

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

урок 87 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→](#)

Упрощенный способ обращения к пространствам имен

Пусть у нас есть следующий класс Controller:

```
1  <?php
2      namespace Admin;
3
4      class Controller
5      {
6
7      }
8  ?>
```

Пусть также есть класс Page, наследующий от класса Controller:

```
1  <?php
2      namespace Admin;
3
4      class Page extends \Admin\Controller
5      {
6
7      }
8  ?>
```

Как вы видите, при наследовании мы указываем имя родителя вместе с пространством имен. В данном примере, однако, есть нюанс: оба класса принадлежат одному и тому же пространству имен. В таком случае при обращении к классу можно просто написать имя этого класса, вот так:

```
1  <?php
2      namespace Admin;
3
4      class Page extends Controller
5      {
6
7      }
8  ?>
```

Задача 87.1

Даны два класса:

```
1  <?php
2      namespace Modules\Shop;
3
4      class Cart
5      {
6
7      }
8  ?>
```

```
1  <?php
2      namespace Modules\Shop;
3
```

```
4 | class UserCart extends \Modules\Shop\Cart
5 | {
6 |
7 | }
8 | ?>
```

Упростите код для наследования класса, учитывая то, что оба класса находятся в одном пространстве имен.

Пространства имен в файлах

Команду **namespace** можно писать не только в файлах с определениями классов, но и вообще в любых других файлах. Например, зададим пространство имен в файле *index.php*:

```
1 | <?php
2 | namespace Admin;
3 | ?>
```

Задача 87.2

Пусть в файле *index.php* создается объект класса Controller:

```
1 | <?php
2 | namespace Admin;
3 |
4 | new \Admin\Controller;
5 | ?>
```

Упростите код для создания объекта, учитывая то, что создание объекта класса происходит в том же пространстве имен, которому принадлежит класс.

Относительные пути

Пусть в файле *index.php* происходит следующий вызов:

```
1 | <?php
2 | namespace Admin\Data;
3 |
4 | new \Core\Controller;
5 | ?>
```

Как вы уже знаете, при обращении к классу перед его пространством имен следует писать обратный слеш. На самом деле это не обязательно. Если этот слеш не написать, то вызываемое пространство имен будет рассчитано относительно текущего пространства. Смотрите пример:

```
1 | <?php
2 | namespace Admin\Data;
3 |
4 | new Core\Controller; // эквивалентно \Admin\Data\Core\Controller
5 | ?>
```

Задача 87.3

Даны два класса:

```
1 | <?php
2 | namespace Modules\Shop\Core;
3 |
```

```
4 |     class Cart
5 |     {
6 |
7 |     }
8 | ?>
```

```
1 | <?php
2 |     namespace Modules\Shop;
3 |
4 |     class UserCart extends \Modules\Shop\Core\Cart
5 |     {
6 |
7 |     }
8 | ?>
```

Упростите код для наследования класса, учитывая то, что оба класса находятся в одном пространстве имен.

Задача 87.4

Даны два класса:

```
1 | <?php
2 |     namespace Core\Data\Controller;
3 |
4 |     class Controller
5 |     {
6 |
7 |     }
8 | ?>
```

```
1 | <?php
2 |     namespace Core\Data\Model;
3 |
4 |     class Model
5 |     {
6 |
7 |     }
8 | ?>
```

Вот так создаются объекты этих классов в файле *index.php*:

```
1 | <?php
2 |     namespace Core\Data;
3 |
4 |     $controller = new \Core\Data\Controller;
5 |     $model      = new \Core\Data\Model;
6 | ?>
```

Упростите код для создания объектов, учитывая пространство имен, в котором создаются объекты наших классов.

