

第6章 面向对象程序设计

+ 自PHP 5发布以来，最大的特性就是引入了面向对象的全部机制。数组和对象都是PHP复合数据类型中的一种。前面我们已经学习过了数组。我们可以感受到PHP的数组是非常强大的。而对象不仅可以存储任意类型的数据，而且可以储存函数。并且对象还可以封装保护类中的成员不受到非法操作。可以通过继承对类进行扩展。本章的核心就是让读者理解面向对象程序设计的思想，达到灵活应用面向对象方法的目的。

6.1 面向对象概述

+ 面向对象程序设计是一种计算机编程架构，它一直是程序开发中的热门话题。使用面向对象的思想设计的程序代码更加简洁，更加易于维护，并且有更强的重用性。这些特性就达到了软件工程的三个目标：重用性、灵活性和扩展性。并且面向对象的思想相对面向过程的思想更加符合人类看待事物的一般规律。PHP虽不是一个真正的面向对象的编程语言，但是面向对象的思想我们是完全可以借鉴，并且大部分都可以很好地实现，这在大型项目的设计中是很有优势的。

6.1.2 面向对象程序设计

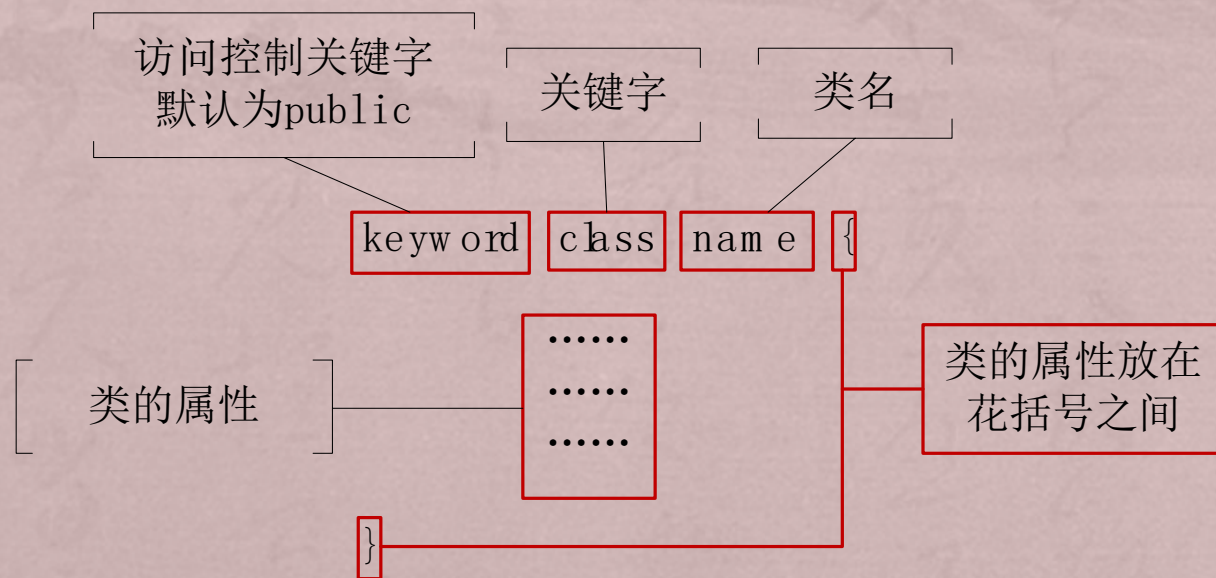
- + 面向对象程序设计。面向对象程序设计是一种程序设计方法，它把现实世界中的事物抽象成对象，对象之间通过消息传递进行交互。面向对象程序设计的主要特点是：封装、继承、多态。面向对象程序设计使得程序的结构更加清晰，代码更加简洁，易于维护和扩展。面向对象程序设计是现代程序设计的主流方法，广泛应用于各种软件开发中。
- + 面向对象程序设计。面向对象程序设计是一种程序设计方法，它把现实世界中的事物抽象成对象，对象之间通过消息传递进行交互。面向对象程序设计的主要特点是：封装、继承、多态。面向对象程序设计使得程序的结构更加清晰，代码更加简洁，易于维护和扩展。面向对象程序设计是现代程序设计的主流方法，广泛应用于各种软件开发中。

6.2 创建一个类

- + 初步使用面向对象程序设计，我们要掌握的就是使用面向对象的思想抽象出一个类，这是一种思想的转化，难点也就是在这里。而这些思想的体现在程序中就是代码。在这里，要写出代码，掌握语法，这些都是很简单的，而能正确地抽象出一个类才是我们的希望读者可以完成的。我们就先以简单的语法开始逐渐深入地讲解他们。

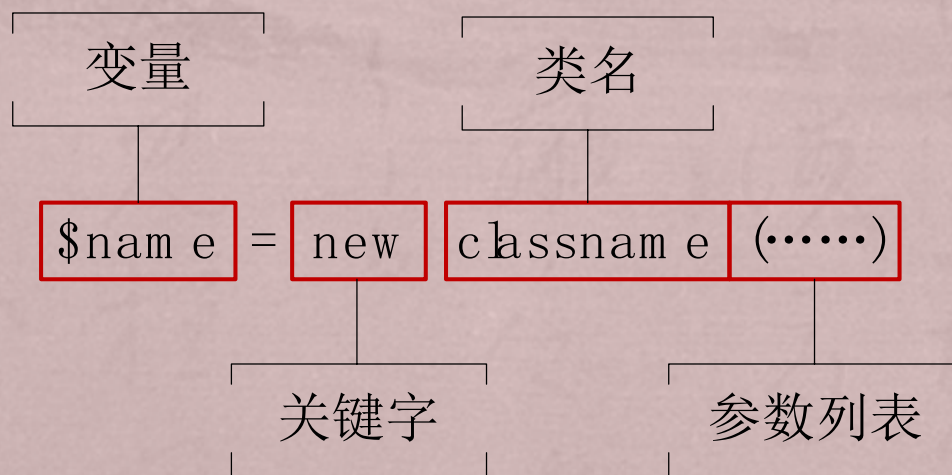
6.2.1 创建类

- + PHP类由三部分构成：第一部分是表示类的关键字class；第二部分是表示类名的PHP合法标示符；第三部分是使用花括号括起来的一组代码。整个类的语法如图所示。



