

FCT/Unesp – Presidente Prudente
Programação Orientada a Objetos – Prova I
Prof. Danilo Medeiros Eler

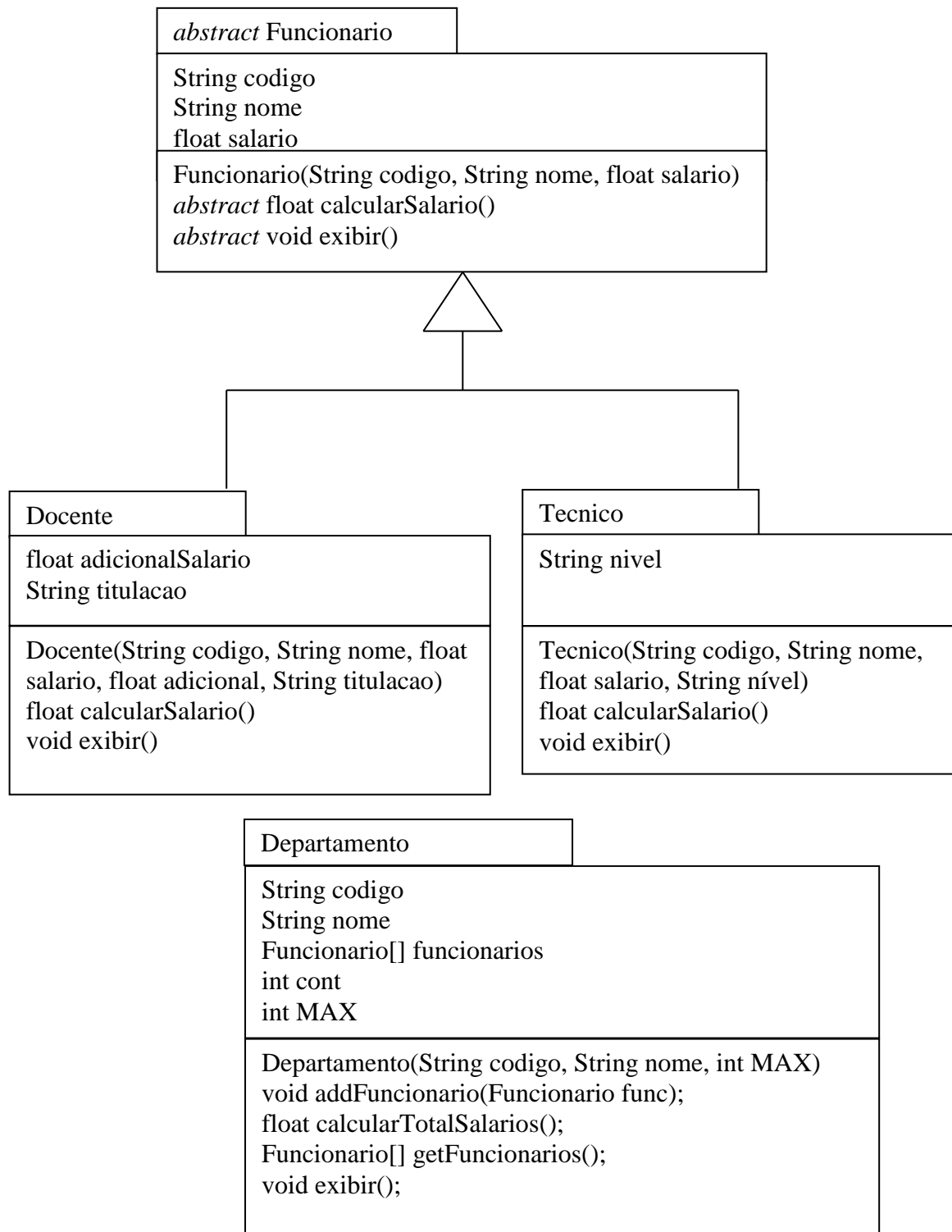
Nome: _____

1) Algumas classes de parte de um sistema de uma universidade estão ilustradas a seguir. Nesse exercício, você implementará apenas a classe Departamento, com seus métodos descritos no diagrama de classe.

