1.学生信息管理系统代码

*#student\_manager\_system*

*#第一步：*

*# 数据模型类：StudentModel*

*# 逻辑控制类：StudentManagerController*

*# 数据：学生列表* *\_\_stu\_list*

*# 行为：获取列表* *stu\_list,*

*# 添加学生* *add\_student*

*# 添加id 整数 递增 类变量*

*# 数据模型类*

**class** StudentModel:

*#id不需要传值 放在最后一位*

**def** \_\_init\_\_(self, name=**""**, age=0, score=0,id=0):

self.id = id

self.name = name

self.age = age

self.score = score

*# 逻辑控制类：*

**class** StudentManagerController:

\_\_stu\_id = 1000

**def** \_\_init\_\_(self):

self.\_\_stu\_list = []

@property

**def** stu\_list(self):

**return** self.\_\_stu\_list

*#1.添加学生*

**def** add\_student(self,stu):

*#为学生设置id 递增*

*# StudentManagerController.\_\_stu\_id += 1*

*#stu.id = StudentManagerController.\_\_stu\_id*

stu.id = self.\_\_generate\_id()

*#将学生添加到学生列表*

self.\_\_stu\_list.append(stu)

*#生成id*

**def** \_\_generate\_id(self):

StudentManagerController.\_\_stu\_id += 1

**return** StudentManagerController.\_\_stu\_id

*# 2.删除学生remove\_student*

*#根据id删除学生*

*#删除后返回结果 成功True/失败False*

**def** remove\_student(self,id):

**for** item **in** self.\_\_stu\_list:

**if** item.id == id:

self.\_\_stu\_list.remove(item)

**return True**

**return False**

*#3.修改学生update\_student*

*# s01 = StudentModel('zs',18,50)*

*# s01 = StudentModel('zs',19,40,1001)*

**def** update\_student(self,stu):

**for** item **in** self.\_\_stu\_list:

**if** item.id == stu.id:

item.name = stu.name

item.age = stu.age

item.score = stu.score

**return True**

**return False**

*# 4.根据成绩排序order\_by\_score。*

*# 你是谁 你要找谁*

**def** order\_by\_score(self):

**for** i **in** range(len(self.\_\_stu\_list)-1):

**for** j **in** range(i+1,len(self.\_\_stu\_list)):

**if** self.\_\_stu\_list[i].score > self.\_\_stu\_list[j].score:

self.\_\_stu\_list[i],self.\_\_stu\_list[j] = \

self.\_\_stu\_list[j],self.\_\_stu\_list[i]

*# 界面视图类：*

**class** StudentManagerView:

*#生成在此类下的逻辑管理器 组合*

**def** \_\_init\_\_(self):

self.\_\_manger = StudentManagerController()

*#1.显示菜单*

**def** \_\_display\_menu(self):

print(**'''**

**学生信息管理系统1.0**

**+-----------------------+**

**| 0)退出管理系统** **|**

**| 1)添加学生信息** **|**

**| 2)显示学生信息** **|**

**| 3)删除学生信息** **|**

**| 4)修改学生信息** **|**

**| 5)按照成绩排序** **|**

**+-----------------------+**

**'''**)

*#2.选择菜单*

**def** \_\_select\_menu(self):

*# import sys*

option = input(**'请选择:'**)

**if** option == **'1'**:

self.\_\_input\_students()

**elif** option == **'2'**:

self.\_\_output\_students()

**elif** option == **'3'**:

self.\_\_delete\_student()

**elif** option == **'4'**:

self.\_\_modify\_student()

**elif** option == **'5'**:

self.\_\_output\_student\_by\_score()

**elif** option == **'0'**:

*# print('谢谢使用,退出系统')*

*# sys.exit()*

exit(**'谢谢使用,退出系统'**)

**else**:

print(**'请正确输入选项!'**)

*#3.入口逻辑*

**def** main(self):

*'''*

*界面入口*

**:return***:*

*'''*

**while True**:

self.\_\_display\_menu()

self.\_\_select\_menu()

*# 4.输入学生*

**def** \_\_input\_students(self):

*#收集学生信息 要求用户输入 姓名年龄 成绩*

*#创建学生对象*

*#在控制器找add\_student方法*

name = input(**'请输入学生姓名:'**)

age = int(input(**'请输入学生年龄:'**))

score = int(input(**'请输入学生成绩:'**))

stu = StudentModel(name,age,score)

*#这样写会重复生成 管理器 列表,因此在视图界面类创造一个管理器*

*# manger = StudentManagerController()*

self.\_\_manger.add\_student(stu)

*#5.输出信息*

**def** \_\_output\_students(self):

**for** item **in** self.\_\_manger.stu\_list:

print(item.id, item.name, item.age, item.score)

*#6.删除学生*

**def** \_\_delete\_student(self):

*#需要用户输入id*

*#调用管理器对象的删除学生方法*

*#如果结果为True 显示删除成功*

*#否则显示删除失败*

id = int(input(**'请输入需要删除的学生ID:'**))

**if** self.\_\_manger.remove\_student(id):

print(**'删除成功!'**)

**else**:

print(**'删除失败!'**)

print(**'删除后学生信息:'**)

self.\_\_output\_students()

*#7.修改学生信息*

**def** \_\_modify\_student(self):

*#手机用户输入的信息保存到对象*

*#调用管理器的修改学生方法*

id = int(input(**'请输入需要修改的学生ID:'**))

name = input(**'请输入修改后学生姓名:'**)

age = int(input(**'请输入修改后学生年龄:'**))

score = int(input(**'请输入修改后学生成绩:'**))

stu = StudentModel(name, age, score,id)

**if** self.\_\_manger.update\_student(stu):

print(**'修改成功!'**)

**else**:

print(**'修改失败!'**)

print(**'修改后学生信息:'**)

self.\_\_output\_students()

*#8.按成绩升序排列*

**def** \_\_output\_student\_by\_score(self):

self.\_\_manger.order\_by\_score()

print(**'按照成绩排序学生:'**)

self.\_\_output\_students()

2.

