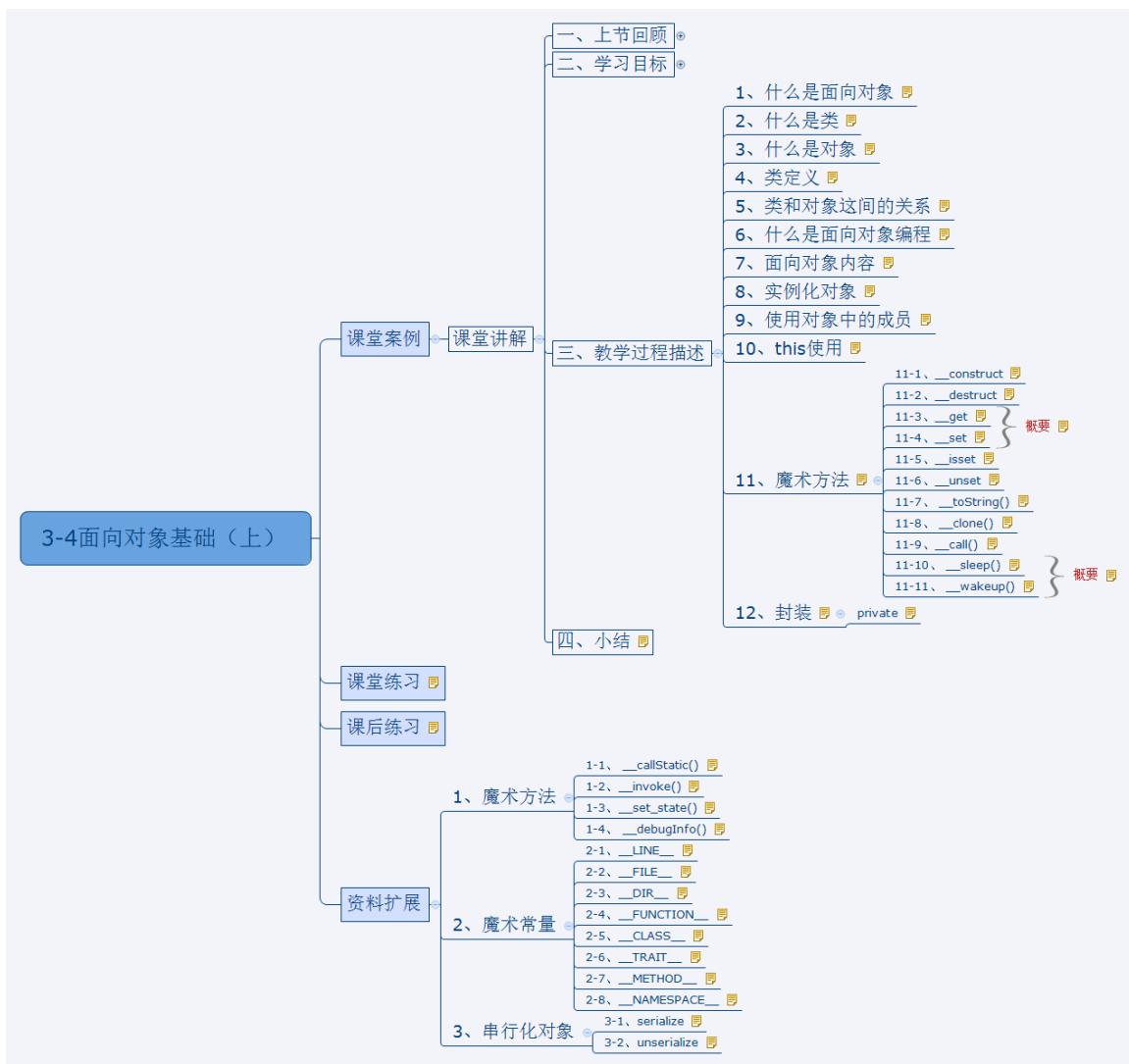


3-4面向对象基础（上）

3-4面向对象基础（上）	1
1. 课堂案例	3
课堂讲解	3
一、上节回顾	3
MySQL-高级应用	3
二、学习目标	3
了解什么是面向对象	3
三、教学过程描述	4
1、什么是面向对象	4
2、什么是类	4
3、什么是对象	4
4、类定义	4
5、类和对象这间的关系	6
6、什么是面向对象编程	7
7、面向对象内容	7
8、实例化对象	8
9、使用对象中的成员	8
10、this使用	11
11、魔术方法	11
12、封装	15
四、小结	16
2. 课堂练习	16
3. 课后练习	16
4. 资料扩展	16
1、魔术方法	16
1-1、__callStatic()	16
1-2、__invoke()	16
1-3、__set_state()	16
1-4、__debugInfo()	16
2、魔术常量	17
2-1、__LINE__	17
2-2、__FILE__	17
2-3、__DIR__	17
2-4、__FUNCTION__	17
2-5、__CLASS__	17

2-6、__TRAIT__	17
2-7、__METHOD__	17
2-8、__NAMESPACE__	17
3、串行化对象	17
3-1、serialize	17
3-2、unserialize	18



1. 课堂案例

课堂讲解

一、上节回顾

MySQL-高级应用

二、学习目标

了解什么是面向对象

三、教学过程描述

1、什么是面向对象

面向对象编程(Object Oriented Programming,OOP,面向对象程序设计)是一种计算机编程架构, OOP的一条基本原则是计算机程序是由单个能够起到子程序作用的单元或对象组合而成, OOP达到了软件工程的三个目标:重用性、灵活性和扩展性。