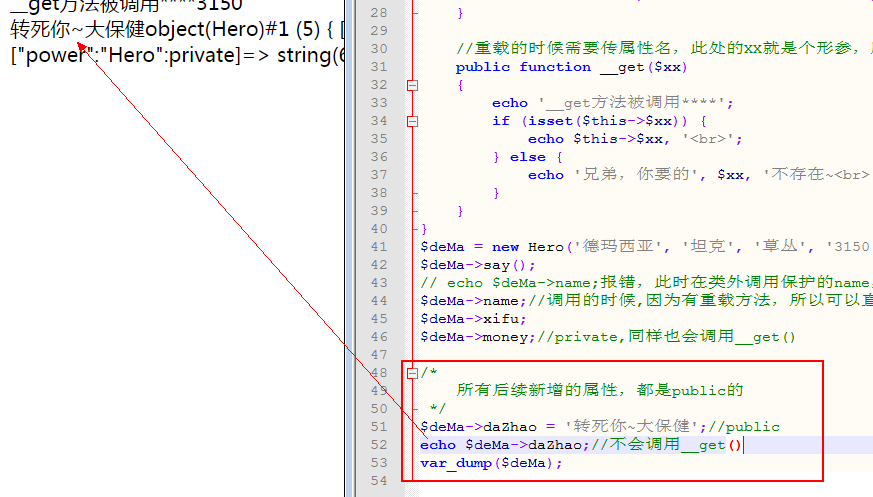
* Oop基础语法
  + 类
    - 抽象的集合就是类
    - 类中有变量，有函数，变量就是成员属性，函数就是成员方法
    - 类的命名
      * 1
      * 2
      * 3
      * 4
      * 5
      * 6
      * 7
  + 对象
    - 类的实例化结果就是对象
    - 怎么实例化对象？
      * $对象名 = new 类名()
  + 对象的复制
    - 传统的引用赋值，直接赋值的结果都是同一个对象，因为对象标识符一样
    - 对象标识符就是在原有的引用关系上多了一个标记，对象的销毁就是xx掉对象标识符
  + 对象的销毁
    - $对象 = null
  + $this
    - 用来在类的内部使用对象，谁是当前对象，$This就是谁
  + 构造函数
    - 特点：对象实例化时，第一个调用，并且自动调用的函数
    - 作用：可以用来初始化赋值
  + 析构函数
    - 特点：对象被销毁时，最后一个调用，并且自动调用的函数
    - 作用：可以用来做资源连接的关闭
  + 封装
    - 目的：隐藏内部细节
    - 如何封装？
      * 通过修饰符
        + Public:公共的
        + Protected：受保护的
        + Private：私有的
      * 作用范围：
        + 只能修饰属性和对象，不能修饰类
      * 访问的范围：
        + Public:任意位置
        + Protected:类内，子类中
        + Private:类内
      * 做了限制之后，如果想在类外访问保护的，私有的方法或者属性，需要三方方法，麻烦。怎么搞？重载

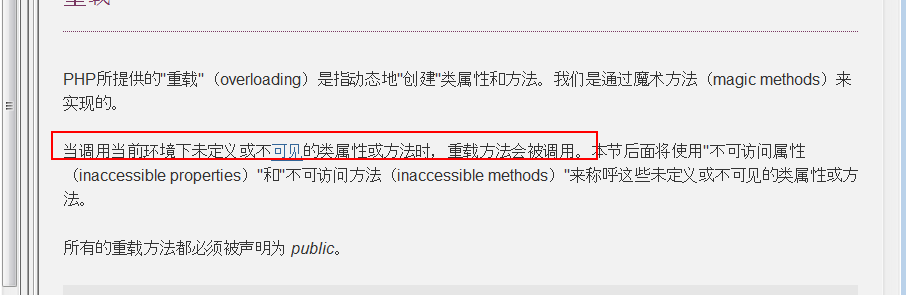
# OOP

## 封装

### 重载

* 分为属性的重载和方法的重载
* 重载是什么意思呢？
  + **就是在当前环境下（类外，子类中）调用不存在或者不可见（私有，保护的）属性和方法时，会自动调用对应的魔术方法，这个就叫重载。也就是说：public的属性和方法永远都不会用到重载**
* **注意：所有的重载方法的修饰都得是public**

