

# Python程序设计报告

设计题目： 商品进销存管理系统

学生姓名： 陈嘉乐

专 业： 计算机科学与技术

班 级： 计算机21-3班

学 号： 2021218152

指导教师： 马学森

完成日期： 2023年11月8日

**（一）需求和规格说明**

1.1 问题描述

这是一款**商品进销存管理**系统软件，进销存管理指的是企业管理过程中采购 （进）—>入库（存）—>销售（销）的动态管理过程。本题需要基于**已有数据文件**建立商品库，不同用户可以对商品库进行不同的操作：**顾客**可以查询商品的信息、库存、价格，以及购买商品操作；**店员**可以查询库中商品库存、商品价格，管理商品的数量，进行进货，售货，查询详细收支操作。

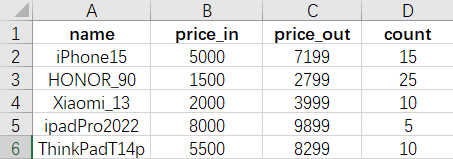
1.2 编程任务

1. 建立商品库。
2. 输出菜单栏，供用户选择功能。
3. 查询货物量、货物价格等信息。
4. 进货，增加对应货物的数量，记录支出。
5. 出售，减少对应货物的数量，记录收入。
6. 查询当前收支情况。

1.3 数据规格

数据文件以.csv的形式存储，更容易进行信息的读取和操作。需要说明的是本项目仅是简易版的商品进销存管理软件，所以提供的数据规格较小（包含五种商品及其名称、进价、售价、数量），但足以完成项目的调试及测试。

下面给出本人构建商品库所使用的数据文件内容截图。

****

**图1 库存商品信息**

**（二）设计**

2.1 设计初衷

由于此次Python大作业为开放性试题，于是本人想设计一个简易的管理系统，以此来巩固自己的Python基础与实操能力，而没有去选择复现他人的大型项目。设计初衷更多的想去根据马老师在课堂上介绍如何查找电子书的方法，以及个人掌握的Python基础知识与编程能力，**独立**实现一个**能够进行文件读写，可交互性的管理系统**。

受模板题目——背单词小软件以及大一学年课设选题的启发与引导，本人想设计一个商品进销存管理软件。此软件能够根据已有的商品信息文件建立商品库，用户能够通过交互式方式进行顾客或店员的角色扮演，从而进行商品的进销存管理，并且将进货、销货信息**模仿**现实生活中的记账本形成文本数据形式的**管理日志**，并且对商品信息文件进行实时更新，保证进销存管理系统能够进行**多次操作和使用**，使其程序健壮性更强。

2.2 设计背景

随着经济的全球化以及中国经济改革的逐渐深化，中小企业面临着越来越激烈的竞争。改善企业内部以及整个供应链各环节的管理、调度及资源配置，迅速适应客户的新需求和市场的新机遇，是中小企业赢得竞争胜利的决定性因素。有效地进行商品进销存管理，已经成为了中小企业生存和发展的头等要事。然而，对于一贯疏于管理的中小企业来说，进行有效的商品进销存管理必然存在一定的难度，必须借助于现代化的管理方法和信息技术——商品进销存管理软件。

了解本题的设计背景和意义后，我们就可以开始进行简易版商品进销存管理系统的设计。

2.3 设计细节

2.3.1设计思想

本题主要考察Python中列表、字典、函数，文件读写以及面向对象中类的设计等相关操作。

首先基于面向对象思想，抽象地设计出类ManagementSystem，后续关于系统中的功能设计基本都在该类体中进行实现。对于实例属性的设置，本人定义了以下**实例属性**：列表goods\_info存储商品信息，sell\_info存储销售信息，purchase\_info存储进货信息，变量funds为总资金，income为总收入，expenditure为总支出，t为管理系统当前时间。这些信息是贯穿整个系统最基础信息，在进行各种功能操作的同时，对这些数据以及相关数据文件进行同步的修改，实现后台的管理。