为了实现整体运算, 每个对象都能够接收信息、处理数据和向其它对象发送信息。面向对象一直是软件开发领域内比较热门的话题, 首先, 面向对象符合人类看待事物的一般规律。其次, 采用面向对象方法可以使系统各部分各司其职、各尽所能。为编程人员敞开了一扇大门, 使其编程的代码更简洁、更易于维护, 并且具有更强的可重用性。有人说PHP不是一个真正的面向对象的语言, 这是事实。PHP是一个混合型语言, 你可以使用OOP, 也可以使用传统的过程化编程。然而, 对于大型项目, 你可能需要在PHP中使用纯的OOP去声明类, 而且在你的项目里只用对象和类

三大特性: 封装、继承、多态

2、什么是类

类是具有相同属性和服务的一组对象的集合。它为属于该类的所有对象提供了统一的抽象描述, 其内部包括属性和服务两个主要部分。在面向对象的编程语言中, 类是一个独立的程序单位, 它应该有一个类名并包括属性说明和服务说明两个主要部分。

3、什么是对象

对象是系统中用来描述客观事物的一个实体, 它是构成系统的一个基本单位。一个对象由一组属性和对这组属性进行操作的一组服务组成。从更抽象的角度来说, 对象是问题域或实现域中某些事物的一个抽象, 它反映该事物在系统中需要保存的信息和发挥的作用; 它是一组属性和有权对这些属性进行操作的一组服务的封装体。客观世界是由对象和对象之间的联系组成的

比如 Animal(动物)

是一个抽象类, 我们可以具体到一只狗跟一只羊, 而狗跟羊就是具体的对象, 他们有颜色属性, 可以写, 可以跑等行为状态。

4、类定义

```
class 类名{ }
```

使用一个关键字class和后面加上一个你想要的类名以及加上一对大括号, 这样一个类的结构就定义出来了, 只要在里面写代码就可以了。

比如说, 一个人就是一个对象, 你怎么把一个你看好的人推荐给你们领导呢? 当然是越详细越好了:

首先, 你会介绍这个人姓名、性别、年龄、身高、体重、电话、家庭住址等等。

然后, 你要介绍这个人能做什么, 可以开车, 会说英语, 可以使用电脑等等。

只要你介绍多一点, 别人对这个人就多一点了解, 这就是我们对一个人的描述, 现在我们总结一下, 所有的对象我们用类去描述都是类似的, 从上面人的描述可以看到, 做出一个类来, 从定义的角度分两部分,

第一是从静态上描述, 第二是从动态上描述, 静态上的描述就是我们所说的属性,

像上面我们看到的, 人的姓名、性别、年龄、身高、体重、电话、家庭住址等等。

动态上也就是人的这个对象的功能, 比如这个人可以开车, 会说英语, 可以使用电脑等等, 抽象成程序时, 我们把动态的写成函数或者说是方法, 函数和方法是一样的。所以, 所有类都是从属性和方法这两方面去写, 属性又叫做这个类的成员属性, 方法叫做这个类的成员方法。

```
class 人{
    □□成员属性:姓名、性别、年龄、身高、体重、电话、家庭住址
    □□成员方法:可以开车, 会说英语, 可以使用电脑
}
```

属性:

通过在类定义中使用关键字"

var

"来声明变量, 即创建了类的属性, 虽然在声明成员属性的时候可以给定初值,

但是在声明类的时候给成员属性初始值是没有必要的, 比如说要是把人的姓名赋上“张三”, 那么用这个类实例出几十个人, 这几十个人都叫张三了, 所以没有必要,

我们在实例出对象后给成员属性初始值就可以了。

```
class Person
{
    //下面是人的成员属性
    var $name; //人的名子
    var $sex; //人的性别
    var $age; //人的年龄

    //下面是人的成员方法
    function say() { //这个人可以说话的方法
        echo "这个人在说话";
    }

    function run() { //这个人可以走路的方法
        echo "这个人在走路";
    }
}
```

为了加强你对类的理解, 我们再做一个类,

做一个形状类, 形状的范围广了点, 我们就做个矩形吧, 先分析一下, 想一想从两方面分析, 矩形的属性都有什么? 矩形的功能都有什么?

```

1      class 矩形
2      {
3          //矩形的属性
4          矩形的长;
5          矩形的宽;
6
7          //矩形的方法
8          矩形的周长;
9          矩形的面积;
10     }
1      <?php
2      class Rect
3      {
4          var $kuan;
5          var $gao;
6
7          function zhouChang() {
8              计算矩形的周长;
9          }
10
11         function mianJi() {
12             计算矩形的面积;
13         }
14     }
15     ?>

```

5、类和对象这间的关系

类与对象的关系就如模具和铸件的关系，类的实例化结果就是对象，而对一类对象的抽象就是类。类描述了一组有相同特性(属性)和相同行为(方法)的对象。上面大概就是它们的定义吧，也许你是刚接触面向对象的朋友，

不要被概念的东西搞晕了，给你举个列子吧，如果你去中关村想买几台组装的PC机，到了那里你第一步要干什么，是不是装机的工程师和你坐在一起，按你提供的信息和你一起完成一个装机的配置单呀，这个配置单就可以想像成是类，它就是一张纸，但是它上面记录了你要买的PC机的信息，如果用这个配置单买10台机器，那么这10台机子，都是按这个配置单组成的，所以说这10台机子是一个类型的，也可以说是一类的。那么什么是对象呢，类的实例化结果就是对象，

用这个配置单配置出来(实例化出来)的机子就是对象，是我们可以操作的实体，
10台机子，10个对象。每台机子都是独立的，只能说明他们是同一类的，对其中一个机做任何动作都不会

