

Инициализация объектов PHP

Ересь Артём Владимирович

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

В данной статье будет раскрыт смысл понятия инициализация как операции для объектов PHP, продемонстрирован процесс её использования.

Ключевые слова: Объект, объектно-ориентированное программирование, PHP

Initialization of objects of PHP

Yeres Artem Vladimirovich

Sholom-Aleichem Priamursky State University

Student

Abstract

In this article the concept sense initialization as operations for PHP objects will be revealed, process of its use is shown.

Keywords: Object, object-oriented programming, PHP

Основой объектно-ориентированного программирования на PHP и главным элементом для разработки новых проектов является класс. Он считается по определению неким шаблоном, необходимым для создания реальных предметов, выполняющих определенные действия, заложенные пользователем на функциональном уровне.

Целью данной работы является определение смысла понятия инициализация как операции для работы с объектами PHP и демонстрация процессов связанных с этим.

В сфере научно-исследовательской деятельности данная тема актуальна и достаточное количество работ представлено в свободный доступ. Авторы А.Н. Ундозерова, О.Н. Близнюк и Ю.А. Плакса рассмотрели основные аспекты разработки программного обеспечения касательно объектно-ориентированного программирования на PHP. В работе выявлены преимущества и недостатки данного [1]. В статье Р.Р. Мухаметзянова рассмотрены важнейшие способы и основные направления в обучении объектно-ориентированному программированию, что позволяет пользователям с низким уровнем опыта получить новые знания в этой сфере [2]. Подобно рассмотрен метод инициализации крупных объектов проекта на PHP, а также предложен данной порядок работы в интернет-ресурсе [3]. В источнике представлена классификация классов и объектов PHP, даны основы

и методы [4]. Подробно изложена информация конкретно для объектов в следующем источнике [5].

Для начала рассмотрим понятие объекта и некоторые его особенности.

Объект – совокупность свойств и методов, формирующихся на основе шаблонного варианта, содержимое которого прописано в структуре класса.

Однако объект является так же видом данных, который создается через специальную команду `new` в паре классовым наименованием. Поэтому можно сделать вывод, что объект формируется по классу.

Для демонстрации рассмотрим следующий класс:

```
<?php
class MyClass {
    public $prop1 = 100;
    public $prop2 = "Hello world";
    public $prop3 = TRUE;

    public function getStr() {
        return "Hello world";
    }
}
```

Рис. 1. Класс пример

Теперь проведем инициализацию классовых объектов PHP. Она имеет вид:

```
$obj1 = new MyClass();
$obj2= new MyClass();
```

Рис. 2. Инициализация

Стоит отметить, что после команды `new` пользователь дает лишь один оператор - наименование класса, для которого вызывается объект. То есть, именно `My Class`, является по сути шаблонным вариантом для пары появившихся объектов типа `My Class`. С виду они идентичны, но однако все же это пара совершенно разных объекта сходные по типу классу создания.

Так как объект это вид данных и внутри его структуры имеется набор операций и информация, то пользователь может вывести на дисплей его внутренние элементы с помощью `var_dump()`:

```
var_dump($obj1);
```

Рис. 3. Вывод содержимого

И пользователю система выдаст:

```
object(MyClass)#1 (3) {  
  ["prop1"]=>  
  int(100)  
  ["prop2"]=>  
  string(11) "Hello world"  
  ["prop3"]=>  
  bool(true)  
}
```

Рис. 4. Результат

В нашем случае элемент идентифицированного объекта, а так же его значение равняется 1. Если запросить информацию о втором объекте то результат будет тот же, сменится лишь номер для идентификации.

Проведем изменения в нашем объекте номер 2, точнее к его \$ prop1, дадим новое значение - "Some String".

```
$obj2->prop1 = "Some String";
```

Рис. 5. Смена значения

Для обеспечения доступности объектных свойств используется "->" вместе переменной и наименованием свойства. Благодаря этому, пользователь имеет возможность проводить манипуляции с объектом как переменной. Выведем на экран информацию об объекте номер 2:

```
object(MyClass)#2 (3) {  
  ["prop1"]=>  
  string(11) "Some String"  
  ["prop2"]=>  
  string(11) "Hello world"  
  ["prop3"]=>  
  bool(true)  
}
```

Рис. 6. Объект 2

В результате объект номер 1 не изменился, а второй получил заданные изменения, несмотря на то что они оба принадлежат одному.

То есть пользователь, после прописывания и определения классовых свойств получает уверенность в том, что в новосозданных объектах его присутствие уже определено изначально.

Таким образом, в работе были рассмотрены основные аспекты инициализации как операции для работы с объектами PHP и продемонстрированы процессы связанных с этим.

Библиографический список

1. Ундозерова А.Н., Близнюк О.Н., Плакса Ю.А. Программирование.

- Объектно-ориентированное программирование // Ярославское высшее военное училище противовоздушной обороны министерства обороны РФ. 2015. С. 412. URL: <https://elibrary.ru/item.asp?id=32416822> (Дата обращения: 20.08.2018)
2. Мухаметзянов Р.Р. Обучение объектно-ориентированному программированию // Информатика и образование. 2017. С. 35-39. URL: <https://elibrary.ru/item.asp?id=30504077> (Дата обращения: 20.08.2018)
 3. PHP и инициализация очень тяжелых объектов URL: <https://toster.ru/q/71296> (Дата обращения: 20.08.2018)
 4. Классы и объекты PHP, основы URL: http://dnzl.ru/view_post.php%3Fid%3D294 (Дата обращения: 20.08.2018)
 5. Объекты PHP URL: http://bourabai.kz/php/php_manual/language.types.object.html (Дата обращения: 20.08.2018)