6.2.2 实例化对象

- + PHP中实例化一个对象类似赋值操作。它使用new操作符加类名和参数列表来实例化一个对象，然后赋值给一个变量。语法如图所示。
- + (1)创建一个ren类并且实例化三个对象。

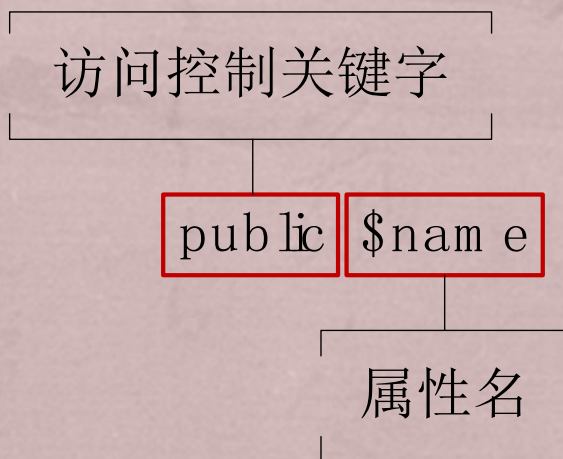


6.3 成员属性

- + 在前面的章节中我们声明了类并且也实例化了对象，但是我们很清楚的知道那些类及其对象是没有任何意义的，因为他们不会做任何事情，也没有任性的特性。学习就是要循序渐进的，类就是要慢慢补充的。成员属性就是类的核心之一。下面我们来开始学习他们。

6.3.1 成员属性

+ 我们知道人都有一定的共有属性，例如姓名、年龄、生日、户籍地等等这些都是一个人常有的必备的一些性质，而它们体现在类中就是类的成员属性。定义成员属性的语法如图所示。

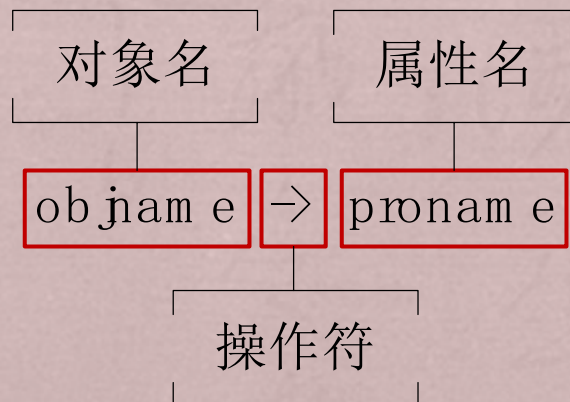


6.3.1 成员属性

- + 在早前的PHP版本中使用如下这种写法：
- + `var $name`
- + 这种写法的含义和用法与
- + `public $name`
- + 是完全一样的，在PHP 5中改为使用public修饰。但是为了兼容以前的程序，这种写法依然可以使用，但是并不是我们推荐的。
- + (1)定义一个含有姓名，性别，年龄，生日，户籍属性的ren类，并且实例化若干个对象。
- + (2)定义一个ren类，在示例6-2中ren类的基础上去掉一个属性，并实例化若干个对象。

6.3.2 访问成员属性

- + 在PHP中我们使用“->”来访问对象的成员属性，它的语法如图6.6所示。
- + (1)创建一个ren类的对象\$xiaoyang，并且用“->”操作符访问它的属性并执行赋值操作，并在最后访问输出这些属性。
- + (2)实例化ren类的两个对象，并且给其中一个对象的xingming属性赋值。然后输出两个对象对应的xingming属性的值。

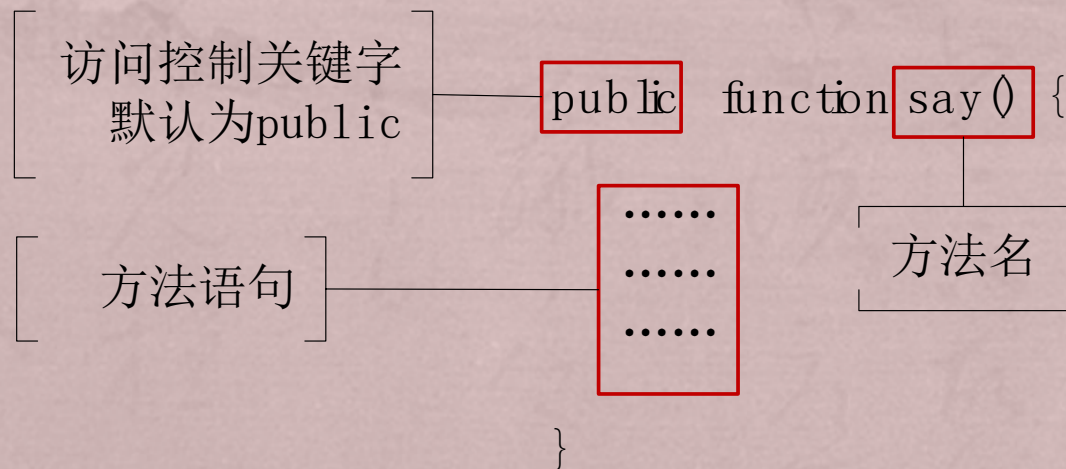


6.4 成员方法

+ 前面的章节中我们已经了解了成员属性，已经学会了访问成员属性的语法。我们可以清楚的体验出，光有成员属性的类是非常单薄和没有多大用处的。这里我们就要再给类添加一种成员——成员方法。成员方法可以操作类里面的成员属性，也可以独立完成一些操作，它为类添加了强大的功能，下面我们就来学习它。