影响其它9台机器,但是我对类修改,也就是在这个配置单上加一个或少一个配件,那么装出来的9个机器都改变了,这是类和对象的关系(类的实例化结果就是对象)。

6、什么是面向对象编程

如果你想建立一个电脑教室,首先要有一个房间,房间里要有N台电脑,有N个桌子, N个椅子, 白板, 投影机等等, 这些是什么, 刚才咱们说了, 这就是对象, 能看到的一个个的实体, 可以说这个电脑教室的单位就是这一个个的实体对象, 它们共同组成了这个电脑教室, 那么我们是做程序, 这和面向对象有什么关系呢? 开发一个系统程序和建一个电脑教室类似, 你把每个独立的功能模块抽象成类形成对象, 由多个对象组成这个系统, 这些对象之间都能够接收信息、处理数据和向其它对象发送信息等等相互作用。就构成了面向对象的程序。

7、面向对象内容

1、类 – 定义了一件事物的抽象特点。类的定义包含了数据的形式以及对数据的操作。

2、对象 – 是类的实例。

3、成员变量

定义在类内部的变量。该变量的值对外是不可见的, 但是可以通过成员函数访问, 在类被实例化为对象后, 该变量即可称为对象的属性。

成员函数 – 定义在类的内部, 可用于访问对象的数据。

4、继承

继承性是子类自动共享父类数据结构和方法的机制, 这是类之间的一种关系。在定义和实现一个类的时候, 可以在一个已经存在的类的基础之上来进行, 把这个已经存在的类所定义的内容作为自己的内容, 并加入若干新的内容。

5、父类 – 一个类被其他类继承, 可将该类称为父类, 或基类, 或超类。

6、子类 – 一个类继承其他类称为子类, 也可称为派生类。

7、多态

多态性是指相同的操作或函数、过程可作用于多种类型的对象上并获得不同的结果。不同的对象, 收到同一消息可以产生不同的结果, 这种现象称为多态性。

8、重载

简单说, 就是函数或者方法有同样的名称, 但是参数列表不相同的情形, 这样的同名不同参数的函数或者方法之间, 互相称之为重载函数或者方法。

9、抽象性

抽象性是指将具有一致的数据结构(属性)和行为(操作)的对象抽象成类。一个类就是这样一种抽象, 它反映了与应用有关的重要性质, 而忽略其他一些无关内容。任何类的划分都是主观的, 但必须与具体的应用有关。

10、封装

封装是指将现实世界中存在的某个客体的属性与行为绑定在一起, 并放置在一个逻辑单元内。

11、构造函数

主要用来在创建对象时初始化对象,

即为对象成员变量赋初始值, 总与new运算符一起使用在创建对象的语句中。

与构造函数相反, 当对象结束其生命周期时(例如对象所在的函数已调用完毕), 系统自动执行析构函数。析构函数往往用来做"清理善后"

的工作(例如在建立对象时用new开辟了一片内存空间, 应在退出前在析构函数中用delete释放)。

8、实例化对象

我们上面说过面向对象程序的单位就是对象, 但对象又是通过类的实例化出来的, 既然我们类会声明了, 下一步就是实例化对象了。

当定义好类后, 我们使用new关键字来生成一个对象。

```
$对象名称 = new 类名称 ();

1      <?php
2      class Person
3      {
4          //下面是人的成员属性
5          var $name; //人的名字
6          var $sex; //人的性别
7          var $age; //人的年龄
8
9          //下面是人的成员方法
10         function say() { //这个人可以说话的方法
11             echo "这个人在说话";
12         }
13
14         function run() { //这个人可以走路的方法
15             echo "这个人在走路";
16         }
17     }
18
19     $p1=new Person();
20     $p2=new Person();
21     $p3=new Person();
22     ?>

$p1=new Person();
```

9、使用对象中的成员

上面看到PHP对象中的成员有两种一种是成员属性,

一种是成员方法。对象我们已经可以声明了, \$p1=new

Person();

怎么去使用对象的成员呢? 要想访问对象中的成员就要使用一个特殊的操作符"-

>"来完成对象成员的访问:

对象->属性 \$p1->name; \$p2->age; \$p3->sex;