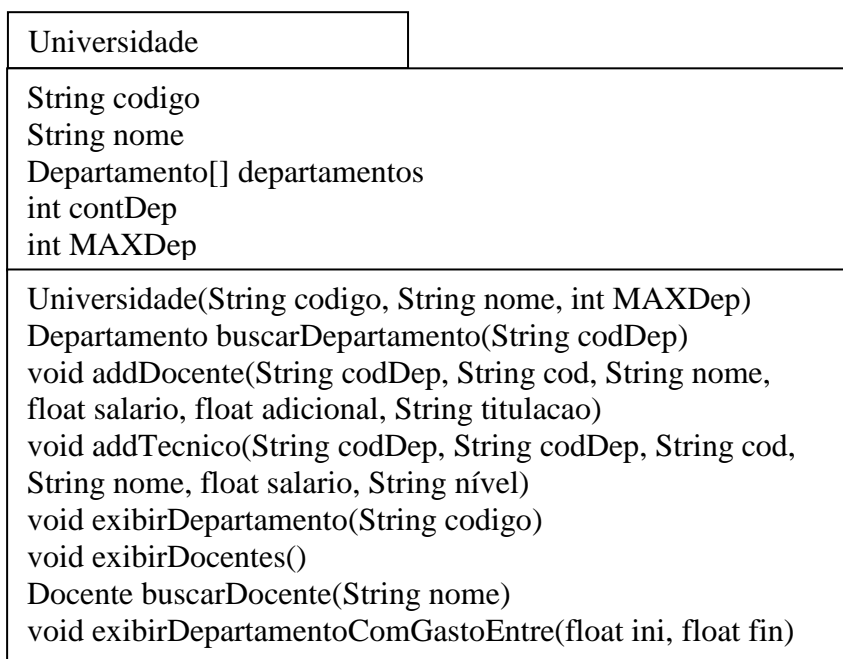
Nessa modelagem, a universidade possui dois tipos de funcionários: Docente e Técnico. Eles diferem na maneira como o seu salário é calculado. Para calcular o valor final do salário existe um método chamado `calcularSalario()`, o qual é implementado de maneira distinta para o Técnico e para o Docente.

O sistema também controla os departamentos, os quais possuem vários funcionários. De modo simplificado, os departamentos estão representados na classe Departamento, que possui um método chamado **`addFuncionario(Funcionario func)`** para adicionar um funcionário ao departamento. Outro método da classe Departamento é o **`calcularTotalSalarios()`** que deve retornar a soma de todos os salários gastos com os funcionários do departamento. **Use o polimorfismo** para a implementação dos métodos que utilizam objetos do tipo funcionário. Também há um método denominado **`exibir()`**, o qual deve exibir na tela todas as informações (atributos) da classe. Devem ser exibidos os dados do departamento, dos funcionários do departamento (use o método `exibir` do funcionário) e o valor total gasto com funcionários. Por fim, o método **`getFuncionarios()`** retorna o vetor de funcionários. **Observação:** você deverá implementar somente a classe Departamento e seus métodos, suponha que as outras classes já estão implementadas.

Lembre-se de implementar o construtor da classe Departamento e também de instanciar o vetor. **Não** é necessário implementar os métodos de **`set`** e **`get`** dos atributos nesta prova. Se quiser, você pode usar *sout* ao invés de `System.out.println`, por exemplo, `sout("Departamento: " + nome);`. **(4.0)**



2) Implemente uma classe chamada **Universidade** para gerenciar os dados e algumas operações de um sistema que utiliza as classes descritas no Exercício 1. Sempre que possível, utilize os métodos implementados no Exercício 1. O diagrama da classe **Universidade** é ilustrado a seguir. (6.0)



O método **buscarDepartamento** recebe como parâmetro o código do departamento e efetua uma busca no vetor **departamentos**. Ele retorna o primeiro departamento que possui o código passado por parâmetro; caso contrário, se não encontrar, retorna *null*.

Os métodos **addDocente** e **addTecnico** adicionam, respectivamente, um funcionário Docente ou Técnico a um departamento específico. Para tanto, são passados como parâmetros os dados dos funcionários (docente ou técnico). Adicionalmente, esses métodos recebem como parâmetro o código do departamento no qual o funcionário será adicionado. Assim, os métodos **addDocente** e **addTecnico** devem criar uma instância da classe Docente ou da classe Tecnico para ser adicionada ao departamento específico. Utilize o método **buscarDepartamento** para recuperar o departamento em que esses funcionários serão adicionados.

O método **exibirDepartamento(String codigo)** exibe os dados de um departamento específico que possui o código passado por parâmetro. Utilize o método **buscarDepartamento** para encontrar o departamento.

O método **exibirDocentes** exibe os dados dos funcionários docentes pertencentes à universidade. Se preferir, você pode supor que existe um atributo **tipo** na classe **Funcionario** para distinguir os objetos de cada classe, ou usar o *instanceof*.

O método **exibirDepartamentoComGastoEntre(float ini, float fin)** deve listar todos os departamentos cujo gasto com funcionários está entre **ini** e **fin**, isto é, **ini** representa um valor inicial e **fin** representa um valor final. Exiba o nome do departamento e o gasto total com funcionários.

O método **buscarDocente(String nome)** retorna o primeiro funcionário docente que possui o nome passado por parâmetro. Se não encontrar, retorna *null*.

Não é necessário implementar *get* e *set*.