

类的继承

1. 概述

简单理解

类A有某些特征，另一类B，也具有A中所有特征，并且可能还具有自己的更多一些特征，此时可实现:B类使用A的特征信息，然后再补充一些自己的特征信息。

```
Class B extends A{  
    echo $this->A定义的某个属性;  
};
```

基本概念

- (1) 继承：一个类从另一个已有类获得其特征，称为继承
- (2) 派生：从一个已有类为模板产生的一个新的类，称为派生类
- (3) 父类/子类：已有类称为父类，新生类称为子类。父类又叫基类，子类又叫派生类
- (4) 单继承：一个类只允许有一个父类，不能同时从多个上级类继承。Php同java c#、oc等都是单继承。C++多继承
- (5) 拓展：加新属性、新方法、新常量，拓展原有类的功能，称为拓展，没了拓展，继承也就没意义。

2. 访问修饰符

- (1) public:公共的，共有的，公开的 var声明默认是public、静态、常量默认为public。可以在任何位置访问，比如类内部、子类内部、类外部。
- (2) protected:受保护的。可在当前类或子类内部访问，不能在类的外部访问
- (3) private:私有的。只可以在当前类内部访问，不能在子类、类外部访问
- (4) parent表示父类，相当于oc中的super，而self表示当前类

```
class D extends C{  
    function showMe() {  
        parent::showMe();  
    }  
    function __construct($p1, $p2) {  
        parent::__construct($p1); //先初始父类原有属性  
        self.$p2 = $p2; //后初始化拓展的属性  
    }  
}
```

3. 重写

(1) 概念：又叫覆盖，从父类继承下来的属性或方法重新定义——就是重写
注：子类覆盖父类的方法，虽然可以调用父类的同名方法会完成一定工作，但不是必须的，如果父类功能不适用于子类，则子类可以完全自己实现具体的方法，而不调用parent的方法实现。

(2) 基本要求

1) 下级属性、方法的访问修饰符权限应该不小于父类的访问控制权限。

父：public 子：public

protected public、protected

private public、protected、private

2) 参数形式应该与父类一致

4. 最终类

通常，一个类无特别声明，可以任意继承，对其拓展
但，如果不希望自己被拓展，可以将其声明为最终类

形式：final class A {};

5. 最终方法

通常，一个方法无特别声明，可以任意被子类重写(覆盖)

但，如果不希望自己的方法被子类重写，那么可以声明为最终方法

形式：final function f1() {};