

Реализация паттерна проектирования Singleton на PHP

Круглик Роман Игоревич

Приамурский государственный университет им. Шолом-Алейхема

Студент

Аннотация

В статье реализуется один из самых популярных паттернов проектирования singleton на языке программирования PHP. В качестве примера реализовано подключение и вывод из БД данных.

Ключевые слова: Singleton, проектирование, паттерн.

Implementation engineering pattern Singleton on PHP

Kruglik Roman Igorevich

Sholom-Aleichem Priamursky State University

Student

Abstract

In article implements one of the most popular singleton engineering patterns on the PHP programming language. As an example, the connection and output from the database are implemented.

Keywords: Singleton, engineering, pattern.

Singleton (Синглтон, одиночка) относится к классу порождающих паттернов. Он используется для создания всего одного экземпляра класса, и гарантирует, что во время работы программы не появиться второй (см. рис. 1).

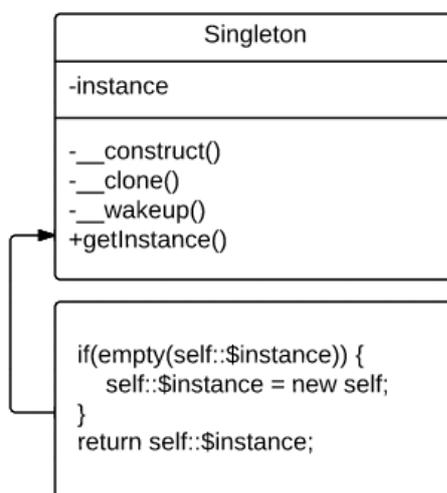


Рисунок 1. Модель паттерна

Преимущества использования, паттерна Singleton заключается в возможности иметь общую точку доступа к внешнему ресурсу. Кроме того, он упрощает инициализацию, проверку состояния, передачу контекста в приложении и управление приложением.

Singleton стоит использовать при подключении к БД, так же когда нужен единственный экземпляр класса, инициализируемый из любой части приложения.

Основными недостатками Singleton является то, что паттерн нарушает принцип единственной ответственности, затрудняет тестирование и имеет скрытые зависимости.

Исследования в области применения различных паттернов проектирования не заканчиваются и по сей день. В статье К.А. Жерденко [1] рассмотрены современные подходы к разработке клиентской части веб-приложений(MVC). С.М. Гардейчик, А.И. Шербаф [2] приводят краткое описание PHP-фреймворка Laravel, реализующего шаблон MVC. В статье О.Н. Симонова [3] рассказывается о шаблонах проектирования MVC как эффективное средство построения архитектуры программной системы. Р.И. Ибраимов [4] создаёт тесты для spring mvc контроллеров. А. Майоров [5] представлен один из вариантов реализации Singleton. В статье П.П. Олейник [6] рассмотрена реализация шаблона проектирования Singleton, представляющая собой атрибут, который можно использовать в языках платформы Net Framework. Проведен анализ существующих реализаций данного паттерна на языке C#, выявлены его достоинства и недостатки. В.К. Хорошкин описывает использование паттерна singleton в языках программирования C++, C#, Java.

В данной статье показана работа паттерна на примере подключения к БД. Рассмотрим главный файл (см. рис. 2).

```
<?php
require_once ('Singleton.php');
require_once ('Db.php');
$db = new Db();
echo '<pre>';
    var_dump($db->getSQL());
echo '</pre>';
```

Рисунок 2. Файл index

В начале файла идёт подключение 2 классов, далее создаётся экземпляр класса с выводом данных из БД. Перейдём к первому классу Singleton.php. (см. рис. 3).

```
<?php
class Singleton
{
    private static $db;
    private function __construct() {}
    private function __clone() {}
    private function __wakeup() {}

    public static function singletonAction()
    {
        if (self::$db == null)
        {
            self::$db = new PDO( dsn: 'mysql:host=localhost;dbname=person', username: 'root', passwd: '' );
        }
        return self::$db;
    }
}
```

Рисунок 3. Класс Singleton

Создаётся статичная приватная переменная. Далее объявляется `construct` для защиты от вызова через `new`, `clone` от клонирования и `wakeup` от `unserialize`. Потом идёт метод, в котором если переменная пуста происходит выполнение подключение к БД, иначе просто возвращается предыдущий результат (см. рис. 4).

```
<?php
class Db
{
    private $db;
    public function __construct()
    {
        $this->db = Singleton::singletonAction();
    }
    public function getSQL()
    {
        $sql = $this->db->query( statement: "SELECT * FROM person" );
        return $sql->fetchAll( fetch_style: PDO::FETCH_ASSOC );
    }
}
```

Рисунок 4. Класс Db

Создаётся приватная переменная. В конструкторе ей приравнивается вызов подключения к БД. Далее метод `getSQL`, в котором переменной `sql` даётся массив через запрос с БД и возвращается преобразованный ассоциативный массив. Посмотрим на результат (см. рис. 5).

```
array(4) {
  [0]=>
  array(2) {
    ["id"]=>
    string(1) "1"
    ["name"]=>
    string(8) "Макс"
  }
  [1]=>
  array(2) {
    ["id"]=>
    string(1) "2"
    ["name"]=>
    string(8) "Вася"
  }
  [2]=>
  array(2) {
    ["id"]=>
    string(1) "3"
    ["name"]=>
    string(14) "Алексей"
  }
  [3]=>
  array(2) {
    ["id"]=>
    string(1) "4"
    ["name"]=>
    string(10) "Мария"
  }
}
```

Рисунок 5. Результат

В результате на экране массив данных из БД. Данный шаблон очень популярен среди разработчиков и может быть использовать при решении различного рода задач.

Библиографический список

1. Жерденко К.А. Применение mvc-паттерна во front-end разработке веб-программ // Синергия Наук. 2018. № 22. С. 653-657.
2. Гардейчик С.М., Шербаф А.И. РНР-фреймворк laravel с использованием архитектурной модели MVC // Перспективные направления развития отечественных информационных технологий Материалы III межрегиональной научно-практической конференции. Научный редактор Б.В. Соколов. 2017. С. 133-135.
3. Симонова О.Н. Шаблон проектирования mvc как эффективное средство построения архитектуры программной системы // Студенческий научный

- форум -2014 VI Международная студенческая электронная научная конференция: Электронное издание. 2014.
4. Ибраимов Р.И., Джемалетдинов А.Б., Шевченко А.А. Spring boot: создание тестов для spring mvc контроллеров // Информационно-компьютерные технологии в экономике, образовании и социальной сфере. 2018. № 4 (18). С. 104-111.
 5. Майоров А. Варианты реализации паттерна singleton // Системный администратор. 2013. № 1-2 (122-123). С. 92-95.
 6. Олейник П.П. Декларативный подход к реализации шаблона проектирования Singleton // Информационные технологии и вычислительные системы. 2006. № 4. С. 45-51.
 7. Хорошхин В.К. Использование паттерна singleton в языках программирования c++, c#, java // Труды Университета. 2011. № 1. С. 82-84.