对象->方法 \$p1->say(); \$p2->run();

```

1      <?php
2      class Person
3      {
4          //下面是人的成员属性
5          var $name; // 人的名子
6          var $sex; // 人的性别
7          var $age; // 人的年龄
8
9          //下面是人的成员方法
10         function say() { // 这个人可以说话的方法
11             echo "这个人在说话";
12         }
13
14         function run() { //这个人可以走路的方法
15             echo "这个人在走路";
16         }
17     }
18
19     $p1 = new Person(); //创建实例对象$p1
20     $p2 = new Person(); //创建实例对象$p2
21     $p3 = new Person(); //创建实例对象$p3
22
23     //下面三行是给$p1对象属性赋值
24     $p1->name = "张三";
25     $p1->sex = "男";
26     $p1->age = 20;
27
28     //下面三行是访问$p1对象的属性
29     echo "p1对象的名子是: " . $p1->name;
30     echo "p1对象的性别是: " . $p1->sex;
31     echo "p1对象的年龄是: " . $p1->age;
32
33     //下面两行访问$p1对象中的方法
34     $p1->say();
35     $p1->run();
36
37     //下面三行是给$p2对象属性赋值
38     $p2->name = "李四";
39     $p2->sex = "女";
40     $p2->age = 30;
41
42     //下面三行是访问$p2对象的属性
43     echo "p2对象的名子是: " . $p2->name;
44     echo "p2对象的性别是: " . $p2->sex;
45     echo "p2对象的年龄是: " . $p2->age;
46
47     //下面两行访问$p2对象中的方法
48     $p2->say();
49     $p2->run();
50
51     //下面三行是给$p3对象属性赋值
52     $p3->name="王五";
53     $p3->sex="男";
54     $p3->age=40;
55
56     //下面三行是访问$p3对象的属性
57     echo "p3对象的名子是: " . $p3->name;
58     echo "p3对象的性别是: " . $p3->sex;
59     echo "p3对象的年龄是: " . $p3->age;
60
61     //下面两行访问$p3对象中的方法
62     $p3->say();
63     $p3->run();
64     ?>

```

从上例中可以看出只是对象里面的成员就要使用对象->属性、对象->方法形式访问，再没有第二种方法来访问对象中的成员了。

10、this使用

现在我们知道如何访问对象中的成员，是通过”对象-

>成员”的方式访问的,这是在对象的外部去访问对象中成员的形式，

那么如果我想在对象的内部，让对象里的方法访问本对象的属性，

或是对象中的方法去调用本对象的其它方法这时我们怎么办？因为对象里面的所有的成员都要用对象来调用，包括对象的内部成员之间的调用，所以在PHP里面给我提供了一个本对象的引用\$this，

每个对象里面都有一个对象的引用\$this来代表这个对象，完成对象内部成员的调用，

this的本意就是“这个”的意思，上面的实例里面，我们实例化三个实例对象\$P1、\$P2、\$P3，这三个对象里面各自存在一个\$this分别代表对象\$P1、\$P2、\$P3。

PHP面向对象(OOP)编程完全教程:7.特殊的引用“\$this“的使用

通过上图我们可以看到，\$this就是对象内部代表这个对象的引用，在对象内部和调用本对象的成员和对象外部调用对象的成员所使用的方式是一样的。

```
$this->属性: $this->name; $this->age; $this->sex;
```

```
$this->方法 : $this->say(); $this->run();
```

```
function say() { //这个人可以说话的方法
    echo "我的名字叫: " . $this->name . " 性别: " . $this->sex . " 我的年龄是: " . $this->age;
}
```

11、魔术方法

__construct(), __destruct(), __call(), __callStatic(), __get(), __set(), __isset(), __unset(), __sleep(), __wakeup(), __toString(), __invoke(), __set_state(), __clone() 和 __debugInfo() 等方法在 PHP 中被称为"魔术方法"(Magic methods)。在命名自己的类方法时不能使用这些方法名，除非是想使用其魔术功能。

