

Lista de Exercício II
Unidade II - Conceitos Básicos de Orientação a Objetos
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 Conceitos de Básicos de Programação Orientado a Objeto

1. Em relação aos conceitos pertinentes à Programação Orientada a Objetos, assinale a Verdadeiro (V) ou Falsa (F) em cada afirmativa a seguir e justifique a(s) alternativa(s) falsa(s):

- () Uma classe é uma instância de um objeto.
- () Os valores armazenados em atributos definem o estado do objeto.
- () Em assinaturas de métodos é necessário especificar, obrigatoriamente, pelo menos um parâmetro (argumento).
- () A classe especifica um conjunto de propriedades e comportamentos de um objeto.

A sequência correta, de cima para baixo, é:

- A) F - V - V - V
- B) V - V - V - F
- C) F - V - F - F
- D) V - V - F - V
- E) F - V - F - V

Justificativa

2. Faça um programa usando a linguagem de programação Java que implemente uma classe Pessoa que possui os atributos nome, altura e idade. A classe Pessoa deve conter um construtor para inicializar os atributos e os seguintes métodos: um método para mostrar os dados da pessoa e outro método para calcular o seu ano de nascimento. A Figura 1 apresenta o diagrama UML da classe Pessoa.

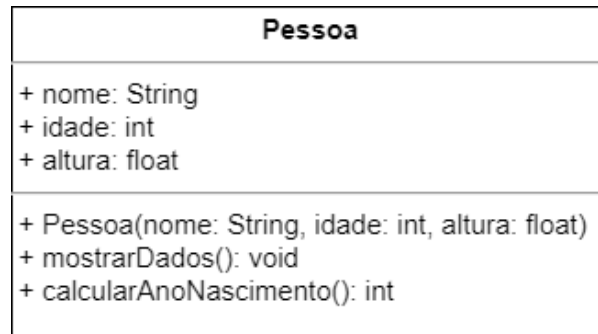


Figura 1. Diagrama da classe Pessoa.

3. Crie uma classe *Televisão* e uma classe *ControleRemoto* que pode controlar o *volume* e trocar os *canais* da televisão. O controle remoto permite as seguintes ações:

- Aumentar ou diminuir o volume de som em uma unidade.
- Aumentar e diminuir o número do canal em uma unidade.
- Trocar para um canal indicado.
- Mostrar o volume de som e o canal selecionado atualmente.

Questões de Relacionamento entre Classes (Associações)

4. Crie um programa que implemente uma classe **Disciplina** que possui os atributos: *nome da disciplina* e *código*. A classe **Disciplina** deve conter um construtor para inicializar os atributos, um método que retorne o seu nome, um método para adicionar um professor e um método para retornar o nome do professor (caso tenha). Implemente também uma classe **Professor** que possui os atributos *nome*, *id*. A classe **Professor** deve possuir um construtor para inicializar os atributos. Uma disciplina pode possuir nenhum ou pelo menos um professor responsável. A Figura 2 apresenta o diagrama UML da associação.

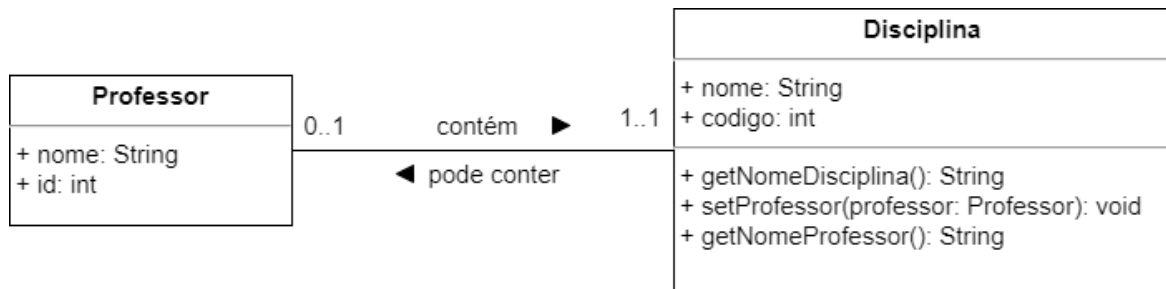


Figura 2. Diagrama UML da associação das classes **Disciplina** e **Professor**.