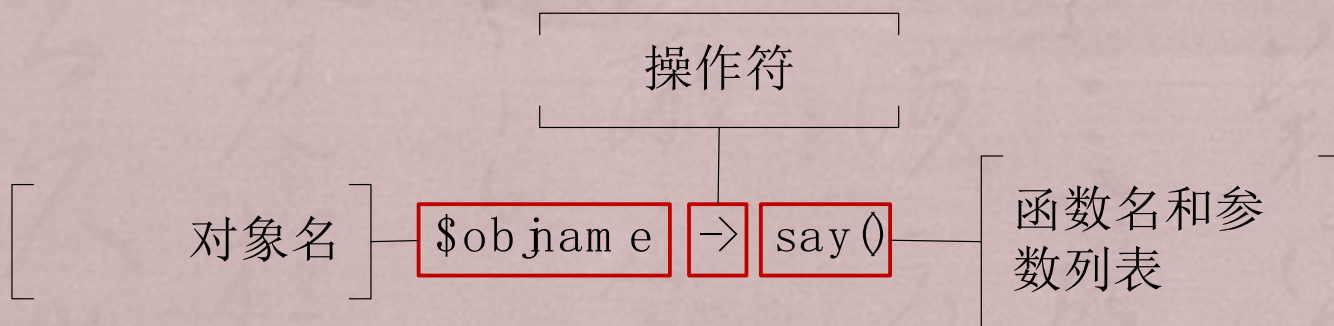
6.4.1 成员方法

- + 成员方法也就是函数。它可以完成一些我们需要的操作。成员方法的定义很简单，只要在函数前面加上访问控制修饰符关键字即可，语法如图所示。
- + (1) 给ren类定一个成员方法say。



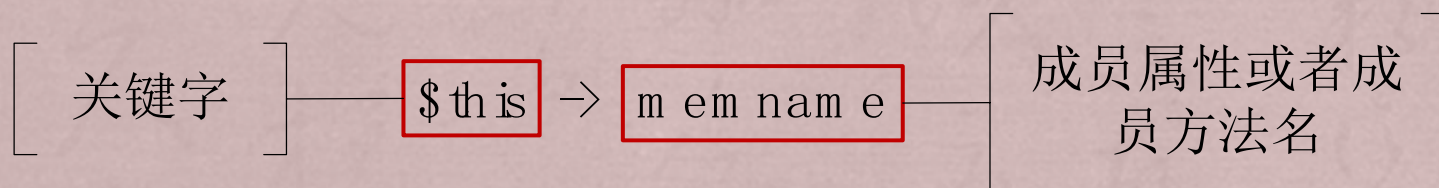
6.4.2 访问成员方法

- + 成员方法的访问与成员变量的访问方法是相同的都是使用“->”操作符来完成的，语法如图所示。
- + (1)访问ren类中的成员方法say。



6.4.6 \$THIS关键字

- + 在定义了成员方法后，我们常常会用到一个成员属性，调用成员方法。要实现这种操作，我们只能使用\$this，它的语法如图所示。
- + 它会把查找范围缩小在一个类以内，也就是说使用了\$this语法后，程序只会在类内寻找这个同名成员，而不会在类范围外的地方去查找。



6.4.6 \$THIS关键字

- + (1)使用\$this在成员方法中访问成员属性。
- + (2)在一个成员方法中使用\$this访问另一个成员方法。

6.4.6 构造方法

- + 在类的方法中有一种特殊的方法就是构造方法。PHP 5 允许开发者在类中定义一个方法作为构造函数。我们先来看它的语法，如图所示。
- + 由于构造方法是由系统自动调用的，我们不可以主动调用，因此不需要加访问控制关键字。这个读者一定要注意。构造函数会在每次创建新对象时被调用，所以非常合适在使用对象之前做一些初始化工作。如果我们不显式地创建构造方法，在类也是有默认的构造方法的。

函数名必须这么写，注意开头是两个下划线

```
function __construct () {  
    .....  
    .....  
    .....  
}
```

构造方法

6.4.6 构造方法

- + (1) 在类中定义一个构造方法用来对对象进行初始化。
- + (2) 给类的构造方法定义默认参数。

6.4.7 析构方法

- + 与构造函数名称类似，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。同样的，析构函数也不需要访问控制关键字。它的语法如图所示。
- + 析构函数允许在销毁一个类之前执行一些操作或完成一些功能。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自动调用析构函数。下面我们来看一个示例。

注意函数名开头
是两个下划线

```
function __destruct() {  
    .....  
    .....  
    .....  
}
```


6.4.7 析构方法

- + (1) 为类定义一个析构方法，在对象不再被使用后，系统会自动调用析构函数并输出函数体中的内容。
- + (2) 实例化一个类的对象后使用unset函数销毁对象名。

6.5 封装性

- + 封装性是面向对象的三大特性之一，封装的最主要作用就是把数据封装起来，不让外界可以随意访问，修改。这样就可以使得数据更加的安全可靠。封装性还是比较好理解和学习的，下面我们就来学习他们。

6.5.1 封装性的含义

+ PHP的封装以有
两个方面含义。
我们先来看第一
个含义。如图所
示。

+ 我们可以看到的我
们使用的类成员函
员属性和被集中的
数都是类封装的
置在类里面，从
这就体现了封装
一个角度的来。

```
class ren {
```

```
public $xingming;  
public $xingbie;  
public $nianling;
```

成员属性

```
function __construct($name){  
    $this->xingming=$name;  
    echo '创建了一个对象。<br/>';  
}  
public function say(){  
    echo '我叫'.$this->xingming.'。<br/>';  
}  
function __destruct(){  
    echo '这个对象被销毁了。';  
}
```

成员方法

结构上的封装

6.5.2 访问控制关键字PUBLIC、PROTECTED、PRIVATE

- + 另一种体现就是在数据上的，我们由生活中的案例就很容易想到，比如我们的个人隐私，手机号码，生日，居住地址之类的数据并不想随意就让别人得到或者修改。有些就是我们不愿公开的。
- + 我们在网上注册一些账号的时候也可以体现出来，就比如生日和联系方式可以设置为自己的好友可见而非好友不可见等等状态。要实现这些，我们知道在学习面向对象程序设计以前是做不到的，所有变量或者函数都是可以随意操作。这在类中可以使用访问关键字来约束它们。访问控制关键字有如下三种：
- + public 由它修饰的成员被称为公共成员，可以在类内部和外部被访问。
- + protected 由它修饰的成员被称为保护成员，可以在类内部被访问。
- + private 由它修饰的成员被称为私有成员，只可以在类内部被访问。
- + 我们前面一直使用的是public关键字修饰成员的，由它修饰的成员可以在类内部和外部访问。由private和protected修饰的成员不可以被外部成员访问。

