

PHP

- 1 PHP简介
- 2 PHP基本语法
- 3 PHP流程控制
- 4 PHP函数
- 5 PHP字符串操作
- 6 PHP数组
- 7 PHP时间和日期
- 8 PHP面向对象
 - 8.1 PHP面向对象简介
 - 8.2 PHP class: 定义类
 - 8.3 PHP实例化对象
 - 8.4 PHP构造函数
 - 8.5 PHP析构函数
 - 8.6 PHP继承
 - 8.7 PHP \$this
 - 8.8 PHP命名空间
 - 8.9 PHP设计模式
 - 8.10 PHP魔术方法
 - 8.11 PHP抽象类和抽象方法
 - 8.12 PHP interface: 接口
 - 8.13 PHP最终类和最终方法
 - 8.14 PHP clone关键字

[🏠 首页](#) > [PHP](#) > [PHP面向对象](#)

阅读: 2,643

PHP设计模式之单例模式与工厂模式



小白入职大厂完全攻略，很硬很肝
学习路线 / 笔试面试 / 升职加薪 / 跳槽技巧

[猛击查看详情](#)

C语言中文网推出辅导班啦，包括「C语言辅导班、C++辅导班、算法/数据结构辅导班」，全部都是**一对一教学**：一对一辅导 + 一对一答疑 + 布置作业 + 项目实践 + 永久学习。QQ在线，随时响应！

设计模式不是一套具体的语言框架，而是一种行之有效的编码规范，是前人经过反复使用并总结出来的编写代码的经验。使用设计模式的目的是为了提高代码可重用性，让代码更容易被他人理解，同时保证代码可靠性。合理使用设计模式有助于我们更加深入地理解面向对象思维。

其实，大家在编码的过程中肯定也用到了一些与这些设计模式类似的编码方法，只是不知道这些方法可以被抽象到“模式”这么一个高度。

本节我们就来简单地了解一下在 PHP 编码过程中经常会用到的一些设计模式。

1、工厂模式

工厂模式是一种专门用来创建其它对象的类（称为“工厂类”），根据传递参数的不同，来创建不同类的对象。我们可以使用工厂类创建对象，而不是直接使用 new。

工厂类中至少有一个公共的静态方法（称为“工厂方法”），静态方法接受一个参数，根据这个参数来创建不同类的对象。

8.15 PHP判断对象是否属于某个类

8.16 PHP自动加载机制

9 正则表达式

10 PHP会话控制

11 PHP错误和异常处理

12 MySQL数据库的基础操作

13 PHP文件目录操作

14 PHP图像处理

【示例】定义加、减、乘、除四个类，并通过工厂类调用它们。

```
01. <?php
02.     // 加法类
03.     class Add {
04.         public function getValue($num1, $num2){
05.             return $num1 + $num2;
06.         }
07.     }
08.
09.     // 减法类
10.     class Sub {
11.         public function getValue($num1, $num2){
12.             return $num1 - $num2;
13.         }
14.     }
15.
16.     // 乘法类
17.     class Mul {
18.         public function getValue($num1, $num2){
19.             return $num1 * $num2;
20.         }
21.     }
22.
23.     // 除法类
24.     class Div {
25.         public function getValue($num1, $num2){
26.             try {
27.                 if ($num2==0){
28.                     throw new Exception("除数不能为0");
29.                 }else {
```

```
30.         return $num1/$num2;
31.     }
32.     }catch (Exception $e){
33.         echo "错误信息: ".$e->getMessage();
34.     }
35. }
36. }
37.
38. // 工程类，主要用来创建对象
39. // 功能：根据输入的运算符号，工厂就能实例化出合适的对象
40. class Factory {
41.     public static function createObj($operate){
42.         switch ($operate){
43.             case '+':
44.                 return new Add();
45.                 break;
46.             case '-':
47.                 return new Sub();
48.                 break;
49.             case '*':
50.                 return new Mul();
51.                 break;
52.             case '/':
53.                 return new Div();
54.                 break;
55.         }
56.     }
57. }
58. $test = Factory::createObj('/');
59. $result = $test->getValue(23,3);
60. echo $result;
```



```
61.  ?>
```

运行结果如下：

```
7.66666666666667
```

2、单例模式

单例模式也叫单子模式，是一种常用的软件设计模式。在应用这个模式时，可以确保一个类只能创建一个对象，这么做可以极大节省内存空间，有利于我们协调系统的整体行为。

使用单例模式创建的类（“单例类”）不能再其它类中直接实例化，只能被其自身实例化。它不会创建实例副本，而是会向单例类内部存储的实例返回一个引用。

单例模式一个主要应用场合就是应用程序与数据库打交道的场景，在一个应用中会存在大量的数据库操作，针对数据库句柄连接数据库的行为，使用单例模式可以避免大量的 new 操作，因为每一次 new 操作都会消耗系统和内存的资源。

实现单例模式的思路（三私一公）：

- 私有的静态的对象实例；
- 私有的构造方法，在类外不能使用 new 创建对象；
- 私有的克隆方法，在类外不能使用 clone 克隆对象；
- 公共的静态的创建对象实例的方法。

【示例】使用单例模式创建一个数据库连接类。

```
01.  <?php
02.      class Mysql{
03.          //该属性用来保存实例
04.          private static $conn;
05.          // 私有的构造函数，防止在类外创建对象
06.          private function __construct() {
```



```
07.         self::$conn = mysqli_connect('localhost','root','root');
08.     }
09.     // 公共的创建对象的方法，如果不存在一个这个类的实例属性，就创建一个，否则就取这个
    实例属性。
10.     public static function getInstance(){
11.         if(!(self::$conn instanceof self)){
12.             self::$conn = new self;
13.         }
14.         return self::$conn;
15.     }
16.     // 私有的克隆方法，防止在类外 clone 对象
17.     public function __clone() {}
18.     //防止反序列化后创建对象
19.     private function __wakeup(){
20.         trigger_error('Unserialized is not allowed !');
21.     }
22. }
23. //只能这样取得实例，不能 new 和 clone
24. $obj1 = Mysql::getInstance();
25. $obj2 = Mysql::getInstance();
26. $obj3 = Mysql::getInstance();
27. var_dump($obj1, $obj2, $obj3);
28. ?>
```

运行结果如下：

```
object(Mysql)#1 (0) {
}
object(Mysql)#1 (0) {
}
object(Mysql)#1 (0) {
}
```

通过运行结果可以看出，我们多次调用 `getInstance()` 方法并没有创建多个对象，所以说使用单例模式可以保证一个类只能创建一个对象，不能创建第二个对象。

关注微信公众号「站长严长生」，在手机上阅读所有教程，随时随地都能学习。本公众号由[C语言中文网站](#)长运营，每日更新，坚持原创，敢说真话，凡事有态度。



微信扫描二维码关注公众号

优秀文章

C语言杨辉三角（两种方法）

Java Swing文本编辑器的实现

汇编语言寄存器模式指令简述

Redis和数据库的结合

Python Django框架是什么？

Java线程的优先级和执行顺序

JSP获取并显示字符串长度

FilterChain（过滤器链）详解

Python `rjust()`字符串右对齐方法详解

不使用三大范式会对数据库造成什么影响？

精美而实用的网站，分享优质编程教程，帮助有志青年。千锤百炼，只为大作；精益求精，处处斟酌；这种教程，看一眼就倾心。

[关于网站](#) | [关于站长](#) | [如何完成一部教程](#) | [联系我们](#) | [网站地图](#)

Copyright ©2012-2022 biancheng.net, 陕ICP备15000209号



biancheng.net