11-1、__construct

大多数类都有一种称为构造函数的特殊方法。当创建一个对象时，它将自动调用构造函数，也就是使用new这个关键字来实例化对象的时候自动调用构造方法。构

造函数的声明与其它操作的声明一样，只是其名称必须是__construct()。

这是PHP5中的变化，以前的版本中，构造函数的名称必须与类名相同，这种在PHP5中仍然可以用，但现在已经很少有人用了，这样做的好处是可以使构

造函数独立于类名，当类名发生改变时不需要改相应的构造函数名称了。为了向下兼容，如果一个类中没有名为__construct()的方法，PHP将搜索一个php4中的写法，与类名相同名的构造方法。

在一个类中只能声明一个构造方法, 而是只有在每次创建对象的时候都会去调用一次构造方法, 不能主动的调用这个方法, 所以通常用它执行一些有用的初始化任务。比如对成属性在创建对象的时候赋初值。

格式: `function __construct() { }`

11-2、__destruct

与构造函数相对的就是析构函数。析构函数是PHP5新添加的内容, 在PHP4中没有析构函数。

析构函数允许在销毁一个类之前执行的一些操作或完成一些功能, 比如说关闭文件, 释放结果集等, 析构函数会在到某个对象的所有引用都被删除或者当对象被显式销毁时执行, 也就是对象在内存中被销毁前调用析构函数。与构造函数的名称类似, 一个类的析构函数名称必须是__destruct()。析构函数不能带有任何参数。

格式: `function __destruct() { }`

注意

由于类实例是以堆栈的形式放在内存中, 所以最后调用 析构函数 的时候, 输出顺序是按 后进先出的原则!

11-3、__get

这个方法用来获取私有成员属性值的, 有一个参数,

参数传入你要获取的成员属性的名称, 返回获取的属性值, 这个方法不用我们手工的去调用, 是在直接获取私有属性的时候自动调用的。因为私有属性已经被封装上了, 是不能直接获取值的(比如: `echo $p1->name`” 这样直接获取是错误的), 但是如果你在类里面加上了这个方法, 在使用`echo $p1->name`”

这样的语句直接获取值的时候就会自动调用__get(\$property_name)方法, 将属性name传给参数\$property_name, 通过这个方法内部执行, 返回我们传入的私有属性的值。

11-4、__set

这个方法用来为私有成员属性设置值的, 有两个参数, 第一个参数为你要设置值的属性名, 第二个参数是要给属性设置的值, 没有返回值。这个方法同样不用我们手工去调用, 它也可以做成私有的, 是在直接设置私有属性值的时候自动调用的, 同样属性私有的已经被封装上

了, 如果没有__set()这个方法, 是不允许的, 比如: `$this->name='zhangsan'`,

这样会出错, 但是如果你在类里面加上了__set(\$property_name,

\$value)这个方法, 在直接给私有属性赋值的时候, 就会自动调用它, 把属性比如name传给\$property_name,

把要赋的值“zhangsan”传给\$value, 通过这个方法的执行, 达到赋值的目的。如果成员属性不封装成私有的, 对象本身就不会去自动调用这个方法。为了不传入非法的值, 还可以在这个方法给做一下判断。

11-5、__isset

__isset()

方法:在看这个方法之前我们看一下“isset()”函数的应用, isset()是测定变量是否设定用的函数, 传入一个变量作为参数, 如果传入的变量存在则传回true, 否则传回false。那么如果在一个对象外面使用“isset()”这个函数去测定对象里面的成员是否被设定可不可以用它呢?分两种情况, 如果对象里面成员是公有的, 我们就可以使用这个函数来测定成员属性, 如果是私有的成员属性, 这个函数就不起作用了, 原因就是因为有私有的被封装了, 在外部不可见。那么我们就不可在对象的外部使用“isset()”函数来测定私有成员属性是否被设定了呢?可以, 你只要在类里面加上一个“__isset()”方法就可以了, 当在类外部使用“isset()”函数来测定对象里面的私有成员是否被设定时, 就会自动调用类里面的“__isset()”方法了帮我们完成这样的操作, “__isset()”方法也可以做成私有的。

