## 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。

class Calculator:  
  
 def add(self,\*args):  
 sum=0  
 for i in args:  
 sum+=i  
 print(sum)  
 *# sum = self.\_\_num + self.\_\_num1  
 # print(self.\_\_num,"+",self.\_\_num1,"=",sum)* def subtract(self,\*args):  
 sub=args[0]  
 for i in args[1:]:  
 sub-=i  
 print(sub)  
 *# sub = self.\_\_num - self.\_\_num1  
 # print(self.\_\_num,"-",self.\_\_num1,"=",sub)* def multiply(self,\*args):  
 mul=1  
 for i in args:  
 mul\*=i  
 print(mul)  
 *# mul = self.\_\_num \* self.\_\_num1  
 # print(self.\_\_num,"×",self.\_\_num1,"=",mul)* def divide(self,\*args):  
 div=args[0]  
 for i in args[1:]:  
 div/=i  
 print(div)  
 *# div = self.\_\_num / self.\_\_num1  
 # print(self.\_\_num,"÷",self.\_\_num1,"=",div)*c=Calculator()  
c.add(23,34,56,78)  
c.subtract(85,43,23,13,12)  
c.multiply(1,2,3,4,5,6)  
c.divide(8,2,2)

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；

class Oldphone:  
 \_\_brand = None  
  
 def \_\_init\_\_(self,brand):  
 self.\_\_brand = brand  
  
 def setBrand(self,brand):  
 self.\_\_brand = brand  
 def getBrand(self):  
 return self.\_\_brand  
  
 def call(self,string):  
 print(**"正在给"**,string,**"打电话"**)  
  
  
  
class Newphone(Oldphone):  
 def \_\_init\_\_(self,brand):  
 super().\_\_init\_\_(brand)  
  
 def call(self,string):  
 super().call(string)  
 print(**"语音拨号中......"**)  
  
 def show(self):  
 print(**"品牌为："**,self.getBrand(),**"的手机很好用！"**)  
  
n=Newphone(**"三星"**)  
n.call(**"曹东雪"**)  
n.show()

## 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；

class Chef:  
 \_\_name = None  
 \_\_age = None  
  
 def \_\_init\_\_(self,name,age):  
 self.\_\_name = name  
 self.\_\_age = age  
  
 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 steamRice(self):  
 print(**"先放水再放米"**)  
  
class Xchef(Chef):  
  
 def aFriedDish(self):  
 print(**"先放油再放菜"**)  
  
class Xxchef(Xchef):  
 def steamRice(self):  
 super().steamRice()  
  
  
 def aFriedDish(self):  
 super().aFriedDish()  
  
  
x=Xxchef(**"曹东雪"**,18)  
print(x.getName(),x.getAge())  
x.steamRice()  
x.aFriedDish()

## 请编程

* + 1. 人：年龄，性别，姓名。

|  |
| --- |
| class Human:  \_\_age = None  \_\_sex = None  \_\_name = None   def \_\_init\_\_(self,age,sex,name):  self.\_\_age = age  self.\_\_sex = sex  self.\_\_name = 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   def \_setName(self,name):  self.\_\_name = name  def getName(self):  return self.\_\_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 Worker(Human):   def \_\_init\_\_(self,age,sex,name):  super().\_\_init\_\_(age,sex,name)   def work(self):  print(**"我叫"**,self.getName(),self.getSex(), **"，今年"**,self.getAge(), **"岁!每天的工作就是搬砖"**)   w=Worker(23,**"女"**,**"曹东雪"**) w.work() |

* + 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 Worker(Human):   def \_\_init\_\_(self,age,sex,name):  super().\_\_init\_\_(age,sex,name)   def work(self):  print(**"我叫"**,self.getName(),self.getSex(), **"，今年"**,self.getAge(), **"岁!每天的工作就是搬砖"**)   w=Worker(23,**"女"**,**"曹东雪"**) w.work() |