

Инкапсуляция как особенность объектно-ориентированного программирования на языке PHP

Ересь Артём Владимирович

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

В данной статье рассмотрена такая важная особенность объектно-ориентированного программирования на языке PHP как инкапсуляция. Будет представлен смысл этого понятия и методы работы с ним.

Ключевые слова: PHP, данные, объектно-ориентированное программирование, защита

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

Yeres Artem Vladimirovich

Sholom-Aleichem Priamursky State University

Student

Abstract

In this article such important feature of object-oriented programming in the PHP language as encapsulation is considered. The sense of this concept and methods of work with it will be presented.

Keywords: PHP, data, object-oriented programming, protection

Объектно-ориентированное программирование подразумевают наличие определенных особенностей при написании исходного кода программы. Одним из важных считается инкапсуляция. Этот метод помогает существенно упростить манипуляции с кодом и отражает идеи по защите информации от стороннего доступа.

Целью данной работы является рассмотрение смысла понятия инкапсуляции и правил для использования в объектно-ориентированном программировании на PHP.

Тема данной работы популярна в сфере научно-исследовательской деятельности. Авторы И.Т. Степаненко и Е.В. Степаненко посвятили свою статью подробному анализу переходного периода от традиционного к объектно-ориентированному программированию. Были выделены стадии, предпосылки и причины этих изменений [1]. Манипуляции с объектами в объектно-ориентированном программировании раскрыла Л.А. Залогова. Автор дает объяснение процессам, происходящим между объектами и приводит наглядные примеры для читателей [2]. Исследователь А.А.

Рудакова раскрывает тему нашей статьи с точки зрения образовательного процесса в высших учебных заведениях, говорит о сложности обучения объектно-ориентированным методам программирования и классифицирует проблемы в этой сфере в современное время [3]. В интернет-источнике рассказывается об объектно-ориентированном программировании, выделяются основные принципы его работы [4]. Подробно останавливается на изучении инкапсуляции и ее особенностях следующий ресурс сети [5].

В начале работы дадим определение понятию инкапсуляции в объектно-ориентированном программировании.

Инкапсуляция – системное свойство имеющее в своем функционале возможность соединения данных и методологий для манипуляций с ними в классах, с целью скрыть от пользователя детали работы.

Далее выделим следующие возможные классовые элементы по типу доступа:

1. Открытые
2. Защищенные
3. Закрытые

Получение доступа ко второму типу возможно лишь методом обращения к его методам и параметрам используя класс, к которому они принадлежат.

```
1 <?php
2
3
4 class Page {
5     protected $title = 'Index Page';
6
7     public function getTitle() {
8         return $this->title;
9     }
10 }
11
12 $index = new Page();
13 //приведет к ошибке
14 echo $index->title;
```

Рис. 1. Защищенный тип

Это означает, что при попытке доступа воспользовавшись параметром title через закрытый спецификатор напрямую к объекту, пользователь получит от системы отказ и увидит на экране ошибку. Поэтому можно считать метод для открытого свойства с целью получения значения необходимого свойства.

```
1 <?php
2
3 class Page {
4     protected $title = 'Index Page';
5
6     public function getTitle() {
7         return $this->title;
8     }
9
10 }
11
12 $index = new Page();
13 echo $index->getTitle();
```

Рис. 2. Закрытый тип

Аналогично можно выделить способ для редактирования значений необходимого свойства.

```
1 <?php
2
3 class Page {
4     protected $title = 'Index Page';
5
6     public function getTitle() {
7         return $this->title;
8     }
9
10    public function setTitle($title) {
11        $this->title = $title;
12    }
13
14 }
15
16 $index = new Page();
17 $index->setTitle("New String");
18
19 echo $index->getTitle();
```

Рис. 3. Метод для редактирования

Стоит отметить, что получить доступ защищенным параметрам можно еще в наследующих классах.

```
1 class Page {
2     protected $title = 'Index Page';
3
4 }
5
6 class IndexPage extends Page {
7     public function getTitle() {
8         return $this->title;
9     }
10
11    public function setTitle($title) {
12        $this->title = $title;
13    }
14 }
15
16 $index = new Page();
17 $index->setTitle("New String");
18
19 echo $index->getTitle();
```

Рис. 4. Получение доступа

Далее для понимания рассмотрим пример ценообразования в интернет-магазинах. Цена здесь формируется из минимальной заложенной системно стоимостью и налоговой ставкой. Реализуем класс:

```
1 class Tovar {
2     public $nalog;
3     public $price;
4
5     public function setNalog($nalog) {
6         $this->nalog = $nalog;
7     }
8
9     public function getPriceTovar() {
10        return $this->price + $this->nalog;
11    }
12 }
```

Рис.5. Класс магазина

Созданный класс можно обозначить как шаблон: создал его один, а использовать могут многие. И в результате можно получить многофункциональное приложение высокого уровня. За доступ здесь отвечает метод в 9 строке кода, и он формирует итоговую стоимость. Значит для выдачи этой стоимости следует:

```
1 $computer = new Tovar();
2 $computer->setNalog(100);
3
4 echo $computer->getPriceTovar();
```

Рис.6. Получение стоимости

Однако в работе есть нюанс: так как спецификатор используется открытый, то система напрямую будет обращаться к цене и налог не будет учитываться. Для корректной выдачи стоимости нужно установить доступ защищенного вида, что исключит возможность неверных обращений. Но здесь снова появляется проблема в виде запрета реализации подклассов.

Следовательно, конечным для нашего примера будет вариант закрытого доступа. Инкапсуляция же выражается здесь в виде сокрытия частей кода используя спецификаторы для доступа к свойствам.

Таким образом, в работе был рассмотрен смысл понятия инкапсуляции и показаны правила для использования в объектно-ориентированном программировании на PHP.

Библиографический список

1. Степаненко И.Т., Степаненко Е.В. Переход от традиционного к объектно-ориентированному программированию // Современные наукоемкие технологии. 2008. №1. С. 26. URL: <https://elibrary.ru/item.asp?id=9926657> (Дата обращения: 10.01.2019)
2. Залогова Л.А. Взаимодействие объектов в объектно-ориентированном программировании // Вестник Пермского Университета. Серия:

- Математика. Механика. Информатика. 2016. №3. С. 94-102. URL: <https://elibrary.ru/item.asp?id=27370711> (Дата обращения: 10.01.2019)
3. Рудакова А.А. Объектно-ориентированный подход и объектно-ориентированные языки: проблемы изучения в вузе // Объектные системы. 2011. №3. С. 32-35. URL: <https://elibrary.ru/item.asp?id=22901330> (Дата обращения: 10.01.2019)
 4. Основные принципы ООП: инкапсуляция, наследование, полиморфизм URL:http://gosit.wikia.com/wiki/Основные_принципы_ООП:_инкапсуляция,_наследование,_полиморфизм (Дата обращения: 10.01.2019)
 5. Инкапсуляция в программировании URL: <http://fb.ru/article/225829/inkapsulyatsiya---eto-cto-takoe-inkapsulyatsiya-v-programmirovanii> (Дата обращения: 10.01.2019)