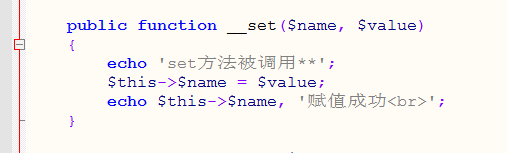






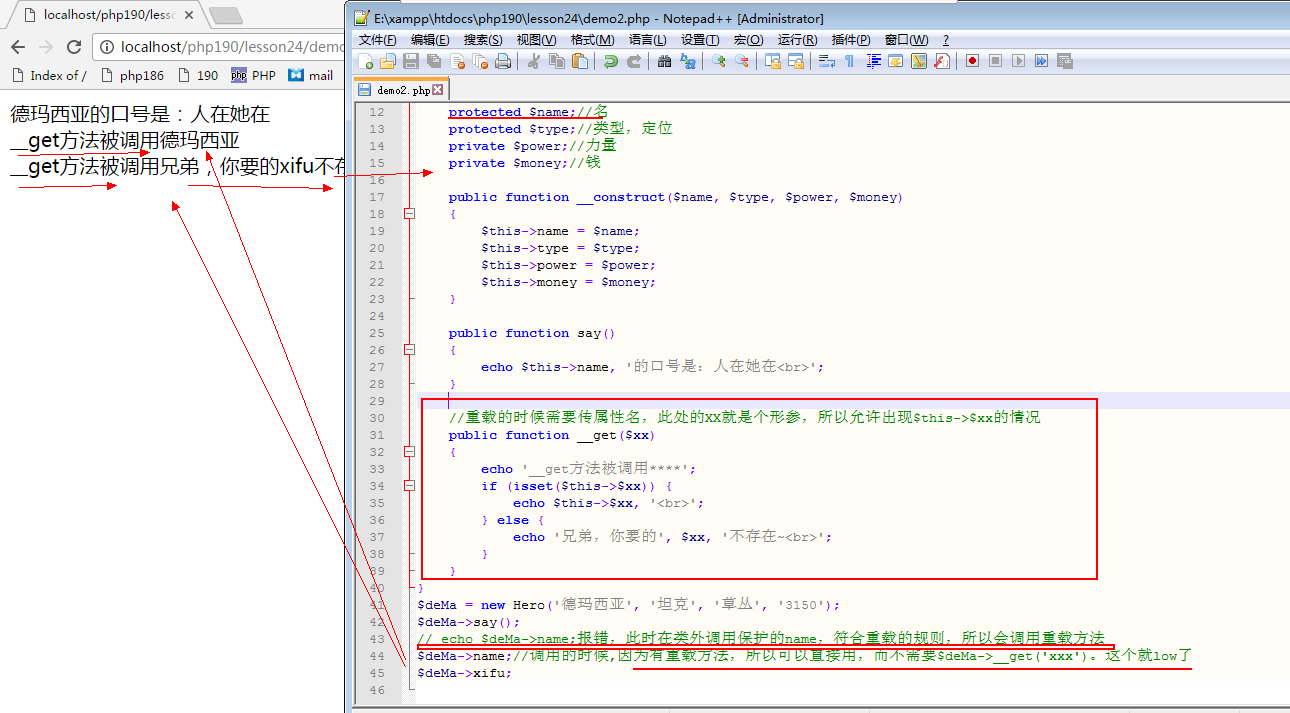
#### \_\_set(属性名的形参，属性值的形参)

* 值的设置



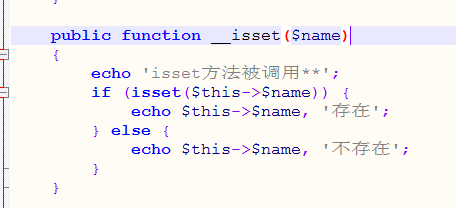
#### \_\_get(属性名的形参)

* 值的获取



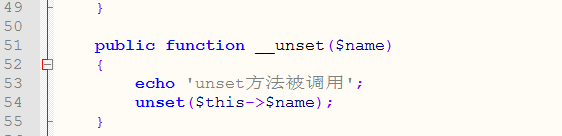
#### \_\_isset(属性名的形参)