11-6、__unset

看这个方法之前呢, 我们也先来看一下“unset()”这个函数, “unset()”这个函数的作用是删除指定的变量且传回true, 参数为要删除的变量。那么如果在一个对象外部去删除对象内部的成员属性用“unset()”函数可不可以呢, 也是分两种情况, 如果一个对象里面的成员属性是公有的, 就可以使用这个函数在对象外面删除对象的公有属性, 如果对象的成员属性是私有的, 我使用这个函数就没有权限去删除, 但同样如果你在一个对象里面加上“__unset()”这个方法, 就可以在对象的外部去删除对象的私有成员属性了。在对象里面加上了“__unset()”这个方法之后, 在对象外部使用“unset()”函数删除对象内部的私有成员属性时, 自动调用“__unset()”函数来帮我们删除对象内部的私有成员属性, 这个方法也可以在类的内部定义成私有的。

11-7、__toString()

类被当成字符串时的回应方法

```
1      <?php
2      // Declare a simple class
3      class TestClass
4      {
5          public $foo;
6
7          public function __construct($foo) {
8              $this->foo = $foo;
9          }
10
11         //定义一个__toString方法, 追加一个成员属性$foo
12         public function __toString() {
13             return $this->foo;
14         }
15     }
16
17     $class = new TestClass('Hello');
18
19     //直接输出对象
20     echo $class;
21     >>
```

上例输出: Hello

11-8、__clone()

有的时候我们需要在一个项目里面，使用两个或多个一样的对象，如果你使用“new”关键字重新创建对象的话，再赋值上相同的属性，这样做比较烦琐而且也容易出错，所以要根据一个对象完全克隆出一个一模一样的对象，是非常有必要的，而且克隆以后，两个对象互不干扰。

在PHP中我们使用“clone”这个关键字克隆对象；

11-9、__call()

在在程序开发中，如果在使用对象调用对象内部方法时候，调用的这个方法不存在那么程序就会出错，然后程序退出不能继续执行。那么可不可以不在程序调用对象内部不存在的方法时，提示我们调用的方法及使用的参数不存在，但程序还可以继续执行，这个时候我们就要使用在调用不存在的方法时自动调用的方法“__call()”。

```
1      <?php
2      //这是一个测试的类，里面没有属性和方法
3      class Test
4      {
5
6      }
7
8      //产生一个Test类的对象
9      $test = new Test();
10
11     //调用对象里不存在的方法
12     $test->demo("one", "two", "three");
13
14     //程序不会执行到这里
15     echo "this is a test<br>";
16     ?>
```

上例出现如下错误，程序退出不能继续执行

```
1      Fatal error: Call to undefined method Test::demo()
```

下面我们加上“__call()”方法，这个方法有2个参数，第一个参数为调用不存在的方法过程中，自动调用__call()方法时，把这个不存在的方法的方法名传给第一个参数，第二个参数则是把这个方法的多个参数以数组的形式传进来。

```

1  <?php
2  //这是一个测试的类，里面没有属性和方法
3  class Test
4  {
5      //调用不存的方法时自动调用的方法，第一个参数为方法名，第二个参数是数组参数
6      function __call($function_name, $args) {
7
8          print "你所调用的函数: $function_name(参数: ";
9          print_r($args);
10         echo ")不存在! <br>
11     ";
12     }
13 }
14
15 //产生一个Test类的对象
16 $test=new Test();
17
18 //调用对象里不存在的方法
19 $test->demo("one", "two", "three");
20
21 //程序不会退出可以执行到这里
22 echo "this is a test<br>";
23 ?>