在进行销售、进货等各种商品处理操作时，主要的方法是创建一个新的字典，字典保存了当前交易的信息，在交易结束的同时，将该信息通过列表append()方法加入到列表中，或者根据序号取出goods\_info列表中储存商品的字典，对其值进行直接修改。最后对于上述实例属性中的相关存储信息列表及文件作同步的修改，将相关操作基于文件读写操作技术记录在相关日志文件中，便于系统的维护以及管理，保证系统的健壮性。

显示货物信息、交易信息时，使用Python中enumerate()函数，将一个可遍历的商品对象good\_info组合为一个索引序列，即index，good\_dict，再进行输出。此处也是深受**马老师上课多次强调的enumerate()函数**的启发才能将此功能较好地设计出来。

交互性菜单也采用无限循环，人为地控制程序的结束，同时因为本软件是基于本地文件(类似于数据库)进行的商品管理，所以支持程序的多次操作。

2.3.2设计表示

在进行本题的代码编写之前，本人先通过**亿图软件和Excel表格**画出了以下设计表示图：

1. 核心程序流程图；
2. ManagementSystem类图；
3. 系统参数表。

这使我在编写程序的时候能够更加专注于代码撰写本身，有了事半功倍的效果，而且不易造成逻辑错误。

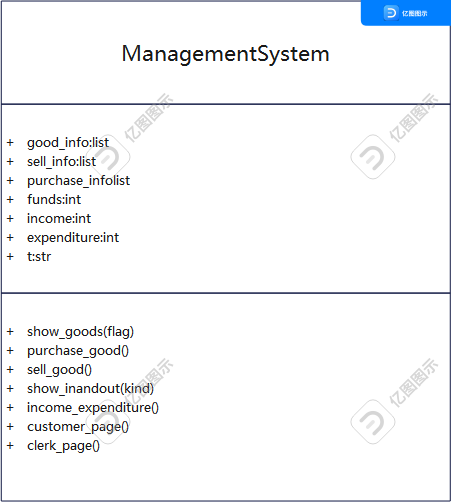
下面是设计表示图的罗列：

本人设计的核心程序流程图如下图所示：



**图2.1 核心程序流程图**

本人设计的ManagementSystem的类图如下图所示：



**图2.2 ManagementSystem类图**

## 本人设计的系统参数表如下：

1. ManagermentSystem类中的实例属性：

**表2.1 ManagementSystem类实例属性表**

|  |  |  |
| --- | --- | --- |
| **类型** | **名称** | **描述** |
| list | good\_info | 商品库 |
| list | sell\_info | 销售信息 |
| list | purchase\_info | 进货信息 |
| int | funds | 总资金 |
| int | income | 总收入 |
| int | expenditure | 总支出 |
| str | t | 系统当前时间 |

1. ManagermentSystem类中的实例方法：

**表2.2 ManagementSystem类实例方法表**

|  |  |
| --- | --- |
| **方法名称** | **描述** |
| show\_goods(flag) | 显示货物信息，为了区别不同用户，当flag==True显示货物的全部信息，当flag==False时只显示货物名称和售价。 |
| purchase\_good() | 进货（店员）功能。包括两种操作：增加新种类货物、增加已有货物库存。同时对good\_info，purchase\_info，funds，expenditure的值进行修改。使用到show\_goods(True)。 |
| sell\_good() | 售货（店员）/购物（顾客）功能。同时对good\_info，sell\_info，funds，income的值进行修改。使用到show\_goods(False)。 |
| show\_inandout(kind) | 显示销售和进货详情（店员）功能。当kind==1时，显示销售记录；当kind==2时显示进货记录。 |
| income\_expenditure() | 收支查询（店员）功能。使用到show\_inandout(kind)函数。 |
| customer\_page() | 顾客菜单页（二级菜单）。 |
| clerk\_page() | 店员菜单页（二级菜单）。 |

1. 外部函数：

**表2.3 外部函数表**

|  |  |
| --- | --- |
| **函数名称** | **描述** |
| log(filename, info) | 用来记录进货，销货信息，并存入日志文件。 |
| revise\_info\_file | 更新商品信息存储文件。 |

