
■ TEMA 10: Introdução à Programação Orientada a Objetos (POO) em PHP

1. O que é POO?

A **Programação Orientada a Objetos** organiza o código em torno de **objetos**, que representam coisas do mundo real.

Um **objeto** é formado por:

- **Atributos** → características (ex: marca, modelo, ano de um carro).
- **Métodos** → ações ou comportamentos (ex: exibir informações).

2. Nosso exemplo: um carro 🚗

Código completo:

```
<?php
// Definindo a classe Carro
class Carro {
    public $marca;
    public $modelo;
    public $ano;

    // Construtor para inicializar o objeto
    public function __construct($marca, $modelo, $ano) {
        $this->marca = $marca;
        $this->modelo = $modelo;
        $this->ano = $ano;
    }

    // Método para exibir informações
    public function exibir() {
        echo "Carro: {$this->marca} {$this->modelo} ({$this->ano})<br>";
    }
}

// Criando objetos (instâncias)
$carro1 = new Carro("Fiat", "Uno", 2010);
$carro2 = new Carro("Toyota", "Corolla", 2020);

// Usando métodos
$carro1->exibir();
$carro2->exibir();
?>
```

3. Entendendo passo a passo

◆ Classe Carro

```
class Carro {  
    public $marca;  
    public $modelo;  
    public $ano;
```

- class Carro → define o **molde** para criar carros.
 - public \$marca; \$modelo; \$ano; → são os **atributos** (características do carro).
-

◆ Construtor

```
public function __construct($marca, $modelo, $ano) {  
    $this->marca = $marca;  
    $this->modelo = $modelo;  
    $this->ano = $ano;  
}
```

- O **construtor** é executado automaticamente quando criamos um objeto.
 - \$this->atributo → representa o atributo **do objeto atual**.
 - Exemplo: quando criamos um carro "Fiat", "Uno", 2010, o construtor guarda esses valores dentro do objeto.
-

◆ Método exibir()

```
public function exibir() {  
    echo "Carro: {$this->marca} {$this->modelo} ({$this->ano})<br>";  
}
```

- Um **método** é uma **função dentro da classe**.
 - O método exibir() imprime as informações do carro.
-

◆ Criando objetos

```
$carro1 = new Carro("Fiat", "Uno", 2010);  
$carro2 = new Carro("Toyota", "Corolla", 2020);
```

- new Carro(...) cria um **objeto da classe Carro**.
 - Cada objeto tem seus próprios valores:
 - \$carro1 → Fiat Uno, 2010.
 - \$carro2 → Toyota Corolla, 2020.
-

◆ Usando métodos

```
$carro1->exibir();  
$carro2->exibir();
```

- O operador -> acessa métodos e atributos de um objeto.
 - Aqui, chamamos o método `exibir()` para mostrar os dados de cada carro.
-

4. Saída esperada

Carro: Fiat Uno (2010)

Carro: Toyota Corolla (2020)

5. O que aprendemos aqui?

- ✓ Como criar uma **classe**.
 - ✓ Como definir **atributos** (dados do objeto).
 - ✓ Como usar o **construtor** para inicializar objetos.
 - ✓ Como criar e usar **métodos**.
 - ✓ Como **instanciar objetos** com `new`.
 - ✓ Como exibir informações de forma organizada.
-

6. Exercícios sugeridos 💡

1. Adicione um terceiro carro (VW Gol, 2015) e exiba.

2. Classe **Torcedor** em PHP:

Crie uma classe chamada **Torcedor** que represente um torcedor de futebol.

1. Atributos

A classe deve ter três atributos públicos:

- **nome** → nome do torcedor (string).
- **time** → time para o qual ele torce (string).
- **idade** → idade do torcedor (inteiro).

2. Construtor

Implemente um **construtor** (`__construct`) que receba os três valores acima e atribua aos atributos do objeto.

3. Método `exibir()`

Crie um **método público chamado `exibir()`** que mostre uma frase com as informações do torcedor.

Exemplo de saída esperada:

Torcedor: Ana, 25 anos, torce para o Flamengo.

4. Instanciação de objetos

- Crie **3 objetos** da classe Torcedor, cada um com valores diferentes.
- Use o método `exibir()` para mostrar os dados dos três torcedores.

👉 **Dica** : Lembre-se de que, para acessar um método de um objeto em PHP, usamos o operador `->`.