* 属性的检测



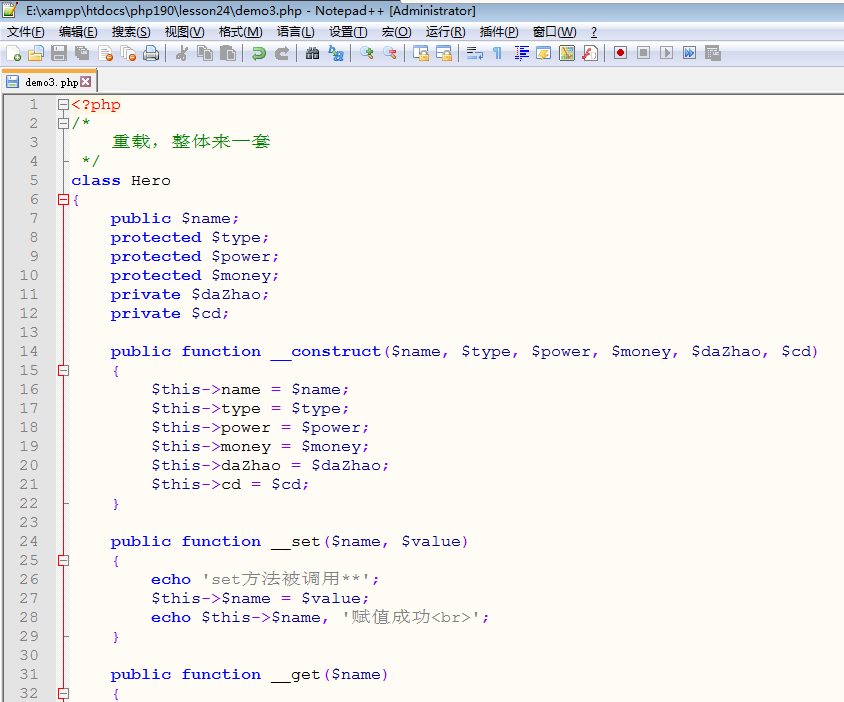
#### \_\_Unset(属性名的形参)

* 属性的销毁



#### 大例子：

详见demo3.php



### 方法的重载

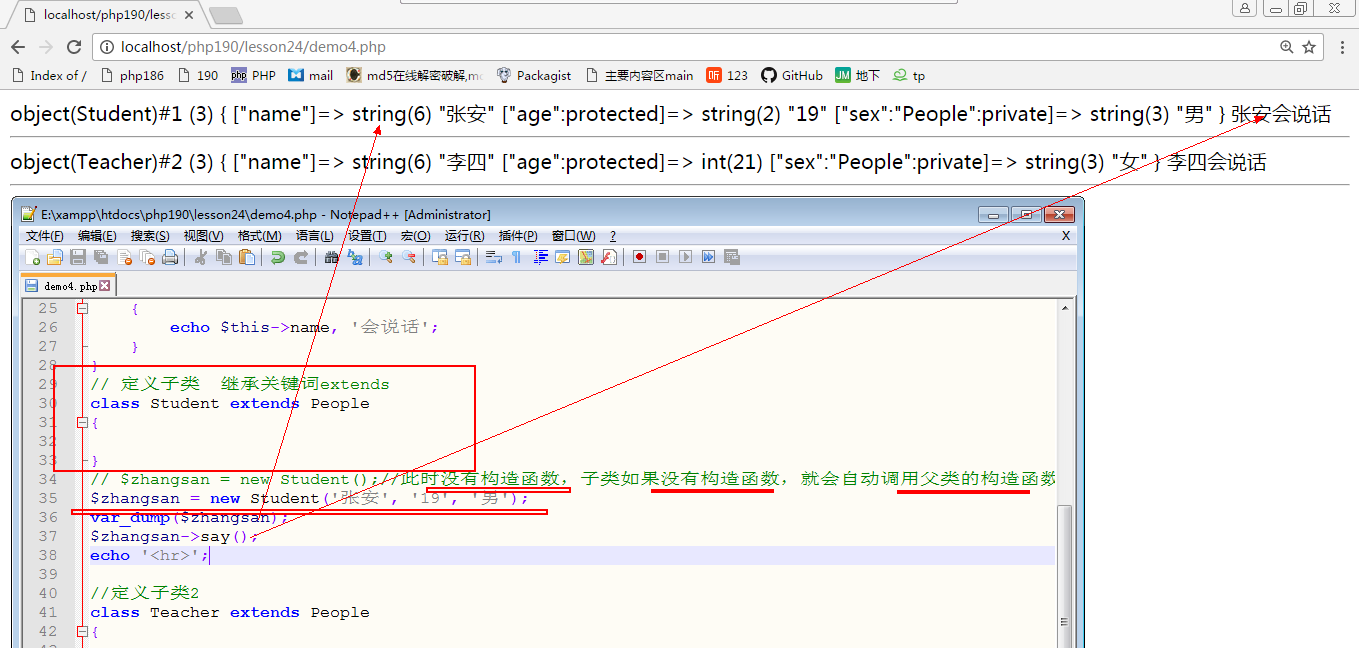
#### \_\_call()