**（三） 用户手册**

程序运行显示一级菜单，提示选择用户类型，键入数字1-顾客、2-店员、3-退出程序。

* **选择1-顾客进入顾客菜单页，提示选择功能，键入数字1-3。**
* 选择**1-查询商品信息**后，打印商品库中的商品信息（名称；售价；库存）。
* 选择**2-购买商品**后，打印商品库中的商品信息，提示输入商品编号，正确输入后，打印所选的商品名称和售价，提示选择是否确认购买，输入1-购买后，根据提示输入购买数量，购买成功；输入2-返回后直接结束购物，返回用户菜单页。
* 选择**3-退出**，返回一级菜单页，键入3结束程序。
* **选择2-店员进入店员菜单页，提示选择功能，键入数字1-5。**
* 选择**1-查询商品信息**后，打印商品库中的商品信息（名称；成本；售价；库存）。
* 选择**2-进货**后，打印商品库中的商品信息，提示输入进货方式，键入1-2。选择1-增加新种类货物，根据提示依次输入商品名称、成本、售价、数量，采购成功，返回店员菜单页；选择2-增加已有货物库存，打印商品库中的商品信息，根据提示选择商品编号，输入采购数量，进货成功，返回店员菜单页。
* 选择**3-售货**后，打印商品库中的商品信息，根据提示进行售货。
* 选择**4-查询收支**后，根据提示选择操作，键入1-4。选择1-销售情况，打印销售记录；选择2-进货情况，打印采购记录；选择3-总财产情况，打印所有交易记录以及总资金；选择4-退出，返回店员菜单页。
* 选择**5-退出**，返回一级菜单页，键入3结束程序。

**（四）调试及测试**

4.1 测试数据

初始总资金80000元。

**表4. 1 初始商品库**

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 成本 | 售价 | 数量 |
| **iPhone15** | 5000 | 7199 | 15 |
| **HONOR\_90** | 1500 | 2799 | 25 |
| **Xiaomi\_13** | 2000 | 3999 | 10 |
| **ipadPro2022** | 8000 | 9899 | 5 |
| **ThinkPadT14p** | 5500 | 8299 | 10 |

**表4. 2 进货信息**

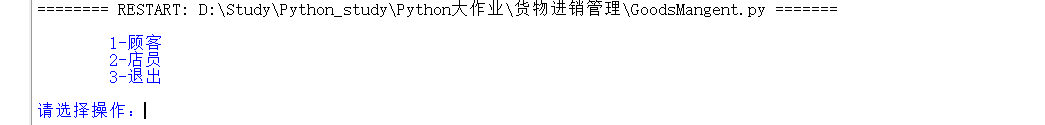
|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 成本 | 售价 | 数量 |
| **HUAWEI\_Mate60** | 6000 | 9399 | 10 |

**表4. 3 顾客购买信息**

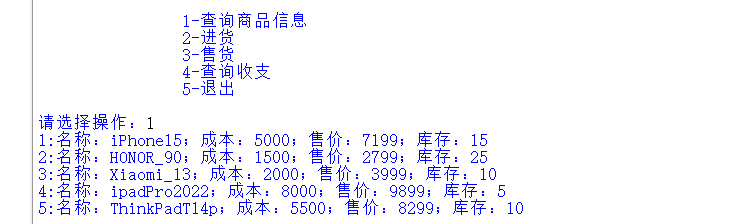
|  |  |  |
| --- | --- | --- |
| 名称 | 售价 | 数量 |
| **iPhone15** | 7199 | 2 |
| **HONOR\_90** | 2799 | 1 |

