PHP面向对象

张三是一位普通人,别人早晨向他打招呼,他会说"早上好",但有一天他被车撞了,脑子受了一些伤,因此,别人早晨向他打招呼时,他有时会说"早上好",但有时会说"晚上好",甚至辱骂对方.

张三(人): 智商,打招呼()

宝马(车): 撞人()

智商是对象的属性 (名词)

而打招呼是方法. (动词)

$iq = 100;

$new\_iq = rand(60, 110);

class human {

public $iq = 100;

public function greet() {

if($this->iq>=100) {

echo '早上好<br />';

} else {

$ci = array('早上好','晚上好','混蛋');

echo $ci[rand(0,2)];

}

}

}

$lisi = new human();

// print\_r($lisi);

$lisi->greet();

class car {

public function hit($people) {

$iq = rand(60,105);

$people->iq = $iq;

}

}

$bmw = new car(); //

$bmw->hit($lisi);

echo '智商:',$lisi->iq,'<br />';

$lisi->greet();

类声明

class human {

public static $leg = 2; // 这是类的属性,属于全类,不属于某一个对象,在内存中有一份.

public $height = 180;

public $house = 180;

public function eat() {

echo '咪西咪西';

}

}

属性前加public 是干吗的?

答:这是对象的属性或方法的权限修饰符.有public protected private这三种(后面详述)

如何在new 对象时,传一些参数.

构造方法,方法名是固定的, 叫 \_\_construct() {}

这个方法在new对象的时候, 自动执行.

析构方法 \_\_destruct,是在对象被销毁时,自动执行一下.

$this是谁？

是当前对象

// 写一个人类,随机制造智商值,在80-120之间

/\*

class human {

public $iq = //'hello' . 'world'; //rand(80,120);

}

在PHP中声明类,属性不能是表达式或者函数的返回值,也不能是计算结果

// 以下几种情况都不行

// public $iq = 'hello ' . 'world';

// public $iq = rand(80,120);

// public $iq = 3 > 2;

箭头是干嘛的？

面向对象的三大特性: 封装,继承,多态 以及[public,protected,private]

什么是封装?

答:封装对象内部的某些方法或属性,为外部不可见.

开放某些接口,来操作受保护的内部方法或属性.

class human {

public $money = 10000;

public $name;

public function \_\_construct ($name) {

$this->name = $name;

}

public function borrow($who,$much) {

$who->money -= $much;

$this->money += $much;

}

}

protected 保护的

类中声明的protected属性,在对象实例化之后

在类外部,调用该属性是不允许的.

在类内部可以控制(不区分对象)

类的public,protectecd,private的权限控制是以类的级别来控制的,

而不是以对象的级别来控制的.

class human {

private $money = 10000;

public $name;

public function \_\_construct ($name,$money) {

$this->name = $name;

$this->money = $money;

}

public function showmoney($who=false) {

if($who) {

return $who->money;

} else {

return $this->money;

}

}

}

$zhang = new human('张三',3000);

$lisi = new human('李四',4000);

echo $zhang->showmoney(),'<br />';

echo $lisi->showmoney(),'<br />';

echo $lisi->showmoney($zhang);

如果某类的某个方法是公有的, 而该公有方法操作私有属性或保护属性.

那么,该公有方法,不仅能操作本对象的私有属性和保护属性,

还能够操作其他对象的私有属性和保护属性.

如上例子中:$lisi通过showmoney公共方法,读到了$zhang的保护属性$money

继承

某个类可以继承另外一个类的属性和方法.

继承的语法 extends

属性和方法的继承与覆盖.

在子类继承父类的过程,是允许声明一个与父类中存在的同名方法,

并且效果是: 该方法覆盖掉父类的同名方法.

在子类中,还可以声明父类中不存在的属性和方法,来扩展类.

继承与权限控制符 public protected private

在类外部: 只能访问public类型

在类内部: 3者都可以访问

在子类中, 可以访问父类public, 还可以访问父类的protected

提示: 在继承中, 子类相比父类的同名属性或同名方法,

权限控制只能越来越宽松/或不变,不能越来越严格

总结:

1:继承时用 extends 关键字

2:继承时,只能继承自一个父类 (但是,父类还有可能是继承自另一个类)

3:继承时,继承来公共/保护的属性+方法,私有的不继承.

4:子类中,可以声明与父类同名的属性或方法,进行覆盖.

5:继承过程,某个方法/属性的权限只能越来越宽松或者不变, 但是不能越来越严格.

6:public protected,private各自的控制的领域. 在哪儿能访问???

多态

接口: 我们只需要调用父类,不必关心子类.

父类 ins = new 五边形()

导致---->父类的对象产生,并且 ,父类中的面积方法被 子类对象的面积方法重写.

父类对象 x ->面积(a,b,c,d,e);

这种称为对象类型上的多态.

在php中, new 一个实例,返回一个对象

无法对该对象进行类型转换,转换成父类对象,并重写父类.

在PHP中,无法实现java中的这种多态特性,但是可以模拟类似效果.