#### \_\_callStatic()

## 继承

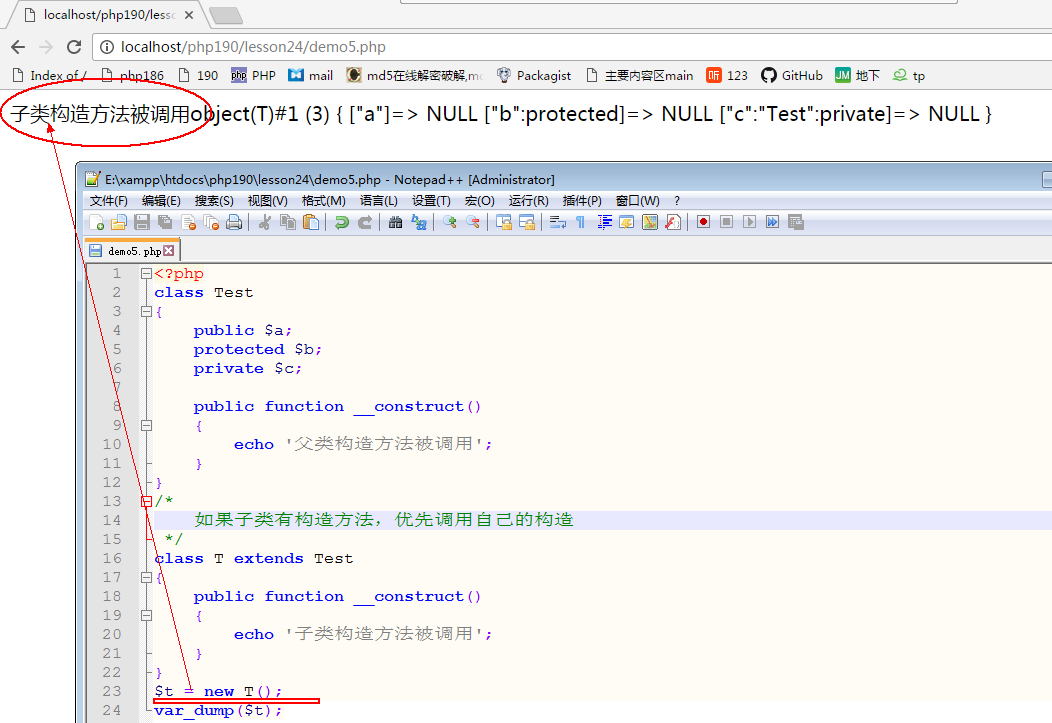
* 特点：体现代码的重用性
* **如果多个类中有同样的属性和方法，那么就完全可以把这些公有的属性和方法抽出来，单独当成一个类，其他的类，继承这个类，这样可以少写代码。这就是继承**
* **注意：php是单继承**
* **父类：具有公共属性和方法类，也叫基类**
* **子类：继承了父类的类，也叫派生类**

