

提高：封装、继承、多态

笔记本：Python提高-1

创建时间：2018/5/6 21:15

作者：985601646@qq.com

更新时间：2018/5/6 22:05

面向对象的3大特性：封装、继承、多态

封装

why?

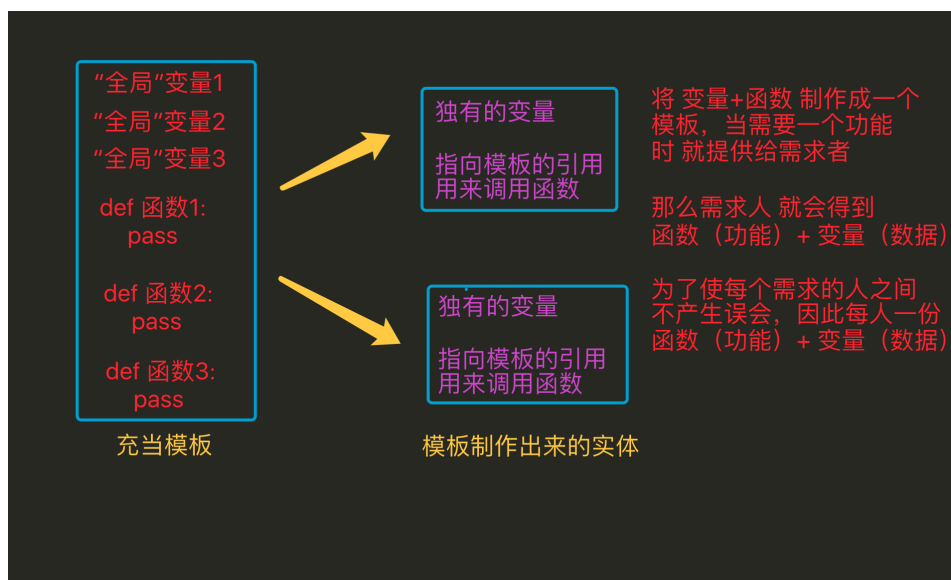
变量：全局的变量任何函数（传递方式）和变量都能使用，导致该变量不断重复的被使用，出现的一些污染问题。而**封装**到（类）起来之后，就只有内部能使用（从类实例化来的对象只能使用该类的内部的变量，保证其他外部无法使用）

好处：

保护公共变量，方便使用。

增加了代码的复用率：需要重复使用时，只需重复调用封装好功能的代码

划分更清晰：让主程序和功能区分严明，各司其职



继承

1，代码级别：减少重复代码的冗余

2，工作层面：团队开发中，当书写的功能和同事的重复时，使用继承同事的模块，可以直接使用相同的部分，不同的部分进行重新父类方法改变，既不会改变同事的源码，也能实现自己的功能。

能够提升代码的复用率，即开发一个类，可以在多个子功能中直接使用

继承能够有效的进行代码的管理，当某个类有问题只要修改这个类就行，而其继承这个类的子类往往不需要就修改

这是一个类有 很多代码

类1

这里要对类1进行功能的扩充
多某些方法进行适当修改

方式1：重新自己写一个 新版本

方式2：在类1的基础上修改

你会选择哪种？？？

继承类1 不同的地方进行重新父类

多态：

多态的发生条件 （缺一不可）

- 1 发生继承**
- 2 子类重写父类方法**
- 3 被重写的父类方法被调用**

补充：一切皆对象

类、实例对象、字符串、变量都是类去创建的实例化对象
而类也是对象 类是由type(原类)实例化来的对象
object只是所有类都继承父类而已 并不是类型

```
In [12]: class_ = 10
In [13]: a = 10
In [14]: a.__class__
Out[14]: int
In [15]: b = "123"
In [16]: b.__class__
Out[16]: str
In [17]: def test():
785669...:     pass
...:
In [18]: test.__class__
Out[18]: function
In [19]: class test:
...:     pass
...:
In [20]: test.__class__
Out[20]: type
In [21]: a.__class__
Out[21]: int
In [22]: a.__class__.__class__
Out[22]: type
In [23]: a.__class__.__class__.__class__
Out[23]: type
In [24]:
```