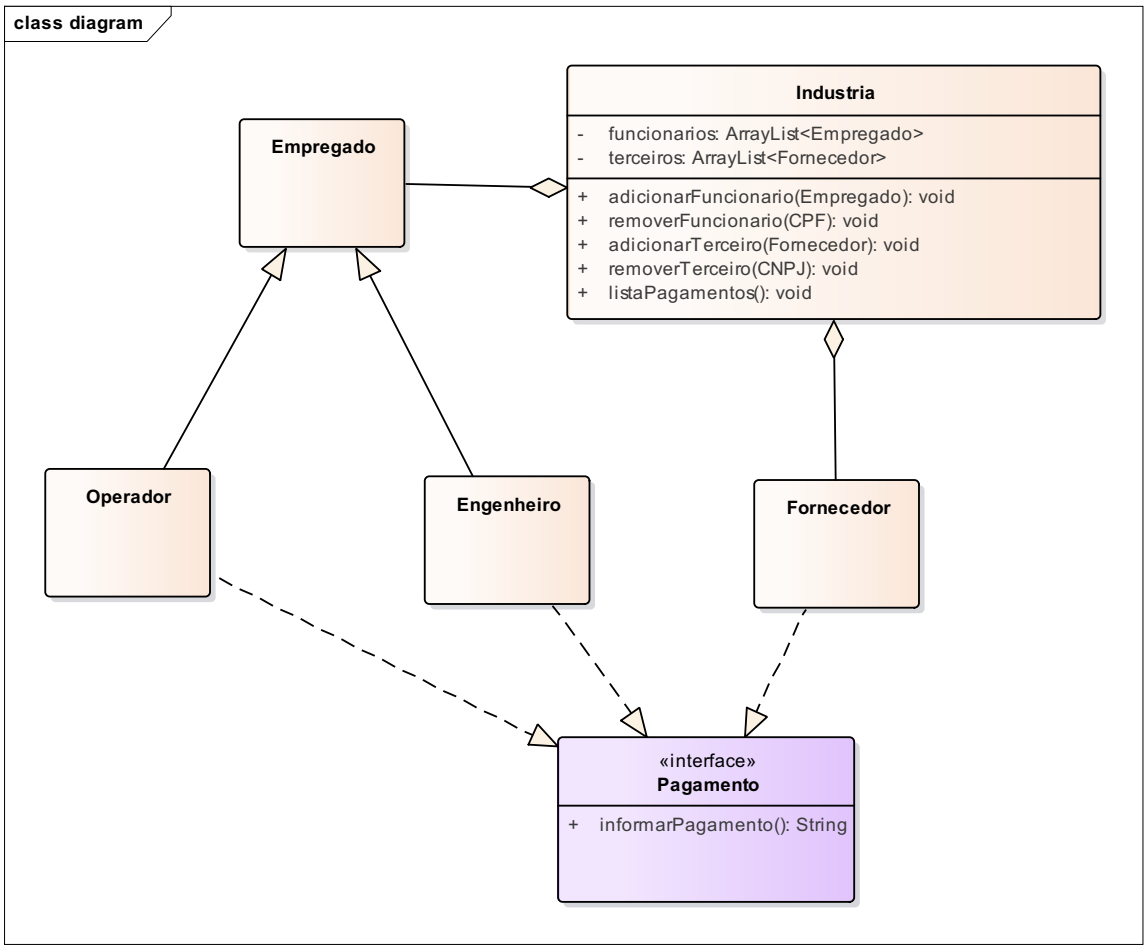
A empresa fornecedora é caracterizada pelo nome (String), nome fantasia (String), CNPJ (String), valor mensal do serviço prestado (double) e detalhes da conta bancária da empresa (código banco, agência e número da conta).

No final do mês a indústria apresenta à contabilidade uma lista de pagamentos a efetuar de empregados e fornecedores. Essa lista é composta por uma relação com nome (do empregado ou da empresa fornecedora), CPF (ou CNPJ, se empresa fornecedora), dados bancários e valor a ser depositado, uma linha para cada pagamento a ser efetuado.

O valor do salário líquido dos empregados é calculado como 80% do (salário base + adicional de tempo), onde o adicional de tempo é proporcional ao tempo de serviço, sendo 10% do salário base para cada ano (só considera ano inteiro). Para empregados do tipo operador existe um adicional de periculosidade de 30% em cima do salario base. Com isso o líquido do operador é calculado como 80% do (salário base + adicional de tempo + adicional de periculosidade). Para empregados do tipo de engenheiro que exercem cargo de chefia é pago um adicional integral de chefia (sem desconto) de 10% em cima do valor líquido previamente calculado. ATENÇÃO: Empregados do tipo engenheiro não podem ser do tipo operador e vice-versa.

Para os fornecedores, a indústria efetua apenas um desconto de 18% em cima do valor mensal do serviço prestado a título de impostos.

A solução proposta obedece ao seguinte diagrama de classe simplificado:



Posto o cenário, agora cabe a você:

1) Crie as classes Empregado, Operador, Engenheiro e Fornecedor com os atributos descritos no cenário, obedecendo a relação de herança proposta no diagrama. Crie também a interface Pagamento com o único método listado. **(2,0)**

2) Crie a classe Industria com os atributos indicados no diagrama, que são os ArrayLists funcionarios e terceiros. Elabore também métodos para adicionar e remover objetos nos ArrayLists, conforme o diagrama. Para adicionar em funcionarios, passar apenas o objeto de Empregado como parâmetro. Para remover em funcionarios, passar apenas o CPF

como parâmetro. Para adicionar em terceiros, passar apenas o objeto de Fornecedor como parâmetro. Para remover em terceiros, passar apenas o CNPJ como parâmetro. **(3,0)**

3) Desenvolva o método listaPagamentos() da classe Industria de modo que ele seja utilizado para emitir a lista de pagamentos para a contabilidade, conforme explicado no cenário **(2,0)**. Para uma melhor reutilização, este método tem que ser desenvolvido de forma que cada classe envolvida (Empregado, Operador, Engenheiro e Fornecedor) tenha um método próprio que reflita sua responsabilidade de exibir sua própria informação de pagamento **(3,0)**.

ATENÇÃO:

- As decisões de existirem ou não atributos e métodos adicionais (ou auxiliares, se aplicáveis) e também de onde os métodos deverão ficar, fazem parte do conhecimento do paradigma OO a ser avaliado.

- Todos os atributos de todas as classes criadas devem ser privados e seu acesso disponibilizado exclusivamente pelos métodos acessores (*get* e *set*).

- Modularize seu código de forma que ele seja o máximo reutilizável, utilizando o máximo dos conceitos vistos de orientação a objeto.