## 题目一：

该题考查点：属性和方法的使用！

定义一个空调类和对应的测试类

**要求：**

1. 空调有品牌和价格两个属性，并且将属性私有化，提供公有的getXxx与setXxx方法对属性赋值和取值；
2. 提供一个无返回值的无参数的开机的方法，内容打印一句话：“空调开机了...”；
3. 提供一个无返回值的带1个int类型参数的定时关机的方法,(int类型的参数表示设定的分钟数)，内容打印一句话：“空调将在xxx分钟后自动关闭...”；
4. 在测试类中创建出空调对象，并给空调的品牌和价格赋任意值；
5. 使用空调对象获取空调的品牌和价格并打印到控制台上；
6. 使用空调对象调用开机方法；
7. 使用空调对象调用定时关机方法，并传递具体数据值，在控制台上可以看到的效果为：空调将在xxx分钟后自动关闭...

其中语句中的“xxx”是调用方法时传递的具体数据值；

*题目一：  
# 该题考查点：属性和方法的使用！  
# 定义一个空调类和对应的测试类  
# 要求：*class AirConditioner:  
 *# 1、空调有品牌和价格两个属性，并且将属性私有化，提供公有的getXxx与setXxx方法对属性赋值和取值；* \_\_brand = **""** \_\_price = 0  
 def setBrand(self,brand):  
 self.\_\_brand = brand  
  
 def getBrand(self):  
 return self.\_\_brand  
  
 def setPrice(self,price):  
 self.\_\_price = price  
  
 def getPrice(self):  
 return self.\_\_price  
*# 2、提供一个无返回值的无参数的开机的方法，内容打印一句话：“空调开机了...”；* def open(self):  
 print(**"空调开机了。。。。"**)  
*# 3、提供一个无返回值的带1个int类型参数的定时关机的方法,(int类型的参数表示设定的分钟数)，内容打印一句话：“空调将在xxx分钟后自动关闭...”；* def timing(self,int):  
 print(**"空调将在"**,int,**"分钟后自动关闭。。。。"**)  
  
class Test:  
 *# 4、在测试类中创建出空调对象，并给空调的品牌和价格赋任意值；* c=AirConditioner()  
 c.setBrand(**"格力"**)  
 c.setPrice(5000)  
*# 5、使用空调对象获取空调的品牌和价格并打印到控制台上；* print(**"品牌为："**,c.getBrand(),**"，价格是"**,c.getPrice(),**"元。"**)  
*# 6、使用空调对象调用开机方法；* c.open()  
  
*# 7、使用空调对象调用定时关机方法，并传递具体数据值，在控制台上可以看到的效果为：空调将在xxx分钟后自动关闭...  
# 其中语句中的“xxx”是调用方法时传递的具体数据值；* c.timing(24\*60)

## 题目二：

该题考查点：self关键字的使用！

定义一个学生类和对应的测试类

**要求：**

1. 学生有姓名和年龄两个属性，并且将属性私有化，提供公有的getXxx与setXxx方法对属性赋值和取值；
2. 提供一个无返回值的无参数的自我介绍的方法，内容打印一句话：

“大家好，我叫xxx，今年xxx岁了！”

1. 提供一个返回值为String类型，参数为学生类型的比较年龄差值的方法，如果当前对象***的年龄比参数中的学生的年龄大***，则返回：“我比同桌大xxx岁！”；如果当前对象的年龄比参数中的学生的年龄小，则返回：“我比同桌小xxx岁！”；如果当前对象的年龄和参数中的学生的年龄一样大，则返回：“我和同桌一样大！”
2. 在测试类中分别创建你和你同桌两个人的对象，并分别给你和你同桌的姓名和年龄属性赋上对应的值；
3. 调用你自己的对象的自我介绍的方法，展示出你自己的姓名和年龄；
4. 用你自己的对象调用比较年龄差值的方法，把你同桌作为参数使用，并打印方法返回的字符串的内容；

