

Exercício: Criando uma Classe Personalizada em JavaScript

Objetivo: Cada aluno deve criar uma classe separada em JavaScript que represente uma entidade do mundo real (como Produto, Usuário, Carro, Livro, etc.). A classe deve incluir atributos, métodos e demonstrar o uso de conceitos como construtores, getters/setters (opcional) e pelo menos um método personalizado.

Instruções

1. **Escolha da Entidade:** Escolha uma entidade do mundo real para representar com a classe (ex.: Produto, Usuário, Carro, Livro, Animal, etc.).
2. **Requisitos da Classe:**
 - Deve ter pelo menos **3 atributos** (propriedades) que descrevam a entidade.
 - Deve incluir um **construtor** para inicializar os atributos.
 - Deve ter pelo menos **2 métodos**:
 - Um método para exibir informações da entidade (ex.: `mostrarDetalhes()`).
 - Um método personalizado que faça algo relevante para a entidade (ex.: calcular preço com desconto, atualizar status, etc.).
 - (Opcional) Incluir **Getters** e/ou **Setters** para controlar o acesso aos atributos.
3. **Implementação:** Escreva o código em JavaScript, usando a sintaxe de classe (`class`). Teste a classe criando pelo menos **2 instâncias** e chamando seus métodos.
4. **Entrega:** O aluno deve:
 - Escrever o código em um arquivo `.js`.
 - Incluir comentários explicando o propósito da classe e dos métodos.
 - Mostrar no console (usando `console.log`) os resultados das instâncias e métodos.

Exemplo de Classe

Segue um exemplo de uma classe que representa um **Livro**:

```
1 // Classe que representa um Livro
2 class Livro {
3   // Construtor para inicializar os atributos
4   constructor(titulo, autor, preco) {
5     this.titulo = titulo;
6     this.autor = autor;
```

```

7     this.preco = preco;
8 }
9
10 // Método para exibir detalhes do livro
11 mostrarDetalhes() {
12     return `Livro: ${this.titulo}, Autor: ${this.autor}, Preço:
13         R${this.preco.toFixed(2)}`;
14 }
15
16 // Método personalizado: aplica desconto no preço
17 aplicarDesconto(percentual) {
18     if (percentual > 0 && percentual <= 50) {
19         const desconto = this.preco * (percentual / 100);
20         this.preco -= desconto;
21         return `Desconto de ${percentual}% aplicado! Novo preço:
22             R${this.preco.toFixed(2)}`;
23     } else {
24         return "Percentual de desconto inválido! Deve ser entre 1% e
25             50%.";
26     }
27 }
28
29 // Testando a classe
30 const livro1 = new Livro("O Senhor dos Anéis", "J.R.R. Tolkien",
31     59.90);
32 const livro2 = new Livro("Dom Casmurro", "Machado de Assis", 29.90);
33
34 console.log(livro1.mostrarDetalhes()); // Livro: O Senhor dos
35     Anéis, Autor: J.R.R. Tolkien, Preço: R$59.90
36 console.log(livro1.aplicarDesconto(10)); // Desconto de 10%
37     aplicado! Novo preço: R$53.91
38 console.log(livro2.mostrarDetalhes()); // Livro: Dom Casmurro,
39     Autor: Machado de Assis, Preço: R$29.90
40 console.log(livro2.aplicarDesconto(60)); // Percentual de desconto
41     inválido! Deve ser entre 1% e 50%.

```

Sugestões de Entidades

- **Produto:** Atributos (nome, preço, quantidade em estoque); Métodos (calcular valor total, repor estoque).
- **Usuário:** Atributos (nome, email, idade); Métodos (exibir perfil, atualizar email).
- **Carro:** Atributos (marca, modelo, ano); Métodos (mostrar detalhes, calcular idade do carro).
- **Animal:** Atributos (espécie, nome, idade); Métodos (emitir som, mostrar informações).

Dicas para os Alunos

- Use nomes claros e descritivos para atributos e métodos.
- Teste os métodos com diferentes valores para garantir que funcionem corretamente.
- Adicione validações (como no exemplo do desconto) para tornar a classe mais robusta.
- (Avançado) Considere usar Getters e Setters para proteger os atributos, como no exemplo abaixo:

```
1 class Livro {  
2     #preco; // Atributo privado  
3     constructor(titulo, autor, preco) {  
4         this.titulo = titulo;  
5         this.autor = autor;  
6         this.#preco = preco;  
7     }  
8     get preco() {  
9         return this.#preco;  
10    }  
11    set preco(novoPreco) {  
12        if (novoPreco > 0) this.#preco = novoPreco;  
13    }  
14 }
```

CrITÉrios de Avaliação

- A classe tem os atributos e métodos solicitados?
- O código está comentado e bem organizado?
- As instâncias e métodos foram testados e exibem resultados corretos no console?
- (Opcional) O aluno usou Getters/Setters ou outras funcionalidades avançadas?