4.2 测试结果：

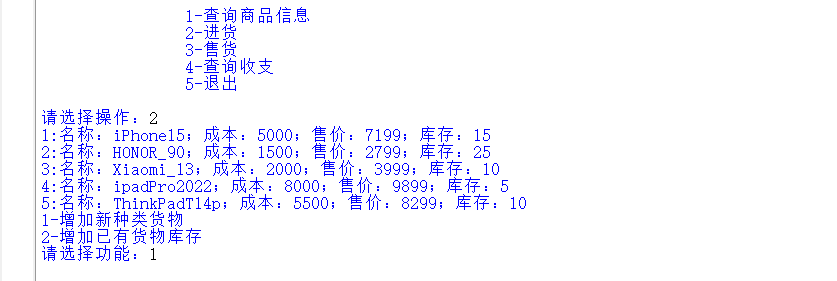
**测试一：进货**

****

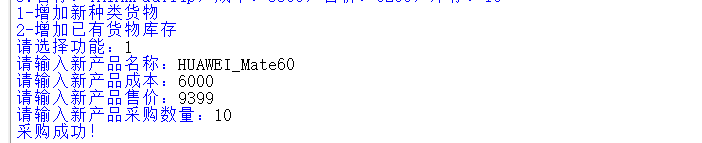
**图4.1 一级菜单——用户选择页**



**图4. 2 二级菜单——店员菜单页**

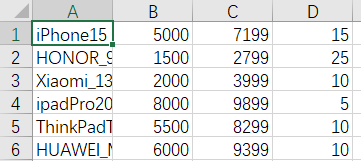


**图4. 3 选择进货类型**

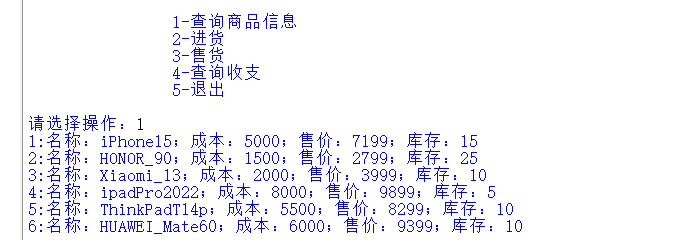


**图4. 4 进货成功**



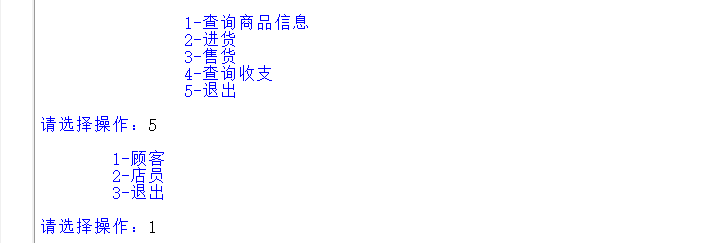


**图4. 5 进货日志记录**

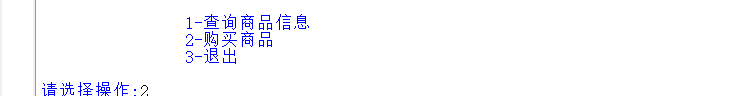


**图4. 6 查询进货后的商品信息库**

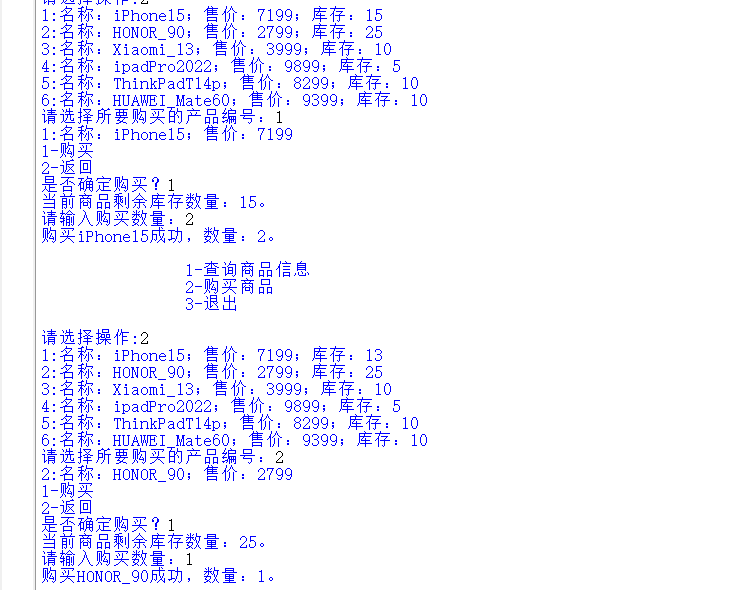
**测试二：顾客购物**



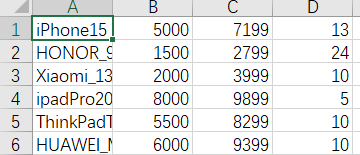
**图4. 7 一级菜单——用户选择页**

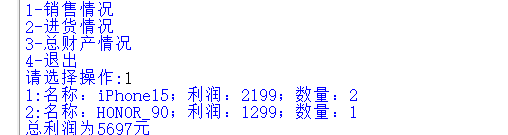


**图4. 8 二级菜单——顾客菜单页**



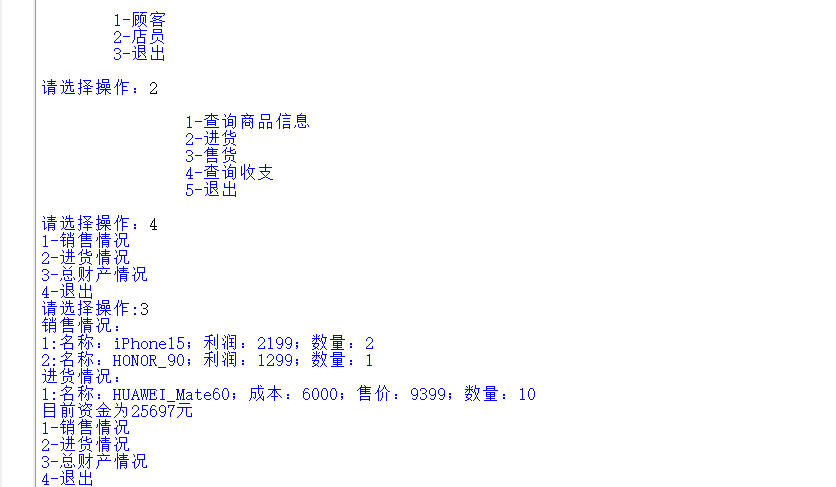
**图4. 9 购买两种商品**





**图4. 10 销货日志记录**

**测试三：查询收支情况**

****

**图4. 11 查询收支情况**

**（五）进一步改进**

（1）界面单一。作为课程大作业来说，仅在控制台上进行交互性操作虽然足够，但是如果能够设计GUI图形界面可能会更好，这也是我以后做课程设计学习与努力的方向。后续可能还会做成网页形式进行操作和展示——分成商家端和客户端，从而实现真正的管理系统。