*#重构shopping:*

*#商品模型* *id name price*

*#订单模型* *cid count*

*#购物车控制台界面视图*

*#入口 显示界面* *1 2 q*

*#选择* *1 2 q*

*#1 执行的函数*

*#2 执行的函数*

*#购物车逻辑控制器*

*#加载商品*

*#添加到订单*

*#生成订单*

*#计算总价*

*#-----------*

**class** Shopping\_info:

**def** \_\_init\_\_(self,name,price,num=0):

self.num = num

self.name = name

self.price = price

**class** Cart\_info:

**def** \_\_init\_\_(self,cnum,count):

self.cnum = cnum

self.count = count

**class** ShoppingManagerController:

\_\_info\_num = 100

**def** \_\_init\_\_(self):

self.shang\_pin\_info = []

self.cart\_info = []

*#1.添加商品信息*

**def** add\_info(self,info):

info.num = self.\_\_generate\_num()

self.shang\_pin\_info.append(info)

*#生成id*

**def** \_\_generate\_num(self):

ShoppingManagerController.\_\_info\_num += 1

**return** ShoppingManagerController.\_\_info\_num

*#2.打印商品信息*

**def** print\_info(self):

print(**'商品信息:'**)

print(**'\*'** \* 50)

**for** item **in** self.shang\_pin\_info:

print(item.num,item.name,item.price)

print(**'\*'** \* 50)

*#检查商品id*

**def** \_\_check\_id(self,sp\_num):

**for** item **in** self.shang\_pin\_info:

**if** item.num == sp\_num:

**return True**

*#3.添加购物车*

**def** add\_cart(self,info):

**if** self.\_\_check\_id(info.cnum):

self.cart\_info.append(info)

**return True**

*#4.打印购物车*

**def** print\_cart\_info\_total\_price(self):

total\_price = 0

print(**'\n购物车商品信息:'**)

print(**'\*'** \* 50)

**for** item **in** self.cart\_info:

**for** item\_s **in** self.shang\_pin\_info:

**if** item.cnum == item\_s.num:

print(item.cnum,item.count,item\_s.price)

total\_price += item.count \* item\_s.price

print(**'\*'** \* 50)

**return** total\_price

*#5.结算*

**def** paying(self,money):

total\_price = self.print\_cart\_info\_total\_price()

**if** money >= total\_price:

change = money - total\_price

**return** change

**class** ShoppingManagerView:

*#1.生成视图界面中的管理器*

**def** \_\_init\_\_(self):

self.\_\_manger = ShoppingManagerController()

*#2.显示界面*

**def** \_\_display\_menu(self):

print(**'''**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**夺命商店1.0**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**按1购买**

**按2结算**

**按q退出**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**'''**)

*#3. 主程序:界面入口*

**def** main(self):

self.\_\_display\_menu()

self.\_\_input\_info()

choice = **None**

**while** choice != **'q'**:

choice = input(**'请输入选项:'**)

**if** choice == **'q'**:

print(**'谢谢使用,退出商店!'**)

**break**

**elif** choice == **'1'**:

self.\_\_output\_info()

self.\_\_add\_cart()

**elif** choice == **'2'**:

self.\_\_output\_cart\_info()

self.\_\_paying()

**else**:

print(**'请输入正确选项!'**)

*#4.加载商品信息*

**def** \_\_input\_info(self):

s01 = Shopping\_info( **'倚天剑'**, 10000)

s02 = Shopping\_info( **'屠龙刀'**, 9000)

s03 = Shopping\_info( **'七伤拳'**, 8000)

s04 = Shopping\_info( **'一阳指'**, 7000)

s05 = Shopping\_info( **'断肠剑'**, 7000)

self.\_\_manger.add\_info(s01)

self.\_\_manger.add\_info(s02)

self.\_\_manger.add\_info(s03)

self.\_\_manger.add\_info(s04)

self.\_\_manger.add\_info(s05)

*#5.显示商品信息*

**def** \_\_output\_info(self):

self.\_\_manger.print\_info()

*#6.添加购物车*

**def** \_\_add\_cart(self):

cnum = int(input(**'\n请输入需要购买的商品编号:'**))

count = int(input(**'请输入购买的商品数量:'**))

cart\_info = Cart\_info(cnum,count)

**if** self.\_\_manger.add\_cart(cart\_info):

print(**'\n添加购物车成功!'**)

**else**:

print(**'\n添加购物车失败!'**)

*#7.打印购物车信息并计算总价:*

**def** \_\_output\_cart\_info(self):

res = self.\_\_manger.print\_cart\_info\_total\_price()

print(**'\n共计%s金元宝.'** % res)

*#8.结算*

**def** \_\_paying(self):

money = int(input(**'\n请支付金元宝:'**))

change = self.\_\_manger.paying(money)

**if** change:

print(**'\n支付%s金元宝,购买成功，找零%.2f金元宝'** % (money,change))

print(**'购买成功!'**)

**else**:

print(**'\n购买失败!'**)

view = ShoppingManagerView()

view.main()