



Viditelnost

PHP 5 umožňuje definovat vlastnostem i metodám viditelnost pomocí zvláštních klíčových slov. PHP tedy dělí atributy (vlastnosti) a metody na `public`, `protected` a `private`.

Vlastnost nebo metoda určená pomocí **public** je přístupná všem uživatelům třídy.

Klíčové slovo **protected** (z anglického "chráněný") před metodou nebo vlastností třídy znamená, že je vlastnost přístupná uživatelům třídy (rodiče) a podtřídy, která tuto třídu dědí.

Soukromá metoda nebo vlastnost předznačená klíčovým slovem **private** je přístupná pouze uvnitř třídy.

Viditelnost není nutné definovat, pokud tak neučiníte, pak nabývá implicitní hodnoty *public*.

```
<?php

class Trida{
    public $verejna = 'Veřejná';
    protected $chranena = 'Chráněná';
    private $soukroma = 'Soukromá';

    function volatVlastnosti(){
        echo $this->verejna;
        echo $this->chranena;
        echo $this->soukroma;
    }
}

$instance = new Trida();
$instance->volatVlastnosti();
echo $instance->verejna;
//echo $instance->chranena;
//echo $instance->soukroma;

?>
```

Jak tato instance funguje? Když voláme metodu `$instance->volatVlastnosti()`, vypíše se všechny vlastnosti definované uvnitř metody `volatVlastnosti()` (uvnitř rodičovské třídy jsou všechny vlastnosti viditelné), když však voláme jednotlivé vlastnosti postupně, dochází k několika změnám.

Zápis `echo $instance->verejna;` bude bez problémů fungovat, vlastnost je veřejná, můžeme ji volat kdekoli. Zápisy `echo $instance->chranena;` a `echo $instance->soukroma;` budou nefunkční, nejsou viditelné mimo třídu.

Přidejme k našemu zápisu nového potomka třídy:

```
<?php

class TridaDedi extends Trida{
    function volatVlastnosti(){
        echo $this->verejna;
        echo $this->chranena;
        echo $this->soukroma;
```

```
}  
}  
  
$instance = new TridaDedi();  
$instance->volatVlastnosti();  
echo $instance->verejna;  
echo $instance->soukroma;  
?>
```

Metoda `$instance->volatVlastnosti()` vypíše pouze vlastnosti `$instance->verejna;` a `$instance->chranena;`, zápis `echo $instance->verejna;` bude bez problémů funkční, `echo $instance->soukroma;` nevypíše nic. Kdybychom chtěli podobným způsobem volat i chráněnou vlastnost, celý skript by skončil fatální chybou. Stejná pravidla platí i pro metody.

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

zpracoval: Mrozek, Jakub (1. 7. 2006)

Komentáře k článku

Od: Jordan

IP: 193.86.251.2

Datum: 2007-03-08 10:07:02

Možná by byl dobrý příklad i s `public`, `private`, `protected` metodami ve třídě, ne jen s atributy.

Od: Ivo

IP: 77.48.74.254

Datum: 2008-08-31 11:50:49

Podle mě by to chtělo doplnit k čemu vlastně to `protected` a `private` slouží a kdy se dají využít. Takto začátečník z toho pochopí akorát, že se nedají použít mimo rodičovskou třídu.

[Přidejte svůj názor!](#)

. ISSN 1212-8651 .

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