（2）可操作性空间不足。由于本设计系统相对简单，功能难免不太齐全。比如客户端可增加退货相关功能；商家对于多个订单的处理逻辑需要进行完善等等...

（3）程序健壮性不足。目前管理信息仅存储在本地磁盘文件中，后续可考虑使用数据库的方式进行存储。

**（六）心得体会**

通过本次大作业，个人感觉是受益匪浅的，首先感谢这次大作业马老师所给予的开放性试题的机会，以及大作业本身给我带来的收获。

这道题并不难，主要就是考察Python中列表、字典、函数，文件读写以及面向对象中类的设计等相关操作，除此之外还有对于Python库中的函数调用的能力。

在进行设计的初期，我只构思了系统的各种功能，并没有考虑用户的类型，在整体功能都完成之后，在此基础上对用户进行了区分，编写了顾客和店员两种用户类型的菜单页面。

在设计好主体程序后，我发现此系统只能进行一次操作，程序的体验性和健壮性过差。于是我用文本文件存储商品信息，将进货、销货操作写进文本文件中，形成日志，真实的模仿了现实中的商品进销货流程，并且能够对其进行多次操作，使程序具有较好的可维护性。

对于此次大作业的收获，我认为自己更加熟练地掌握了关于Python判断分支结构、循环结构、函数；变量、字符串、列表、列表推导式、切片、元组、生成器推导式、字典、集合的应用；面向对象相关语法以及文件的读写操作。Python代码的精炼，语法的严密，其强大的内嵌数据库让我在编写程序的时候十分轻松和灵活。

在本次设计中，我也通过马老师在课堂上介绍如何在合肥工业大学文泉学堂上查找电子书等方式从书中了解到更多更丰富的库，例如enumerate() 函数，文件读写的mode代表什么意义以及如何其在具体场景下的使用。电子书和各大平台资源（GitHub、CSDN）对于大学生来说是非常重要的，它能够让你学到课堂因学时有限所学不到的内容，从而让你获得编码能力上的提升。我想这也是马老师在课堂上向我们介绍如何从网络上获取资源以及布置此次大作业的意义所在。

