

Programação Orientada a Objetos em PHP

A Programação Orientada a Objetos (POO) tem como pilar teórico o conceito de **objeto**. Um sistema orientado a objeto é um conjunto de objetos que representam seres e coisas do mundo real, interagindo computacionalmente com as mesmas características e comportamentos reais.

A estrutura computacional que define o modelo de um objeto do mundo real é chamada de **classe**. A partir dela, podemos criar múltiplas **instâncias** (objetos reais no sistema). As características dos objetos são chamadas de **atributos** e seus comportamentos são chamados de **métodos**.

Exemplo de modelagem

Um carro no mundo real tem propriedades como: Cor, Ano, Número de portas e Tipo de combustível. E comportamentos como: Acelerar, Frear, Virar à esquerda. No PHP, podemos representar isso criando uma classe Carro, que terá atributos e métodos correspondentes.

Diferença entre Classe e Objeto

- Classe: é o molde ou modelo (como a planta de uma casa).
- Objeto: é a instância real criada a partir da classe.

O uso de classes permite **reutilização de código** e facilita a manutenção.

Métodos

Em PHP:

- Um método pode ou não retornar valores.
- Pode ou não receber parâmetros.
- Pode ser estático (pertence à classe e não a um objeto específico).

Exemplo prático: Classe em PHP

```
<?php

class RegistraAluno {
    // Atributos privados
    private string $nome;
    private string $endereco;
    private int $idade;
    private float $notaMatematica;
    private float $notaPortugues;
    private float $notaCiencias;

    // Atributo de classe (estático)
    private static int $contadorEstudante = 0;

    // Construtor
    public function __construct() {
        self::$contadorEstudante++;
    }

    // Métodos GET
    public function getNome(): string {
        return $this->nome;
    }
    public function getEndereco(): string {
```

```

        return $this->endereco;
    }

    public function getIdade(): int {
        return $this->idade;
    }

    public function getMedia(): float {
        return ($this->notaMatematica + $this->notaPortugues + $this->notaCiencias) / 3;
    }

    public static function getQuantidadeAlunos(): int {
        return self::$contadorEstudante;
    }

    // Métodos SET
    public function setNome(string $nome): void {
        $this->nome = $nome;
    }

    public function setEndereco(string $endereco): void {
        $this->endereco = $endereco;
    }

    public function setIdade(int $idade): void {
        $this->idade = $idade;
    }

    public function setNotaMatematica(float $nota): void {
        $this->notaMatematica = $nota;
    }

    public function setNotaPortugues(float $nota): void {
        $this->notaPortugues = $nota;
    }

    public function setNotaCiencias(float $nota): void {
        $this->notaCiencias = $nota;
    }
}
?>

```

Criando e utilizando objetos

```

<?php

require_once 'RegistraAluno.php';

// Criar objetos
$ana = new RegistraAluno();
$beto = new RegistraAluno();
$carlos = new RegistraAluno();

// Definir valores
$ana->setNome("Ana Machado");
$beto->setNome("Roberto da Silva");
$carlos->setNome("Carlos Alberto");

// Exibir nome da Ana
echo $ana->getNome() . PHP_EOL;

// Exibir quantidade de alunos
echo "Contador: " . RegistraAluno::getQuantidadeAlunos() . PHP_EOL;

```

?>

Conceitos aplicados

- Encapsulamento: atributos são privados e acessados apenas por métodos get e set.
- Atributos de classe (static): contadorEstudante é compartilhado entre todas as instâncias.
- Métodos estáticos: getQuantidadeAlunos() pode ser chamado sem criar um objeto.