### 继承使用的关键词是extends

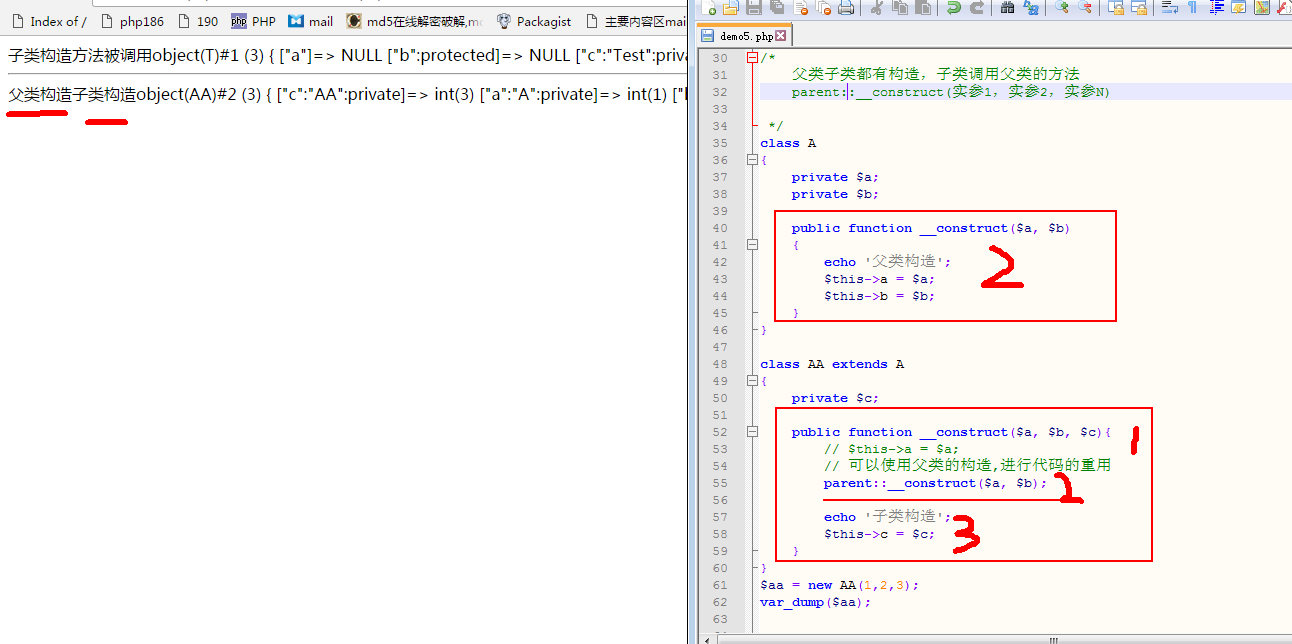


### 扩展1：构造方法的调用

* **如果子类没有构造方法会自动调用父类的构造方法，如果子类有构造方法，会先调用自己的构造方法**



* **如果子类父类都有构造方法，子类能否使用父类的构造？可以。使用关键词：parent.父类能否使用子类的构造？不行**