6.5.2 访问控制关键字PUBLIC、PROTECTED、PRIVATE

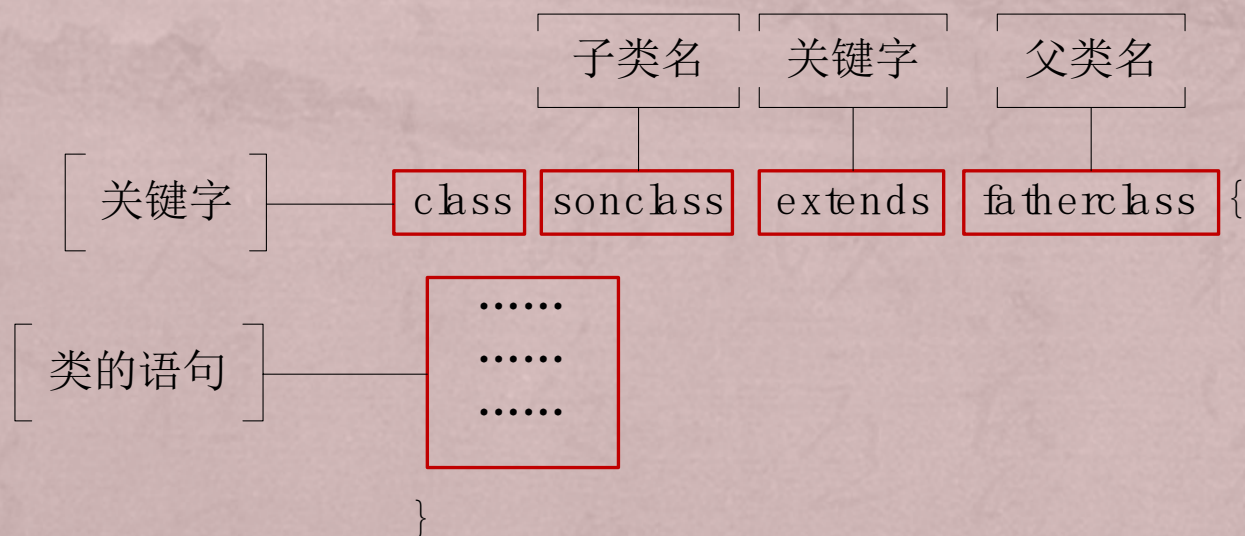
- + (1)演示private和protected修饰的成员不可以在类外被访问。
- + (2)使用接口函数访问类的私有成员属性。
- + (3)在接口函数中加入判断，防止非法访问私有属性。

6.6 继承性

+ 继承性是面向对象编程的三大特性之一，我们知道人类有遗传，有继承财产这些的说法。类中的继承性就是这些特性的抽象，正是由于继承性而增强了面向对象的可扩展性，下面我们就来学习这个重要的特性。

6.6.1 继承概述

- + PHP 的对象模型也使用了继承。继承将会影响到类与类，对象与对象之间的关系。继承对于功能的设计和抽象是非常有用的，而且对于类似的对象增加新功能就无须重新再写这些公用的功能。继承使用 extends 关键字进行的，语法如图所示。



6.6.2 继承PUBLIC成员

- + 我们知道类的成员有不同的访问权限，是由访问控制关键字来控制的。在PHP中所有的public成员均可以被继承。
- + (1)演示儿子类继承父亲的高个子、高智商的属性，继承了父亲绝活的函数。

6.6.3 继承PROTECTED成员

- + 很多情况下有些类继承的属性是不想被在类外部访问的，及比如儿子继承了父亲100万的存款，而父亲不希望外人知道孩子的存款。就可以把存款成员声明为一个保护成员也就使用protected修饰的成员。保护成员的性质就是不可以被在类外部访问到，但是可以在子类内部访问到。也就是说我们可以在子类设置一个接口函数来访问保护成员。
- + (1)在类外部不可以访问到类的protected成员，而可以在类内部访问到。

6.6.4 PRIVATE成员

- + 使用private修饰的成员被称为私有成员。父类中的私有成员不会被子类继承，因此不会被子类访问到。
- + (1)演示父类中的私有成员不会被子类继承和访问。

6.6.5 继承的扩展

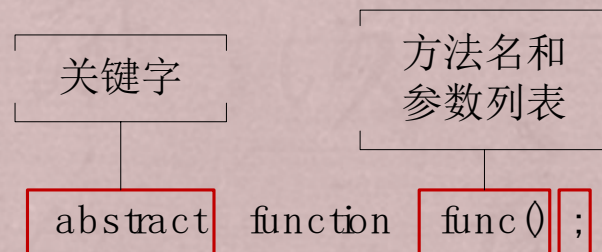
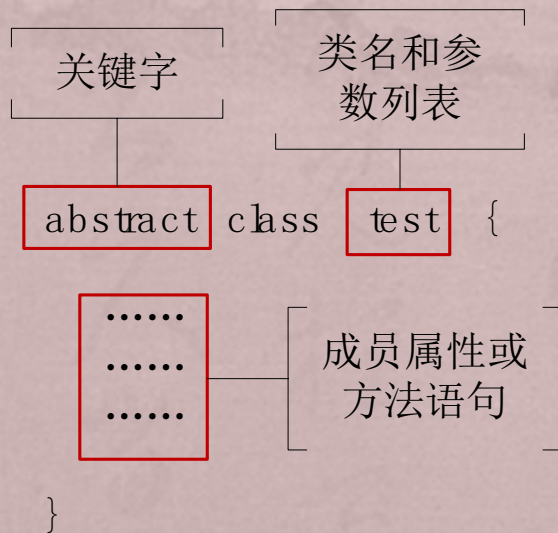
- + 所谓继承的扩展，两个方面的扩展，一方面就是成员的扩展，一方面就是成员函数的覆盖。这个还是比较容易理解的。成员的扩展就好似在继承了父类的属性后，自己还可以有自己独特的成员属性和方法。而如成员方法如果和父类继承下来的成员方法同名，那么就会子类中的方法就会覆盖从父类继承的方法。这就像把父亲那里学来的手艺加了改进一样。
- + (1) 演示在子类中可以有新的属性和方法。

