

Introdução à Programação Orientada a Objetos

Prof. Elder Rizzon Santos

Universidade Federal de Santa Catarina

Sistemas de Informação

## Prova 2 - 2022/2

Elabore soluções na linguagem Python para as seguintes questões (Q1: 4 pontos,, Q2.1: 3 pontos, Q2.2: 3 pontos). Utilize somente recursos da linguagem abordados em aula, em caso de dúvida sobre poder ou não utilizar algum recurso, pergunte.

1. Desenvolva uma **função** a qual recebe uma matriz quadrada (caso não seja, retorne uma lista vazia) e **retorna uma lista contendo os índices das diagonais acima da secundária que possuem todos os valores diferentes entre si**. Verifique apenas as diagonais com mais do que 1 elemento. Considere que a diagonal mais próxima da secundária tem índice 1 e os índices das demais aumentam de 1 em 1. Caso nenhuma diagonal tenha todos seus elementos diferentes entre si, retorne uma lista vazia.

### Exemplos de parâmetro e retorno:

Parâmetro	Retorno	Parâmetro	Retorno
1 1 -4 2 -1 -2 3 -2 1	[1]	8 -3 1 4 -4 2	[]
1 2 4 2 1 2 3 1 1	[]	1 2 3 4 6 1 2 4 5 7 1 2 5 7 8 0 5 7 1 9 1 2 5 7 8	[2, 3]

**2.1 Construa a classe ProdutoImportado de acordo com as especificações a seguir.** A classe ProdutoImportado é uma subclasse da classe Produto (código especificado a seguir).

Atributos: a classe ProdutoImportado, além dos atributos herdados, tem os atributos país e taxa, ambos protegidos.

a) Construtor: possui parâmetros para inicializar todos os atributos da classe. Todos os parâmetros são opcionais, sendo que os valores padrão são Japão para país (tipo texto) e 13% para taxa (tipo ponto flutuante). É obrigatório executar o construtor da superclasse (usando o *super*) dentro desse construtor para inicializar

os atributos vindos da superclasse, os demais, são inicializados conforme os argumentos recebidos.

b) Sobrescreva o método calcularCusto da superclasse de modo a retornar o custo do produto importado, o qual aplica o percentual da taxa de importação (atributo *taxa*) ao custo normal do produto (calculado com o mesmo método da superclasse). Exemplo: custo: 100, taxa: 15% retorno: 115.

## 2.2 Construa os métodos especificados nas letras a) e b) na classe ECommerce (código especificado a seguir).

a) Método custoProduto: recebe um *ean* como parâmetro e, caso o cadastro (atributo da classe) contenha esse *ean*, o método retorna o custo do respectivo produto (independente se o produto for da classe Produto ou ProdutoImportado). Caso o cadastro não contenha o *ean*, retorna falso - tipo lógico.

b) Método produtosFabricante: recebe um nome de *fabricante* como parâmetro e retorna uma lista com os todos os produtos (objetos) deste *fabricante* disponíveis no cadastro (atributo da classe). Caso o cadastro não contenha o *fabricante*, retorna uma lista vazia.

As classes Produto e ECommerce não podem ser modificadas (apenas os métodos adicionais da questão 2.2 em ECommerce).

```
class Produto:
    def __init__(self, ean='', fabricante='', custoFabrica=0):
        self._ean = ean #texto
        self._fabricante = fabricante #texto
        self.__custoFabrica = custoFabrica #ponto flutuante

    def calcularCusto(self):
        return self.__custoFabrica * 1.12

    def getEan(self): #retorna um texto
        return self._ean

    def getFabricante(self): #retorna um texto
        return self._fabricante

class ECommerce:
    def __init__(self):
        self.__cadastro = {}

    def adicionaProduto(self, prod):
        self.__cadastro[ prod.getEan() ] = prod
```