

```

1  #201014_面向对象高级
2  #私有属性,在属性的前面加__就是私有属性
3  #私有方法,在方法的前面加__就是私有方法
4  #私有属性和私有方法不能被外部直接访问,也不能被子类直接继承
5  # class Cls1:
6  #     __a=10  #私有属性
7  #     def __init__(self):
8  #         pass
9  #     def abc(self):
10 #         print(self.__a)  #私有属性可以被类当中的方法访问
11 #         self.__bbc()  #私有方法可以被类当中的其他方法访问
12 #     def __bbc(self):  #私有方法,不能被直接访问
13 #         print('这是一个私有方法')
14 # cls1=Cls1()
15 # cls1.abc()
16
17 # class Cls2(Cls1):
18 #     pass
19 # cls2=Cls2()
20 # cls2.__bbc()  #不可以直接访问这个方法
21 # cls2.abc()  #可以间接使用__bbc方法
22
23 #python中所有的类,其实都是object的子类,无论是否声明继承object,其实都继承了
24 # class Class1(object):
25 #     '''
26 #     注释1
27 #     '''
28 #     pass
29 #
30 # class Class2:
31 #     '''
32 #     注释2
33 #     '''
34 #     pass
35
36 # print(Class2.__doc__)  #显示类的注释
37 # print(Class2.__name__)  #显示类的名称
38 # print(Class2.__bases__)  #显示类的父类的名称
39
40 #多继承
41 # class Airen:
42 #     def dayin(self):
43 #         print('矮人')
44 # class Shengqishi:
45 #     def dayin2(self):
46 #         print('圣骑士')
47 #
48 # class AirenShengqishi(Airen,Shengqishi):
49 #     pass
50 # airenshengqishi=AirenShengqishi()
51 # airenshengqishi.dayin()  #方法名冲突时,按优先级顺序继承
52 # airenshengqishi.dayin2()
53
54 #多态
55 #狗,猫,它们都是动物

```

```
56 # class Animal:
57 #     def say(self):
58 #         pass
59 # class Dog(Animal):
60 #     def say(self):
61 #         print('汪汪')
62 # class Cat(Animal):
63 #     def say(self):
64 #         print('喵喵')
65 # def animal_say(obj):
66 #     obj.say()
67 # dog=Dog()
68 # cat=Cat()
69 # animal_say(cat)
70
71 class Fanguan:
72     pass
73 class Yuxiangrousi(Fanguan):
74     def caidan(self):
75         print('鱼香肉丝')
76 class Gongbaojiding(Fanguan):
77     def caidan(self):
78         print('宫保鸡丁')
79 class Qingjiaotudousi(Fanguan):
80     def caidan(self):
81         print('青椒土豆丝')
82 def fuwuyuan(obj):
83     obj.caidan()
84 guke1=Yuxiangrousi()
85 guke2=Gongbaojiding()
86 guke3=Qingjiaotudousi()
87 fuwuyuan(guke3)
88
89 #上节课思考题:
90 class sanjiaoxing:
91     def __init__(self,a,b,c):
92         self.a=a
93         self.b=b
94         self.c=c
95     def zhouchang(self):
96         return self.a+self.b+self.c
97     def mianji(self):
98         p=(self.a+self.b+self.c)/2
99         return ((p-self.a)*(p-self.b)*(p-self.c)*p)**0.5
```