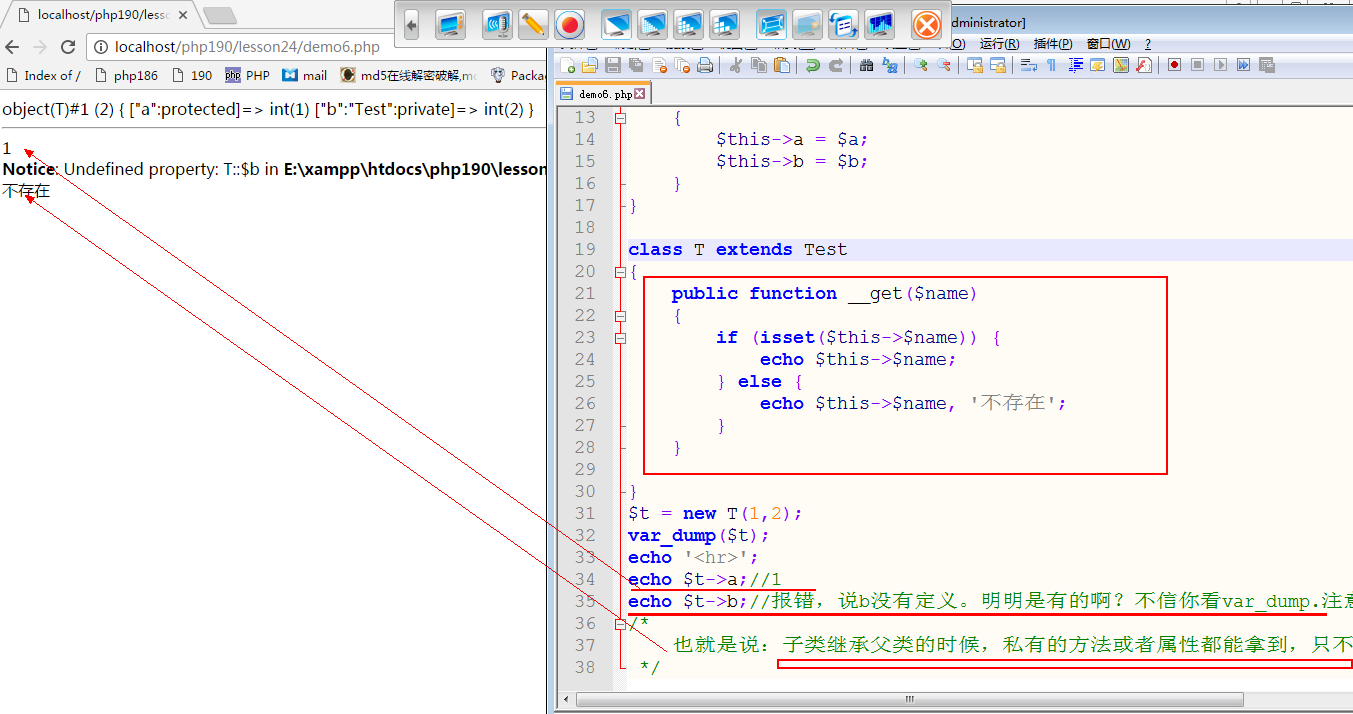


**思考？析构呢？一样的道理。子有用子，子没有，用父，都没有，那就不用呗，子有想用父，parent::\_\_destruct()**

## 扩展2：访问修饰

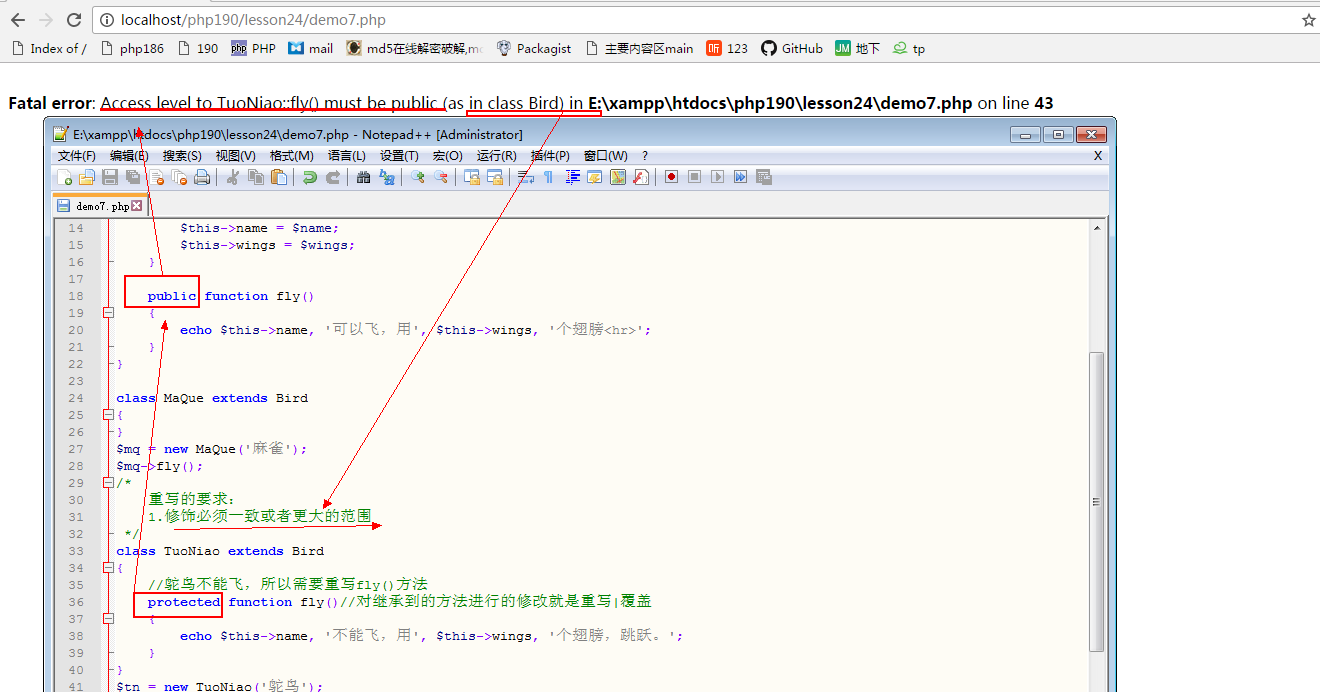
**注意：子类的继承是可以拿到父类的所有属性和方法，但是私有的方法或者属性，能看不能玩儿。**

