**《Java技术》实验报告**

**实验三： 2020 年5 月 5日**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **学院** | | 计算机与信息学院 | | | **专业班级** | |  | **姓名** |  | | **成绩** | |  |
| **课程**  **名称** | | Java技术 | | | | **实验项目**  **名 称** | 货物进销管理系统 | | | **指导 教师** | | 路强 | |
| 教师评语 |  | | | 教师签名：  年 月 日 | | | | | | | | | |
|  | | | **一、实验目的**  1．掌握Java中文件的读写操作。  2．学会使用Java提供的实用类（Vector, ArrayList）来完成特定的功能。  3．掌握字符串类（String, StringBuffer）的使用。  4．掌握用面向对象的方法分析和解决复杂问题。  **二、实验原理**  编写一个Inventory.java完成以下功能（没有学过Java文件处理之前，各位同学可以使用硬编码将数据放进两个Vector变量里。等学过Java文件处理之后，再补充数据文件读取部分）：  1．程序首先打开并读取Inventory.txt中记录的所有库存记录，然后读取Transactions.txt，处理这个文件中包含的事务，记录发货记录到Shipping.txt，并记录错误信息到Errors.txt中。最后更新库存到另外一个文件NewInventory.txt中。  2．文件Inventory.txt和NewInventory.txt的每行包含一个存货记录，没条记录包含下面一些字段息，这些字段之间用一个tab分开（见后面的文件格式）：     |  |  | | --- | --- | | 字段 | 格式和含义 | | Item number | 字符串型，货物编号 | | Quantity | 整型，货物数量 | | Supplier | 字符串型，供应商编号 | | Description | 字符串型，货物描述 |   3．字段Items按照从小到大的顺序写入文件的。注意Item号不必连续，如Item号为752的后面可能是800。  4．文件Transactions.txt包含几个不同的处理记录（每行一条记录）。每条记录前面以一个大写字母开头，表示这条记录是什么类型的事务。在不同的大写字母后面是不同的信息格式。所有的字段也是以tab键分开的（见Transactions.txt文件格式）。  5．以'O'开头的事务表示这是一个发货订单，即某一种货物应该发给特定的客户。Item number和Quantity的格式如上面表格定义。Custom编号和上面的Supplier编号一致。处理一条定单记录（以'O'开头的事务）意味着从减少库存记录中相应货物的数量（减少的数量=发货单中的数量），记录发货信息到Shipping.txt中。注意：Inventory.txt中的quantity不应该小于0，如果对于某一种货物，库存的数量小于发货单的数量的话，系统应该停止处理发货单，并记录出错信息到Errors.txt。如果对于某一种货物有多个发货单，而且库存总量小于这些发货单的总和的话，系统应该按照发货单中的数量从小到大的有限原则满足客户。也就是说，对于某一种货物如果一个数量Quantity少的发货单没有处理之前，数量Quantity多的发货单永远不会被处理。（这种处理原则不受发货单记录在Transactions.txt的先后顺序影响）  6．以'R'开头的事务表示这是一个到货单记录，在'R'后面是Item number和它的数量Quanlity。处理一条到货单意味着增加库存中相应货物的数量（增加的数量=到货单中的数量）。注意：如果在Transactions.txt文件中，到货单出现在发货单之后，到货单中的货物数量可以用来填补发货单中的数量（可以理解成在Transactions.txt中，优先处理到货单）。  7．以'A'开头的事务表示向库存中增加一种新的货物（即这种货物以前库存中没有），在'A'后面是Item number，供应商supplier以及货物的描述description。处理一个新增货物记录意味着向库存中增加一个数量Quantity为0的新的Item。你可以假设在一个Transactions.txt中，新增货物记录总是出现在第一个到货单之前。  8．以'D'开头的事务表示从库存中删除一种货物，在'D'后面是Item号。