# Пространства имён

# Лабораторная: определение пространства имён

Пространства имен (Namespace) – механизм группировки связанных констант, функций и классов, логически связанных между собой.

1. Откройте файл lab.php
2. Распечатайте текущее пространство имён

echo \_\_NAMESPACE\_\_;

1. Убедитесь, что в браузере ничего не вывелось
2. Над строкой с оператором echo впишите пространство имён Specialist:

namespace Specialist

1. Убедитесь, что в браузере отобразилось слово «Specialist». Закомментируйте вывод.
2. Внимательно изучите код

const SPEC\_URL = "http://specialist.ru";

function spec\_func() { return \_\_FUNCTION\_\_; }

class Specialist {

static function getClass(){

return \_\_CLASS\_\_;

}

}

echo "<p>",SPEC\_URL;

echo "<p>",\Specialist\SPEC\_URL;

echo "<p>",spec\_func();

echo "<p>",Specialist::getClass();

1. Вставьте код в lab.php, сохраните файл
2. Изучите результат работы. Должен получиться результат:

«http://specialist.ru

http://specialist.ru

Specialist\spec\_func

Specialist\Specialist»

1. Создайте в файле lab.php функцию substr():

function substr(){

echo " <p> test";

}

substr();

1. Обратите внимание! Несмотря на то, что substr() – существующая функция PHP в нашем пространстве имён Specialist, она является самостоятельной функцией.
2. Добавьте перед вызовом функции символ обратного слеша «\».
3. Посмотрите результат в браузере. Функция будет требовать два параметра. Удалите слеш перед функцией.

# Лабораторная: загрузка файлов по пространству имён

1. В нижней части скрипта в файле lab.php создайте экземпляр класса User (в текущем файле этот класс не описан, нам нужно просто попытаться создать экземпляр).

$user = new User();

echo "<p>",$user->name;

1. Должна показаться ошибка: «**Fatal error**: Class 'Specialist\User' not found in…». То есть в **текущем** пространстве имён нет такого класса.
2. Нам нужно подключить класс User из папки vendor. Папка с таким названием обычно содержит сторонние библиотеки/классы, которые мы можем загружать.

Добавьте обратный слеш перед названием класса при создании объекта:

$user = new \User();

1. Должна показаться ошибка: «**Fatal error**: Class 'User' not found in…». То есть в **глобальном** пространстве имён нет такого класса.
2. В верхней части файла lab.php подключите autoload.php:

require \_\_DIR\_\_ . '/vendor/autoload.php';

1. Создайте поочередно экземпляры класс Foo

$foo = new \Some\Foo\Foo(); //а потом просто new \Some\Foo ()

$foo->getInfo();

1. Сравните работу в случае с \Some\Foo\Foo() и \Some\Foo(). Найдите в папке vendor файлы, соответствующие вызываемым в коде.
2. Примечание: во многих фреймворках такая организация загрузки классов упрощает построение веб-приложения

# Дополнительно задание для самостоятельной работы

1. Изучите способы загрузки файлов с классами в Laravel, Yii и Zend Framework 2