* Protected:子类也可用
* Private:子类不可用

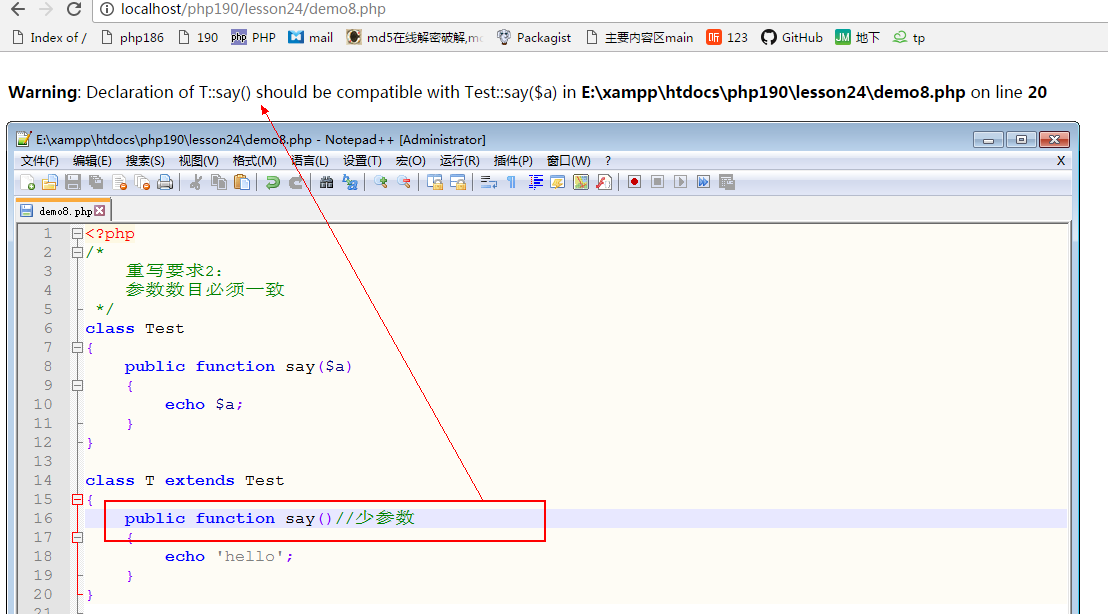


### 重写

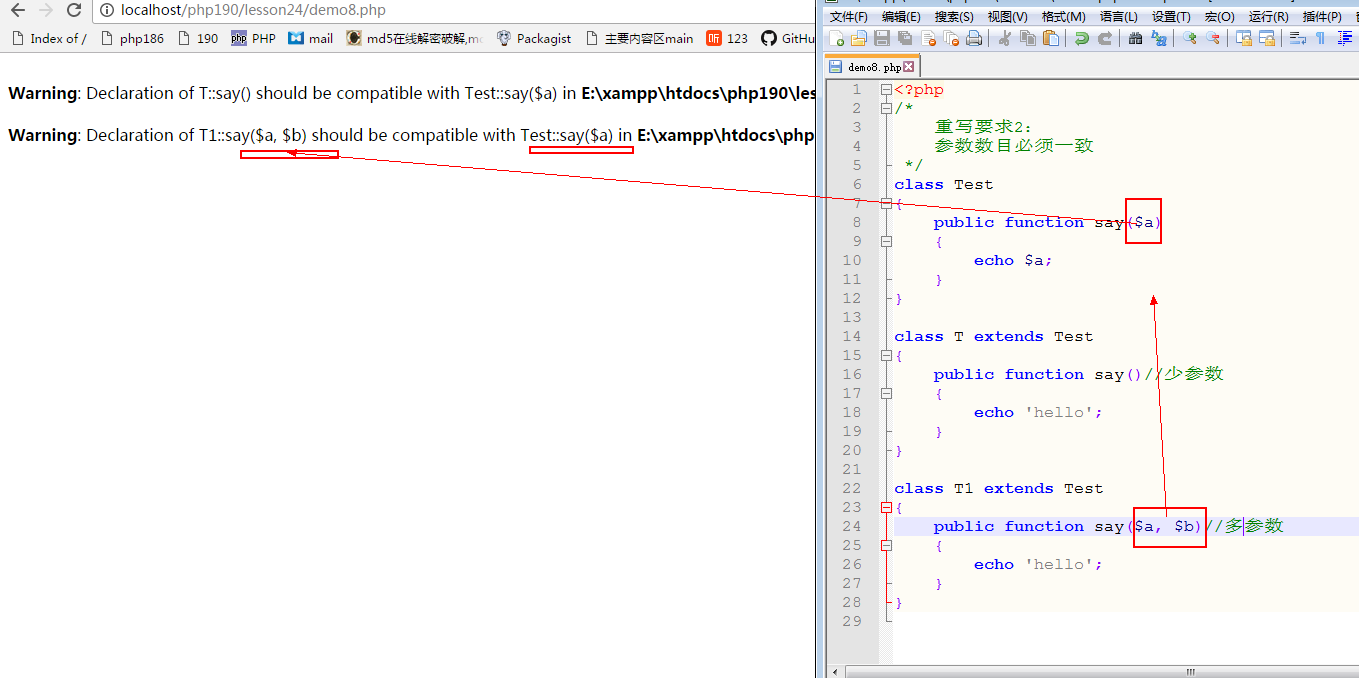
* **也就是覆盖，子类继承到的父类的方法并不合适，所以需要修改。这个修改就是重写**
* **要求：**
  + **1.子类重写的方法|属性的修饰必须和父类一致，或者比父类的修饰范围更大，也就是说：**
    - **父：public,子：必须public**
    - **父：protected，子：public,protected**
    - **父：private,子：任意**



