## 书写一个计算器，有加减乘除四大方法。写一个测试类对计算器的加减乘除进行调用。

numerator = Numerator()  
while True:  
 num1 = float(input(**"请输入第一个数："**))  
 num2 = float(input(**"请输入第二个数："**))  
 print(**"1:加法 2：减法 3：乘法 4：除法 5:退出"**)  
 choose = input(**"请输入选项："**)  
 if choose == **'1'**:  
 print(**"结果为："**, numerator.jia(num1, num2))  
 elif choose == **'2'**:  
 print(**"结果为："**, numerator.jian(num1, num2))  
 elif choose == **'3'**:  
 print(**"结果为："**, numerator.cheng(num1, num2))  
 elif choose == **'4'**:  
 print(**"结果为："**, numerator.chu(num1, num2))  
 else:  
 break

class Numerator:  
 def jia(self,num1, num2):  
 return num1 + num2  
 def jian(self,num1, num2):  
 return num1 - num2  
 def cheng(self,num1, num2):  
 return num1 \* num2  
 def chu(self,num1, num2):  
 return num1 / num2

按要求定义类

考查知识点：super关键字的使用

**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String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；

测试类

phone1 = Oldphone()  
phone2 = Newphone()  
print(**"旧手机："**)  
phone1.call(1234556)  
print(**""**)  
phone2.setSign(**"oppo"**)  
print(**"新手机："**)  
phone2.call(1234124)

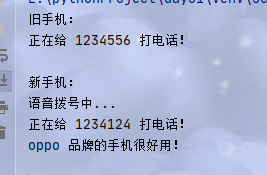
旧手机类

class Oldphone:  
 \_\_sign = None  
 def setSign(self,sign):  
 self.\_\_sign = sign  
 def getSign(self):  
 return self.\_\_sign  
 def call(self,number):  
 print(**"正在给"**,number,**"打电话！"**)

新手机类

class Newphone(Oldphone):  
 def call(self,number):  
 print(**"语音拨号中..."**)  
 super().call(number)  
 print(super().getSign(),**"品牌的手机很好用！"**)

结果：



## 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；

2、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；

3、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；

4、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；

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；

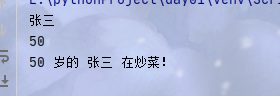
测试类

grandson = Grandson()  
grandson.setName(**"张三"**)  
grandson.setAge(50)  
print(grandson.getName())  
print(grandson.getAge())  
grandson.cook()

厨师类

class Chef:  
 \_\_name = None  
 \_\_age = None  
 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  
class Son(Chef):  
 def cook(self):  
 print(super().getName(),**"在炒菜！"**)  
class Grandson(Son):  
 def cook(self):  
 print(super().getAge(), **"岁的"**, super().getName(), **"在炒菜！"**)

结果：



## 请编程

* + 1. 人：年龄，性别，姓名。

|  |
| --- |
| class People:  \_\_name = None  \_\_age = None  \_\_sex = None  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 setSex(self,sex):  self.\_\_sex = sex  def getSex(self):  return self.\_\_sex |

* + 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 WorkPeople(People):  def work(self):  print(super().getAge(),**"岁的工人"**,super().getName(),**"在干活。"**) |

* + 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(People):  def study(self):  print(super().getAge(), **"岁的"**,super().getSex(),**"学生"**, super().getName(), **"在学习。"**)    def sing(self):  print(super().getAge(), **"岁的"**, super().getSex(), **"学生"**, super().getName(), **"在唱歌。"**) |