

Desenvolvimento para Servidores 1

Prof. Orlando Saraiva Júnior
orlando.nascimento@fatec.sp.gov.br

PHP

Orientação a Objetos

A relação entre os objetos

Relacionamento entre objetos

- Associação
- Composição
- Agregação
- Herança

Associação é a relação mais comum entre objetos. Na associação, um objeto faz uma referência a outro objeto.

Ver código
`associacao.php`

A composição é uma relação entre objetos de duas classes conhecidas como relação todo/parte. O relacionamento tem esse nome porque conceitualmente um objeto (todo) contém outros objetos (parte). A composição permite combinar diferentes tipos de objetos em um objeto mais complexo.

Ver código
`composicao.php`

Agregação também é um tipo de relação entre objetos todo/parte. Na agregação, um objeto agrega outro objeto, ou seja, torna um objeto externo parte de si mesmo pela utilização de um dos seus métodos.

Ver código
[agregacao.php](#)

Quando uma classe herda características de uma classe pai.

Um dos maiores benefícios que encontramos na utilização deste paradigma é o reúso. A possibilidade de reutilizar partes de códigos já definidas é o que nos dá mais agilidade, além de eliminar a necessidade de eventuais duplicações ou reescrita de códigos.

Ver código
Conta.php
ContaCorrente.php
ContaPoupanca.php

Formas de definir a visibilidade das propriedades e dos métodos de um objeto.

`public` → Poderão ser acessados livremente

`private` → Poderão ser acessados dentro da classe

`Protected` → somente podem ser acessados dentro da própria classe e a partir de classes descendentes.

Ver código
private1.php
private2.php
protected.php
public.php

PHP

Orientação a Objetos
Tópicos complementares

É o princípio que permite que classes derivadas de uma mesma superclasse tenham métodos iguais (com a mesma nomenclatura e os mesmos parâmetros), mas comportamentos diferentes, redefinidos em cada uma das classes filhas.

Ver código
`poli.php`

São classes que nunca serão instanciadas na forma de objetos, somente suas filhas serão.

Ver código
`classe_abstrata.php`

Uma classe final é uma classe que não pode ser superclasse, ou seja, não pode ser base para construção de outra classe em uma estrutura de herança.

Ver código
`classe_final.php`

É o princípio que permite que classes derivadas de uma mesma superclasse tenham métodos iguais (com a mesma nomenclatura e os mesmos parâmetros), mas comportamentos diferentes, redefinidos em cada uma das classes filhas.

**Ver código
poli.php**

Um método abstrato consiste na definição de uma assinatura de método, ou seja, na definição de seu nome e de seus parâmetros, não de sua implementação.

Observe o método retirar() na classe Conta.

Ver código
conta.php
metodo_abstrato.php

Há situações em que escrevemos determinados métodos, mas não queremos que eles sejam sobrescritos em classes filhas. Sempre que quisermos que um método seja a implementação definitiva e não seja mais especializado em classes filhas, devemos marcá-lo como um método final.

Observe o método `retirar()` na classe `ContaCorrenteEspecial`.

Ver código
conta.php
metodo_final.php

Atributos estáticos são atributos pertencentes a uma classe, não a um objeto específico. São dinâmicos como os atributos de um objeto, mas estão relacionados à classe.

Ver código
`propriedade_estatica.php`
`metodo_estatico.php`

Para manipular atributos estáticos, podemos usar métodos estáticos. Métodos estáticos podem inclusive ser executados diretamente a partir da classe sem a necessidade de criar um objeto para isso.

Ver código
`propriedade_estatica.php`
`metodo_estatico.php`

Dúvidas

Prof. Orlando Saraiva Júnior
orlando.nascimento@fatec.sp.gov.br

PHP

Desafio anterior

Crie uma classe Pessoa com atributos e métodos que acreditar ser importante.

Crie cinco objetos do tipo Pessoa.

Crie uma classe SalaVirtual, com atributos e métodos que acreditar ser importante.

Crie dois objetos do tipo SalaVirtual.

PHP Desafio

Baseado no desafio anterior, crie uma relação entre as classes **Pessoa** e **SalaVirtual**.

Crie e adapte os métodos e atributos criados na aula anterior.