调参能力与修改bug能力是我们计算机科学与技术专业的学生所要面对的问题，而在大作业设计中这些能力也得到较大的提升，实践才能出真知！在规定的时间，自行设计并尽可能地完成它，我觉得这次的大作业设计能反映我目前个人较好的编码能力。如果代码和功能存在一些致命性问题，还望老师多多批评指正，我会虚心接受，努力修改。未来可期，希望有更多的大作业机会来锻炼我，让我的编码能力得到进一步的提升！最后感谢马老师的Python课程以及本次大作业机会，让我受益匪浅！

**（七）附录——源代码及数据文件**

|  |
| --- |
| import csv import time   def log(filename, info):  with open(filename, encoding='utf-8', mode='a') as f:  f.write(info)   def revise\_info\_file(goods\_info):  with open('product\_info.csv', encoding='utf-8', mode='w+') as f:  for item in goods\_info:  f.write('{},{},{},{}\n'.format(item['name'], item['price\_in'], item['price\_out'], item['count']))   class ManagementSystem:  def \_\_init\_\_(self, funds=80000):  self.goods\_info = [] # 商品信息  self.sell\_info = [] # 销售信息  self.purchase\_info = [] # 进货信息  self.funds = funds # 总资金  self.income = 0 # 总收入  self.expenditure = 0 # 总支出  self.t = time.strftime('%Y-%m-%d %H:%M:%S', time.localtime()) # 管理系统当前时间   def creat\_goods\_info(self, filename): # 创建商品信息  with open(filename, encoding='utf-8') as f:  f\_csv = csv.reader(f)  for row in f\_csv:  self.goods\_info.append(  {'name': row[0], 'price\_in': int(row[1]), 'price\_out': int(row[2]), 'count': int(row[3])})   def show\_goods(self, flag): # 展示商品信息  if flag:  for index, good\_dict in enumerate(self.goods\_info):  print('{}:名称：{}；成本：{}；售价：{}；库存：{}'.format(index + 1, good\_dict['name'], good\_dict['price\_in'],  good\_dict['price\_out'], good\_dict['count']))  else:  for index, good\_dict in enumerate(self.goods\_info):  print('{}:名称：{}；售价：{}；库存：{}'.format(index + 1, good\_dict['name'], good\_dict['price\_out'],  good\_dict['count']))   def purchase\_good(self): # 进货  self.show\_goods(True) # 先展示现有商品信息  print("1-增加新种类货物")  print("2-增加已有货物库存")  choices = int(input("请选择功能："))  if choices == 1:  new\_good\_name = input("请输入新产品名称：")  new\_good\_price\_in = int(input("请输入新产品成本："))  new\_good\_price\_out = int(input("请输入新产品售价："))  new\_good\_count = int(input("请输入新产品采购数量："))  good\_dict = {'name': new\_good\_name, 'price\_in': new\_good\_price\_in, 'price\_out': new\_good\_price\_out,  'count': new\_good\_count}  self.goods\_info.append(good\_dict)  self.purchase\_info.append(good\_dict)  info = '{} 进货{}，其成本为{}，售价为{}，采购数量为{}\n'.format(self.t, new\_good\_name, new\_good\_price\_in,  new\_good\_price\_out, new\_good\_count)  log('purchase\_info.txt', info) # 进货日志记录  revise\_info\_file(self.goods\_info)  self.expenditure += good\_dict['price\_in'] \* good\_dict['count']  self.funds -= good\_dict['price\_in'] \* good\_dict['count']  else:  self.show\_goods(True)  index = int(input("请输入货物编号：")) - 1  good\_dict = self.goods\_info[index]  purchase\_count = int(input("请输入新采购的货物数量："))  good\_dict['count'] += purchase\_count  info = '{} 进货{}，其成本为{}，售价为{}，采购数量为{}\n'.format(self.t, good\_dict['name'], good\_dict['price\_in'],  good\_dict['price\_out'], good\_dict['count'])  log('purchase\_info.txt', info) # 进货日志记录  revise\_info\_file(self.goods\_info)  self.purchase\_info.append(good\_dict)  self.expenditure += good\_dict['price\_in'] \* purchase\_count  self.funds -= good\_dict['price\_in'] \* good\_dict['count']  print("采购成功！")   