

Exercício Aplicação do POO

Questão 1. Crie uma classe chamada *Document*, que deve possuir as propriedades *title*, *id* (um número único gerado automaticamente, começando em 1 e incrementando a cada novo documento) e *content*. Em seguida, crie uma classe *Folder*, que permite adicionar novos *documents*, remover *documents* com base no *id* e ordenar os *documents* por título em ordem alfabética. ou seja, a classe *Folder* deve gerenciar os *documents*, oferecendo as funcionalidades de adição, remoção e ordenação conforme especificado.

Dica: Um campo estático pode ser utilizado para controlar o **id** do próximo documento a ser criado, garantindo que ele seja gerado automaticamente e de forma incremental; E para comparar ignorando pontuação e maiúsculo e minúsculo use [localeCompare](#);

Questão 2. Implemente uma classe chamada **Temperature** que permita converter temperaturas entre as unidades Celsius, Fahrenheit e Kelvin. A classe ao ser inicializada deverá ser informado o valor da temperatura e a unidade (Celsius, Fahrenheit ou Kelvin).

- **toFahrenheit()**: retornará a temperatura em Fahrenheit.
- **toKelvin()**: retornará a temperatura em Kelvin.
- **toCelsius()**: retornará a temperatura em Celsius.
- **compare(otherTemperature)**: Compara a temperatura atual com outra temperatura fornecida, independentemente da unidade em que a outra temperatura esteja expressa. O método retorna: 1 se a temperatura atual for maior que a fornecida; 0 se ambas as temperaturas forem iguais; -1 se a temperatura atual for menor que a fornecida.

Segue a relação de conversão:

$$\frac{T_C}{5} = \frac{T_F - 32}{9} = \frac{T_K - 273}{5}$$

Onde

T_C = Temperatura em graus Celsius;

T_F = Temperatura em graus Fahrenheit;

T_K = Temperatura em Kelvin (não possui a denominação/notação grau).

Questão 3. Uma **tupla** é uma sequência ordenada de elementos, onde a ordem dos elementos é importante e pode haver repetição de valores. Uma tupla é frequentemente representada como um conjunto de elementos entre parênteses, como (a_1, a_2, \dots, a_n) , onde a_1, a_2, \dots, a_n são os elementos da tupla e n é o número de elementos (chamado de aridade ou dimensão da tupla). Em uma tupla, os elementos podem ser de diferentes tipos ou de tipos iguais, e, ao contrário de um conjunto, os elementos de uma tupla têm uma ordem definida, o que significa que $(a_1, a_2) \neq (a_2, a_1)$, a menos que os elementos sejam idênticos. Uma dupla é uma tupla com **aridade 2**, ou seja, uma tupla que contém somente 2 elementos.

Crie uma classe chamada **Duple** que, ao ser inicializada, receberá dois elementos de tipo primitivo. Esses elementos devem ser **imutáveis**, ou seja, não poderão ser

alterados após a criação da instância. Além disso, a classe deve conter um método para **comparar** a instância atual com outra dupla, verificando se ambos os elementos são iguais ou não. lembre-se: $(a_1, a_2) \neq (a_2, a_1)$

Questão 4. Um cache com **TTL** (Time to Live) é uma técnica de armazenamento temporário de dados, onde os dados são mantidos por um período específico de tempo antes de expirarem. O TTL define o tempo em que os dados podem ser armazenados no cache, após o qual eles são automaticamente removidos ou marcados como obsoletos.

Crie uma classe chamada **Cache** que receberá o TTL em milissegundos ao ser inicializada, a classe armazenará o cache em uma estrutura de chave-valor e conterá os seguintes métodos:

- **set(key, value)**: adiciona uma chave e um valor ao cache, iniciando a contagem do TTL (Time to Live).
- **get(key)**: retorna o valor associado a uma chave. Se o valor não for encontrado, retorna **undefined**. Se o TTL tiver expirado, o valor será removido e o método retornará **undefined** (lazy strategy). Se o valor existir e o TTL não tiver expirado, o valor será retornado.
- **remove(key)**: remove um valor a partir de sua chave.
- **removeExpired()**: remove todos os valores cujo TTL tenha expirado.

Obs: documente a classe com JsDoc (pode usar o chatGPT para isso).

Dica: Use [Map - Javascript](#); e use `Date.now()` para pegar o tempo atual em milissegundos