6.7 抽象类和接口

+ 前面我们已经学习了继承，子类可以继承父类的成员方法和成员属性，也可以修改继承来的成员方法。而抽象类就是在类中定义了一个父类不曾实现的方法，而规定子类必须实现。这也是一种很有用的方法，只要看到父类中的方法名，就可以不必看子类而直接调用子类中的方法了。

6.7.1 抽象类和抽象方法

- + 在前面我们学过类可以被继承，这里就一个吧继承做到极致的类，那就是抽象类。抽象类和方法使用abstract修饰，语法如图所示。
- + 任何一个类，如果它里面至少有一个方法是被声明为抽象的，那么这个类就必须被声明为抽象类。
- + 抽象类不能被直接实例化，必须先继承该抽象类，然后再实例化子类。继承一个抽象类的时候，子类必须实现抽象类中的所有抽象方法；另外，这些方法的可访问性必须和抽象类中一样（或者更为宽松）。如抽象类中某个抽象方法被声明为protected，那么子类中实现的方法就应该声明为protected或者public，而不能定义为private。



特别注意抽象方法要以分号结尾，并且不可以有功能实现部分

6.7.1 抽象类和抽象方法

- + (1) 通过 fangxing 类继承并实现抽象类的 mianji 的抽象方法并计算长方形的面积。
- + (2) 演示抽象类中的非抽象方法会继承。

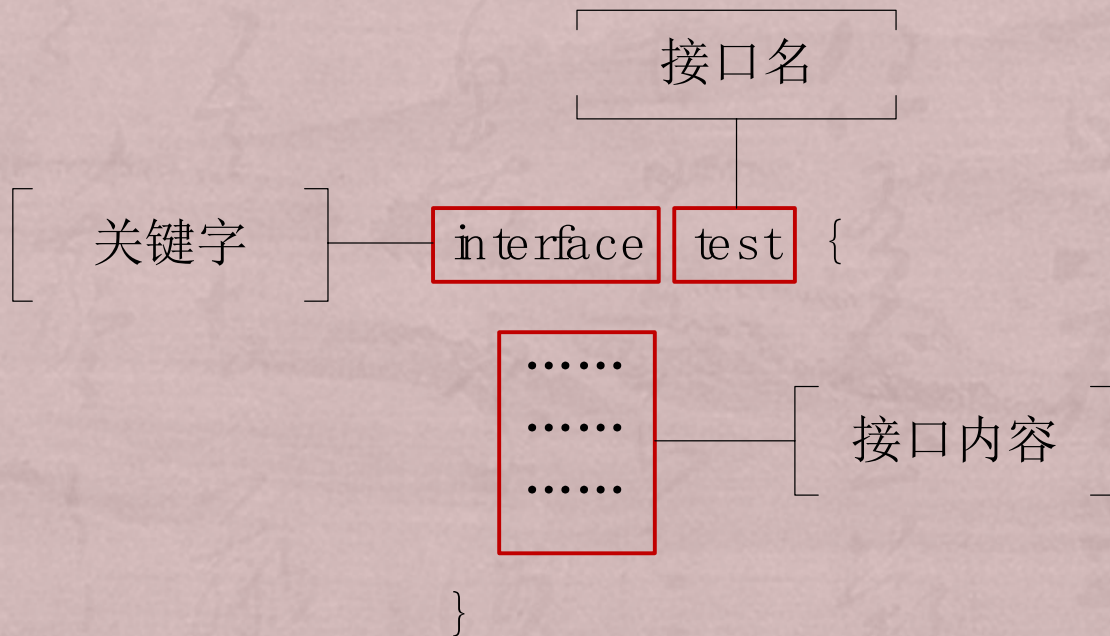
6.7.2 接口

+

本节我们并不要学习新的抽象思想，接口不是全部成员。接口是抽象类的一个特例，它的方法变量都必须有public的访问权。它的作用呢？其有时，接口类起到了作用了。实际上接口类说白了就是一个类的模板，一个类的规定。如果你属于这类，你必须遵守我的规定，但是具体你做什么，那是你的事。

+

那接口类的作用呢？其有时，接口类起到了作用了。实际上接口类说白了就是一个类的模板，一个类的规定。如果你属于这类，你必须遵守我的规定，但是具体你做什么，那是你的事。



6.7.2 接口

- + (1) 演示实现一个接口类。
- + (2) 演示接口类继承接口类实现功能的扩展。
- + (3) 在继承的同时实现多个接口。

6.8 多态性

- + PHP 由于是弱类型语言的原因。因此PHP并不是支持完全的多态性。在PHP中我们的多态我们理解为一段程序能够处理多种类型对象的能力。
- + (1)演示多态性的应用。

6.9 静态成员

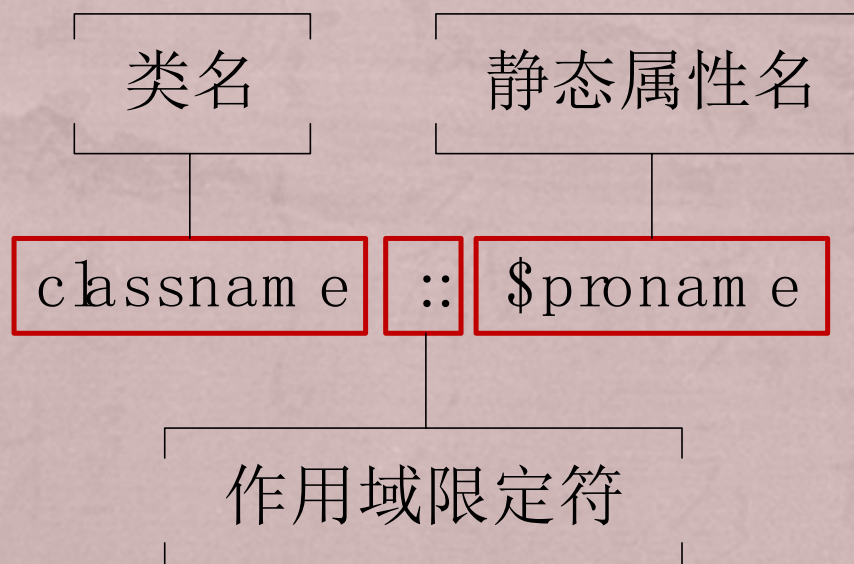
- + 静态成员是一种特殊的成员，它在类中可以被所有对象共享。静态成员是属于类的成员，且并不会被实例化到对象中。这样的优势就在于可以不实例化一个对象就可以通过类来访问到这些成员。下面我们就来学习这些知识。

