

PHP orientado à objetos – parte 1

Postado por [Abraão Levi](#) em 19 de dezembro de 2010 @ 15:56 em [Artigos e tutoriais,PHP](#)



Nesta série de artigos você vai aprender e ver que orientação à objetos não é um bicho de sete cabeças e que seu uso pode poupar muito trabalho dependendo do tipo de projeto que você pode estar desenvolvendo.

Apesar deste artigo utilizar a linguagem PHP, o conceito pode ser aplicado à outras linguagens. Então vamos começar!

Entendendo Objetos e Classes

Antes de começarmos a nos aprofundar na Orientação à objetos, é muito importante nós sabermos as diferença entre **Classes** e **Objetos**. Neste primeiro artigo vamos ver exatamente

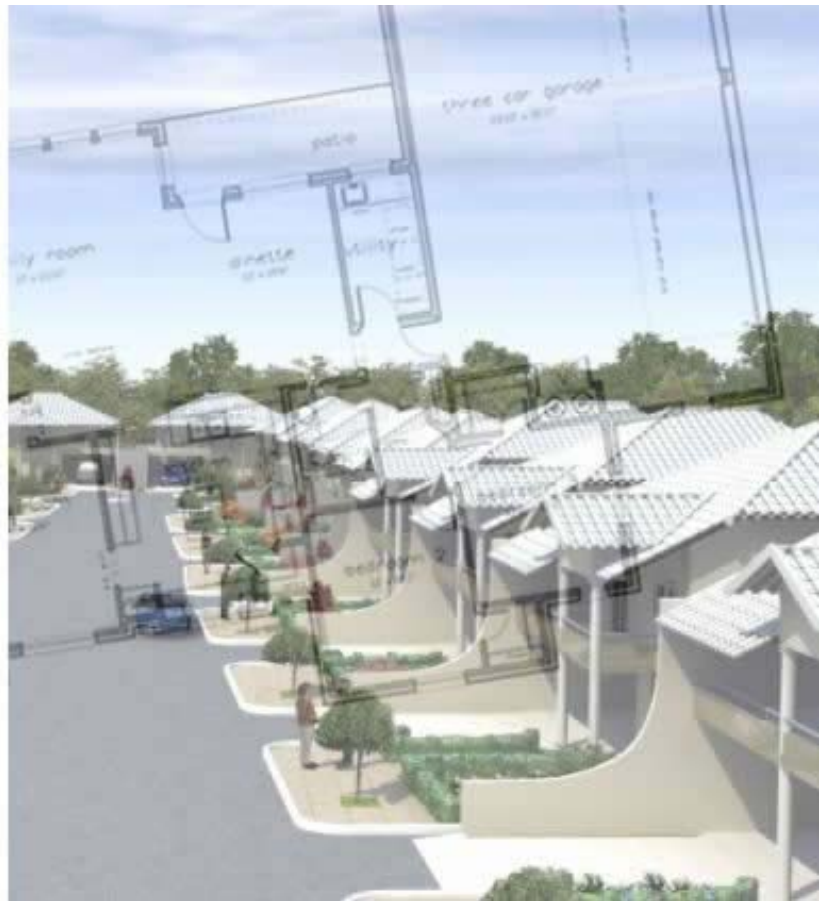
isto.

Identificando as diferenças entre objetos e classes

Muitas vezes existe uma certa confusão na OOP (*Object-oriented programming, ou em português programação orientada à objetos*): desenvolvedores experientes começam a falar sobre objetos e classes, e eles parecem ser termos intercambiáveis (coisas diferentes podem ser usadas alternadamente com o mesmo propósito sem o que o resultado seja prejudicado). Este não é o caso, talvez a diferença pode ser difícil de se perceber mas logo mais você vai entender.

Uma classe, por exemplo, é como **uma planta de uma casa**. Ele define a forma da casa no papel, com as relações entre as diferentes partes da casa claramente definidas e planejadas, mesmo que a casa não existe.

Um objeto, então, é como a **casa real** construída de acordo com esse modelo. Os dados armazenados no objeto, como a madeira, fios e concreto que compõem a casa, sem ser montado de acordo com o projeto, é apenas um monte de coisas. No entanto, quando tudo vem junto e organizado, torna-se uma casa.



Classes formam a estrutura de dados e as ações e usa essa informação para construir objetos. Mais de que um objeto pode ser construído a partir da mesma classe, mas ao mesmo tempo, cada um independente do outro. Continuando com nossa analogia da construção, imagine 150 casas construídas a partir de uma planta em condomínio, inicialmente parecem todos iguais, mas são diferentes: famílias diferentes e decorações diferentes no interior das casas.

Ou seja, uma classe pode criar mais que um objeto e que cada objeto é independente de seus "irmãos".

Estruturando Classes

A sintaxe para criar uma classe é bastante simples: declarar uma classe usando a palavra-chave `class`, seguido pelo nome da classe e um conjunto de chavetas (`{}`):

```
<!--?php

class MyClass {
    // As propriedades e métodos vão aqui
}

?-->
```

Depois de criar a classe, ela pode ser instanciada e armazenados em uma variável usando palavra-chave **new**:

```
$obj = new MyClass;
```

Para ver o conteúdo da classe, você pode usar a função `var_dump()`:

```
var_dump($obj);
```

Experimente este processo, colocando todo o código anterior, em um novo arquivo, aqui vou chamar de `teste.php` e vou colocar no diretório principal do meu servidor local:

```
<!--?php

class MyClass {
```

```
// As propriedades e métodos vão aqui
}

$obj = new MyClass;

var_dump($obj);

?-->
```

Se você carregar a página em seu navegador, por exemplo <http://localhost/test.php>, o seguinte deverá apresentar:

```
object(MyClass)#1 (0) { }
```

Na sua forma mais simples, você acabou de completar seu primeiro script OOP.

Nos próximos artigos vamos ver um pouco sobre propriedades (atributos) e métodos.

Até a próxima!

Veja também:

- [PHP orientado à objetos – parte 4](#) ^[1]
- [PHP orientado à objetos – parte 3](#) ^[2]
- [PHP orientado à objetos – parte 2](#) ^[3]
- [Como encurtar url com PHP](#) ^[4]
- [Otimização, Otimização, Otimização...](#) ^[5]

Artigo imprimido de Falando TI: <http://falandoti.com.br>

URL do Artigo: <http://falandoti.com.br/php-orientado-a-objetos-parte-1/>

Copyright © 2010 Falando TI. All rights reserved.