



Vzor Factory

Takzvaný "vzor Factory" umožňuje jednoduché vytváření objektů uvnitř jiných tříd.

Podívejte se na následující příklad:

```
<?php
function __autoload($soubor){
    include_once $soubor.'.php';
}

function factory($skript){
    if($skript == 'MySQL'){
        return new Mysql;
    }else if($skript == 'SQLite'){
        return new SQLite;
    }else{
        return FALSE;
    }
}

$ins = factory('MySQL');
?>
```

Funkce `factory` přijímá jako argument jméno třídy, kterou chcete volat. Následně voláme instanci vybrané třídy, ta bude zahrnuta pomocí automatického nahrávání tříd (autoloading).

Pozn. red.: Zdůrazňujeme, že tento text se týká PHP 5.

zpracoval: Mrozek, Jakub (15. 11. 2006)

Komentáře k článku

Od: DavesMan

IP: 212.71.177.1

Datum: 2007-03-16 14:59:28

Jako Factory se zpravidla používá celá třída, nikoliv jen funkce.

Od: Václv Mach

IP: 147.231.251.16

Datum: 2007-10-30 16:15:56

v článku je vzor factory zřetně omezen jen na Factory method aniž by to bylo zmíněno. Často se používá abstract Factory ke tvorbě celé sady souvisejících objektů, na což naráží předchozí příspěvek.

Od: FantomX1

IP: 78.98.210.168

Datum: 2007-11-03 14:59:46

Co bude v premennej soubor? aky soubor to ma volat?

syntax __autoload sa pokial viem pouziva len v triedach pri konstruktoroch nie?

kedy sa teda spusti funkcia __autoload a aky bude jej parameter ked zavolam triedu

\$ins = factory('MySQL');? vďaka

Od: Ondřej Brejla

IP: 89.233.130.135

Datum: 2007-12-03 12:13:10

__autoload() se spustí právě v případě, když se snažím vytvořit instanci třídy, kterou nemám nikde

definovanou... __autoload() se ji pokusí naincludovat. Jako parametr je právě jméno třídy, tedy Mysql, který se vkládá automaticky.

Od: Megaloman

IP: 147.229.216.182

Datum: 2008-07-14 11:30:15

Jen bych dodal, že **není** pravda, že funkce __autoload() musí být v konstruktoru třídy.

Stačí dát na začátek hlavního skriptu:

```
function __autoload($nazev_tridy)
{
    require_once('cesta_k_mym_tridam' . DIRECTORY_SEPARATOR . $nazev_tridy . '.pripona');
}
```

Pokaždé, když se potom pokusíte použít třídu nebo interface, který ještě nebyl definovaný, tak PHP zavolá funkci __autoload() s parametrem, ve kterém bude název třídy/rozhraní.

Například na začátku skriptu je:

```
function __autoload($class)
{
    require_once('tridy' . DIRECTORY_SEPARATOR . $class . '.inc');
}
```

a v průběhu skriptu se pokusíte vytvořit instanci zatím nedefinované třídy 'dummy':

```
$obj = new dummy();
```

tak skript neskončí chybou, ale zavolá funkci

```
__autoload('dummy');
```

která načte tridy/dummy.inc (*pokud ovšem existuje požadovaný soubor a obsahuje definici třídy dummy*) a následně vytvoří instanci třídy dummy.

Samozřejmě může být kód uvnitř funkce __autoload() mnohem komplikovanější a uvažovat "naming conventions" vašich tříd/rozhraní, algoritmus pro procházení adresáři a hledání definičního souboru... viz diskuze na

<http://cz.php.net/manual/en/language.oop5....>

Přidejte svůj názor!

, ISSN 1212-8651 ,

© Zoner software, s.r.o., všechna práva vyhrazena, tento server dodržuje právní předpisy o ochraně osobních údajů.