def sell\_good(self):  self.show\_goods(False) # 先展示现有商品信息  choices = int(input("请选择所要购买的产品编号："))  good\_dict = self.goods\_info[choices - 1]  print('{}:名称：{}；售价：{}'.format(choices, good\_dict['name'], good\_dict['price\_out']))  print("1-购买")  print("2-返回")  choices = int(input("是否确定购买？"))  if choices == 1:  cnt = good\_dict['count']  print('当前商品剩余库存数量：{}。'.format(cnt))  buy\_count = int(input("请输入购买数量："))  while buy\_count <= 0 or buy\_count > cnt:  buy\_count = int(input('请输入正确的购买数量：'))  cnt = cnt - buy\_count  good\_dict['count'] = cnt  # 如果剩余库存数量为0，可以将该产品l字典从列表中移除。  if cnt == 0:  self.goods\_info.remove(good\_dict)  print('购买{}成功，数量：{}。'.format(good\_dict['name'], buy\_count))  sell\_dict = {'name': good\_dict['name'], 'profit': good\_dict['price\_out'] - good\_dict['price\_in'],  'count': buy\_count}  info = '{} 销货{}，其成本为{}，售价为{}，销售数量为{}，此单收入为{}\n'.format(self.t, good\_dict['name'],  good\_dict['price\_in'],  good\_dict['price\_out'],  buy\_count,  sell\_dict['profit'] \* buy\_count)  log('sell\_info.txt', info) # 销货日志记录  revise\_info\_file(self.goods\_info)  self.sell\_info.append(sell\_dict)  self.income += sell\_dict['profit'] \* buy\_count  self.funds += sell\_dict['profit'] \* buy\_count  else:  return   # 显示销售和进货详情  def show\_in\_and\_out(self, kind):  if kind == 1:  for index, sell\_dict in enumerate(self.sell\_info):  print('{}:名称：{}；利润：{}；数量：{}'.format(index + 1, sell\_dict['name'], sell\_dict['profit'],  sell\_dict['count']))  elif kind == 2:  for index, purchase\_dict in enumerate(self.purchase\_info):  print(  '{}:名称：{}；成本：{}；售价：{}；数量：{}'.format(index + 1, purchase\_dict['name'],  purchase\_dict['price\_in'],  purchase\_dict['price\_out'], purchase\_dict['count']))   # 收支查询  def income\_expenditure(self):  while True:  print("1-销售情况")  print("2-进货情况")  print("3-总财产情况")  print("4-退出")  choices = int(input("请选择操作:"))  if choices == 1:  self.show\_in\_and\_out(1)  print("总利润为{}元".format(self.income))  elif choices == 2:  self.show\_in\_and\_out(2)  print("进货总支出为{}元".format(self.expenditure))  elif choices == 3:  print("销售情况：")  self.show\_in\_and\_out(1)  print("进货情况：")  self.show\_in\_and\_out(2)  print("目前资金为{}元".format(self.funds))  else:  break   # 顾客功能页  def customer\_page(self):  while True:  print("""  1-查询商品信息  2-购买商品  3-退出  """)  choices = int(input("请选择操作:"))  if choices == 1:  self.show\_goods(False)  elif choices == 2:  self.sell\_good()  else:  break   # 店员功能页  def clerk\_page(self):  while True:  print("""  1-查询商品信息  2-进货  3-售货  4-查询收支  5-退出  """)  choices = int(input("请选择操作："))  if choices == 1:  self.show\_goods(True)  elif choices == 2:  self.purchase\_good()  elif choices == 3:  self.sell\_good()  elif choices == 4:  self.income\_expenditure()  else:  break   if \_\_name\_\_ == '\_\_main\_\_':  management = ManagementSystem()  management.creat\_goods\_info('product\_info.csv')  while True:  print("""  1-顾客  2-店员  3-退出  """)  choice = int(input("请选择操作："))  if choice == 1:  management.customer\_page()  elif choice == 2:  management.clerk\_page()  else:  break |

**数据文件：**

****