1.静态成员属性的定义

- + 静态成员属性是一个特殊的成员属性。它被类的所有对象共享，的语法如图6.24所示。
- + 静态属性能被初始化为一个字符值或一个常量，不能使用表达式。
- + (1)给ren类定义一个静态成员属性。

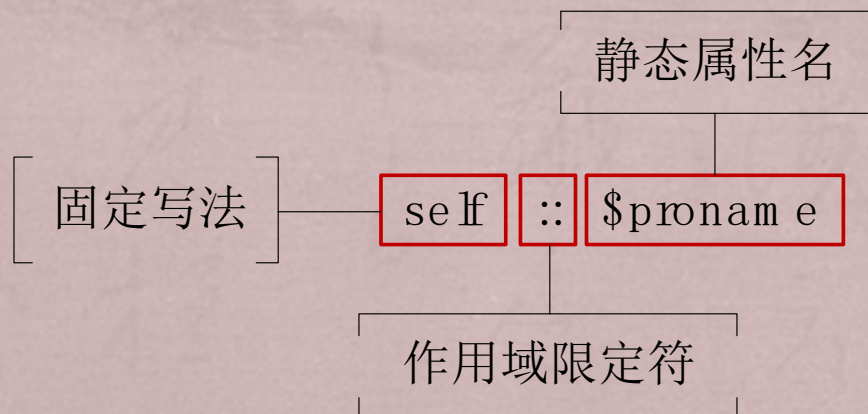
2.访问静态成员属性

- + 由于静态成员属性是属于类的属性。因此只能通过类访问。PHP中使用作用域限定符“::”来访问类的静态成员属性。语法如图所示。



2. 访问静态成员属性

虽然在函数内部也可以使用作用域限定符来访问类的静态变量，但是为了使增强灵活性，通常在类内部使用self来代替类名，这样的好处是不会被类名的变化影响到访问。它会写成如图所示的形式。

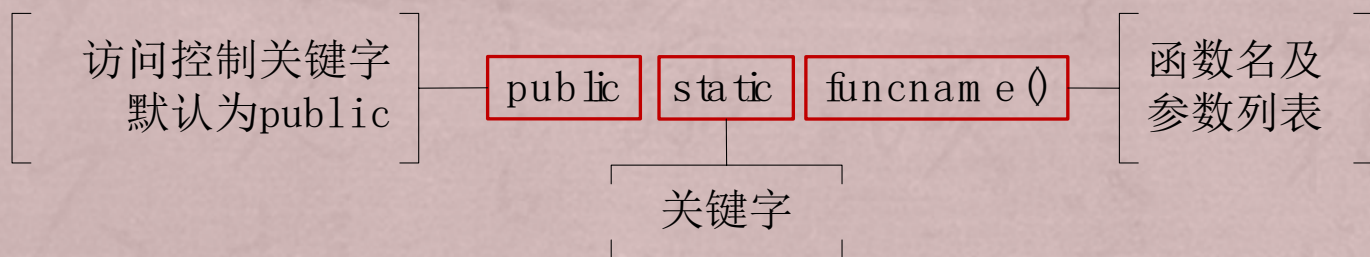


2.访问静态成员属性

- + (1)在若干个类中定义若干个静态成员方法，使用作用域限定符来访问它们。

1.静态成员方法

- + 成员属性可以是静态的，同样的，成员方法也可以是静态成员方法。它的语法如图所示。
- + (1)为ren类定义一个静态成员方法renlei()。



2.访问静态成员方法

- + 静态成员方法的访问语法和访问静态属性的语法是一样的，都是使用类名和作用域限定操作符来完成的。它的语法如图所示。
- + (1)访问ren类中的静态成员方法renlei()。

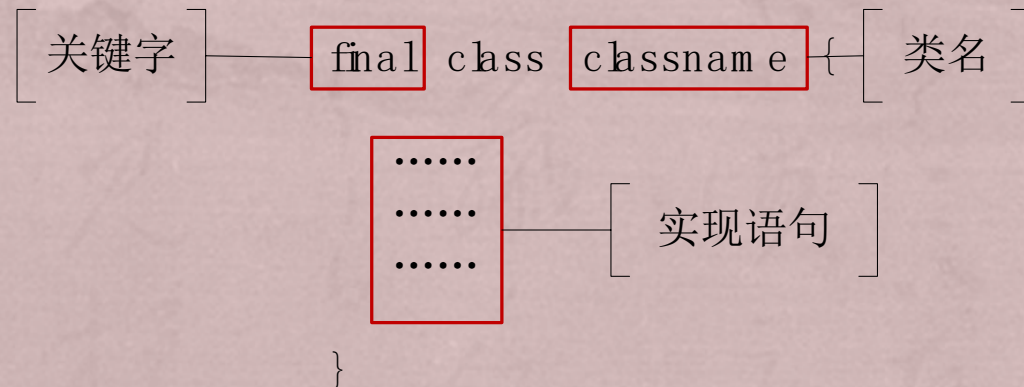


6.10 常见关键字和魔术方法

- + 在前面的学习中我们已经学习了很多在面向对象编程中非常用的关键字，比如public、protected、static以及描述类的abstract和interface。在面向对象编程中，还有其他一些有用的关键字和方法会被使用到。下面我们就来讲解他们。

1.FINAL关键字

- + final关键字是在PHP 5后新增的。final有“最终的”含义。如果父类中的方法被声明为final, 则子类无法覆盖该方法; 如果一个类被声明为final, 则不能被继承。final关键字修饰一个类的语法如图所示。