|  |
| --- |
| **class** Student:  \_\_username = **None** \_\_age = **None   def** setUsername(self,username):  self.\_\_username = username   **def** getUsername(self):  **return** self.\_\_username   **def** setAge(self,age):  **if** age > 120 **or** age < 0:  print(**"您年龄输入非法！"**)  **else**:  self.\_\_age = age   **def** getAge(self):  **return** self.\_\_age   **def** showMe(self):  print(**"大家好，我叫"**,self.\_\_username,**"，今年"**,self.\_\_age,**"岁了！"**)   **def** compare(self,student):*# self代表我自己 student代表另一个人* **if** self.\_\_age > student.getAge():  print(**"我比同桌大"**,(self.\_\_age - student.getAge()),**"岁！"**)  **elif** self.\_\_age < student.getAge():  print(**"我比同桌小"**, ( student.getAge()- self.\_\_age),**"岁！"**)  **else**:  print(**"我俩一样大！"**)  s = Student() s.setUsername(**"旺财"**) s.setAge(55)  s1 = Student() s1.setUsername(**"李四"**) s1.setAge(56)  s.compare(s1) *# 旺财要和李四比较* s1.compare(s) |

## 题目三：打电话业务逻辑

|  |
| --- |
| class Person:  name = **""** sex = **""** age = 0  thePhone = 0  brand = **""** battery = **""** size = **""** hour = **""** integral = 0   *# 功能：  # 发短信（要求参数传入短信内容）。* def texting(self,text):  print(text)  *# 打电话（要求传入要打的电话号码和要打的时间长度。程序里判断号码是否为空或者本人的话费是否小于1元， # 若为空或者小于1元则报相对应的错误信息，否则的话拨通。结束后， # 按照时间长度扣费并返回扣费（0~10分钟：1元/钟、15个积分/钟，10~20分钟：0.8元/钟、39个积分/钟， # 其他：0.65元/钟、48个积分/钟））* def Make\_a\_phone\_call(self,num,time,thePhone,integral):  if num == 0:  print(**"号码不能为空！！！"**)  elif thePhone < 1:  print(**"您的余额不足！！！"**)  else:  print(**"正在拨打中！！！"**)  if time <= 10:  print(**"消费"**,time\*thePhone,**"元话费"**,**"获得积分"**,15\*time,**"分"**)  elif 10<time<=20:  print(**"消费"**,time\*0.8,**"元话费"**,**"获得积分"**,39\*time,**"分"**)  else:  print(**"消费"**,time\*0.65,**"元话费"**,**"获得积分"**,48\*time,**"分"**)  a = Person() a.name = **"linsa"** a.sex = **"男"** a.age = 27 a.thePhone = 50 a.brand = **"三星"** a.battery = **" 6000毫安"** a.size = **" 428 x 926"** a.hour = **"30分钟"** a.integral = 10000 a.Make\_a\_phone\_call(18284847592,10,50,25) |
| class Person:  name = **""** sex = **""** age = 0  thePhone = 0  brand = **""** battery = **""** size = **""** hour = **""** integral = 0   *# 功能：  # 发短信（要求参数传入短信内容）。* def texting(self,text):  print(text)  *# 打电话（要求传入要打的电话号码和要打的时间长度。程序里判断号码是否为空或者本人的话费是否小于1元， # 若为空或者小于1元则报相对应的错误信息，否则的话拨通。结束后， # 按照时间长度扣费并返回扣费（0~10分钟：1元/钟、15个积分/钟，10~20分钟：0.8元/钟、39个积分/钟， # 其他：0.65元/钟、48个积分/钟））* def Make\_a\_phone\_call(self,num,time,thePhone,integral):  if num == 0:  print(**"号码不能为空！！！"**)  elif thePhone < 1:  print(**"您的余额不足！！！"**)  else:  print(**"正在拨打中！！！"**)  if time <= 10:  print(**"消费"**,time\*thePhone,**"元话费"**,**"获得积分"**,15\*time,**"分"**)  elif 10<time<=20:  print(**"消费"**,time\*0.8,**"元话费"**,**"获得积分"**,39\*time,**"分"**)  else:  print(**"消费"**,time\*0.65,**"元话费"**,**"获得积分"**,48\*time,**"分"**)  a = Person() a.name = **"linsa"** a.sex = **"男"** a.age = 27 a.thePhone = 50 a.brand = **"三星"** a.battery = **" 6000毫安"** a.size = **" 428 x 926"** a.hour = **"30分钟"** a.integral = 10000 a.Make\_a\_phone\_call(18284847592,10,50,25) |

