

PYTHON

编程基础

方法重写和鸭子类型

方法重写



方法重写

是指子类可以对从父类中继承过来的方法进行重新定义，从而使得子类对象可以表现出与父类对象不同的行为。



例：

方法重写示例。

方法重写

1 `class Person: #定义Person类`

2 `def __init__(self, name): #定义构造方法`

3 `self.name=name #将self对象的name属性赋为形参name的值`

4 `def PrintInfo(self): #定义PrintInfo方法`

5 `print('姓名 : %s'%self.name)`

方法重写

6 `class Student(Person):` #以Person类作为父类定义子类Student

7 `def __init__(self, sno, name):` #定义构造方法

8 `self.sno=sno` #将self对象的sno属性赋为形参sno的值

9 `self.name=name` #将self对象的name属性赋为形参name的值

10 `def PrintInfo(self):` #定义PrintInfo方法

11 `print('学号 : %s , 姓名 : %s'%(self.sno,self.name))`

方法重写



方法重写

```
12     def PrintPersonInfo(person): #定义普通函数PrintPersonInfo
13         print('PrintPersonInfo函数中的输出结果', end='#')
14         person.PrintInfo() #通过person调用PrintInfo方法
```

方法重写



方法重写

```
15     if __name__ == '__main__':  
16         p=Person('李晓明') #创建Person类对象p  
17         stu=Student('1810100','李晓明') #创建Student类对象stu  
18         p.PrintInfo()  
19         stu.PrintInfo()  
20         PrintPersonInfo(p)  
21         PrintPersonInfo(stu)
```

方法重写

姓名

▶ 李晓明

学号

▶ 1810100 , 姓名 : 李晓明

PrintPersonInfo函数中的输出
结果#姓名 :

▶ 李晓明

PrintPersonInfo函数中的输出
结果#学号 :

▶ 1810100 , 姓名 : 李晓明

提示：多态，是指在执行同样代码的情况下，系统会根据对象实际所属的类去调用相应类中的方法。

鸭子类型



在鸭子类型中，关注的不是对象所属的类，而是一个对象能够如何使用。

例：鸭子类型示例。

在Python中编写一个函数，传递实参前其参数的类型并不确定，在函数中使用形参进行操作时只要传入的对象能够支持该操作程序就能正常执行。

鸭子类型



示例

```
1 class Person: #定义Person类
2     def CaptureImage(self): #定义CaptureImage方法
3         print('Person类中的CaptureImage方法被调用！')
4 class Camera: #定义Camera类
5     def CaptureImage(self): #定义CaptureImage方法
6         print('Camera类中的CaptureImage方法被调用！')
```

鸭子类型



示例

```
7     def CaptureImageTest(arg): #定义CaptureImageTest方法
8         arg.CaptureImage() #通过arg调用CaptureImage方法
9     if __name__ == '__main__':
10         p=Person() #定义Person类对象p
11         c=Camera() #定义Camera类对象c
12         CaptureImageTest(p)
13         CaptureImageTest(c)
```

鸭子类型

Person类中的CaptureImage方法被调用！

Camera类中的CaptureImage方法被调用！

提示



实际上，Python中的多态也是借助鸭子类型实现，与C++、Java等语言中的多态并不是同一含义。