Команда use и пространства имен в ООП на PHP

Пусть у нас есть следующий класс Data:

```php
<?php
  namespace \Core\Admin;

  class Data
  {
    public function __construct($num)
    {

    }
  }
?>
```

Пусть также есть класс Page, создающий внутри себя объекты класса Data:

```php
<?php
  namespace Users;

  class Page
  {
    public function __construct()
    {
      $data1 = new \Core\Admin\Data('1');
      $data2 = new \Core\Admin\Data('2');
    }
  }
?>
```

Как вы видите, оба наших класса находятся в совсем разных пространствах имен, поэтому вызовы класса Data упростить нельзя, подобно тому, как мы это делали в предыдущем уроке.

Эти вызовы, однако, очень длинные и неудобные, так как в каждом вызове класса Data приходится указывать его длинное пространство имен.

Для решения подобной проблемы существует специальная команда **use**. С помощью этой команды достаточно один раз подключить класс по его полному имени, и после этого можно будет обращаться к этому классу просто по имени класса. Смотрите пример:

```php
<?php
  namespace Users;
  use \Core\Admin\Data; // подключаем класс

  class Page extends Controller
  {
    public function __construct()
    {
      $data1 = new Data('1'); // вызываем просто по
        имени
      $data2 = new Data('2'); // вызываем просто по
        имени
    }
  }
?>
```

Задача 88.1

Упростите следующий код с использованием **use**:

```php
<?php
  namespace Project;

  class Test
  {
    public function __construct()
    {
      // Создаем 3 объекта одного класса:
      $data1  = new \Core\Users\Data('user1');
      $data2  = new \Core\Users\Data('user3');
      $data3  = new \Core\Users\Data('user3');
    }
  }
?>
```

Задача 88.2

Даны следующие классы:

```php
<?php
  namespace Core\Admin;

  class Controller
  {

  }
?>
```

```php
<?php
  namespace Users;

  class Page extends \Core\Admin\Controller
  {

  }
?>
```

Упростите код наследования класса, применив команду **use**.

Подключение нескольких классов

Если нужно подключить несколько классов, то каждый из них подключается своей командой **use**:

```php
<?php
  namespace Users;
  use \Core\Admin\Data1; // подключаем класс
  use \Core\Admin\Data2; // подключаем класс

  class Page extends Controller
  {
    public function __construct()
    {
      $data1 = new Data1; // вызываем просто по
```

```
        имени
11      $data2 = new Data2; // вызываем просто по
        имени
12    }
13  }
14  ?>
```

Задача 88.3

Упростите следующий код с использованием **use**:

```
1  <?php
2    namespace Project;
3
4    class Test
5    {
6      public function __construct()
7      {
8        $model = new \Core\Admin\Model;
9        $data  = new \Core\Users\Storage\Data;
10     }
11   }
12 ?>
```

Команда use и относительные пути

При использовании команды use можно указывать относительные пути, подобно тому, как мы это делали в предыдущем уроке. Давайте посмотрим на примере.

Пусть мы подключаем некоторый класс:

```
1  <?php
2    namespace Core\Admin;
3    use \Core\Admin\Path\Router; // подключаем класс
4
5    class Controller extends Router
6    {
7
8    }
9  ?>
```

Как вы видите, начало пространства имен подключаемого класса совпадает с текущим пространством. Это значит, что мы можем эту часть при подключении нашего класса, убрав при этом начальный обратный слеш:

```
1  <?php
2    namespace Core\Admin;
3    use Path\Router; // делаем относительный путь
4
5    class Controller extends Router
6    {
7
8    }
9  ?>
```

Задача 88.4

Упростите следующий код с использованием **use**:

```
1  <?php
2    namespace Core\Storage;
3
```

```
4   class Model
5   {
6     public function __construct()
7     {
8       $database  = new \Core\Storage\DataBase;
9     }
10  }
11 ?>
```