

Exercícios Classes:

01 – Complete a classe Carro para que os atributos da mesma sejam acessados somente por métodos (Getters/ Setters) e tenha um método para calcular a idade do carro com base no ano atual. Crie uma instância de Carro

02 – Desenvolva uma classe Calculadora em TypeScript

Crie uma classe Calculadora que encapsula funcionalidades matemáticas básicas. Esta classe deve possuir os seguintes requisitos:

- Dois atributos numéricos (valor1 e valor2) para armazenar os operandos das operações.
- Um construtor que aceite dois números como parâmetros e os atribua aos respectivos atributos.
- Métodos para realizar as seguintes operações matemáticas:
 - Soma: retorna a adição dos dois valores.
 - Subtração: retorna a diferença entre os dois valores.
 - Multiplicação: retorna o resultado da multiplicação dos dois valores.
 - Divisão: retorna o resultado da divisão do primeiro valor pelo segundo valor. Atenção: evite divisões por zero.
 - Porcentagem: retorna a porcentagem do primeiro valor em relação ao segundo valor.
 - Getters e Setters

Certifique-se de que a classe lida corretamente com situações de erro, como a divisão por zero.

Teste todos os métodos.

03 – Desenvolva uma classe Produto em TypeScript

Crie uma classe Produto que represente um item disponível em um catálogo de uma loja virtual. A classe deve ter as seguintes características:

Atributos:

- nome: uma string que representa o nome do produto.
- preco: um número que representa o preço do produto.
- quantidadeEmEstoque: um número inteiro que indica a quantidade disponível em estoque do produto.

Construtor:

- Deve aceitar três parâmetros: nome, preço e quantidadeEmEstoque, e atribuí-los aos respectivos atributos da classe.

Métodos:

- `calcularValorTotalEmEstoque()`: retorna o valor total em estoque do produto, calculado multiplicando o preço pelo quantidade em estoque.
- `reporEstoque(quantidade: number)`: adiciona a quantidade especificada ao estoque do produto.
- `vender(quantidade: number)`: remove a quantidade especificada do estoque do produto. Se a quantidade desejada for maior do que a disponível em estoque, deve ser lançado um erro indicando que não há estoque suficiente.
- Getters e Setters

Certifique-se de que a classe lida corretamente com situações de erro, como tentar vender mais produtos do que há em estoque. **Teste todos os métodos.**