

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

урок 48 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→](#)

Модификаторы доступа и трейты

Следует обратить ваше внимание на то, что использование трейтов - это не наследование. При наследовании наследования приватные методы и свойства не наследуются.

В трейтах же все наоборот: в использующем трейт классе будут доступны как публичные, так и приватные методы и свойства класса.

Посмотрим на примере:

```
1  <?php
2  trait TestTrait
3  {
4      // Приватный метод:
5      private function method()
6      {
7          return '!!!';
8      }
9  }
10
11 class Test
12 {
13     use TestTrait; // подключаем трейт
14
15     public function __construct()
16     {
17         // Используем приватный метод трейта:
18         echo $this->method(); // выведет '!!!'
19     }
20 }
21
22 new Test;
23 ?>
```

Изменения прав доступа к методам трейта

Внутри трейта можно использовать любой модификатор доступа для методов (то есть public, private или protected). При необходимости, однако, в самом классе можно этот модификатор поменять на другой. Для этого в теле **use** после ключевого слова **as** нужно указать новый модификатор.

Посмотрим на примере:

```
1  <?php
2  trait TestTrait
3  {
4      // Приватный метод:
5      private function method()
6      {
7          return '!!!';
8      }
9  }
10
11 class Test
12 {
13     use TestTrait {
14         TestTrait::method as public; // меняем мет
15         од на публичный
16     }
17 }
```



```
17  
18     $test = new Test;  
19     echo $test->method(); // выведет '!!!'  
20     ?>
```

