

UNIVERSIDADE DO OESTE DE SANTA CATARINA -
UNOESC

Curso de Ciência da Computação

Programação II

Fundamentos da Programação Orientada a Objetos em PHP

Guia Prático com Exemplos e Atividades

Autor: Leandro Otavio Cordova Vieira

Videira - SC
11 de agosto de 2025

Capítulo 1

Fundamentos da Programação Orientada a Objetos em PHP

1.1 Introdução

Este capítulo apresenta o conteúdo abordado na aula do dia 11/08/2025, cujo tema central foi a **Definição de Classes e Objetos** no contexto da Programação Orientada a Objetos (POO) com PHP. O objetivo principal é fornecer uma compreensão sólida sobre como criar classes, instanciar objetos e utilizar métodos e atributos, preparando a base para o aprofundamento em conceitos mais avançados.

1.2 Objetivos de Aprendizagem

Ao final deste capítulo, o leitor deverá ser capaz de:

- Entender o conceito de **classe** como molde para criação de objetos.
- Compreender o que é um **objeto** como instância de uma classe.
- Conhecer a sintaxe básica para criação de classes e objetos em PHP.
- Implementar e executar exemplos simples utilizando POO.

1.3 Revisão de Conceitos

A Programação Orientada a Objetos organiza o código em torno de **objetos**, que são estruturas que combinam *dados* (atributos) e *comportamentos* (métodos). No PHP moderno (PHP 7 e PHP 8), a POO é plenamente suportada, oferecendo recursos robustos para encapsulamento, herança, polimorfismo, interfaces e traits.

1.4 Definição de Classe e Objeto

- **Classe:** Modelo que define os atributos e métodos que os objetos criados a partir dela terão.
- **Objeto:** Instância concreta de uma classe.

1.4.1 Sintaxe Básica de Classe em PHP

```

1 class NomeDaClasse {
2     // atributos
3     // m todos
4 }

```

1.5 Exemplos Práticos

1.5.1 Classe Pessoa – Estrutura Básica

```

1 <?php
2 class Pessoa {
3     public $nome;
4     public $idade;
5 }

```

1.5.2 Instanciando e Atribuindo Valores

```

1 <?php
2 require_once "Pessoa.php";
3
4 $p1 = new Pessoa();
5 $p1->nome = "Maria";
6 $p1->idade = 25;
7
8 $p2 = new Pessoa();
9 $p2->nome = "João";
10 $p2->idade = 30;
11
12 echo "Pessoa 1: $p1->nome, $p1->idade anos\n";
13 echo "Pessoa 2: $p2->nome, $p2->idade anos\n";

```

1.5.3 Adicionando um Método à Classe

```

1 <?php
2 class Pessoa {
3     public $nome;
4     public $idade;
5
6     function apresentar() {
7         echo "Olá, meu nome é $this->nome e tenho $this->idade
8             anos.\n";
9     }
10 }

```

1.5.4 Usando o Método

```
1 <?php
2 require_once "Pessoa.php";
3
4 $p = new Pessoa();
5 $p->nome = "Carlos";
6 $p->idade = 28;
7
8 $p->apresentar();
```

1.5.5 Classe Produto – Estrutura e Método

```
1 <?php
2 class Produto {
3     public $nome;
4     public $preco;
5     public $quantidade;
6
7     function valorTotal() {
8         return $this->preco * $this->quantidade;
9     }
10 }
```

1.5.6 Instanciando Produto

```
1 <?php
2 require_once "Produto.php";
3
4 $prod = new Produto();
5 $prod->nome = "Caneta";
6 $prod->preco = 2.5;
7 $prod->quantidade = 10;
8
9 echo "Produto: $prod->nome\n";
10 echo "Valor total em estoque: R$ " . $prod->valorTotal();
```

1.6 Atividades Propostas

1.6.1 Classe Carro

```
1 <?php
2 class Carro {
3     public $marca;
4     public $modelo;
5     public $ano;
6 }
```

```

7     function exibirInfo() {
8         echo "Carro: $this->marca $this->modelo ($this->ano)\n";
9     }
10 }
11
12 $c = new Carro();
13 $c->marca = "Toyota";
14 $c->modelo = "Corolla";
15 $c->ano = 2022;
16 $c->exibirInfo();

```

1.6.2 Classe Aluno

```

1 <?php
2 class Aluno {
3     public $nome;
4     public $media;
5
6     function verificarAprovacao() {
7         if ($this->media >= 7) {
8             return "$this->nome est aprovado.";
9         } else {
10            return "$this->nome est reprovado.";
11        }
12    }
13 }
14
15 $a = new Aluno();
16 $a->nome = "Fernanda";
17 $a->media = 8.5;
18 echo $a->verificarAprovacao();

```

1.7 Resumo

Neste capítulo, foram explorados os fundamentos da criação de classes e objetos em PHP, incluindo a definição de atributos, métodos, instanciação e utilização de objetos. Os exemplos práticos apresentados formam a base para a compreensão dos conceitos mais avançados que serão abordados nas próximas seções do curso.

1.8 Próximos Passos

O próximo tópico a ser estudado será **Métodos e Atributos em Profundidade**, abordando boas práticas, visibilidade (public, private, protected) e uso de construtores.