

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

урок 79 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 TagHelper в ООП на PHP

В предыдущих уроках мы с вами делали классы, каждый из которых был предназначен для создания множества объектов. Давайте теперь потренируемся делать классы, представляющие собой набор методов для решения некоторой задачи.

Сделаем, к примеру, класс **TagHelper**, с помощью которого можно будет создавать HTML теги:

```
1  <?php
2  class TagHelper
3  {
4      public function open($name, $attrs = [])
5      {
6          $attrsStr = $this->getAttrsStr($at
7              trs);
8          return "<$name$attrsStr>";
9      }
10     public function close($name)
11     {
12         return "</$name>";
13     }
14
15     // Формируем строку с атрибутами:
16     private function getAttrsStr($attrs)
17     {
18         if (!empty($attrs)) {
19             $result = '';
20
21             foreach ($attrs as $name => $value) {
22                 if ($value === true) {
23                     $result .= " $name";
24                 } else {
25                     $result .= " $name=\"\$value\"";
26                 }
27             }
28
29             return $result;
30         } else {
31             return '';
32         }
33     }
34 }
35 ?>
```

Воспользуемся нашим классом, к примеру, для того, чтобы сделать тег div с текстом:

```
1  <?php
2  $th = new TagHelper();
3  echo $th->open('div') . 'text' . $th->clo
4      se('div'); // <div>text</div>
5  ?>
```

А теперь используем наш класс для создания формы:

```
1  <?php
2  $th = new TagHelper();
3
```

```
4 | echo $th->open('form', ['action' => 'test.php', 'method' => 'GET']);
5 | echo $th->open('input', ['name' => 'year', 'type' => 'text']);
6 | echo $th->open('input', ['type' => 'submit']);
7 | echo $th->close('form');
8 | ?>
```

В результате получится следующий HTML код:

```
1 | <form action="test.php" method="GET">
2 |   <input name="year">
3 |   <input type="submit">
4 | </form>
```

Задача 79.1

Самостоятельно, не подсматривая в мой код, реализуйте описанный класс **Tag**.

Задача 79.2

Добавьте в класс **Tag** метод `show`, одновременно создающий открывающий и закрывающий теги, а также текст между ними. Метод должен принимать параметром имя тега и текст.

