

Programação Orientada a Objetos (POO) - Aula 16

Professor: Jerffeson Gomes Dutra

Métodos e Atributos estáticos

Os atributos e métodos estáticos (`static`) são associados à própria classe, e não a instâncias específicas da classe. Isso significa que você pode acessar atributos e métodos estáticos diretamente através da classe, sem precisar criar uma instância dela.

Exemplo

```
class Class {  
    static classesDisponiveis: string[] = [];  
  
    static adicionarClass(nome: string): void {  
        this.classesDisponiveis.push(nome);  
    }  
  
    static listarClass() {  
        this.classesDisponiveis.forEach(classe => console.log(classe))  
    }  
}  
  
Class.adicionarClass("Guerreiro");  
Class.adicionarClass("Mago");  
  
Class.listarClass()
```

Como acessar?

```
class Class {  
    static classesDisponiveis: string[] = [];  
  
    static adicionarClass(nome: string): void {  
        this.classesDisponiveis.push(nome);  
    }  
}
```

```
Class.classesDisponiveis;
```

```
Class.adicionarClass("Guerreiro");
```

```
class Guerreiro {  
    nome: string;  
  
}
```

```
let g = new Guerreiro();  
g.nome;
```

Quando usar um atributo estático?

É usados quando você precisa de uma propriedade ou variável que pertence à própria classe em vez de a uma instância específica da classe.

Isso significa que o valor do atributo é compartilhado entre todas as instâncias da classe.

Exercício 01

- Crie uma classe chamada Pessoa com o atributo CPF;
- Crie uma classe chamada PessoaUtils com um método chamado validarCPF que recebe como parâmetro o cpf da pessoa;
- Retornar a mensagem de CPF válido ou inválido;

Exercício 02

- Crie uma classe Setup que tem os atributos resolucao, dificuldade, raytracing.
- Esses atributos devem ser estáticos;
- O usuário deve poder informar esses atributos;
- Permitir que o usuário liste essas informações;

Opções disponíveis:

- resolucao: HD, FullHD, 2k, 4k;
- dificuldade: Fácil, Normal e Difícil;
- raytracing: true ou false

Exercício 03 - Database de monstros

- Criar uma classe chamada Monstro com os atributos nome, força, defesa e raridade;
 - Ao criar um monstro, se não existir o monstro na Database, adicionar o monstro.
- Criar a classe MonstoDatabase com um atributos estático listaMonstros;
 - Criar um método chamado exibir database que exibe o nome dos monstros;
 - Criar o método que retornar o nome e a raridade do monstro selecionada;

Exercício 04 - Classes de Personagem

- Criar uma classe chamada Personagem;
 - os personagens tem os atributos com valores base: nome, hp(100), mp(70), forca(60), defesa(40) e agilidade(25);
- Nessa classe existe 3 atributos estáticos, uma para cada tipo de personagem que é possível criar no jogo;
- Permitir que o usuário selecione o tipo do Personagem que ele vai criar;
- Permitir que o usuário informe o nome do personagem;
- Se o personagem for:
 - Mago ganha um buff de: +30 de mp
 - Guerreira ganha um buff: +30 de hp e +15 de força
 - Assassino ganha um buff: +15 de agilidade
 - Arqueiro: +5 agilidade, +7 de forca e +5 de hp