УДК 004.624

Наследование как особенность объектно-ориентированного программирования на языке PHP

Ересь Артём Владимирович
Приамурский государственный университет имени Шолом-Алейхема
Студент

Аннотация
В данной статье рассмотрена особенность объектно-ориентированного программирования на языке PHP под названием наследование. Будет раскрыт смысл этого понятие и методика для реализации.
Ключевые слова: PHP, объектно-ориентированное программирование, web

Inheritance as feature of object-oriented programming in the PHP language

Yeres Artem Vladimirovich
Sholom-Aleichem Priamursky State University
Student

Abstract
In this article the feature of object-oriented programming in the PHP language under the name inheritance is considered. The sense of it a concept and a technique for realization will be revealed.
Keywords: PHP, object-oriented programming, web

Объектно-ориентированное программирование строит свою деятельность на трех основных особенностях, являющихся эффективными методами для написания различных программ. Одной из таких важных методик является наследование при формировании классов. Этот метод отражает связь классовых элементов, где на основе одного строятся все остальные.

Целью данной работы является рассмотрение смысла понятия наследования в объектно-ориентированном программировании на PHP.

Тема данной работы популярна в сфере научно-исследовательской деятельности. В статье Р.М. Магомедова поднимаются вопросы перспектив развития объектно-ориентированного программирования относительно обучения в школах. Предлагаются пути решения проблем сложного перехода к данному типу программирования [1]. Автор А.А. Рафанович в своей работе касается темы разработки экспертных систем с использованием всех особенностей объектно-ориентированного программирования [2]. Исследователи Н.Р. Ахметов и А.А. Макаров рассмотрели вопросы автоматизации с использованием объектно-ориентированных методов разработким программного обеспечения [3]. Обо всех принципах объектно-

ориентированного программирования повествует ресурс сети Интернет [4]. Подробно останавливает на наследовании классов следующий источник [5].

В объектно-ориентированном программировании существует три основных принципа: инкапсуляция, полиморфизм и наследование. Рассмотрим, что же такое наследование и как правильно с ним работать.

Наследование в объектно-ориентированном программировании на PHP –создание всех подклассов на основе одного главного класса, именуемого родительским. Итогом этого процесса можно считать доступ подклассов к параметрам главного, что является удобным свойством при условии необходимости обобщения функций работы.

Рассмотрим пример, когда стоит цель в разработке системы состоящей из трех категорий – товар, пользователи, информация. Для каждой из них нужно создать класс для правильного отображения на пользовательский экран

Создадим первый класс, код которого размещен внизу (Рис. 1).

```php
<?php

class User {

 public $header;
 public $footer;
 public $content;

 public $db;

 public function __construct() {
 $this->db = new mysqli('localhost','root','','my_db');
$this->footer = "Footer Blog";
$this->header = "Header Blog";

 }

 public function getHeader() {

 return $this->header;
 }

 public function getFooter() {
 return $this->footer;
 }

 public function render_body() {

 $statti = $this->get_content();
 $str = '';
 foreach($statti as $item) {
 $str .= "<div>";
 $str .= '<h3>'.$item['title'].'</h3>';
 $str .= '<p>'.$item['text']."</p>";
 $str .= "</div>";
 }
 return $str;
 }

 public function get_content() {
 $query = "SELECT * FROM `users` WHERE active == '1'";
 $this->content = $this->db->query($query)->fetch_all(MYSQLI_ASSOC);

 return $this->content;
 }

 public function show() {
 return $this->getHeader().$this->render_body().$this->getFooter();
 }
}

$obj = new Blog();
echo $obj->show();
```

Рис. 1. Первый класс

На рисунке мы видим простой запрос с использованием метода «get content», а конструктор производит связь с базой данный на сервере, которая и дает информацию. Шаблоном является «render body», далее команда «show», объединяет все и выводит на пользовательский экран.

