

Lista de Exercício III
Unidade III - Herança
Centro Universitário Christus

Nome:	Matrícula:
Professor(a): Maurício Moreira Neto	Semestre: 2021.1
Disciplina: Linguagens de Programação II (Orientado a Objetos)	
Curso: Sistemas de Informação	
Instruções: <ul style="list-style-type: none">• A lista de exercícios é individual.• As questões práticas devem ser implementadas usando a linguagem de programação Java.• Os arquivos das resoluções devem ser compactados em um arquivo para entrega (.zip ou .rar)	

Questões de Herança

1. Em relação aos conceitos pertinentes ao conceito de herança em Programação Orientada a Objetos, assinale verdadeiro (V) ou falsa (F) em cada alternativa a seguir.

- a. () Uma subclasse é uma classe especializada que herda métodos e atributos de uma superclasse.
- b. () A palavra reservada *extends* define a herança entre classes.
- c. () É possível fazer sobrescrita de método sem herança.
- d. () Todos os atributos e métodos definidos em uma superclasse, independentes dos modificadores de visibilidade, podem ser acessados nas subclasses desta superclasse.

I) V - F - V - F

II) V - V - V - F

III) V - V - F - F

IV) F - F - V - F

V) F - V - V - F

2. Analise o diagrama de classes da Figura 1 e indique qual a alternativa contém a afirmação correta a respeito deste diagrama. Justifique sua resposta às alternativas falsas.

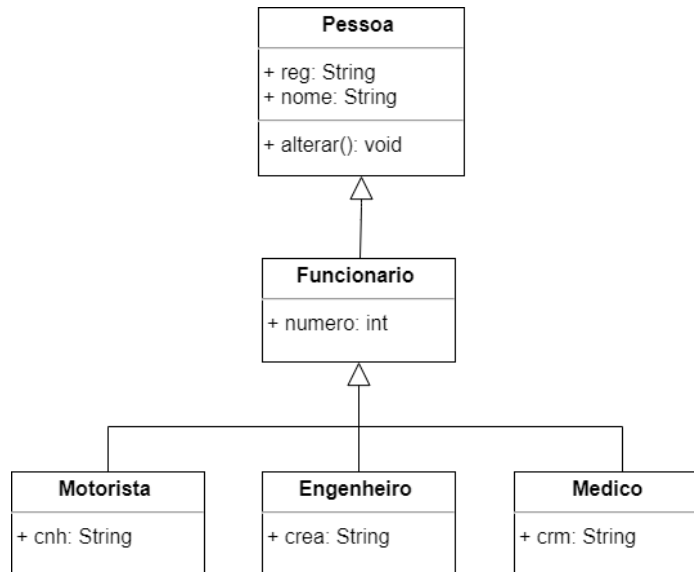


Figura 1. Diagrama de Classes da Questão 2.

- a. ☐ Ao criar um objeto da classe Motorista, ele não terá acesso ao valor do atributo nome da classe Pessoa.
- b. ☐ A classe Engenheiro tem acesso ao atributo cnh.
- c. ☐ A classe Engenheiro estende a classe Funcionario.
- d. ☐ A classe Médico pode acessar os atributos das subclasses de mesma hierarquia.

Justificativa

3. Implemente uma superclasse *Pessoa* que possua os seguintes atributos: *id* do tipo inteiro, *nome* do tipo cadeia de caracteres (String) e *email* do tipo cadeia de caracteres (String). O atributo *id* só pode ser diretamente visível e acessado dentro da própria classe *Pessoa*. Já o atributo *nome* só pode ser diretamente visível e acessado dentro da classe *Pessoa* e das subclasses que herdam de *Pessoa*. Já o atributo *email* pode ser diretamente visível e acessado dentro da classe *Pessoa*, dentro das subclasses de *Pessoa* e por meio dos objetos instanciados da classe *Pessoa* ou das subclasses de *Pessoa*. Como o atributo *nome* só pode ser diretamente acessado pela classe *Pessoa* e pelas subclasses de *Pessoa*, implemente um método que permita a leitura do valor do atributo *nome* por meio de objetos instanciados a partir da classe *Pessoa* e de suas classes derivadas. Implemente também, pelo menos, uma subclasse de *Pessoa*. Por fim, implemente uma classe *Principal* que possua o método *main* e instancie um objeto da superclasse *Pessoa* e dois objetos da subclasse de *Pessoa*. A resposta desta questão pode ser feita na linguagem Java.

