

phpoop:PHP高级OOP技术演示

疯狂代码 <http://CrazyCoder.cn/> <http://CrazyCoder.cn/Php/Article52674.html>

如果你了解基本OOP概念那么我将向你展示更高级技术

序列化(Serializing)

PHP不支持永久对象在OOP中永久对象是可以在多个应用引用中保持状态和功能对象这意味着拥有将对象保存到文件或数据库中能力而且可以在以后装入对象这就是所谓序列化机制PHP 拥有序列化思路方法它可以通过对象进行序列化思路方法可以返回对象串表示然而序列化只保存了对象成员数据而不包括思路方法

在PHP4中如果你将对象序列化到串\$s中然后释放对象接着反序列化对象到\$obj你可以继续使用对象思路方法！我不建议这样去做(a)文档中没有保证这种行为在以后版本中仍然可以使用(b) 这个可能导致种误解在你把个序列化后版本保存到磁盘并退出脚本时当以后运行这个脚本时你不能期待着在反序列化个对象时对象思路方法也会在那里串表示根本就不包括思路方法

总而言之的PHP 进行序列化对于保存对象成员变量非常有用(你也可以将相关和序列化到个文件中)

例子:

```
<?php
$obj = Classfoo;
$str = serialize($obj);
    //保存$str到磁盘上
//几个月以后
//从磁盘中装入str
```

`$obj2 = unserialize($str)` 你恢复了成员数据但是不包括思路方法(根据文档所说)这导致了只能通过类似于使用`$obj2->x`来存取成员变量(你没有别思路方法!)唯办法所以不要在家里试它

有些办法可以解决这个问题我把它留着对这篇简洁文章来说他们太不好我会很高兴地欢迎在PHP后续版本中有全序列化特性

使用类进行数据存储PHP和OOP件非常好事情就是你可以很容易地定义个类来操作某件事情并且无论何时你想用时候都可以相应类假设你有个HTML表单用户可以通过选择产品ID号来选择个产品在数据库中有产品信息你想把产品显示出来显示它价格等等你拥有区别类型产品并且同个动作可能对区别产品具有区别意思例如显示个声音可能意味着播放它但是对于其它种类产品可能意味着显示个存在数据库中图片你可以使用OOP或PHP来减少编码并提高质量

定义个产品类定义它应该有思路方法(例如:显示)然后定义对每种类型产品类从产品类派后出来(SoundItem类ViewableItem类等等)覆盖在产品类中思路方法使它们按你想法动作

根据数据库中每种产品类型(type)字段给类命名个典型产品表可能有(id, type, price, description, 等等字段)...然后在处理脚本中你可以从数据库中取出type值然后例子化个名为type对象:

```
<?php
$obj= $type;
$obj->action;    这是PHP个非常好特性你可以不用考虑对象类型$obj显示思路方法或其它思路方法使用这个技术你不需要修改脚本去增加个新类型对象只是增加个处理它类
```

这个功能很强大只要定义思路方法而不去考虑所有对象类型在区别类中按区别思路方法实现它们然后在主脚本中对任意对象使用它们没有...也不需要两个员只有高兴

现在你同意编程是容易维护是便宜可重用是真吗?

如果你管理组员分配工作就是很简单了每个人可能负责个类型对象和处理它类

可以通过这个技术实现国际化根据用户所选语言字段应用相应类就可以了等等

拷贝和克隆

当你创建个\$obj对象时你可以通过\$obj2=\$obj来拷贝对象新对象是\$obj个拷贝(不是个引用)所以它具有\$obj在当时状态有时候你不想这样你只是想生成个象obj类样个新对象可以通过使用语句来类构造在PHP中也可以通过序列化和个基类来实现但所有其它类都要从基类派生出来

进入危险区域

当你序列化个对象你会得到某种格式串如果你感兴趣你可以调究它其中串中有类名字(太好了!)你可以把它取出来象:

```
<?php
$herring=serialize($obj);
```

```
$vec=explode(':',$herring);  
$nam=str_replace("","",$vec[2]);    所以假设你创建了个"Universe"类并且强制所有类都必须从universe扩  
展你可以在universe 中定义个clone思路方法如下:
```

```
<?php  
function clone {  
$herring=serialize($this);  
$vec=explode(':',$herring);  
$nam=str_replace("","",$vec[2]);  
$ret= $nam;  
$ret;  
}  
}  
//然后  
$obj= Something;  
//从Universe扩展  
$other=$obj->clone;    你所得到的新Something类对象它同使用思路方法构造创建出对象样我不知道这  
个对你是否有用但是Universe类可以知道派生类名字是个好经验想象是唯一限制
```

这个语句就是写入当前时间

2009-1-12 1:47:13

疯狂代码 <http://CrazyCoder.cn/>