按要求定义类

考查知识点：super关键字的使用和继承中方法的调用

**要求：**

1、定义老手机类，有品牌属性，且属性私有化，提供相应的getXxx与setXxx方法，提供无返回值的带一个Str类型参数的打电话的方法，内容为：“正在给xxx打电话...”

2、定义新手机类，继承老手机类，重写父类的打电话的方法，内容为2句话：“语音拨号中...”、“正在给xxx打电话...”要求打印“正在给xxx打电话...”这一句调用父类的方法实现，不能在子类的方法中直接打印；提供无返回值的无参数的手机介绍的方法，内容为：“品牌为：xxx的手机很好用...”

3、定义测试类，创建新手机对象，并使用该对象，对父类中的品牌属性赋值；

4、使用新手机对象调用手机介绍的方法；

5、使用新手机对象调用打电话的方法；

import time  
class OldPhone:  
 \_\_brand = **""** def setBrand(self,brand):  
 self.\_\_brand = brand  
 def getBrand(self):  
 return self.\_\_brand  
 def call(self,num,):  
 print(**"品牌为： "**,self.\_\_brand,**"的手机很好用"**,**"正在给"**,num,**"打电话"**,end=**""**)  
 for i in range(8):  
 print(**"."**,end=**""**)  
 time.sleep(1)  
 i = i + 1  
 def \_\_init\_\_(self,brand):  
 self.\_\_brand = brand  
class NewPhone(OldPhone):  
 def call(self,phone):  
 print(**"正在打电话中"**,end=**""**)  
 for i in range(8):  
 print(**"."**,end=**""**)  
 time.sleep(1)  
 i = i + 1  
  
 print(**"品牌为： "**,phone,**"的手机很好用"**,end=**""**)  
 for i in range(8):  
 print(**"."**,end=**""**)  
 time.sleep(1)  
 i = i + 1  
*# c = NewPhone()  
# c.call("菜鸟")*class Test(OldPhone):  
 pass  
  
test = Test(**"华为"**)  
test.call(123456)  
print(Test.\_\_mro\_\_)

## 题目一：

考查知识点：继承的传递性

按要求定义类

**要求：**

1. 定义厨师类，有姓名和年龄的属性，且属性私有化，提供相应的getXxx与setXxx方法，提供无返回值的无参数的蒸饭方法；

class Cooker:  
 \_\_name = **""** \_\_age = 0  
 def setName(self,name):  
 self.\_\_name = name  
 def getName(self):  
 return self.\_\_name  
 def setAge(self,age):  
 self.\_\_age = age  
 def getAge(self):  
 return self.\_\_age  
 def cook(self):  
 print(**"厨师正在烤羊肉！！！！！"**)

1. 定义厨师的子类，该类中要求只能写一个无返回值的无参数的炒菜的方法，其他的方法不能写；

class StirFry（cooker）:  
 def stir\_fry(self):  
 print(**"只要把面煮熟了就行！！！！！"**)

1. 定义厨师的子类的子类，重写所有父类的方法，每个方法的内容只需打印一句话描述方法的功能即可；

class Cooker:  
 \_\_name = **""** \_\_age = 0  
 def setName(self,name):  
 self.\_\_name = name  
 def getName(self):  
 return self.\_\_name  
 def setAge(self,age):  
 self.\_\_age = age  
 def getAge(self):  
 return self.\_\_age  
 def cook(self):  
 print(**"厨师正在烤羊肉！！！！！"**)

class StirFry(Cooker):  
 def stir\_fry(self):  
 print(**"只要把面煮熟了就行！！！！！"**)

c =StirFry()  
c.cook()  
c.stir\_fry()

1. 定义测试类，创建厨师的子类的子类（厨师的孙子类）对象，使用该对象，对厨师类中的姓名和年龄属性赋值，并获取赋值后的属性值打印到控制台上；

class Cooker:  
 \_\_name = **""** \_\_age = 0  
 def setName(self,name):  
 self.\_\_name = name  
 def getName(self):  
 return self.\_\_name  
 def setAge(self,age):  
 self.\_\_age = age  
 def getAge(self):  
 return self.\_\_age  
 def cook(self):  
 print(**"厨师正在烤羊肉！！！！！"**)  
 def \_\_init\_\_(self,name,age):  
 print(**"姓名："**,name,**"年龄："**,age,end=**""**)

class Children(Cooker):  
 pass  
c =Children(**"李四"**,89)

5、使用厨师的孙子类对象调用该对象除了getXxx与setXxx以外的其他方法；

class Cooker:  
 \_\_name = **""** \_\_age = 0  
 sex = **""** high = **""** def setName(self,name):  
 self.\_\_name = name  
 def getName(self):  
 return self.\_\_name  
 def setAge(self,age):  
 self.\_\_age = age  
 def getAge(self):  
 return self.\_\_age  
 def cook(self):  
 print(**"厨师正在烤羊肉！！！！！"**)  
 *# def \_\_init\_\_(self,name,age):  
 # print("姓名：",name,"年龄：",age,end="")* def el(self,sex,high):  
 print(sex,high)

class Children(Cooker):  
 pass  
c =Children()  
c.el(**"男"**,**"156"**)

## 请编程

* + 1. 人：年龄，性别，姓名。

|  |
| --- |
| class Person:  age = 0  sex = **""** name = **""** |

* + 1. 现在有个工种，工人：年龄，性别，姓名 。行为：干活。请用继承的角度来实现该类。

|  |
| --- |
| class Workers(Person):  def behavior(self,work):  print(**"工人在"**,work)  w = Workers() w.sex = **"男"** w.age = 35 w.name = **"lisa"** w.behavior(**"开车"**) |

* + 1. 现在有学生这个工种，学生：年龄，性别，姓名，学号。行为：学习，唱歌。请结合上面的几个题目用继承的角度来实现。

|  |
| --- |
| class Student(Person):  ID = **""** def behavior(self,name,singing,dancing):  print(name,**"同学，不但会"**,singing,**"还会跳舞"**,dancing)  s = Student() s.age = 23 s.name = **"lisi"** s.sex = **"女"** s.ID = **"123456789"** s.behavior(**"lisa"**,**"唱歌"**,**"跳舞"**) |