

Orientação a Objetos com PHP

Wendell Bento Geraldes

Instituto Federal de Goiás campus Luziânia

17 de junho de 2019

Sumário

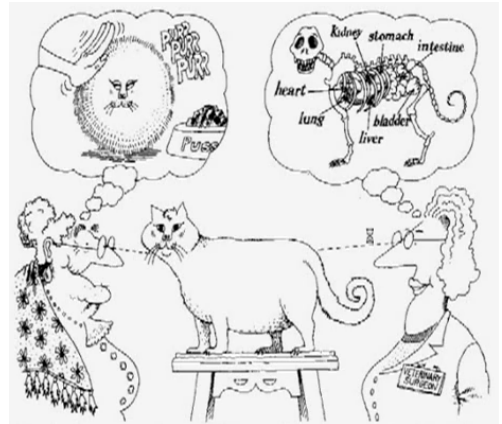
- 1 Abstração
- 2 Classes abstratas
- 3 Classes finais
- 4 Métodos abstratos
- 5 Métodos finais
- 6 Referências bibliográficas

Orientação a Objetos com PHP

Abstração

No paradigma de orientação a objetos se prega o conceito da abstração. Abstrair é separar mentalmente, considerar isoladamente, simplificar. Para construir um sistema orientado a objetos, não devemos projetar o sistema como sendo uma grande peça monolítica; devemos separá-lo em partes, concentrando-nos nas peças mais importantes e ignorando os detalhes (em primeiro momento), para podermos construir peças bem definidas que possam ser reaproveitadas mais tarde, formando uma estrutura hierárquica. (DALL'OGGIO, 2007)

Figura: Abstração



Orientação a Objetos com PHP

Classes abstratas

São classes que nunca serão instanciadas na forma de objetos, somente suas filhas serão. Nestes casos, é interessante marcar essas classes como sendo classes abstratas, de modo que cada classe abstrata é tratada diferentemente pela linguagem de programação, a qual irá automaticamente impedir que se instanciem objetos a partir dela. (DALL'OGGIO, 2007)

Código-Fonte 1: Conta.class.php

```
<?php
abstract class Conta {
    // ...
    // conteudo da classe
    // ...
}
?>
```

Orientação a Objetos com PHP

Classes abstratas

Código-Fonte 2: classe-abstrata.php

```
<?php  
include_once 'classes/Conta.class.php'; //insere a classe  
$conta = new Conta; //cria um objeto  
?>
```

Orientação a Objetos com PHP

Classes finais

A classe final não pode ser uma superclasse, ou seja, não pode ser base para uma estrutura de herança. Se definirmos uma classe como final pelo operador FINAL, ela não poderá ser mais especializada. (DALL'OGGIO, 2007)

Código-Fonte 3: Conta.class.php

```
<?php
final class ContaPoupanca extends Conta {
    // ...
    // conteudo da classe
    // ...
}
?>
```

Orientação a Objetos com PHP

Classes finais

Código-Fonte 4: classe-final.php

```
<?php
include_once 'classes/Conta.class.php';
include_once 'classes/ContaPoupanca.class.php';

class ContaPoupancaUniversitaria extends ContaPoupanca {

    // ... sobescrita de metodos

}
?>
```

Orientação a Objetos com PHP

Métodos abstratos

Um método abstrato consiste na definição de uma assinatura na classe abstrata. Este método deverá conter uma implementação na classe filha, mas não deve possuir implementação na classe em que ele é definido. (DALL'OGGIO, 2007)

Código-Fonte 5: Conta.class.php

```
<?php
abstract class Conta {
    // ...
    // conteúdo da classe
    // ...
    abstract function Transferir ($Conta, $Valor);
}
```


Orientação a Objetos com PHP

Métodos abstratos

Código-Fonte 6: abstrato.php

```
<?php
include_once 'classes/Pessoa.class.php';
include_once 'classes/Conta.class.php';
include_once 'classes/ContaPoupanca.class.php';

$carlos = new Pessoa(10,"Carlos da Silva",1.85,25,72,"Ensino Medio"
    ,650.000);
$carlos = new ContaPoupanca(6677,"CC.1234.56","10/07/02",$carlos,
    9876, 500.00,"10/07");
?>
```

Orientação a Objetos com PHP

Métodos abstratos

Código-Fonte 7: ContaPoupanca.class.php

```
<?php
class ContaPoupanca extends Conta {
    // ...
    // conteúdo da classe
    // ...
    function Transferir ($conta, $valor) {
        if ($this->Retirar($valor)){
            $conta->Depositar($valor);
        }
    }
}
?>
```

Orientação a Objetos com PHP

Métodos finais

Um método final não pode ser sobrescrito, ou seja, não pode ser redefinido na classe filha. Para marcar um método como final, basta utilizar o operador `FINAL` no início da sua declaração. (DALL'OGGIO, 2007)

Orientação a Objetos com PHP

Métodos finais

Código-Fonte 8: ContaCorrente.class.php


```
<?php
class ContaCorrente extends Conta {
    var $TaxaTransferencia = 2.5;
    // conteúdo da classe
    final function Transferir ($conta, $valor) {
        if ( $this->Retirar($valor) {
            $conta->Depositar($valor);
        }
        if ( $this->Titular != $conta->Titular) {
            $this->Retirar($this->TaxaTransferencia);
        }
    }
}
```

Orientação a Objetos com PHP

Métodos finais

Código-Fonte 9: metodo-final.php

```
<?php
include_once 'classes/Conta.class.php';
include_once 'classes/ContaCorrente.class.php';
class ContaCorrenteEspecial extends ContaCorrente {
    function Depositar($valor) {
        echo "Sobrescrevendo metodo Depositar.\n";
        parent::Depositar($valor);
    }
    function Transferir($conta, $valor) {
        echo "Sobrescrevendo metodo Transferir.\n";
        parent::Transferir($conta, $valor);
    }
}
?>
```

 DALL'OGGIO, P. **PHP: programando com orientação a objetos**. 1. ed. São Paulo: Novatec Editora, 2007. 580 p.