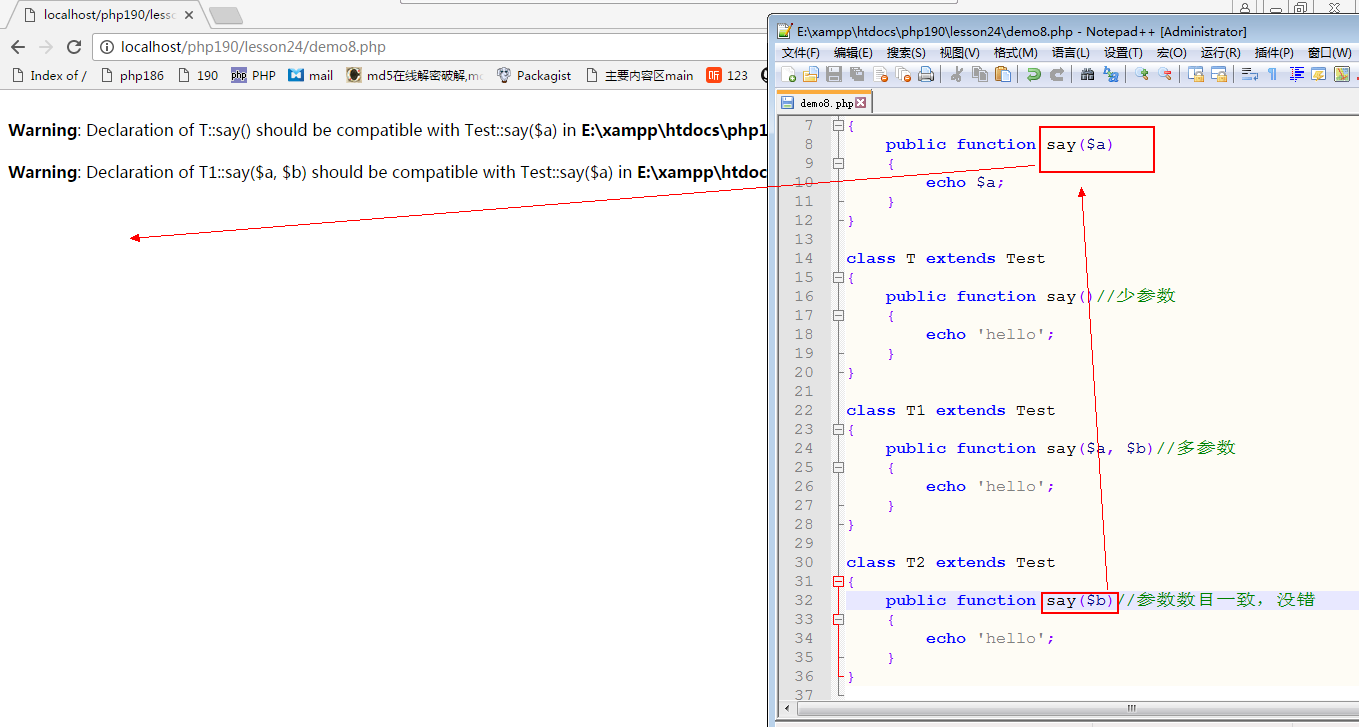
* **2.参数必须数目一致，构造函数除外，切记**
  + **示例1：少参数**



**示例2：多参数**



* **示例3：正常**



### 文件上传类的封装

详见lesson24/Upload.class.php