Создание следующих классов будет выглядеть аналогично, и код по своему строению идентичен за исключение лишь смены данных запроса.

```php
<?php


class User {

 public $header;
 public $footer;
 public $content;

 public $db;

 public function __construct() {
 $this->db = new mysqli('localhost','root','','my_db');
$this->footer = "Footer Users";
$this->header = "Header Users";

 }

 public function getHeader() {
 return $this->header;
 }

 public function getFooter() {
 return $this->footer;
 }

 public function render_body() {

 $statti = $this->get_content();
 $str = '';
 foreach($statti as $item) {
 $str .= "<div>";
 $str .= '<h3>'.$item['title'].'</h3>';
 $str .= '<p>'.$item['text']."</p>";
 $str .= "</div>";
 }
 return $str;
 }

 public function get_content() {
 $query = "SELECT * FROM `users` WHERE active == '1'";
 $this->content = $this->db->query($query)->fetch_all(MYSQLI_ASSOC);

 return $this->content;
 }

 public function show() {
 return $this->getHeader().$this->render_body().$this->getFooter();
 }
}

$obj = new Blog();
echo $obj->show();
```

Рис. 2. Второй класс

При сравнении рисунков выше можно заметить, что они практически продублированы одна в другую, что является не очень правильным с точки зрения разработки. Поэтому мы можем использовать метод наследования и в один главный родительский класс внести общие функции, а в подклассах указать только специфические.

```php
<?php

class Page {
 public function __construct() {
 $this->db = new mysqli('localhost','root','','my_db');
 }

 public function getHeader() {
 return $this->header;
 }

 public function getFooter() {
 return $this->footer;
 }

 public function render_body() {

 $statti = $this->get_content();
 $str = '';
 foreach($statti as $item) {
 $str .= "<div>";
 $str .= '<h3>'.$item['title'].'</h3>';
 $str .= '<p>'.$item['text']."</p>";
 $str .= "</div>";
 }
 return $str;
 }

 public function show() {
 return $this->getHeader().$this->render_body().$this->getFooter();
 }
}

class Blog extends Page {

 public function __construct() {
 parent::__construct();
 $this->header = "Header Blog";
 $this->footer = "Footer Blog";
 }

 public function get_content() {
 $query = "SELECT * FROM `statti`";
 $this->content = $this->db->query($query)->fetch_all(MYSQLI_ASSOC);

 return $this->content;
 }

}

class User extends Page {

 public function __construct() {
 parent::__construct();
 $this->header = "Header User";
 $this->footer = "Footer User";
 }

 public function get_content() {
 $query = "SELECT * FROM `user` WHERE active = '1'";
 $this->content = $this->db->query($query)->fetch_all(MYSQLI_ASSOC);

 return $this->content;
 }

}

$obj = new Blog();
echo $obj->show();
```

Рис. 3. Наследование

На рисунке 3 в созданном классе мы видим вынесение всех обобщенных функций, а остальная информация о подклассах при их создании записывается как «extends» при их записи. В примере описывает как родительский класс «Page» дает информацию подклассам «Blog» и «User». Этот процесс демонстрирует работу метода наследования в объектно-ориентированном программировании.

Таким образом, в работе был рассмотрен смысл понятия наследования и показаны методики создания классов в объектно-ориентированном программировании на PHP.

Библиографический список

1. Магомедов Р.М. Объектно-ориентированное программирование – инновационный путь в программировании в школе // Известия Чеченского Государственного Педагогического Института. 2009. №2. С. 108-113. URL: https://elibrary.ru/item.asp?id=19129300 (Дата обращения: 22.01.2019)
2. Рафанович А.А. Разработка экспертных систем в объектно-ориентированном программировании // Вектор науки Тольяттинского Государственного Университета. 2013. №2. С. 54-58. URL: https://elibrary.ru/item.asp?id=20417884 (Дата обращения: 22.01.2019)
3. Ахметов Н.Р., Макаров А.А. Объектно-ориентированные расширения в программировании систем автоматизации // Молодой ученый. 2015. №13. С. 96-100. URL: https://elibrary.ru/item.asp?id=23735662 (Дата обращения: 22.01.2019)
4. Основные принципы ООП: инкапсуляция, наследование, полиморфизм URL:http://gosit.wikia.com/wiki/Основные_принципы_ООП:_инкапсуляция,_наследование,_полиморфизм (Дата обращения: 22.01.2019)
5. Наследование в ООП URL: https://studfiles.net/preview/1577658/page:5/ (Дата обращения: 22.01.2019)