

Запись на курсы по HTML, CSS, JavaScript, PHP, фреймворкам и CMS,  
а также: помощь в поиске работы и заказов, стажировка на реальных проектах→

урок 85 из 107

[Верстка](#) [JavaScript](#) [PHP](#) [NodeJs](#) [Vue](#) [React](#) [Laravel](#) [WordPress](#) [AJAX](#) [Парсинг](#)[Бесплатные курсы по React для новичков. Начало 4-го ноября→](#) [Конкурс CSS картинок. Тема: Хэллоун. Призовой фонд: 100\\$. Подробности→](#)

# Класс DatabaseShell в ООП на PHP

Реализуем оболочку для работы с базой данных. Такая оболочка позволит нам не писать сырые SQL запросы, а делать их средствами PHP. Зачастую это занимает гораздо меньше кода и проще в обслуживании.

Здесь следует сказать, что подобные оболочки для работы с базой зачастую занимают более 1000 строк. Мы не будем делать такого монстра, а просто реализуем некоторый упрощенный вариант, чтобы вы поняли принцип. Например, вот такой:

```
1  <?php
2  class DatabaseShell
3  {
4      private $link;
5
6      public function __construct($host, $user, $password, $database)
7      {
8          $this->link = mysqli_connect($host, $user, $password, $database);
9          mysqli_query($this->link, "SET NAMES 'utf8'"); // устанавливаем кодировку
10     }
11
12     public function save($table, $data)
13     {
14         // сохраняет запись в базу
15     }
16
17     public function del($table, $id)
18     {
19         // удаляет запись по ее id
20     }
21
22     public function delAll($table, $ids)
23     {
24         // удаляет записи по их id
25     }
26
27     public function get($table, $id)
28     {
29         // получает одну запись по ее id
30     }
31
32     public function getAll($table, $ids)
33     {
34         // получает массив записей по их id
35     }
36
37     public function selectAll($table, $condition)
38     {
39         // получает массив записей по условию
40     }
41 }
42
43 // Создаем объект для работы:
44 $db = DatabaseShell('localhost', 'root', 'root', 'test');
45 ?>
```



Давайте посмотрим, как мы будем пользоваться нашим классом.

Сохраним нового юзера:

```
1 | <?php
2 |     $db->save('users', ['name' => 'user1', 'age
   |         ' => '23']);
3 | ?>
```

Удалим юзера с id равным 3:

```
1 | <?php
2 |     $db->del('users', 3);
3 | ?>
```

Удалим юзеров с заданными id:

```
1 | <?php
2 |     $db->delAll('users', [1, 2, 3, 5]);
3 | ?>
```

Получим юзера с id равным 3:

```
1 | <?php
2 |     $user = $db->get('users', 3);
3 | ?>
```

Получим юзеров с заданными id:

```
1 | <?php
2 |     $users = $db->getAll('users', [1, 2, 3,
   |         5]);
3 | ?>
```

Получим юзеров по условию:

```
1 | <?php
2 |     $users = $db->getAll('users', 'where id
   |         >= 3');
3 | ?>
```

## Задача 85.1

Реализуйте описанный класс DatabaseShell. Проверьте его работу.

