

# Programování v PHP

Katedra softwarového inženýrství  
Fakulta informačních technologií  
České vysoké učení technické v Praze

© Pavel Štěpán, Helena Wallenfelsová, 2013

**Objekty – další možnosti**  
BI-PHP



```

<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Objekty a tridy hlouběji</title>
    </head>
    <body>
        <?php
            class Zam{
                // staticke metody a promenne (na urovni TRIDY,
                // nikoli instance)
                // public static $LastID = 0;    // staticka promenna
                private static $LastID = 0;    // private - pristupne
                                                // POUZE uvnitr kodu tridy

                public static function GetLastID(){ // staticka metoda
                    return self::$LastID; // pouziti staticke promenne UVNITR
                                          // kodu tridy
                }

                // instancni promenne a metody

                // definice vlastnosti - 2. varianta - pristup
                // prostrednictvim "gettru" a "setteru"
                private $ID;
                private $Jmeno;

                public $Mesto = "Praha";
                public $Stat = "CR";

                private $Telefon = 1234;

                public function getJmeno(){ // getter
                    // ... // zde mozno dekomprimovat, formatovat, ...
                    return $this->Jmeno;
                }

                public function setJmeno($Hodnota){ // setter
                    // ... // zde mozno OTESTOVAT, komprimovat, upravit, ...
                    $this->Jmeno = $Hodnota;
                }

                // read-only property ID - chybi setter
                public function getID(){
                    // ...
                    return $this->ID;
                }

                // definice konstruktoru - 1. varianta
                /*
                public function construct($ID,$Jmeno){
                    $this->ID = $ID;
                    $this->Jmeno = $Jmeno;
                }
                */
            }
        </?php>
    </body>
</html>

```

```

// konstruktor 2. varianta - pouziti staticke promenne
public function __construct($Jmeno){
    $this->ID = ++self::$LastID;
    $this->Jmeno = $Jmeno;
}

// __set - k osetreni neexistujici nebo nepristupne vlastnosti
// popr. "dynamicke" vytvoreni novych property ("pretezovani")
// funkce __set se zavola, pokud se pouzije neexistujici
// (nepristupna) vlastnost
public function __set($name,$value){
    // echo "Vlastnost $name neexistuje<br>";
    $this->$name = $value; // dynamicky rozsiri object o
                          // zatim neexistujici vlastnost
}

// __get - slouzi k nacteni hodnoty neexistujici nebo
// nepristupne promenne
public function __get($name){
    if ($name == "Telefon")
        return $this->Telefon;
    else
        return "Vlastnost $name neexistuje<br>";
}

// definice metody
public function SetCase($Upper){
    if ($Upper)
        $this->Jmeno = strtoupper($this->Jmeno);
    else
        $this->Jmeno = strtolower($this->Jmeno);
}

// definice destruktoru:
public function __destruct(){
    echo "Prave se rusi objekt". $this->Jmeno. "<br>";
}
}

// volani staticke promenne nebo metody VNE tridy
// echo "LastID=".Zam::$LastID."<br>";
echo "LastID=".Zam::GetLastID()."<br>";
// $zam1 = new Zam(1,"Franta");
$zam1 = new Zam("Franta");
// echo "LastID=".Zam::$LastID."<br>";
echo "LastID=".Zam::GetLastID()."<br>";

$zam1a = $zam1;

$zam1cloned = clone $zam1; // provede kopii dat na heapu
                          // (shallow - 1. uroven)
// __clone - je mozne definovat vlastni funkci clone - napr. kopie
// do libovolne hloubky

```

```

$zam1->SetCase(true);

$zam1->Titul = "Ing."; // pristup k neexistujici vlastnosti
                      //(pouzije se funkce __set)

echo "Titul=".$zam1->Titul.", Telefon=".$zam1->Telefon."<br>";

// $zam2 = new Zam(2,"Venca");
$zam2 = new Zam("Venca");
// echo "LastID=".$zam2->LastID."<br>";
echo "LastID=".$zam2->GetLastID()."<br>";

echo "Vypis vlastnosti:<br>";
// vypis vlastnosti a jejich hodnot
foreach ($zam1 as $vlastnost => $hodnota){
    echo "\$zam1->$vlastnost = $hodnota<br>";
}

echo "<br>Vypis zamestnancu:<br>";

// echo "Prvni: ID=".$zam1->ID.", Jmeno=".$zam1->Jmeno."<br>";
echo "Prvni: ID=".$zam1->getID().", Jmeno=".$zam1->getJmeno()."<br>";
echo "Prvni cloned: ID=".$zam1->getID().", Jmeno=".$zam1->getJmeno()."<br>";
$zam1 = $zam1->clone();
echo "Druhy: ID=".$zam2->getID().", Jmeno=".$zam2->getJmeno()."<br>";

?>
</body>
</html>

```