# **[设计模式--原型模式](https://segmentfault.com/a/1190000022960229)**

IMG_256

原型模式，通俗点来说就是允许我们复制一个可定制化的对象，而不用去考虑对象的实际创建过程。举个栗子：小明在最大的同性交友论坛看到某个开源的项目很适合自己新项目，然后就把该项目下载下来（可定制化的对象）拿过来，然后对里面配置进行修改（对对象进行定制化操作），开始做开发了。（1.这个例子够通俗吧；2.最大的同性交友论坛是github~）这个就是原型模式。

我们来看下代码：  
我们首先需要有个原型：

**class** **Art** **implements** **Prototype**{

**private** $\_name;

**public** **function** **\_\_construct**($name)

{

**$this**->\_name = $name;

}

*/\*\**

*\* 设置名字*

*\**

*\* @param $name*

*\*/*

**public** **function** **setName**($name)

{

**$this**->\_name = $name;

}

*/\*\**

*\* 获取名字*

*\**

*\* @return mixed*

*\*/*

**public** **function** **getName**()

{

**return** **$this**->\_name;

}

*/\*\**

*\* 浅拷贝*

*\*/*

**public** **function** **shallowCopy**()

{

**return** **clone** **$this**;

}

*/\*\**

*\* 深拷贝*

*\*/*

**public** **function** **deepCopy**()

{

*//对对象进行序列化*

$serialize\_obj = serialize(**$this**);

*//进行反序列化*

$clone\_obj = unserialize($serialize\_obj);

**return** $clone\_obj;

}

}

这个原型允许浅拷贝和深拷贝（这个要记住，我们待会讲）。

我们调用过程：

**class** **Client**{

*/\*\**

*\* 浅拷贝*

*\*/*

**public** **static** **function** **shallow**()

{

$cd = **new** CD();

$cd->content = "music";

$music = **new** Art($cd);

$movie = $music->shallowCopy();

p('$music->getName()',$music->getName());

p('$movie->getName()',$movie->getName());

$cd->content = "movie";

p('$music->getName()',$music->getName());

p('$movie->getName()',$movie->getName());

}

*/\*\**

*\* 深拷贝*

*\*/*

**public** **static** **function** **deep**()

{

$cd = **new** CD();

$cd->content = "music";

$music = **new** Art($cd);

$movie = $music->deepCopy();

p('$music->getName()',$music->getName());

p('$movie->getName()',$movie->getName());

$cd->content = "movie";

p('$music->getName()',$music->getName());

p('$movie->getName()',$movie->getName());

}

}

这个里面有个CD的类，这个类不是原型模式必备的，我放在这里只是为了辅助大家理解什么是浅拷贝和深拷贝（好人一生平安~）。

CD类如下：

**class** **CD**{

**public** $content;

}

执行结果：

\*\*\*\*\*\*\*

$music->getName()=**object**(CD)**#2 (1) {**

[**"content"**]=>

**string**(5) "music"

}

\*\*\*\*\*\*\*

$movie->getName()=**object**(CD)**#2 (1) {**

[**"content"**]=>

**string**(5) "music"

}

\*\*\*\*\*\*\*

$music->getName()=**object**(CD)**#2 (1) {**

[**"content"**]=>

**string**(5) "movie"

}

\*\*\*\*\*\*\*

$movie->getName()=**object**(CD)**#2 (1) {**

[**"content"**]=>

**string**(5) "movie"

}

\*\*\*\*\*\*\*

$music->getName()=**object**(CD)**#4 (1) {**

[**"content"**]=>

**string**(5) "music"

}

\*\*\*\*\*\*\*

$movie->getName()=**object**(CD)**#5 (1) {**

[**"content"**]=>

**string**(5) "music"

}

\*\*\*\*\*\*\*

$music->getName()=**object**(CD)**#4 (1) {**

[**"content"**]=>

**string**(5) "movie"

}

\*\*\*\*\*\*\*

$movie->getName()=**object**(CD)**#5 (1) {**

[**"content"**]=>

**string**(5) "music"

}

这个例子中的movie是music对象的拷贝，在创建movie的过程中，不用考虑movie对象的创建过程，只需要使用movie对象即可。

同时，我们也注意到结果中，和music->getName()的不同，这个里面涉及另外要注意的问题：浅拷贝和深拷贝；

****浅拷贝：****被拷贝对象的所有变量都含有与原对象相同的值，并且对其他对象的引用（对象的地址）也是一致的，换句话说：浅拷贝只负责当前对象实例，对引用的对象不做拷贝。所以我们修改原来music对象的时候，也会修改movie对象中相应的值。

****深拷贝****：被拷贝对象的所有的变量都含有与原来对象相同的值，但是那些引用其他对象的变量将指向一个被拷贝的新对象，而不再是原来那些被引用的对象。即深拷贝把要拷贝的对象所引用的对象也拷贝了一次（也叫做****间接拷贝****）。

原型模式是先创建好一个原型对象，然后通过clone原型对象来创建新的对象。这种模式特别适合复杂对象的创建，如果复杂对象每次new就会消耗很大，原型模式仅需内存拷贝即可。

phper 在进阶的时候总会遇到一些问题和瓶颈，业务代码写多了没有方向感，不知道该从那 里入手去提升，对此我整理了一些资料，包括但不限于：分布式架构、高可扩展、高性能、 高并发、服务器性能调优、TP，laravel，Redis，Swoole、Swoft、Kafka、Mysql 优化、Docker、微服务、Nginx 等多个知识点高级进阶干货需要的可以免费分享给大 家 需要可以加下我 qq:1047403189 或者微信：PHPty123获取哦

这些的话也在下面的直播课中讲解过很多期，并且在不断更新中，是结合企业的一些应用场景讲解的，能够帮助学员突破思维或者带着实战，来帮助大家掌握学习的电脑 腾讯课堂客户端

手机 腾讯课堂APP

网页版直播地址：https://ke.qq.com/course/328509?tuin=6

2020年 6月23日 晚上20-23点 主讲老师介绍:

# 六星教育-Peter老师

六星PHP学院院长、六星教育合伙人，9年PHP研发经验，曾就职于拓维、新浪等上市公司，经历过大型项目的攻坚与调优，擅长领域：接口服务设计，微服务架构、swoole网络编程、机器学习，熟悉Laravel等框架的优化；对大数据业务、高并发数据优化拥有丰富的经验。