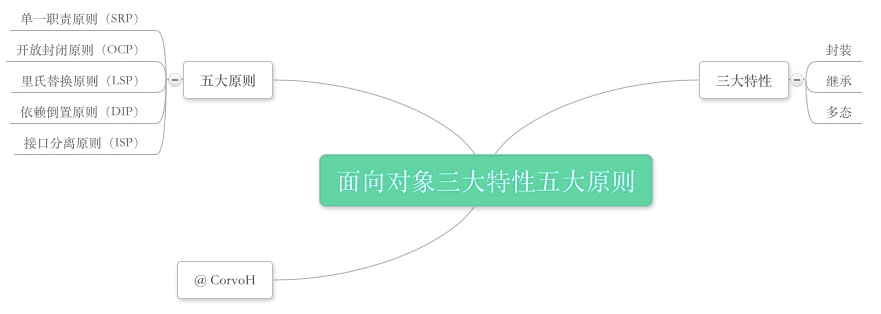
面向对象

**面向对象三大特性五大原则 + 低耦合高内聚**

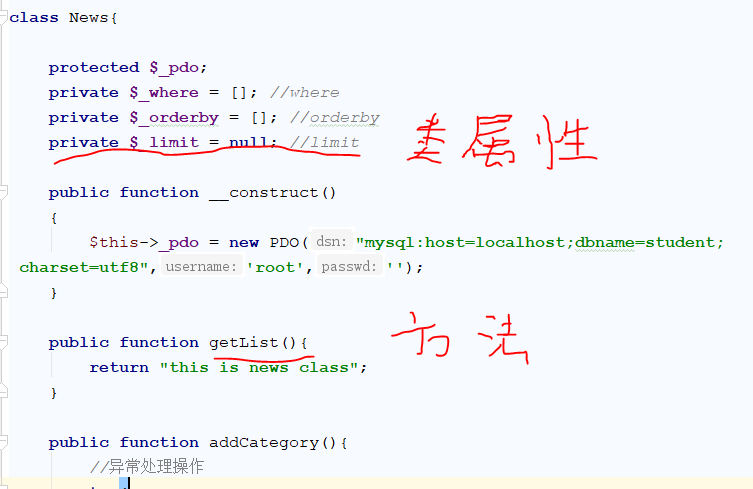
面向对象的三大特性是"封装、"多态"、"继承"，五大原则是"单一职责原则"、"开放封闭原则"、"里氏替换原则"、"依赖倒置原则"、"接口分离原则"。



什么是面向对象

面向对象(Object Oriented,OO)是软件开发方法。面向对象的概念和应用已超越了程序设计和软件开发，扩展到如数据库系统、交互式界面、应用结构、应用平台、分布式系统、网络管理结构、CAD技术、人工智能等领域。面向对象是一种对现实世界理解和抽象的方法，是计算机编程技术

拿 PHP 的 OOP 举个编程实例。



抽象类:

在面向对象的概念中，我们知道所有的对象都是通过类来描绘的，但是反过来却不是这样。并不是所有的类都是用来描绘对象的，如果一个类中没有包含足够的信息来描绘一个具体的对象，这样的类就是抽象类。抽象类往往用来表征我们在对问题领 域进行分析、设计中得出的抽象概念，是对一系列看上去不同，但是本质上相同的具体概念的抽象。

比如：如果我们进行一个图形编辑软件的开发，就会发现问题领域存在着圆、三角形这样一些具体概念，它们是不同的，但是它们又都属于形状这样一个概念，形状这个概念在问题领域是不存在的，它就是一个抽象概念。正是因为抽象的概念在问题 领域没有对应的具体概念，所以用以表征抽象概念的抽象类是不能够实例化的。

在面向对象领域，抽象类主要用来进行类型隐藏。我们可以构造出一个固定的一组行为的抽象描述，但是这组行为却能够有任意个可能的具体实现方式。这个抽象描述就是抽象类，而这一组任意个可能的具体实现则表现为所有可能的派生类。

所有抽象类和抽象方法都需要abstract关键字修饰

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的

不能定义其具体的功能实现。继承一个抽象类的时候，子类必须定义父类中的所有抽象方法

这些方法的访问控制必须和父类中一样（或者更为宽松）。例如某个抽象方法被声明为受保护的，那么子类中实现的方法就应该声明为受保护的或者公有的，而不能定义为私有的

//编写抽象类，子类，实例化对象，以及调用规则

接口，就是规定，里面只能放“抽象方法”和“常量”的一种类似类的语法结构；

——可见，接口就是“比抽象类更抽象的”一种语法结构。

接口:

使用接口（interface），可以指定某个类必须实现哪些方法，但不需要定义这些方法的具体内容。

接口是通过 *interface* 关键字来定义的，就像定义一个标准的类一样，但其中定义所有的方法都是空的。

接口中定义的所有方法都必须是公有，这是接口的特性。

要实现一个接口，使用 *implements* 操作符。类中必须实现接口中定义的所有方法，否则会报一个致命错误。类可以实现多个接口，用逗号来分隔多个接口的名称。

实现多个接口时，接口中的方法不能有重名。

接口也可以继承，通过使用extends操作符.

类要实现接口，必须使用和接口中所定义的方法完全一致的方式。否则会导致致命错误.

**区别:**

1.对接口的继承使用implements,抽象类使用extends.

2.接口中不可以声明变量,但可以声明类常量.抽象类中可以声明各种变量

3.接口没有构造函数,抽象类可以有

4.接口中的方法默认为public,抽象类中的方法可以用public,protected,private修饰

5.一个类可以继承多个接口,但只能继承一个抽象类

三大基本特性：封装，继承，多态

**封装**

封装，就是把客观事物封装成抽象的类，并且类可以把自己的数据和方法只让可信的类或者对象操作，对不可信的进行信息隐藏。一个类就是一个封装了数据以及操作这些数据的代码的逻辑实体。在一个对象内部，某些代码或某些数据可以是私有的，不能被外界访问。通过这种方式，对象对内部数据提供了不同级别的保护，以防止程序中无关的部分意外的改变或错误的使用了对象的私有部分。

**继承**

继承，指可以让某个类型的对象获得另一个类型的对象的属性的方法。它支持按级分类的概念。继承是指这样一种能力：它可以使用现有类的所有功能，并在无需重新编写原来的类的情况下对这些功能进行扩展。 通过继承创建的新类称为“子类”或“派生类”，被继承的类称为“基类”、“父类”或“超类”。继承的过程，就是从一般到特殊的过程。要实现继承，可以通过 “继承”（Inheritance）和“组合”（Composition）来实现。

继承概念的实现方式有二类：实现继承与接口继承。

实现继承是指直接使用 基类的属性和方法而无需额外编码的能力；

接口继承是指仅使用属性和方法的名称、但是子类必须提供实现的能力。-------implements

**多态**

多态，是指一个类实例的相同方法在不同情形有不同表现形式。

多态机制使具有不同内部结构的对象可以共享相同的外部接口。

这意味着，虽然针对不同对象的具体操作不同，但通过一个公共的类，它们（那些操作）可以通过相同的方式予以调用。

-----------------------------------------

多态实现的前提：必须是类与类之间要有关系，要么继承，要么实现，存在重写（override），其实就是抽象函数或接口。

多态的好处：大大提高程序的扩展，增强系统的灵活性，降低模块间的耦合。

多态的实现必须要满足三个条件：一是子类继承父类，二是子类重写父类的方法，三是父类引用指向子类对象。

抽象类的实现多态。