删除操作总是在所有的事物处理之后才被处理，以保证对于可能出现的同一种货物的发货单的操作能在删除之前被正确处理。如果要删除的某种货物的库存量Quantity不为0的话，系统应该向Errors.txt记录出错信息。  9．文件Shipping.txt中的每一行代表给某一客户的发货信息。Shipping.txt中的每一行分别是客户编号、Item号、货物数量，它们之间用tab键分隔。如果发货单中有两条客户编号和Item编号一样的记录，在Shipping.txt中应该将这两条发货信息合并（即将它们的数量相加）。  10．Errors.txt文件包含未发送的发货记录和库存量大于0的删除记录。Errors.txt每一行包含Custom编号、Item编号以及发货单上的数量Quantity。对于删除操作，Custom编号为0，数量Quntity为库存中的Quantity.  **三、使用硬件、软件环境**  PC笔记本一台，CPU 8G，1T内存。装有Windows10操作系统。  使用到了JDK1.8，Intellij IDEA等相关软件   1. **实验过程、步骤及原始记录(算法、原程序、测试结果，分析等)** 2. **设计具体的模块。**这里我设计了三个class。首先是主类Inventory，再就是为了储存订单和库存数据的InventoryData和TransactionData类。这里我将主功能集成在主类里面，包括读取Inventory.txt的库存记录的readInventory（）、读取Transactions. txt的事务的readTransactions()、处理事务的process()、记录发货记录到Shipping.txt的ship(int i)、记录错误信息到Errors.txt的error(int i,char Type)、更新信息到NewInventory.txt的updateInventory()这几个方法。 3. **将上一步具体类和方法完善** 4. 读取类方法   这里我分为了两个方法、其实一个方法也是可以的。利用FileReader和BufferdReader对文件中的数据进行循环录入InventoryList、TransactionList中。这里有几个细节性的问题。一个是读取Transaction时存在split的分割段数不同的问题，采用定位取类型的方法分开处理，这里重载了InventoryData的构造方法。   1. 记录类方法   这一部分的用处在于便捷简化处理方法中的操作，让每一次处理事件时，都可以用这里的方法，主要是ship（int i）、error（int i,char Type）、updateInventory（）这三个方法。这里使用了FileWriter和BufferedWriter的流类进行文件录入，并且允许续写。录入格式参考题目要求，对InventoryData和TransactionData的内容进行选择行的输入。  其中error的处理因为存在两种错误类型，在这里进行区别并进行对应的处理。   1. 处理事件方法   这是代码的核心处理部分，针对不同文件开头的处理顺序：A->R->O->D，这里采用了四次处理，根据相应的情况对已经录入到List中的Inventory和Transaction数据进行逐条处理和匹配。  **注：源程序具体参见附录文件夹exp03，编译环境为Intellij IDEA**  **实验结论、分析、思考题与心得体会**   1. 实验前和实验中一定要对相关的方法和内容进行调查，要了解涉及到的相关系统提供的算法。比如分割字符串的split、查找特定位置字符的charAt（）、List常用的比较函数CompareTo等等。 2. 模块化的设计有利于分开进行代码编写，但也使文件的结构变得复杂。如果利用合适和安排合理，会使结构更加清晰，反之则会出问题，不要为模块化而模块化，而是在分割出来合适复用代码时在这样做。 3. 在设计程序前先对内容进行安排和规划，这样在进行代码书写是会少走许多弯路。 4. 在代码书写过程中，注释的书写十分重要。代码注释的书写不仅对下次程序继续完善提供了思路，还可以记录错误、记录时和安排、重要信息参数等等。这极大的提高了代码的可读性和操作性。这点在我修改代码时深有体会。模块化的设计可以简化设计流程。将内容包装成类，数据封装为成员变量、操作封装为方法。只需要预留好相应的接口使用位置，较低的耦合度更有利于代码的设计和修改。这也让我意识到了面向对象编程的一大长处。 5. 在测试的过程中要熟练的采用debug的手段设置关键位置的断点，通过step into进行跟进、并参考相关步骤的参数进而改正问题。 | | | | | | | | | | |