



OOP static methods and classes

PHP WebDevelopment 2019

Милена Томова
Vratsa Software

<https://vratsasoftware.com/>

Table of Contents



1. Static properties and methods
2. Constants





Static properties

Static properties

Properties that belong to the class rather than to the instance.

Used without any class instance.

```
class Car {  
    static public $type = 'car';  
}
```



Static properties

A static property is accessed
inside a class -

```
public function get_car_type(){  
    return self::$type;  
}
```

A static property is accessed
outside a class -

```
echo Car::$type;
```



Static properties

Static properties values in the classes that extend a class

```
Class Toyota extends Car {  
  
    static public $type = 'toyota';  
  
    public function get_toyota_type(){  
        echo self::$type;  
    }  
}  
  
$toyota = new Toyota();  
$toyota->get_car_type();//car  
$toyota->get_toyota_type();//toyota
```

```
Class Car {  
  
    static public $type = 'car';  
  
    public function get_car_type(){  
        echo static::$type;  
    }  
}  
  
$toyota = new Toyota();  
$toyota->get_car_type();//toyota  
$toyota->get_toyota_type();//toyota
```



Static methods

Static properties



Methods that belong to the class, not to the class instances.

Static methods are called on the class, not on the class instances.

Static methods use only static properties.

```
class Car {  
  
    static public $type = 'car';  
  
    static public function get_car_type(){  
  
        return self::$type;  
  
    }  
}
```

Car::get_type();//car

Toyota::get_type();//car





Late Static Binding

Static properties

Replacing **self** with **static** keyword allows to work with the current class's value of a static property.

```
Class Car {  
  
    static public $type = 'car';  
  
    public function get_car_type(){  
        echo static::$type;  
    }  
}  
  
class Toyota {  
    static public $type = 'toyota';  
}  
  
Car::get_car_type();//car  
Toyota::get_car_type();//toyota
```

Static properties and methods

Task: Using static properties and/or methods implement displaying the current instance number.

Describe the process - the methods to be defined and properties to be used.



Constants

constants

Constants are defined by

```
define('OPERATOR', 'driver');
```

in procedural programming.

Constant names are in **uppercase** by covention.

In a class constants are defined by

```
class Car {  
  
    const OPERATOR = 'driver';  
  
}
```

using the keyword **const**.



constants

Constants can be used inside of the class -

```
public get_const() {  
  
    echo self::OPERATOR  
  
}
```

or outside of the class

Carr::OPERATOR



A class constant is visible and can be used in the classes that inherit the class where the constant is defined.

```
Class ToyotaCar extends Car{  
  
    public function toyota_description(){  
  
        echo 'Usually a Toyota is driven by an ' .  
  
            self::OPERATOR;  
  
    }  
  
}  
  
echo ToyotaCar::OPERATOR
```



constants

A common example of using constants in OOP is a class that implements a database connection.

The data values needed to connect a Data base are stored in constants.

```
const dbHost =
```

```
const dbName =
```

```
const dbUsername =
```

```
const dbPassword =
```



Questions?



Гнездото
Coworking

Цялостен
курс по
програми
ране

Дизайн
курс

Курс по
дигит.
маркетинг

MindHub



Partners



**Telerik
Academy**



MindHub

ПРОМЯНАТА

Trainings @ Vratsa Software



- Vratsa Software – High-Quality Education, Profession and Jobs
 - www.vratsasoftware.com
- The Nest Coworking
 - www.nest.bg
- Vratsa Software @ Facebook
 - www.fb.com/VratsaSoftware
- Slack Channel
 - www.vso.slack.com