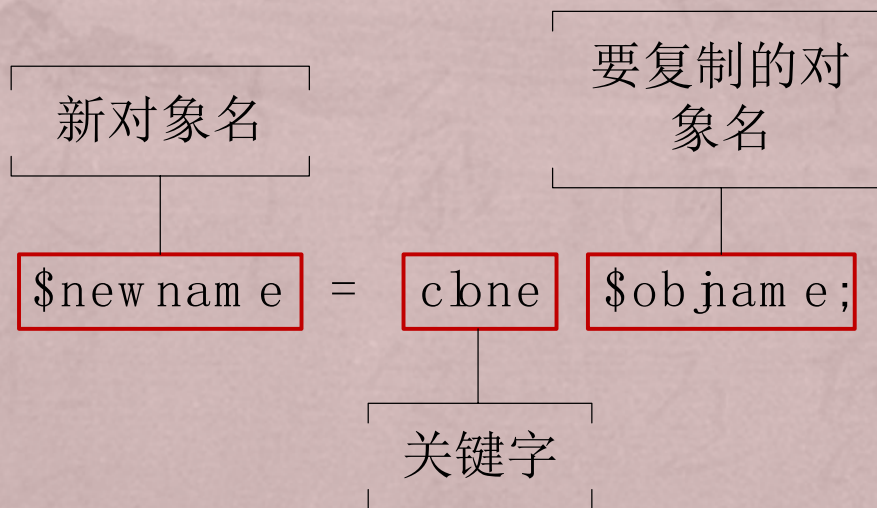


1.FINAL关键字

- + (1)演示一个final修饰的类不会被继承。
- + (2)演示子类继承final成员方法并访问final成员方法。
- + (3)尝试覆盖子类中的final成员方法。
- + (4)把类中的方法使用final修饰，以达到防止被覆盖的目的。

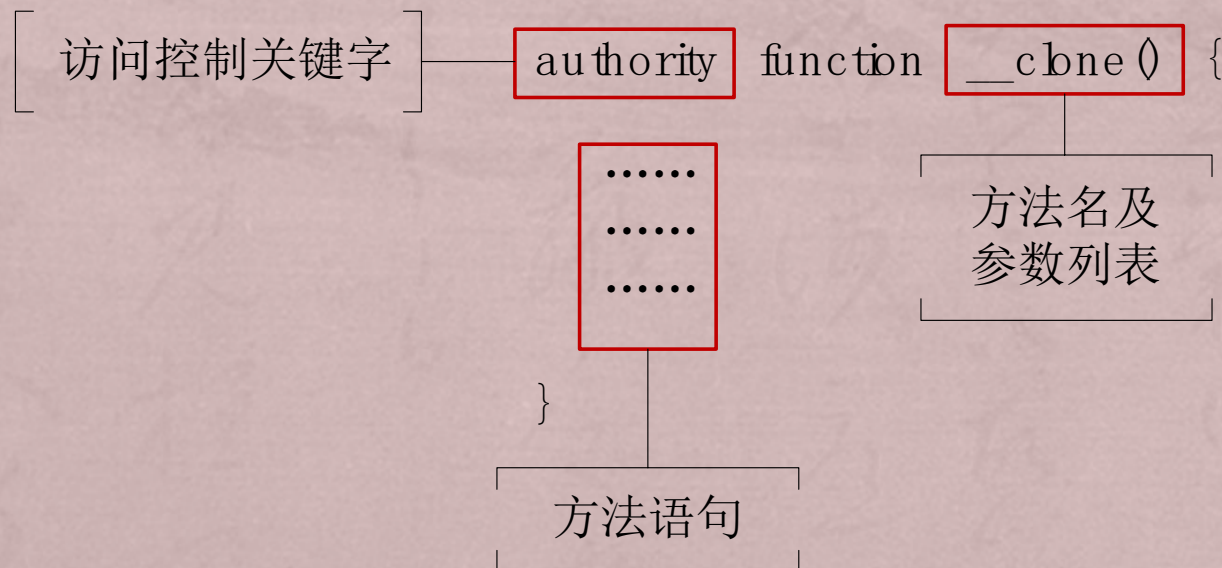
2.CLONE关键字

- + 在编程的过程中有的情况下我们需要建立一个对象的副本，希望对原对象的修改不会影响到副本对象。实际上我们可以再实例化一个对象再给属性添加相同的值。但是这是比较繁琐的，特别是属性很多的时候。这里我们就可以使用clone关键字来复制一个一模一样的对象。它的语法如图所示。
- + (1)复制一个对象并赋值给一个新的对象名，并输出它们的信息。