```

上例输出结果为：

```

你所调用的函数: demo(参数: Array ( [0] => one [1] => two [2] => three ) )不存在!
this is a test

```

11-10、__sleep()

执行serialize()时，先会调用这个函数

11-11、__wakeup()

执行unserialize()时，先会调用这个函数

12、封装

封装性是面向对象编程中的三大特性之一，封装性就是把对象的属性和服务结合成一个独立的相同单位，并尽可能隐蔽对象的内部细节，包含两个含义：

- 1、把对象的全部属性和全部服务结合在一起，形成一个不可分割的独立单位（即对象）。
- 2、信息隐蔽，即尽可能隐蔽对象的内部细节，对外形成一个边界[或者说形成一道屏障]，只保留有限的对外接口使之与外部发生联系。

封装的原则在软件上的反映是：要求使对象以外的部分不能随意存取对象的内部数据（属性），从而有效的避免了外部错误对它的“交叉感染”，使软件错误能够局部化，大大减少查错和排错的难度。

在比如说，个人电脑都有一个密码，不想让其它人随意的登陆，在你电脑里面拷贝和粘贴。还有就是像人这个对象，身高和年龄的属性，只能是自己来增减，不可以让别人随意的赋值等等。

private

通过private就可以把人的成员(成员属性和成员方法)封装上了。封装上的成员就不能被类外面直接访问了,只有对象内部自己可以访问。

私有的成员是不能被外部访问的,因为私有成员只能在本对象内部自己访问

四、小结

1、面相对象是什么?就像你一个人生活什么事情都需要自己处理,这是面相过程,而对象就是,你找到了一个女朋友,你叫她做饭洗衣父给你按摩,什么都是他处理

2、类是什么?类就是很多方法的集合

3、如何用面相对象编程

4、this的使用

5、魔术方法

5、面相对象的三大特征:封装、多态、继承

2. 课堂练习

1、了解什么是面向对象

2、怎么用面相对象编程

3. 课后练习

使用面相对象写一个,“我有一只小狗,他的名字叫作旺财”的的输出

4. 资料扩展

1、魔术方法

1-1、__callStatic()

用静态方式中调用一个不可访问方法时调用

1-2、__invoke()

调用函数的方式调用一个对象时的回应方法

1-3、__set_state()

调用var_export()导出类时,此静态方法会被调用。

1-4、__debugInfo()

当转储对象以获取应显示的属性时,此方法由var_dump()调用。如果方法未在对象上定义,那么将显示所有public, protected和private属性。

2、魔术常量

2-1、__LINE__

文件中的当前行号

2-2、__FILE__

文件的完整路径和文件名

2-3、__DIR__

文件所在的目录

2-4、__FUNCTION__

函数名称

2-5、__CLASS__

类的名称, PHP5返回的结果是区分大小写的

2-6、__TRAIT__

Trait的名字

2-7、__METHOD__

类的方法名

2-8、__NAMESPACE__

当前命名空间的名称

3、串行化对象

3-1、serialize

产生一个可存储的值的表示。

serialize() 返回字符串, 此字符串包含了表示 value 的字节流, 可以存储于任何地方。

这有利于存储或传递PHP的值, 同时不丢失其类型和结构。

想要将已序列化的字符串变回PHP的值, 可使用unserialize()。serialize()可处理除了resource

之外的任何类型。甚至可以serialize()那些包含了指向其自身引用的数组。你正serialize()的数组／对象中的引用也将被存储。

当序列化对象时, PHP将试图在序列动作之前调用该对象的成员函数__sleep()。这样就允许对象在被序列化之前做任何清除操作。类似的, 当使用 unserialize()恢复对象时, 将调用__wakeup()成员函数。

3-2、unserialize

从已存储的表示中创建 PHP 的值

对单一的已序列化的变量进行操作, 将其转换回 PHP 的值。