

Introdução à Programação Orientada a Objetos

Prof. Elder Rizzon Santos

Universidade Federal de Santa Catarina

Sistemas de Informação

Prova 2 - 2022/1

Elabore soluções na linguagem Python para as seguintes questões (Q1: 2 pontos, Q2: 3 pontos, Q3.1: 2 pontos, Q3.2: 3 pontos). Utilize somente recursos da linguagem abordados em aula, em caso de dúvida sobre poder ou não utilizar algum recurso, contate o professor.

1. Desenvolva uma **função** que receba um número inteiro maior do que 1 (não é necessário verificar) por parâmetro, e **retorna uma matriz** contendo uma figura conforme os exemplos abaixo. Note que o parâmetro determina a altura e largura da figura e, por consequência, as dimensões da matriz. Nas posições da matriz em que não houver o caractere '*', atribua o caractere ' ' (espaço em branco).

Exemplos de parâmetro e retorno:

Parâmetro	Retorno	Parâmetro	Retorno
2	<pre> ** **</pre>	3	<pre> *** *** ***</pre>
5	<pre> ***** ***** ***** ***** ***** *****</pre>	8	<pre> ***** ***** ***** ***** ***** ***** ***** *****</pre>

2. 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 abaixo da secundária que possuem todos os valores iguais**. 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 iguais, retorne uma lista vazia.

Exemplos de parâmetro e retorno:

Parâmetro	Retorno	Parâmetro	Retorno
<pre>1 1 -4 2 -1 -2 3 -2 1</pre>	[1]	<pre>8 -3 1 4 -4 2</pre>	[]

1 1 4 2 1 2 3 1 1	[]	1 2 3 4 -1 3 4 5 1 2 5 7 0 5 7 -1	[1, 2]
-------------------------	----	--------------------------------------------	--------

3.1 Construa a classe PlayerCS de acordo com as especificações a seguir. A classe PlayerCS é uma subclasse da classe Player (código especificado a seguir). Atributos: a classe PlayerCS, além dos atributos herdados, tem os atributos timeFavorito e armaFavorita, ambos privados.

a) Construtor: possui parâmetros para inicializar todos os atributos da classe. Todos os parâmetros são opcionais, sendo que os valores padrão são contra-terrorista para timeFavorito e rifle para armaFavorita (ambos do tipo string). 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 `__str__` da superclasse de modo a retornar as seguintes informações do objeto em uma única string: id, créditos, nome do jogo (sempre CS), time favorito e arma favorita. Exemplo e formato de retorno: 'player id: 123, créditos: 87, jogo: CS, time favorito: contra-terrorista, arma favorita: sniper'.

3.2 Construa uma classe chamada CadastroPlayers, a qual possui apenas um atributo chamado cadastro. Esse atributo deve ser privado e armazenar um dicionário.

a) Construtor: não recebe nenhuma informação e apenas cria e inicializa o atributo com um dicionário vazio.

b) Método adicionaPlayer: recebe um objeto do tipo Player ou PlayerCS (não é necessário verificar o tipo) como parâmetro e adiciona-o no atributo cadastro. Caso o objeto já esteja presente no dicionário, o método retorna False e o objeto não é inserido. Caso o objeto não esteja presente, adicione-o no cadastro utilizando o player id do objeto como chave e o próprio objeto recebido como valor. O player id é único para cada jogador.

```
class Player:
    def __init__(self, pl_id=0, cr=0):
        self.__playerID = pl_id
        self.__creditos = cr
    def __str__(self):
        return
            f'player id: {self.__playerID}, créditos: {self.__creditos}'
    def getPlayerID(self):
        return self.__playerID
```