## 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；

|  |
| --- |
| **class** Air:  \_\_brand = **''** \_\_price = 0  \_\_time = 0   **def** setBrand(self, brand):  self.\_\_brand = brand   **def** getBrand(self):  **return** self.\_\_brand   **def** setPrice(self, price):  **if** price <= 0:  print(**'价格不能为零为负！'**)  **else**:  self.\_\_price = int(price)   **def** getPrice(self):  **return** self.\_\_price   **def** setTime(self, time):  **if** time < 0:  print(**'时间不能为负！！'**)  **else**:  self.\_\_time = int(time)   **def** power\_on(self):  print(**'空调开机了...'**)   **def** shut\_down(self):  print(**'空调将在'**, self.\_\_time, **'分钟后自动关闭...'**)   **def** show(self):  print(**'空调品牌是'**, self.\_\_brand, **'空调价格为'**, self.\_\_price, **'元！'**)   p = Air() p.power\_on() p.setBrand(input(**'输入空调品牌：'**)) p.setPrice(int(input(**'输入空调价格：'**))) p.show() p.setTime(int(input(**'空调在多少分钟后关机：'**))) p.shut\_down() |

## 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

|  |
| --- |
| 人类：  属性:  姓名，性别，年龄，所拥有的手机剩余话费，手机品牌，手机电池容量，手机屏幕大小，手机最大待机时长，所拥有的积分。  功能：  发短信（要求参数传入短信内容）。  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）） |
| class People:  \_\_name = ''  \_\_gender = ''  \_\_age = 0  \_\_cost = 0 # 剩余话费  \_\_brand = '' # 品牌  \_\_battery = 0 # 电池容量  \_\_size = 0 # 屏幕大小  \_\_standby = 0 # 最大待机时长  \_\_integral = 0 # 积分  def setName(self, name):  self.\_\_name = name  def getName(self):  return self.\_\_name  def setGender(self, gender):  self.\_\_gender = gender  def getGender(self):  return self.\_\_gender  def setAge(self, age):  if age <= 0 or age >= 120:  print('年龄非法！')  else:  self.\_\_age = int(age)  def getAge(self):  return self.\_\_age  def setCost(self, cost):  self.\_\_cost = float(cost)  def getCost(self):  return self.\_\_cost  def setBrand(self, brand):  self.\_\_brand = brand  def getBrand(self):  return self.\_\_brand  def setBattery(self, battery):  if battery < 0:  print('电池容量不能为负！')  else:  self.\_\_battery = float(battery)  def getBattery(self):  return self.\_\_battery  def setSize(self, size):  if size <= 0:  print('屏幕大小输入非法！')  else:  self.\_\_size = int(size)  def getSize(self):  return self.\_\_size  def setStandby(self,standby):  self.\_\_standby = int(standby)  def getStandby(self):  return self.\_\_standby  def setIntegral(self, integral):  if integral < 0:  print('积分不能为负！')  else:  self.\_\_integral = int(integral)  def getIntegral(self):  return self.\_\_integral  def show(self):  print('姓名', self.\_\_name, '\n性别', self.\_\_gender, '\n年龄', self.\_\_age,'\n所拥有的手机剩余话费',  self.\_\_cost, '元！\n手机品牌', self.\_\_brand,'\n手机电池容量', self.\_\_battery, '%\n屏幕大小',  self.\_\_size, '寸\n最大待机时长',self.\_\_standby, '分钟\n所拥有积分：', self.\_\_integral)  p = People()  p.setName(input('输入姓名'))  p.setGender(input('输入性别'))  p.setAge(int(input('输入年龄')))  p.setCost(float(input('输入手机剩余话费')))  p.setBrand(input('输入手机品牌'))  p.setBattery(float(input('输入电池容量')))  p.setSize(int(input('输入手机屏幕大小')))  p.setStandby(int(input('输入手机最大待机时长')))  p.setIntegral(int(input('输入拥有积分')))  p.show()  cc = p.getCost()  dd = p.getIntegral()  while True:  a = int(input('需要打电话还是发短信：（1或2）'))  if a == 1:  a\_1 = input('输入短信内容：')  print('短信内容为：\n', a\_1)  elif a == 2:  a\_1 = int(input('输入电话号码：'))  a\_2 = int(input('输入打多长时间：'))  if a\_1 == None: print('不能为空！')  elif a\_1 <= 1: print('电话费不够了！')  else:  print('电话已拨通！')  if a\_2 >= 0 and a\_2 <= 10:  if dd >= a\_2 \* 15:  dd -= a\_2 \* 15  else:  cc -= a\_2 \* 1  elif a\_2 > 10 and a\_2 <=20:  if dd >= a\_2 \* 39:  dd -= a\_2 \* 39  else:  cc -= a\_2 \* 0.8  else:  if dd >= a\_2 \* 48:  dd -= a\_2 \* 48  else:  cc -= a\_2 \* 0.65  print('剩余话费为：', cc)  print('剩余积分为：', dd) |

## 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）

|  |
| --- |
| class student:  \_\_id = ""  \_\_name = ""  \_\_age = 0  \_\_sex = ""  \_\_height = 0.0  \_\_weight = 0.0  \_\_result = 0.0  \_\_home = ""  \_\_tel = 0  def set(self,id,name,age,sex,height,weight,home,tel):  self.\_\_id = id  self.\_\_name = name  self.\_\_age = age  self.\_\_sex = sex  self.\_\_height = height  self.\_\_weight = weight  self.\_\_home = home  self.\_\_tel = tel  def study(self):  into\_sd = input("请输入学习时长（小时）:")  print(self.\_\_name,"已经学习了",into\_sd,"小时了")  def game(self):  into\_gm = input("请输入游戏名称：")  print(self.\_\_name,"玩的游戏是：",into\_gm)  def charm(self):  into\_cm = input("请输入代码行数：")  print(self.\_\_name,"编写了",into\_cm,"行代码")  def sum(self):  num = 0  into\_yw = float(input("请输入语文成绩："))  into\_sx = float(input("请输入数学成绩："))  into\_yy = float(input("请输入英语成绩："))  num = into\_yw + into\_sx + into\_yy  all = '''  语文成绩：%s  数学成绩：%s  英语成绩：%s  总和：%s  '''  print(self.\_\_name,"的成绩如下：",all%(into\_yw,into\_sx,into\_yy,round(num,2)))  learn = student()  learn.set("s001","张三",22,"男",182,200,13579246810,"北京市朝阳区霄云路36号国航大厦")  learn.study()  learn.game()  learn.charm()  learn.sum() |

* + 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）

|  |
| --- |
|  |