

面向对象编程：

T1:

什么是类：

个人理解，就是c++中的“结构体”，通过一个统一的命名，统筹一系列变量、函数。（好像说的不是很清楚？）

什么是对象：

个人认为，是一种在“类”的统筹下产生的“自定义”变量，这个自定义变量可以运用相应类的函数和属性，类比c++下的结构体变量

两者联系：

- 1.对象是附属于类的，没有类就没有对象。对象可以视作“类”的具体实现
- 2.一个类下可以定义多个不同属性的对象。
- 3.（感觉好像就1、2两点？）

T2 (3.1) :

观察会发生什么，并说明原因：

观察到会报错：AttributeError: 'Student' object has no attribute '__scores'

原因是Python 对以双下划线开头的属性会进行名称修饰，实际存储的属性名会被改变：

`__scores`被自动重命名为`Student__scores`

T3:

什么是继承：

可以认为是附属在类下的一个更细节的类，比如Animal下可以有Cat、Dog等等。继承的类可以拥有父类的属性和函数，也可以在父类的基础上增加属性和函数。

好处：

- 1.可以更加细化类，从而条分缕析
- 2.可以在原本没有的基础上增加属性和函数，更好地完成题目要求

T4:

什么是多态:

可以说: 不同的对象对同一消息做出不同的响应

就像给的例题一样, 不同的speak有不同的回应

多态需要建立在继承上比较好

T5:

什么是迭代器:

我的印象中 (c++) , 迭代器可以视作一种指针, 可以访问元素储存的位置, 一般用于遍历。一般是一次性的, 使用完就废掉。

要实现需要实现__iter__()和 __next__()

可迭代对象:

可以多次使用, 只能用在迭代器上, 如列表、元组、字典、集合、字符串等.

只需要实现__iter__()

iter和next:

iter(): 获取可迭代对象的迭代器

next(): 从迭代器中获取下一个元素

简单代码:

```
my_list=[1,2,3,4]
iterator=iter(my_list)
while True:
    try:
        element=next(iterator)
        print(element)
    except StopIteration:
        break
```