5. Uma **Universidade** possui diversos **Cursos**. Cada **Curso** pode possuir nenhum ou diversos **Alunos** matriculados. Implemente um sistema de acordo com o diagrama UML da Figura 3.

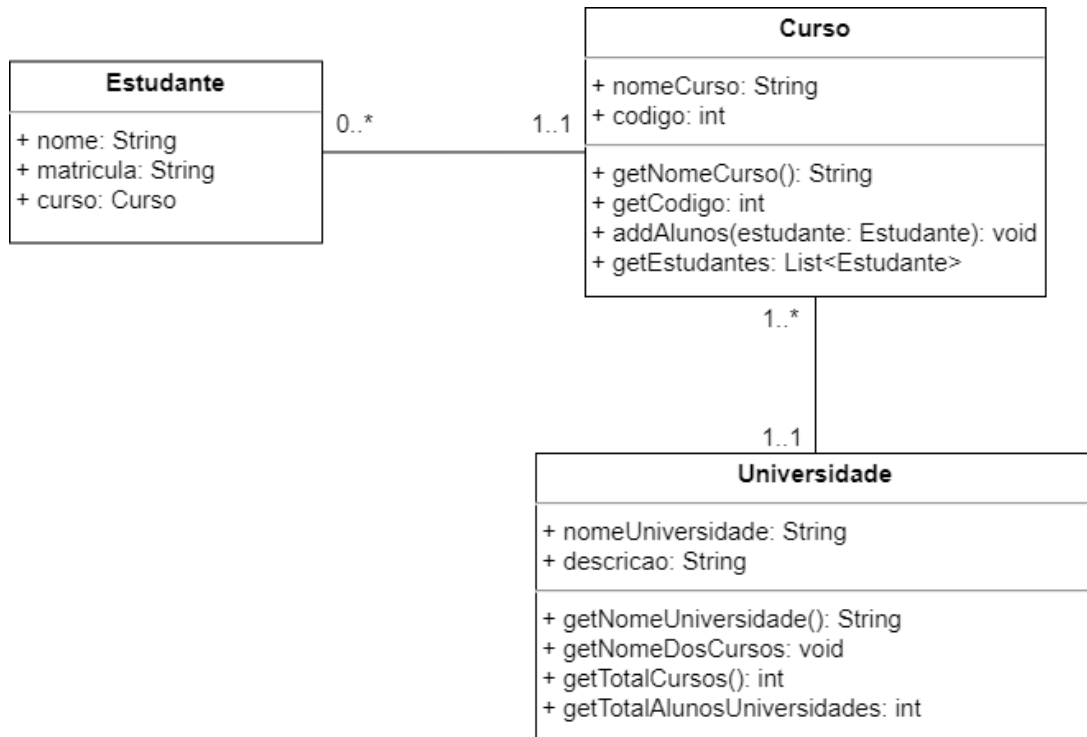


Figura 3. Diagrama UML da relação entre as classes Estudante, Curso e Universidade.

6. A Figura 4 apresenta um exemplo de relação entre as classes Cliente e Pedido.



Figura 4. Associação entre as classes Cliente e Pedido.

Verifique as afirmativas corretas de acordo com o diagrama da Figura 4 e justifique a(s) alternativa(s) errada(s).

- I. O diagrama apresenta uma associação um para muitos.
- II. O diagrama apresenta nome da associação, direção de leitura e multiplicidade.
- III. De acordo com associação do diagrama, um cliente deve fazer pelo menos um pedido.
- IV. Um pedido deve estar associado obrigatoriamente por um cliente.

- A) I, III, IV
- B) I, II, IV
- C) I, IV
- D) II, III
- E) I, III, IV

Justificativa