

# Автозагрузка классов в ООП на PHP

Вы уже знаете, что чтобы воспользоваться каким-нибудь классом, мы должны подключить его через require. Когда в проекте очень много классов, то получается очень много команд require, которые очень неудобно писать и трудно контролировать.

Для решения такой проблемы в PHP была добавлена автозагрузка классов. Автозагрузка позволяет PHP автоматически загружать файл с классом при попытке обращения к этому классу в коде. Однако, просто так классы загружаться не будут - их необходимо разместить по папкам сайта специальным образом, следуя какому-нибудь соглашению об именах файлов и папок классов. Можно пользоваться встроенным в PHP соглашением, либо придумать свое и пользоваться им.

Давайте для начала разберем стандартное соглашение PHP. Это соглашение заключается в то, что если у нас есть некоторый файл с классом, то путь по папкам сайта к этому файлу должен совпадать с пространством имен файла, а имя файла должно совпадать с именем хранящегося в нем класса. При этом имена папок и файла должны быть в нижнем регистре.

Посмотрим на примере. Пусть у нас есть класс Core\Admin\PageController. Это значит, что данный класс должен располагаться в папке /core/admin/ в файле *pagecontroller.php*.

Для включения автозагрузки по умолчанию в начале файла, в котором вызываются классы, следует вызвать функцию spl\_autoload\_register.

Пусть, например, у нас есть файл index.php, а в нем класс Core\Admin\PageController, файл которого расположен в соответствии с соглашением. Давайте создадим объект этого класса в файле index.php, не подключая этот класс через require, а используя автозагрузку:

```
1 | <?php
2 |     spl_autoload_register(); // включаем автозагрузку
3 |
4 |     $obj = new Core\Admin\PageController; // спокойно создаем обь
      ект
5 | ?>
```

## Задача 90.1

В файле index.php включите автоматическую загрузку классов. Следуя соглашению об именах папок и файлов сделайте класс Core\User, класс Core\Admin\Controller и класс Project\User\Data. В файле index.php создайте объекты этих классов.

## Замечания

Функция spl\_autoload\_register пишется только один раз в начале файла. Затем можно создавать столько объектов разных классов, сколько угодно, главное, чтобы их имена следовали соглашению.

Функция spl\_autoload\_register пишется только в начале того файла, который вызывается через адресную строку браузера. То есть, если какой-то класс вызывает внутри себя другой класс или наследует от кого-то, а сам класс вызывается в другом файле, то spl\_autoload\_register нужно написать только в этом другом файле - в самом файле класса дублировать эту функцию не надо.