## 题目四：需求编程

* + 1. 定义了一个学生类：属性:学号，姓名，年龄，性别，身高，体重，成绩，家庭地址，电话号码。行为：学习（要求参数传入学习的时间），玩游戏（要求参数传入游戏名），编程（要求参数传入写代码的行数），数的求和（要求参数用变长参数来做，返回求和结果）

|  |
| --- |
|  |

* + 1. 车类：属性：车型号，车轮数，车身颜色，车重量，油箱存储大小 。功能：跑（要求参数传入车的具体功能，比如越野，赛车）

创建：法拉利，宝马，铃木，五菱，拖拉机对象

|  |
| --- |
|  |

* + 1. 笔记本：属性：型号，待机时间，颜色，重量，cpu型号，内存大小，硬盘大小。 行为：打游戏（传入游戏的名称）,办公。

|  |
| --- |
|  |

* + 1. 猴子类：属性：类别，性别，身体颜色，体重。行为：造火（要求传入造火的材料：比如木棍还是石头），学习事物（要求参数传入学习的具体事物，可以不止学习一种事物）

|  |
| --- |
| *# 题目四：需求编程 # i.定义了一个学生类：属性:学号，姓名，年龄，性别，身高，体重，成绩，家庭地址，电话号码。 # 行为：学习（要求参数传入学习的时间），玩游戏（要求参数传入游戏名），编程（要求参数传入写代码的行数）， # 数的求和（要求参数用变长参数来做，返回求和结果） # class Student: # unum = 0 # name = "" # age = 0 # high = 0 # weight = 0 # results = 0 # address = "" # number = 0 # # def behavior(self,time,gameName,programming,\*sum): # print("学习了",time,"小时，","游戏名称为： ",gameName,"，代码的行数",programming,"求和为：",sum) # return sum # a = len(open(file="C:\\Users\\86187\\PycharmProjects\\pythonProject2\\day09\\day9.py",mode="r",encoding="utf-8").readlines()) # # student = Student() # student.unum = 123456789 # student.name = "lisa" # student.age = 18 # student.high = 1.79 # student.weight = 120 # student.results = 100 # student.address = "云南" # student.number = 18273849572 # # student.behavior(10,"绝地求生",a)  # # ii.车类：属性：车型号，车轮数，车身颜色，车重量，油箱存储大小 。功能：跑（要求参数传入车的具体功能，比如越野，赛车） # 创建：法拉利，宝马，铃木，五菱，拖拉机对象 # # # class Car: # model = "" # num = 0 # color = "" # weight = 0 # size = 0 # def run(self,car): # print(car,"在奔跑！！！") # # a = Car() # a.model = "CA7200" # a.num = 4 # a.color = "红色" # a.weight = 1000 # a.size = 200 # a.run("法拉利") # # c = Car() # c.model = "宝马x5" # c.num = 4 # c.color = "绿色" # c.weight = 1050 # c.size = 200 # c.run("宝马") # # d = Car() # d.model = "M109" # d.num = 4 # d.color = "蓝色" # d.weight = 1550 # d.size = 100 # d.run("铃木") # # e = Car() # e.model = "1.0l-1010plnnf单排 " # e.num = 4 # e.color = "白色" # e.weight = 1550 # e.size = 100 # e.run("五菱") # # f = Car() # f.model = "304" # f.num = 4 # f.color = "白色" # f.weight = 1550 # f.size = 100 # f.run("拖拉机")      # iii.笔记本：属性：型号，待机时间，颜色，重量，cpu型号，内存大小，硬盘大小。 # 行为：打游戏（传入游戏的名称）,办公。 # # class Notebook\_Computer: # model = "" # time = 0 # color = "" # weight = 0 # cpu\_model = "" # size = 0 # disk = "" # def behavior(self,game,office): # print("这台笔记本电脑可以打",game,"的游戏","可以处理",office,"文件。") # # # n = Notebook\_Computer() # n.model = "MacBookAir" # n.time = 24 # n.color = "红色" # n.weight = 15 # n.cpu\_model = "Pentium" # n.size = "8GB" # n.disk = "100GB" # n.behavior("和平精英","word")     # iv.猴子类：属性：类别，性别，身体颜色，体重。 # 行为：造火（要求传入造火的材料：比如木棍还是石头），学习事物（要求参数传入学习的具体事物， # 可以不止学习一种事物） # # class monkey: # category = "" # sex = "" # color = "" # weight = 0 # # def behavior(self,stone,smoking): # print("指猴用",stone,"生火以后，就用火学会了",smoking) # b = monkey() # b.category = "指猴" # b.sex = "男" # b.color = "黄色" # b.weight = 120 # b.behavior("石头","抽烟")* |