概述

# 接口

1、P**HP类是单继承，不支持多继承。当一个类需要多个类的功能时，继承就无能为力了，为此PHP引入了接口技术。**

2、对代码进行统一的规范。多人一起开发一个项目时，可能都会去调用别人写的一些类，接口可以规范成员属性、成员方法等。

注意

1. 接口中的方法都是抽象方法。
2. 接口中可以定义常量。
3. 接口中的方法必须是public。
4. 接口可以被继承，继承后也是接口（继承用于实现接口的扩充）。

## 定义

使用关键字：interface。

例：

interface Shop

{

public function buy($gid);

public function sell($gid);

public function view($gid);

}

## 实现

使用关键字：implements

例：

Class a implements shop{

Public function buy($a){...}

Public function sell($a){...}

Public function view($a){...}

//其他实现方法

}

接口可以实现多个接口（解决PHP单继承的缺点）

Class a implements shop,shop2,shop3{}

# 抽象类

1. 抽象类只能单继承，当一个子类需要实现的功能需要继承多个父类时，就必须使用接口
2. 类只要至少含有一个抽象方法，就必须声明为抽象类。
3. 抽象方法，不能够含有函数体
4. 继承抽象类的子类，实现抽象方法的，必须跟该抽象方法具有相同或者更低的访问级别
5. 继承抽象类的子类，如果不实现所有抽象方法，那么该子类也为抽象类

## 定义

使用关键字：abstract

例：

abstract class AbstractClass {  
  // 强制要求子类定义这些方法  
     abstract protected function getValue();  
     abstract protected function prefixValue($prefix);  
  
     // 普通方法（非抽象方法）  
     public function printOut() {  
         print $this->getValue() . "\n";  
     }  
 }

## 实现

使用关键字：extends

例：

class ConcreteClass1 extends AbstractClass {  
     protected function getValue() {  
        return "ConcreteClass1";  
     }  
  
     public function prefixValue($prefix) {  
         return "{$prefix}ConcreteClass1";  
     }  
 }