第 18 章 面向对象的基础

学习要点：

1.什么是面向对象
2.OOP 的特点
3.关键的 OOP 概念
4.创建 OOP

主讲教师：李炎恢
合作网站：http://www.ibeifeng.com
讲师博客：http://hi.baidu.com/李炎恢

许多语言本身就是面向对象（OOP）的，而 PHP 用了几年才引入了这类功能。面向对象的诞生是开发范型一次的重大改变，编程的注意力重新从应用程序的逻辑回到其数据上来。换句话说，OOP 将焦点从编程的过程性事件转向最终建模的真实实体。这使得应用程序更接近于我们周围的现实世界。

一．什么是面向对象

面向过程

这就好比你是公司的一名员工，今天有个任务要在公司组装一批电脑。那么你就开始采购、讨价还价、运输回公司、开始组装、布线网络、调试机器、完成。也就是说，面向过程就是具体化的实现，细节明确。

面向对象

这就好像你是公司的总裁，你布置给一名员工一个组装一批电脑的任务。完毕。也就是说，面向对象就是抽象化的执行，具体还是由那名员工来完成。而细节方面，总裁不需要去考虑。这样的好处是显而易见的，在有管理高层的公司可以协调作业，而没有管理高层的公司，只有一些具体实现功能的员工，会乱做一团。

二．OOP 的特点

封装

隐藏对象的属性和实现细节，仅对外公开接口,控制在程序中属性的读和修改的访问级别；将抽象得到的数据和行为（或功能）相结合，形成 一个有机的整体，也就是将数据与操作数据的源代码进行有机的结合，形成“类”，其中数据和函数都是类的成员。

继承

继承是从一个基类得到一个或多个类的机制。

继承自另一个类的类被称为该类的子类。这种关系通常用父亲和孩子来比喻。子类将继承父类的特性。这些特性由属性和方法组成。子类可以增加父类之外的新功能，因此子类也被称为父类的“扩展”。

多态

多态是指 OOP 能够根据使用类的上下文来重新定义或改变类的性质或行为，或者说接口的多种不同的实现方式即为多态。把不同的子类对象都当作父类来看，可以屏蔽不同子类对象之间的差异，写出通用的代码，做出通用的编程，以适应需求的不断变化。

三．关键的 OOP 概念

类（class）

类是对某个对象的定义。它包含有关对象动作方式的信息，包括它的名称、方法、属性和事件。实际上它本身并不是对象，因为它不存在于内存中。当引用类的代码运行时，类的一个新的实例，即对象，就在内存中创建了。虽然只有一个类，但能从这个类在内存中创建多个相同类型的对象。

对象（object）

对象是一件事、一个实体、一个名词，可以获得的东西，可以想象有自己的标识的任何东西。对象是类的实例化。一些对象是活的，一些对象不是。

比如这辆汽车、这个人、这间房子、这张桌子、这株植物、这张支票、这件雨衣。 概括来说就是：一切皆对象。

例如：类是对象的抽象定义，说白了，如果这个对象是电脑，类可以创建出许多对象，类可以生成很多电脑，再白一点，类可以当成一个电脑生产厂，可以生成出很多很多台电脑。

字段（filed）

字段是用于描述类的某方面的性质，它与一般的 PHP 变量非常相似，只是有一些细微的差别。

例如：电脑品牌，电脑的型号等特性。

属性（attribute）

通过方法来访问和操作字段，一方面可以保护字段，同时还允许访问公共字段一样访问数据。

例如：获取电脑品牌，设置电脑品牌等操作。

方法（method）

方法与函数非常相似，只不过方法是用来定义类的行为。与函数一样，方法可以接受输入参数，可以向调用者返回一个值。

例如：打开电脑，输入文本，运行程序。

四．创建 OOP

类的创建：

```
class Computer {
    //类的字段(成员)
    //类的方法
}
```

对象的声明：

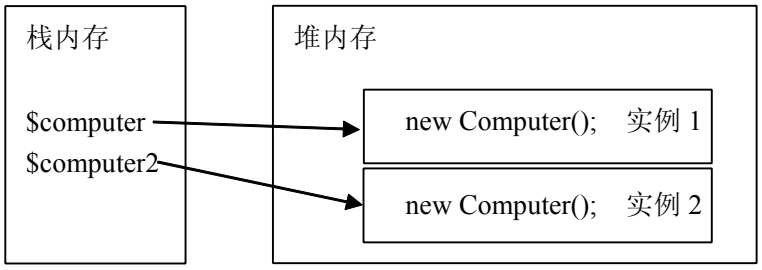
$computer = new Computer();

new 标识符是为了在内测中创建一个对象（实例），而 Computer()就是那个类所生成的实例。

$computer 是一个变量，而且又是生成实例的引用。

有时，你可能需要创建多个对象。

$computer2 = new Computer();



使用 var_dump()函数可以打印变量的相关信息。

字段（成员）的添加：

```
class Computer {
    //类的字段(成员)
    public $_name = '联想120';
    public $_model = 'LX';
}
```

1.声明字段：public $_name = '联想 120'

a) public 是修饰符，表示这是一个公共字段，可以通过外部直接访问。
b) $_name 是变量名。
c) '联想 120'是变量的值。

2.读取字段

```
echo $computer->_name;
```

3.设置字段

```
$computer->_name = 'IBM110';
```

方法的创建

```
class Computer {
    //类的方法
    function run () {
        echo '我成功的运行了！ ';
    }
}
```

执行方法

```
$computer->run();
```

也可以在方法添加一些参数，执行的时候，传入这些参数。

```
class Computer {
    //类的方法
    function run ($_what) {
        echo $_what.'成功的运行了！';
    }
}
$computer = new Computer();
$computer->run('电脑');
```

构造方法

所谓构造方法，也是方法，只不过是一种特殊的方法。而方法名必须和类名一致，并且不需要像普通方法一样，必须通过调用才能执行，只需要实例化即完成调用过程。一般来说，构造方法总是在做一些初始化的工作。

```
class Computer {
    //构造方法
    function Computer() {
        echo '我是构造方法！';
    }

}
new Computer(); //这样即完成了调用
```

在 PHP5 我们可以通过__construct 的内置方法来识别构造方法，而不用再需要和类名相同了。

```
class Computer {
    //构造方法
    function __construct() {
        echo '我是构造方法！';
    }
}
```

相对应构造方法，还有一种内置的方法是析构方法，它的用途在整个类使用完毕都执行。一般可用于清理内存中对象(脚本执行完毕的时候会自动清理)。而如果有脚本执行完毕后并没有清理的，比如数据库数据等，就有必要使用析构方法。

```
class Computer {
    //析构方法
    function __destruct() {
        echo '我是析构方法';
    }
}
```

感谢收看本次教程！

本课程是由北风网(ibeifeng.com)

瓢城 Web 俱乐部(yc60.com)联合提供：

本次主讲老师：李炎恢

我的博客：hi.baidu.com/李炎恢/

我的邮件：yc60.com@gmail.com