

<link rel="stylesheet" type="text/css" href="http://www.zawebis.com/css/noscript.css" /> Для полной работы сайта необходимо, чтоб в вашем браузере был включен JavaScript.

Здесь вы найдете [инструкции](#) как включить JavaScript в вашем браузере.

- 
- [HTML & CSS](#)
- [PHP & SQL](#)
- [JavaScript](#)
- [Linux](#)
- [На Заметку](#)
- [Написать](#)

## МЕНЮ

- [Главная Страница](#)
- [HTML & CSS](#)
- [PHP & SQL](#)

+

- [PHP](#)
- [SQL](#)
- **CMS**

+

- [Joomla](#)
- [Wordpress](#)
- [Opencart](#)
- [Magento](#)

- **FRAMEWORK**

+

- [Yii](#)

- [JavaScript](#)
- [Linux](#)
- [На Заметку](#)
- [Написать Мне](#)
- [support@zawebis.com](mailto:support@zawebis.com)

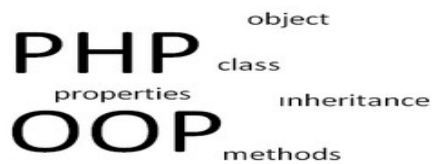
[zaWBis](#)

- 
- 
- 
- 
- 
- 
-

Поискать инфу...

> PHP & SQL > [PHP](#) > Шпаргалка по ООП в PHP

## Шпаргалка по ООП в PHP



17 сентября, 2014 PHP & SQL

Объектно-Ориентированное программирование давно уже шагнуло в жизнь программистов. Оно везде используется. И кодить на процедурке - это уже даже и не модно. Но начинающим программистам очень сложно перейти на новый уровень - перейти от процедурного программирования к ООП. Порою пугают эти не понятные наборы символов, и не понятно от куда что берется. Да еще это странное переименование переменной в свойство, а функции в метод — вообще отбрасывает всякое желание изучать ООП и многие прогеры ломаются на этом и становятся "непрогерами".

Процедурное программирование оно хорошо для мелких задач, но если уже речь идет о большом проекте, то лучше ООП. Тем более что все фреймворки используют ООП в своей архитектуре. А FrameWork это скорость. А если нужна скорость, то без ООП увы никак.

И так. Тут я приведу список основ PHP ООП, которые всегда должны быть под рукой у начинающего PHP ООП прогера. Ибо когда вы начнете читать код, то должны иметь представление о том, что там написано.

**!!!правило 1** (это нужно запомнить навсегда)

свойство объекта — это переменная

метод объекта — это функция

### создание объекта:

```
$obj = new CassName(); $obj->methodClass(); // это обращение к методу класса $obj->svoistvoClass;
// это обращение к свойству класса
```

### конструктор и деструктор

```
function __construct(){} // это метод, который срабатывается во момент создания объекта
function __destruct(){} // это метод, который срабатывается во момент удаления объекта //!!!важно!!! //в
конструктор можно передавать переменные //в деструктор нельзя, так как если передать переменную,
которая будет удалена, то это приведет к ошибке
```

### вспомогательные константы

```
echo __CLASS__; // выведет название класса echo __METHOD__; // выведет название объект::метод
```

### спецификаторы доступа или (инкапсуляция)

```
public //- могу получить доступ из кода и из классов(родительский и наследуемый) protected //-
могу получить доступ только из классов(родительский и наследуемый) private //- могу получить
доступ только из родительского класса
```

### указание на родительский метод

```
parent::classMethod();
```

### перебор свойств объекта

```
foreach ($objName as $name=>$value) { echo "$name: $value"; } // $objName можно заменить на $this
(если я использую это в классе), чтоб указать, что можно использовать любой объект, а не какой-то
конкретно
```

## константы классов

```
const CONST_NAME = 'данные константы'; // объявление константы в классах echo USER::CONST_NAME; //
вывод константы вне класса, где USER это название класса, а CONST_NAME это название константы echo
self::CONST_NAME; // вывод константы внутри класса, где CONST_NAME это название константы
```

## проверяем если объект является экземпляром класса

```
if ($object_name instanceof ClassName) {}
```

## исключения (они нужны для отлова ошибок)

```
try { // some code if ($error) throw new Exception("Тут ошибка"); } catch(Extension $e) { echo $e-
>getMessage(); }
```

## перегрузка стандартных PHP классов (то есть наследование класса)

```
class MyException extends Exception { function __construct($msg) { parent::__construct($msg); } }
```

## МЕЛКИЕ ВАЖНЫЕ ПОЛЕЗНЫЕ ФИШКИ

**абстрактные классы** и интерфейсы используются для описания наших классов, то есть здесь мы задаем шаблоны по которым будут работать наши наследуемые классы

абстрактный класс (abstract) - это шаблон, который содержит в себе объявление методов если в абстрактном классе есть абстрактный метод, то он должен быть перегружен

```
abstract class MyClass {}
```

