

Lista de Exercícios 03

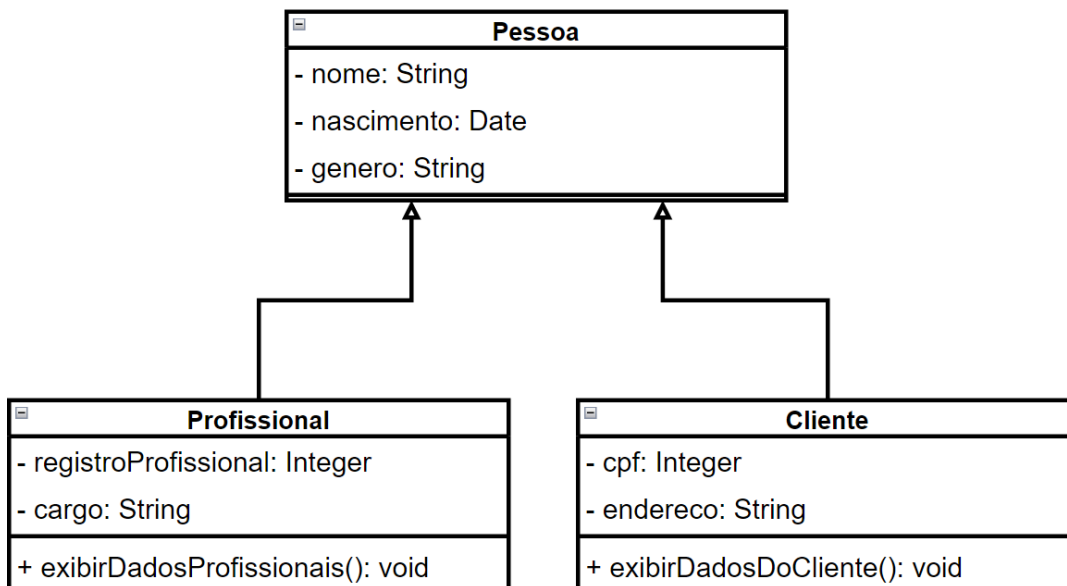
Herança

Cada aluno deverá, individualmente, resolver os exercícios abaixo:

Orientações:

- Para cada exercício faça:
 - a) Elabore ou complemente o diagrama de classes utilizando UML quando o diagrama não for dado;
 - b) Implemente ou complemente o código em Java quando o código não for dado;
 - c) Implemente um aplicativo específico para demonstração dos códigos.

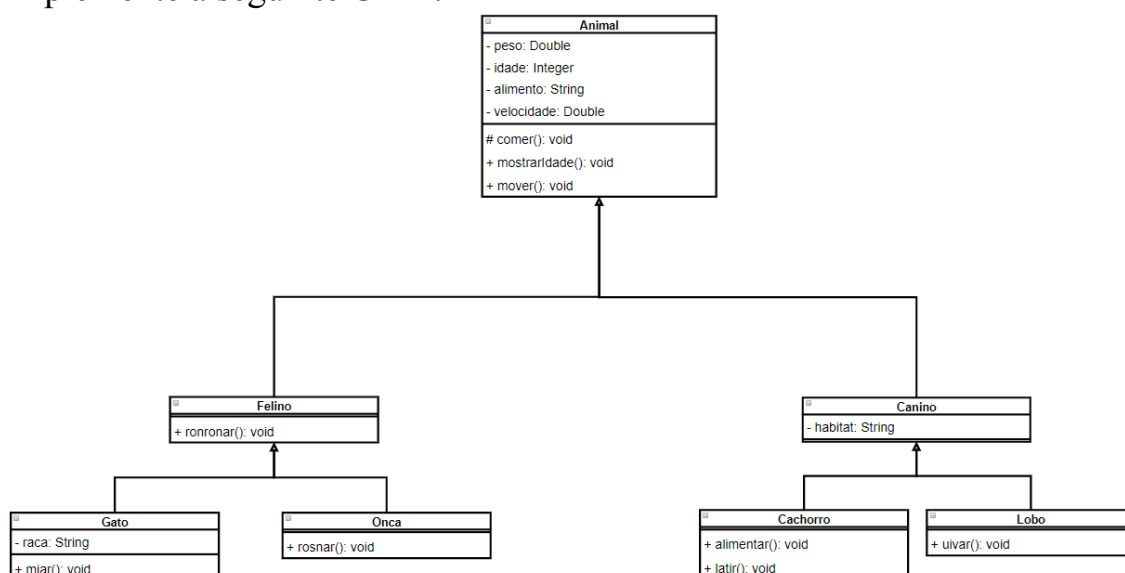
1. Para a seguinte UML faça:



- a. Implemente as classes em Java empregando corretamente o conceito de herança.

- Para cada classe, crie construtores que recebam por parâmetro valores iniciais para todos os atributos da classe em si e também de sua classe pai
- Implemente nos métodos **exibirDadosProfissionais** e **exibirDadosDoCliente**, a apresentação dos valores de todos os atributos
- Crie um aplicativo que demonstre o funcionamento das classes.

2. Implemente a seguinte UML:



- Evolua o exercício 2 criando uma classe **Peixe** que herde da classe **Animal** e obedeça à seguinte descrição:
 - Possua um atributo **caracteristica** do tipo String;
 - Crie um método construtor que receba por parâmetro os valores iniciais de cada um dos atributos, incluindo os atributos da classe Animal e atribua-os aos seus respectivos atributos.
 - Crie ainda os métodos **get** e **set** para o atributo **caracteristica**.
 - Crie um método **dadosDoPeixe()** sem parâmetro e do tipo **void**, que, quando chamado, imprime na tela uma espécie de relatório informando os dados do peixe incluindo os dados da classe pai.
- Modele a UML e implemente o respectivo código em Java para o seguinte exemplo:
 - Crie uma classe **Imovel** com os atributos **endereço** e **preço**.

- b. Crie uma classe **Novo**, que herda **Imovel** e possua um adicional no preço.
 - c. Crie métodos de acesso e impressão tanto para o valor inicial e como para o adicional.
 - d. Crie uma classe **Velho**, que herda **Imovel** e possui um desconto no preço.
 - e. Crie métodos de acesso e impressão tanto para o valor inicial como para este desconto.
 - f. Todas as classes devem possuir construtores para todos os atributos dela e da(s) classe(s) herdada(s).
5. Crie uma classe chamada **Ingresso** que possua um valor em reais, data, nome do evento e os métodos **imprimeValor** e **imprimeDadosDoEvento**.
- a. Crie uma classe **VIP**, que herda **Ingresso** e possua um valor adicional.
 - b. Crie um método que retorne o valor do ingresso **VIP** com o adicional incluído.
 - c. Crie uma classe **Normal**, que herda **Ingresso** e possua um método que imprime: *"Ingresso Normal"*.
 - d. Crie uma classe **CamaroteInferior**, que herda **VIP** e possua a localização do ingresso e métodos para acessar e imprimir esta localização.
 - e. Crie uma classe **CamaroteSuperior**, que herda **VIP** e que é mais cara. Além da localização deve também possuir valor adicional e um método para retornar o valor do ingresso.
 - f. Todas as classes devem possuir construtores para todos os atributos dela e da(s) classe(s) herdada(s).

Bom trabalho.