

Atividade Acadêmica: Algoritmos e Programação – Estruturas Lineares

Professor: Guilherme Silva de Lacerda (guilhermeslacerda@gmail.com – gslacerda@unisinos.br)

Laboratório 3

Importante! Regras a serem seguidas

- 1) Todo e qualquer código deve ter seu teste equivalente (classe para testar o código)

Exemplo: Classe Conta tem que ter uma ContaTeste que realize os testes nela

- 2) Respeite as convenções de código

(<https://www.oracle.com/technetwork/java/codeconventions-150003.pdf>)

- 3) Estruture o projeto com pacotes, conforme exemplo

Nome do Projeto: Laboratorio2_Cinema

pacote1

Classe1

Classe2

pacote2

Classe3

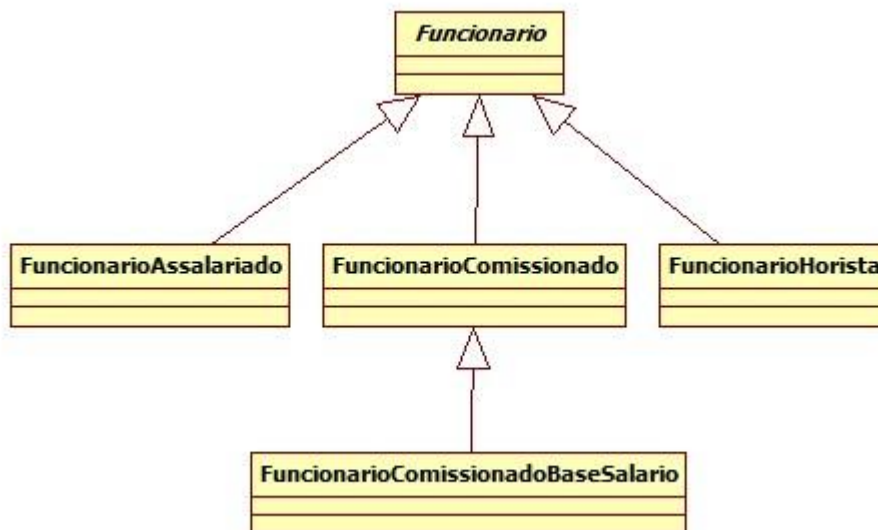
Classe4

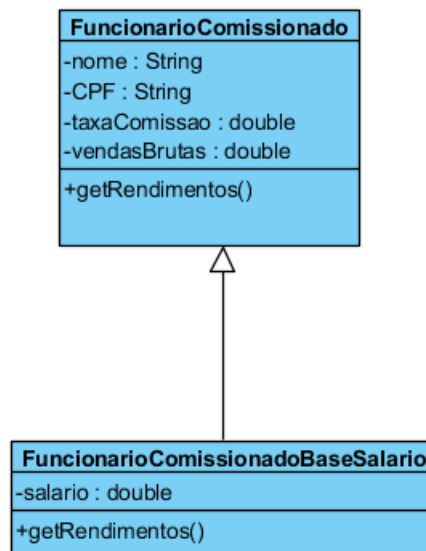
Importante:

- Atividade desenvolvida em duplas
- Atividade a ser entregue no ambiente EAD
- Formato de entrega: Arquivo ZIP dos códigos (sugestão de nome: **Lab3_NomeAluno1_NomeAluno2.zip**)

Enunciado

Observe os seguintes diagramas:





Escrever um programa Java que realize a computação da folha de pagamentos de uma empresa. Nesta empresa, os funcionários são comissionados recebem uma porcentagem de suas vendas (**FuncionarioComissionado**), enquanto funcionários comissionados com salário base recebem o salário base mais um percentual de suas vendas (**FuncionarioComissionadoBaseSalario**).

Para calcular os rendimentos para o **FuncionarioComissionado** isso é feito pela comissão X o percentual de vendas. Para **FuncionarioComissionadoBaseSalario**, usem também o salário base + (comissão X o percentual de vendas).

O **FuncionarioAssalariado** possui um salário semanal. Para o **FuncionarioHorista**, a regra do cálculo de salário é: *Se as horas trabalhadas na semana for menor que 40, salário por hora X horas trabalhadas na semana. Se for mais que 40 horas, 40 X salário por hora + (horas trabalhada – 40) X salário por hora X 1,5.*

Use os conceitos OO para organizar o projeto. Neste sentido, procure estruturar a hierarquia de classes, aproveitando atributos e métodos. Métodos get/set foram suprimidos.

Pode simular vários empregados nos testes, com situações distintas. Também não esqueçam de usar as boas práticas de programação discutidas em aula e os testes para cada tipo de funcionário.