

Matheus Ávila Moreira de Paula
Programação Orientada a Objetos
Lista 1 - Classes e Coleções

1. **Crie uma classe** `Pessoa` **com os atributos** `nome`, `idade` **e** `sexo`. Implemente métodos para:
 - a. Obter(get) e definir(set) cada atributo.
 - b. Apresentar os dados da pessoa.
 - c. Garanta que cada Pessoa só possa ter como sexo uma letra maiúscula.
2. **Crie uma classe** `Livro` **com os atributos** `titulo`, `autor`, `editora`, `status`(Boolean, int, char ou String) **e** `anoDePublicacao`. Implemente métodos para:
 - a. Obter(get) e definir(set) cada atributo.
 - b. Emprestar o livro (alterar o atributo para indicar que o livro está emprestado).
 - c. Devolver o livro (alterar o atributo para indicar que o livro está disponível).
 - d. Pesquisar uma palavra no título do livro.
3. **Crie uma classe** `Data` **com os atributos** `dia`, `mes`, `ano`. Implemente métodos para:
 - a. Verificar se é uma data válida. (valores positivos e respeitando número de dias no respectivo mês).
 - b. Verificar se a data do objeto que chamou o método vem antes ou depois da data passada como parâmetro.
 - c. Calcular a diferença entre a data que chamou o método e a data passada como parâmetro.
 - d. Imprimir a data no formato: DD/MM/AAAA.
4. **Crie uma classe** `Carro` **com os atributos** `marca`, `modelo`, `anoDeFabricacao` **e** `placa`. Implemente métodos para:
 - a. Obter(get) e definir(set) cada atributo.
 - b. Colocar a placa com letra maiúscula.
 - c. Verificar se a placa é válida. (3 letras seguidas de "-" e 4 números).

5. **Crie uma classe** `Cliente` **com os atributos** `nome`, `cpf`, `endereco` e `telefones` **(uma lista de strings)**. Implemente métodos para:
 - a. Obter e definir cada atributo.
 - b. Receber o CPF como um long int e tratar para coloca-lo como uma String no formato: "999.999.999.99".
6. **Crie uma classe** `Produto` **com os atributos** `nome`, `preco`, `descricao` e `estoque`. Coloque todos os produtos instanciados em um map usando um código int incremental como chave. Ou seja: o primeiro produto terá `codigo = 0`, o segundo terá `codigo = 1`, e assim por diante. Implemente métodos para:
 - a. Obter(get) e definir(set) cada atributo.
 - b. Vender um produto (diminuir o estoque).
 - c. Calcular o valor total de determinado produto em estoque.
 - d. Calcular o valor total em estoque de todos os produtos.
7. **Crie uma classe** `Funcionario` **com os atributos** `nome`, `salario`, `cargo` e `dataDeAdmissao`. Insira os funcionários em um HashMap usando o `cpf` (String apenas com os números) como chave.
 - a. Implemente método para obter(get) e definir(set) cada atributo.
 - b. Faça uma operação de busca por um CPF no map.
 - c. Liste todos os funcionários no main.
8. Crie um programa que recebe uma frase como entrada e conta quantas vezes **cada palavra** aparece. Utilize um HashMap para armazenar as palavras e suas contagens.
9. Escreva um programa que conta quantas vezes um elemento específico aparece em um ArrayList do tipo String.
10. Escreva um programa que conta quantos dígitos um número inteiro possui.
11. Escreva um programa que calcula a média de um conjunto de notas (double) armazenadas em um ArrayList.
12. Escreva um programa que verifica se duas palavras são anagramas (contêm as mesmas letras, em ordens diferentes).
13. Escreva um programa que conta a frequência de **cada caractere** em uma string.

14. Escreva um programa que recebe uma **lista de Strings** e remove todas as duplicatas.
15. Escreva um programa que conta quantas vezes cada **caractere** aparece em uma string e armazena os resultados em um HashMap.
16. Escreva um programa que recebe uma frase (String) como entrada e conta quantas vogais e quantas consoantes ela possui.
17. Escreva um programa que verifica se uma String é um palíndromo (lê-se igual de trás para frente).
18. Escreva um programa que mistura duas listas de strings, intercalando elementos de cada lista.
19. Escreva um programa que recebe uma frase (String) como entrada e inverte a ordem das palavras.