2.CLONE关键字

- + 下面我们来看__clone的语法，如图所示。
- + (1)使用__clone方法在复制对象的时候做一些初始化工作。

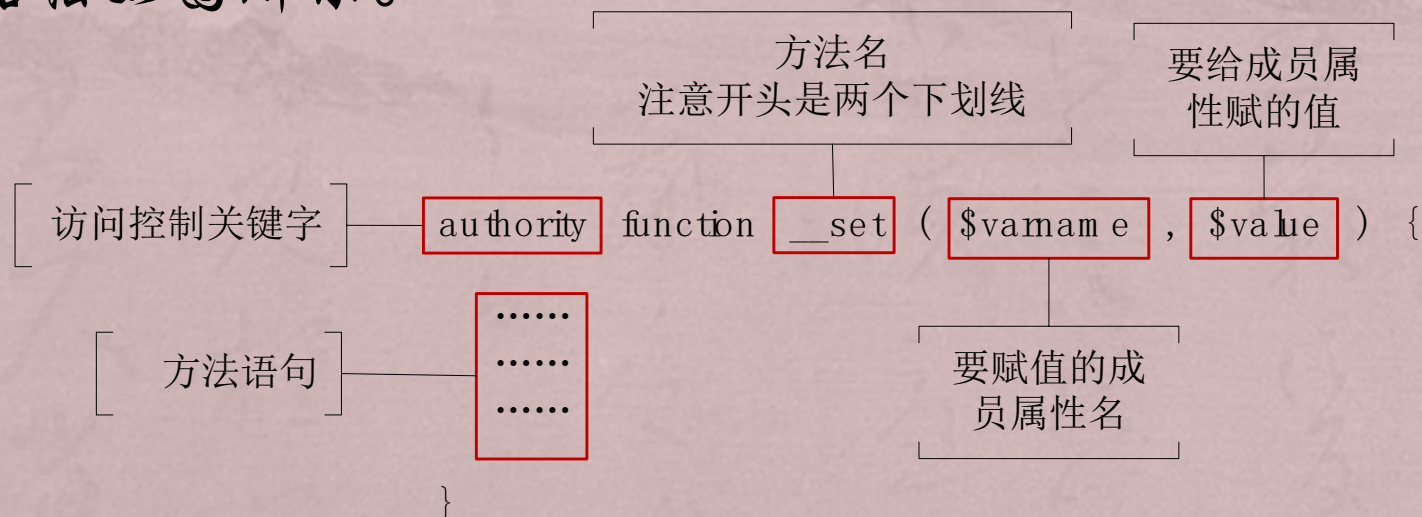


6.10.2 常用魔术方法

+ 所谓魔术方法，就和我们前面学习过的类的构造函数 `__construct` 和复制对象时候自动调用的函数 `__clone`。这些函数的共同特点就是执行某一操作时候会被自动调用。通常和一个关键字相关联。就像使用 `new` 关键字实例化一个对象的时候，系统会自动调用构造函数。使用 `clone` 函数复制一个对象的时候系统会自动调用 `__clone` 函数一样。下面我们就来学习一些其他常用的魔术方法。

1. __SET()、__GET()、__ISSET()、__UNSET()

+ 通常情况下在类中会定义为私有属性，这是为了保护数据而使用的措施，但是给我也带来了一点麻烦。就是如果我们想要修改它们的值的时候，是需要通过调用我们定义的接口函数来完成的。每次需要修改变量都要调用不同的接口这是比较麻烦的。于是在PHP系统中就为我们提供了一个魔术方法__set()。来方便我们的赋值操作。它的语法如图所示。



1. __SET()、__GET()、__ISSET()、 __UNSET()

+ (1) 演示使用 __set() 控制成员属性赋值。

1. __SET()、__GET()、__ISSET()、 __UNSET()

- + 类似的另一个魔术方法就是__get()方法，它的作用是在类的外部访问类的私有成员属性的时候，会自动调用__get()方法返回属性的值，它有一个参数用来接收属性名，它的语法如图所示。
- + (1)演示在类外部自动调用__get()方法访问类的私有属性。

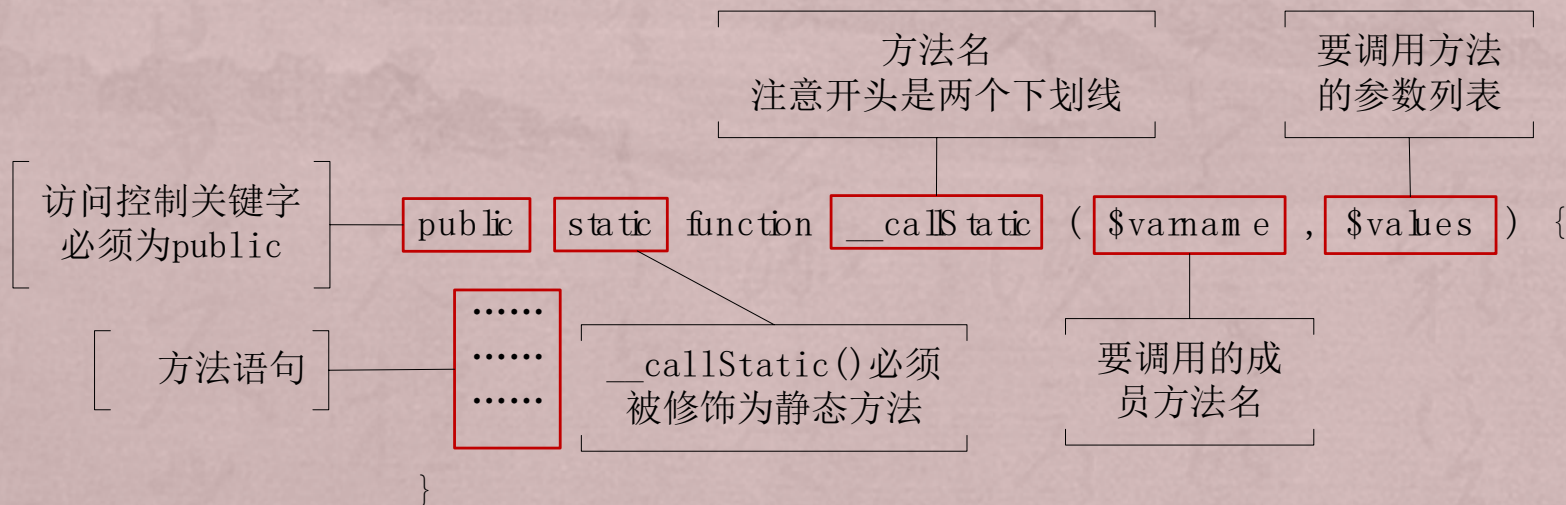
2. CALL()和CALLSTATIC()

- + call()方法用于对象访问的函数不存在或者没有足够的访问权限的时候会被系统自动调用，它有两个参数，第一个参数用于接收要访问的方法名，第二个参数用于接收方法名的参数列表。它的语法如图所示。
- + (1)演示调用不存在或者不可见的成员方法时，系统会自动调用call()。



2. __CALL()和__CALLSTATIC()

- + 在学习类的时候我们还学习过了不会被对象实例化的函数——静态函数。`__callStatic()`就是针对调用static函数而定义的函数。它的功能和`__call()`是一样的。只不过是针对静态函数的。下面我们就来看`__callStatic()`的语法。如图所示。
- + (1)演示调用不存在或者不可见的静态函数时，系统会自动调用`__callStatic()`。



6.11 小结

么，面一完，了越要此的这俗是把来承做过我们因面章通就会化继子通我，后本单的学例像例，码及在简要要实就的识代以了，了重说过，单例知了，了过最是通申简的佳码，束通章就在引常单质记代结量本也后的非简本是的就尽。然类了些最不中习也解想，些过这到而例学们讲思类一通弃学法示的我来的个了都嫌以方写象中字象一习。要可了手对解例抽成学识不才握亲向讲个象还知者子学要面的解这抽后些读例者定的识理象物然这。的读一章知易对事。口解单是者题本多容向个成接讲简的读习