

PHP面向对象小笔记

1.在类中可以定义类的静态属性和方法，所谓的“静态”就是定义的属性和方法与类的对象无关，只跟类本身有关，静态的属性和方法一般用来包含类要封装的数据和功能，可以为该类所有的对象所共享。

2.访问类的静态属性和方法需要用到范围解析符：

如： `classname::$a`

`classname::A()`

3.只有静态属性和方法才能用范围解析符：：，不能用范围解析符访问非静态的属性和方法。

4.在PHP5中，构造函数的名称为__construct,析构函数的名称为__destruct, "construct"与“destruct”前面是两根下划线，**与类同名的方法不再被看成构造函数。**

5.PHP不支持多继承，即一个子类只能继承一个父类，“extends”关键字，如：`class B extends A { }`。

6.要在子类中调用父类的方法，除了用“`$this->`”外，还可以用`parent::functionname()`，建议使用第二种，而对于父类的属性，只能用“`$this->`”来调用，属性是不区分父类和子类的。

7.`final`在声明类时使用这个关键字，将使**此类不能被继承**，如`final class A`

如果将`final`关键字用于声明类中的方法时，此方法将不能被子类重载。

8.抽象类和接口：

1.**抽象类**：用关键字“abstract”来定义，一个抽象类中至少包含一个抽象方法，抽象方法也是用关键字“abstract”来定义。

抽象方法只包含方法的定义而不包含方法的实现。 **包含抽象方法的类必须是抽象类。**

抽象类不能用于创建对象，只能通过继承来使用，继承抽象类的子类必须**重载**抽象类中的所有抽象方法才能被实例化。

抽象类允许非抽象的方法和属性。

2.接口：为了解决PHP单继承的问题，PHP5引入了接口的概念，接口是特殊的抽象类，他用“interface”关键字代替class关键字，接口中**不允许非**

抽象的属性和方法，所有定义的方法都是抽象方法，在**接口中不能使用属性，但是可以使用关键const字定义的常量。**

在接口中定义抽象方法不需要abstract关键字，接口中所有抽象方法都要用public定义（也可以省略public）。

接口也支持继承。

定义了接口之后可以将其实例化，接口的实例化称为接口的实现，接口的实现需要一个子类来实现接口中的所有抽象方法，定义接口的子类用关键

“implements”，另外，一个子类可以实现多个接口，这样就解决了多继承的问题。

一个子类还可以同时继承一个父类和多个接口，如：class A extends B implements C,D,E,.....{ }。

