

## **Lista de Exercícios de Python usando paradigma OO**

Nesta lista, vocês devem usar orientação a objetos. Não usem funções complexas além das básicas, nem bibliotecas externas sofisticadas, entre outras coisas.

**Exercícios (em todos os exercícios, vocês devem criar a(s) classe(s) pedida(s) em arquivos separados do arquivo que contém a main):**

- 1) Faça uma classe, em Python, que represente um ponto no plano cartesiano bidimensional. Essa classe deve armazenar os atributos relevantes e ter um método que calcula a distância entre dois pontos.
- 2) Faça uma classe, em Python, que represente um ponto no plano cartesiano tridimensional. Estenda a classe do exercício 1.
- 3) Faça uma classe abstrata que representa um Produto (no contexto de um mercado), com nome, quantidade no estoque e preço. Crie também uma classe que representa um pedido de um cliente, composto de itens (produto e quantidade). Faça um programa que cria um mercado com 3 produtos e receba (do teclado) uma lista de pedidos feitos em ordem cronológica. Garanta que haja estoque para cada pedido.
- 4) Faça um programa de agenda telefônica, com as classes Agenda e Contato.
- 5) Faça um programa para controle de empréstimo de livros, com as classes Empréstimo, Livro e Pessoa.
- 6) Faça uma classe abstrata que representa uma figura geométrica com métodos para cálculo de área e perímetro. Implemente quatro tipos de figura geométrica: quadrado, retângulo, triângulo e círculo. Faça um programa com um menu com 3 opções: cadastro, cálculo de área e cálculo de perímetro (apenas para figuras cadastradas).
- 7) Faça uma classe para representar contas bancárias. Em seguida, faça um programa em que um Usuário pode depositar, sacar e consultar saldo até escolher sair.
- 8) Implemente as seguintes classes: Aluno (nome, matrícula, método de calcular CRA – média ponderada das notas), Disciplina (nome, código), Matrícula (aluno, disciplina, nota). Faça um programa com um menu onde é possível cadastrar alunos, disciplinas, matrículas e calcular CRA de alunos cadastrados.
- 9) Implemente uma classe “Lâmpada” que pode estar em 3 estados: apagada, acesa e meia-luz. Ela deve possuir um método “trocaDeEstado”, que altera o estado na ordem “apagada → meia-luz → acesa → apagada”.
- 10) Faça uma Lâmpada com 4 estados. Os 3 do exercício anterior, além do estado “queimada”. Sugira formas de
- 11) Generalize a classe Lâmpada para que ela possa representar uma lâmpada onde a luminosidade pode ser ajustada com qualquer valor entre 0% (apagada) e 100% (acesa). Dica: em vez de operações para possibilitar o ajuste para cada um dos estados, descreva uma operação que receba um valor de ajuste

12) Faça uma classe Matriz que armazena uma matriz de qualquer tamanho. Ela deve possuir métodos de soma, subtração e multiplicação de matrizes (de duas formas: elemento a elemento e a multiplicação matricial padrão) e divisão entre elementos de matrizes.

13) Faça uma classe QuadradoMagico cujo construtor recebe uma Matriz e, caso ela não seja um quadrado mágico (Um quadrado mágico é uma matriz quadrada de valores inteiros onde a soma dos valores em cada linha, coluna ou diagonal principal é a mesma, e não existem elementos repetidos), lance uma exceção ou erro. Caso seja um quadrado mágico, o objeto deve ser criado normalmente.

14) Faça uma classe SolucaoSudoku que recebe uma matriz 9x9 e, caso ela não seja uma solução válida de sudoku, lance uma exceção ou erro. Caso seja um quadrado mágico, o objeto deve ser criado normalmente.

15) Implemente uma classe que fornece um algoritmo de ordenação qualquer.

16) Implemente um time de futebol onde cada jogador pode apresentar posições. O programa deve ter:

a) Uma classe time com 11 jogadores titulares e até 12 jogadores reservas. A classe deve ter um método que retorna a formação do time. Exemplo: um time com 4 defensores titulares, 3 meias titulares e 3 atacantes titulares possui formação “4-3-3”.

b) Uma classe Jogador que tem uma posição (goleiro, defensor, meio-campista e atacante), um número, um nome e um . Crie um construtor para inicializar esses atributos.

A classe principal do programa deve conter apenas o método principal (main). O time será cadastrado no sistema, e em seguida os jogadores, que não podem ter número repetido. Após cadastro, imprimir o esquema tático (formação) do time.

17) Faça uma classe que representa uma árvore genealógica.

18) Faça uma classe que realiza a integração numérica de qualquer função matemática utilizando a regra do trapézio. A função deve ser passada como parâmetro do construtor da classe. Ela deve possuir um método “calcular” que recebe o início e fim do intervalo, bem como o número de subdivisões a ser considerado.

19) Escreva a classe Criptografia, que conterá alguns métodos estáticos para codificação e decodificação de strings. Escreva nessa classe o método codificaRot13, que receberá uma string como argumento e retornará uma string codificada com o algoritmo “rot13”, que substitui cada caracter da string pelo valor do caracter mais treze, subtraindo vinte e seis caso o resultado seja maior que a última letra, de forma que “abCde” seja substituída por “noPqr”, “kLmnoPq” seja substituída por “xYzabCd”, e “UVWxyz” seja substituída por “HIJklm”. Somente os caracteres alfabéticos não-acentuados devem ser modificados. Por exemplo, se a string “Revolução de 1930” for passada como argumento para esse método, ele retornará “Eribyhçãb qr 1930”. Uma característica interessante do algoritmo “rot13” é que, se uma string codificada por ele for passada de novo pelo próprio algoritmo, a string original será retornada. Escreva também um método decodificaRot13 que seja somente uma chamada para o método codificaRot13.

20) Implemente um autômato finito que reconhece a expressão regular “(a|b)\*abb”. Proponha as classes necessárias para isso.