

工厂模式和单例模式

1. 设计模式

定义：解决一些问题的“常规性做法”，是一种较好的经验总结。面对不同的问题，可能有不同的解决方法。此时可能有不同的设计模式。

2. 工厂模式

实际项目中，总需要实例化很多很多类，一一取到对象。如果这些对象为同一类或有所关联。则可考虑设计一个工厂，按需生产指定对象。比如，通过指定一个类名，就可以得到该对象。

```
Class factory{
    Static function getInstance($className) {
        If(file_exists(“./class/”.$className.”.class.php”)){
            Return new $className();
        }else{
            die();
        }
    }
}
```

3. 单例模式

对于某些类，使用时希望程序运行到结束，只使用同一个对象，就可以完成所有的任务，则可以考虑使用单例模式

单例：只允许其创建出一个对象，即使多次创建，也只能得到同一个对象

```
$obj1 = new A();
```

实现：

```
class B{
    public $v1 = 10;
    private static $instance;
    private function __construct(){
    }
    private function __clone(){
    }
    public static function getInstance(){
        if(!isset(self::$instance)){
            self::$instance = new self();
        }
        return self::$instance;
    }
}
```

通过1-4步实现一个单例，其中2、3是为了禁止调用new、clone方法创建对象

```
$b = B::getInstance();
var_dump($b);

$c = new B();
$c = clone $b;
var_dump($c);
```

尝试使用new、clone创建对象，报错啦