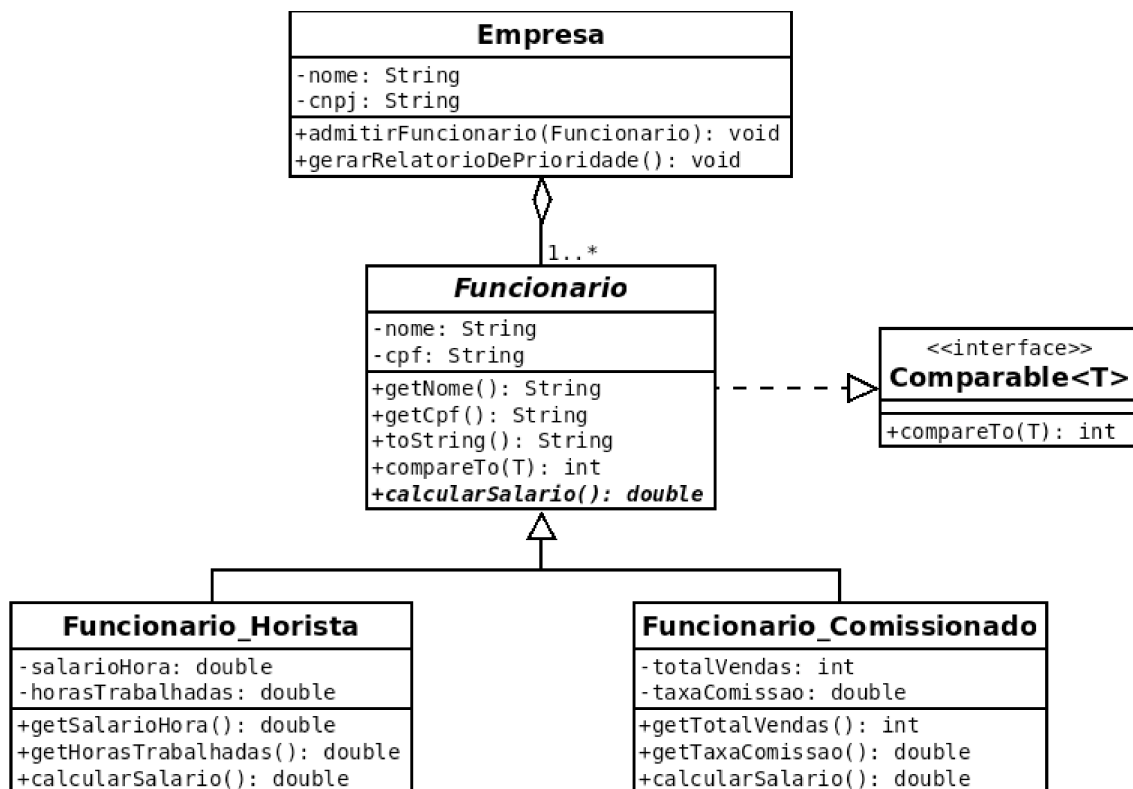


## GCC178 – Práticas de Programação Orientada a Objetos

Uma empresa possui o setor financeiro que é responsável por conceder reajuste de salário para seus trabalhadores. A política desse setor é dar prioridade para reajuste de salário para aqueles trabalhadores que têm menor salário. Para auxiliar essa empresa nessa tarefa, você foi contratado para implementar um sistema que gere um relatório contendo os dados dos funcionários da empresa ordenados de acordo com a prioridade de reajuste de salário, ou seja, nesse relatório deverão aparecer primeiro os funcionários que têm menores salários.

Dado o diagrama de classes a seguir, **seguindo as boas práticas em programação** orientada a objetos, complete o código Java apresentado a seguir de modo que o mesmo implemente a modelagem apresentada no diagrama e atenda os requisitos listados a seguir. Vale observar que a classe *TesteEmpresa*, que não está representada no diagrama de classes, não deve sofrer qualquer alteração, inclusão ou remoção de código.



Os seguintes requisitos devem ser atendidos para essa implementação:

a) Apresente uma solução funcione para qualquer quantidade de funcionários instanciados e passados por parâmetro na chamada do método *admitirFuncionario* na função *main*, ou seja, nenhuma outra alteração no código deverá ser necessária para se ajustar a diferentes quantidades de funcionários.

b) O valor do salário do funcionário horista é calculado multiplicando-se o valor do salário/hora pela quantidade de horas trabalhadas. No caso dos objetos instanciados na classe *TesteEmpresa*, o valor do salário/hora do funcionário Luiz é de R\$ 100,00 e a quantidade de horas trabalhadas igual a 160,0. Já o salário do funcionário comissionado é dado pela multiplicação da quantidade total de vendas pela taxa de comissão. No caso dos objetos instanciados na classe *TesteEmpresa*, o funcionário Zé vendeu 300 itens com uma taxa de comissão de R\$ 50,00.

c) Todos os trechos de código apresentados anteriormente deverão fazer parte da sua implementação sem que qualquer palavra seja alterada (apenas inclusões podem ser realizadas). Além disso, eles devem ter alguma utilidade na sua implementação, ou seja, seu código não pode

ter um método que nunca é utilizado.

d) A chamada ao método *gerarRelatorioDePrioridade* na função *main* deve resultar na seguinte saída (impressão na tela):

Nome: Zé  
CPF: 987654321-00  
Salário: 15000.0

Nome: Luiz  
CPF: 123456789-10  
Salário: 16000.0

```
public class TesteEmpresa {
    public static void main(String[] args) {
        Empresa empresa = new Empresa("EmpresaX", "123456789/1000");
        Funcionario func1 = new FuncionarioHorista("Luiz", "123456789-10", 100.0, 160.0);
        Funcionario func2 = new FuncionarioComissionado("Zé", "987654321-00", 300, 50.0);
        ;
        empresa.admitirFuncionario(func1);
        empresa.admitirFuncionario(func2);
        ;
        empresa.gerarRelatorioDePrioridade();
    }
}

import java.util.*;
public class Empresa {

    public void gerarRelatorioDePrioridade() {

        for(Funcionario f: funcionarios) {
            System.out.println(f);
        }
    }
}

public class Funcionario {

    public String getNome() {
        return nome;
    }

    public String getCpf() {
        return cpf;
    }
}
```