4. Implemente o sistema apresentado pelo diagrama de classes da Figura 2 usando a linguagem de programação Java.

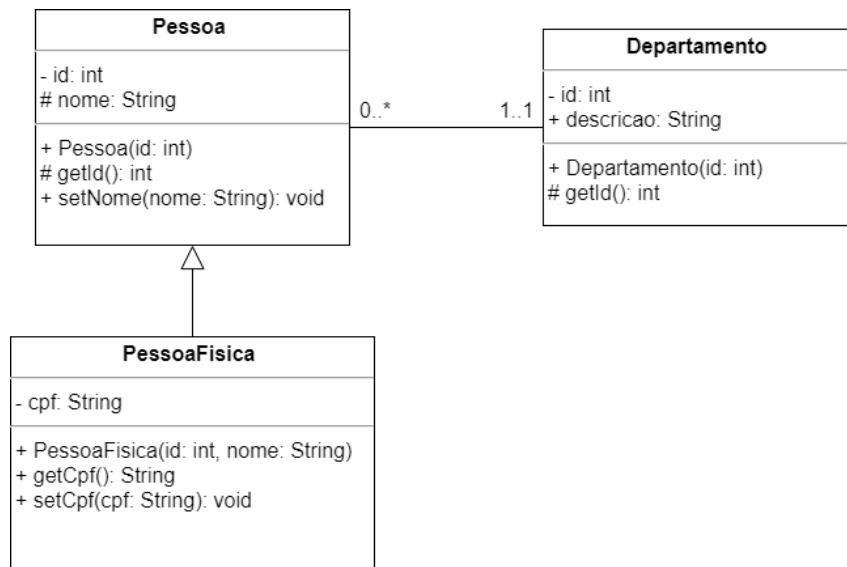


Figura 2. Diagrama de classes da questão 4.

5. O mecanismo de herança em programação orientado a objetos permite que uma nova classe (subclasse) nasça a partir de uma classe já existente (superclasse). Com base no conceito de herança, implemente, usando a linguagem Java, o sistema representado pelo diagrama da Figura 3.

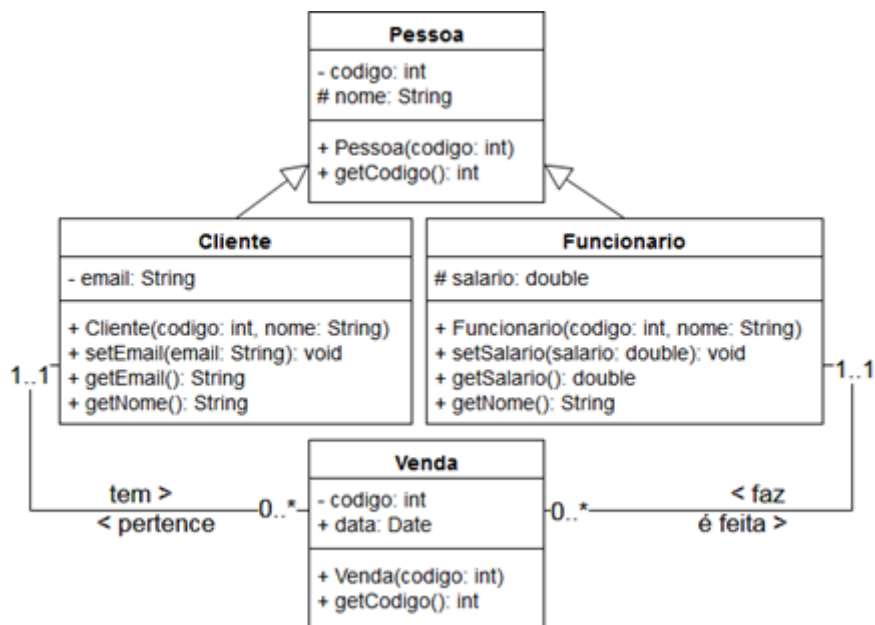


Figura 3. Diagrama de classes da questão 5.