

Mar 07

Implementando Polimorfismo com PHP

Postado em : Sábado, 7 de Março de 2009 | Autor: Paulo Teixeira

Como alguns amigos sabem, já a algum tempo, estou desenvolvendo um framework em PHP. Esse framework é para uso pessoal, mas já foi utilizado em diversas situações não pessoais.

Bom, desde então, comecei a enfrentar um problema pois precisava implementar o framework compatível com vários bancos de dados. Nesse momento logo veio a necessidade de implementar um Polimorfismo.

Comecei a pesquisar como fazer com PHP, achei poucas fontes que serviam de alguma ajuda. Então depois de uns testes, resolvi discutir com um amigo meu Leandro Borges vulgo "Garotinho" sobre tal implementação e chegamos a um denominador comum.

Vamos ao código comentado:

```
<?php
// classe base
abstract class Operador {
    // método da classe base que será reescrito nas classes filhas
    // nesse método fizemos uma implementação para servir como exemplo
    // de como implementar uma modificação no mesmo através de suas classes
filhas
    // O mesmo poderia ser simplesmente assim: abstract function
calcular($x,$y);
    // sem nenhuma implementação, servido apenas como base para as classes
demais
    protected function calcular($x,$y)
    {
        return $x + $y;
    }
}

// classe filha extendendo da classe base
class Soma extends Operador {
    // método reescrito (override) da classe base
    public function calcular($x,$y) {
        return $x + $y;
    }
}

// classe filha extendendo da classe base
class Subtracao extends Operador {
    // método reescrito (override) da classe base
    public function calcular($x,$y)
    {
parent e
        // neste momento executamos o método calcular base utilizando o
        // fazemos uma sobrecarga em sua implementação
        // como na segunda opção do método da classe pai, esse método filho
poderia
        // ser escrito dessa forma: return $x - $y, caso o método da classe
pai
        // não tivesse nenhuma implementação
        return parent::calcular($x, $y) * ($x - $y);
    }
}

// agora criamos duas implementações, Soma() e Subtracao()
$obj1 = new Soma();
$obj2 = new Subtracao();

// aqui executamos os métodos das duas classes filhas
echo $obj1->calcular(2,2);
echo "<br />";
echo $obj2->calcular(8,3);
?>
```

Bom dessa maneira chegamos a forma mais plausível para implementar polimorfismo. Eu não sacava muito de polimorfismo, e como o PHP não é uma linguagem tipada, fica bem mais complicado de entender. Mas acho que agora ficou legal.

Devo os méritos ao meu amigo Garotinho, pela força que sempre deu nessa área de OOP entre outras.

[]s