**интерфейсы** - это тоже абстрактный класс, но который содержит только абстрактные методы

**так же нужно перегружать методы**

```
interface MyClass{} class MyNaslednik implements MyClass{}
```

**метод - FINAL - нельзя перегрузить**

```
final function MyMethod(){} 
```

## автоматическая подгрузка классов

то есть если представить себе ситуацию, что вы используете несколько экземпляров разных классов и чтоб избавиться от бесконечных include (require), лучше один раз указать !!!ФУНКЦИЮ!!! \_\_autoload

```
// это функция, а не метод function __autoload($className) { include($className); }
```

## магический методы \_\_set() и \_\_get()

**\_\_set()** - срабатывает автоматически в том случае, когда мы присваиваем какое-то значение не известному(не объявленному) свойству

```
function __set($objName, $objValue) { $this->someVar[$objName] = $objValue; }
```

**\_\_get()** - срабатывает автоматически в том случае, когда мы пытаемся получить значение не известного(не объявленного) свойства

```
function __get($objName) { return $this->someVar[$objName]; }
```

**магический метод \_\_call()** срабатывает автоматически в том случае, когда мы пытаемся вызвать не известный(не объявленный) метод

```
function __call($methodName, $params){}
```

**магический метод `__toString()`** срабатывает автоматически в том случае, когда мы пытаемся вывести объект

```
$obj = new Class(); echo $obj; function __toString() { // что-то сделать... }
```

---



**Об авторе:**

[Маковкин Александр](http://makovkin.info)

- 
- 
- 
- 
- 
- 
- 
- 

Занимаюсь веб-программированием с 2010 года. Имею опыт работы с Joomla, Wordpress, Opencart, CodeIgniter и Magento. В последнее время перешел на создание сайтов, используя php фреймворк Yii с версткой на Bootstrap'e и "оживлением" сайтов при помощи jQuery. Хостятся мои проекты на DigitalOcean. Активно использую и рекомендую всем Linux. Более подробно можно узнать на <http://makovkin.info>



**Добавить комментарий**

Имя:

Email:

**сах uk**

кликните на картинку,

чтоб обновить код

Введите этот странный

Коментарий:

Добавить комент

Отменить

## Свеженькое



•

[Форма сброса пароля вместо белого экрана](#)

10 мая, 2016



•

[Решение проблемы с отправкой писем в magento 1.9](#)

10 мая, 2016



•

[Как подменить отсутствующее изображение](#)

03 мая, 2016

## Облачно, тэги

[html](#) [css](#) [javascript](#) [jquery](#) [windows](#) [php](#) [sql](#) [wordpress](#) [opencart](#) [joomla](#) [magento](#) [yii](#) [photoshop](#) [framework](#) [cms](#) [xml](#) [seo](#) [видео](#)  
[bootstrap](#) [linux](#) [bash](#) [защита](#)

## Будь в курсе

<input type="text" value="Имя"/>	<input type="text" value="Email"/>
<input type="button" value="Подписаться"/>	

- [Книги](#)
- [Линки](#)
- [Софт](#)
- [Yii. Сборник рецептов](#)
- [Читаемый код или Программирование как искусство.](#)
- [Объектно-ориентированное программирование на PHP](#)
- [PHP глазами хакера](#)
- [Гении и аутсайдеры. Почему одним все, а другим ничего.](#)

вы сами знаете где еще книги можно скачать

## Летопись

- [2016 - 34 статьи](#)
- [2015 - 3 статьи](#)
- [2014 - 12 статей](#)
- [2013 - 12 статей](#)
- [2012 - 12 статей](#)
- [2011 - 10 статей](#)

## Что за сайт

Этот сайт представляет унифицированную базу, где инфа собрана в единое целое, а не разбросанна по многочисленным сайтам, записям на листах бумаги или головам людей; а так же доступна круглосуточно не зависимо от кого-либо или чего-либо.

## Будь в курсе

<input type="text" value="Имя"/>	<input type="text" value="Email"/>	<input type="button" value="Подписаться"/>
----------------------------------	------------------------------------	--

## Облачно, тэги

[html](#) [css](#) [javascript](#) [jquery](#) [windows](#) [php](#) [sql](#) [wordpress](#) [opencart](#) [joomla](#) [magento](#) [yii](#) [photoshop](#) [framework](#) [cms](#) [xml](#) [seo](#) [видео](#)  
[bootstrap](#) [linux](#) [bash](#) [защита](#)

## Интересненько



- [php trends показывает:](#)

прикольные PHP библиотеки для ваших проектов



- 

[php TheRightWay](#) указывает:

место где начинается путь PHP самурая



- 

GEEKBRAINS

[GeekBrains](#) проводит::

вебинары на IT'шные темы

[makovkin.info](#)

[DigitalOcean](#)