DSA - Assignment 1

Sale Management System

using linked list data structure

INTRODUCTION
Your **first assignment in this block** will be using linked list data struture for implementing a small Sale Management System (SMS) in Java language.
SMS manages information about products, customers and product ordering items. These information are:
About a product:

1. pcode (string): the code of the product (this should be unique for the product).
2. pro_name (string): the name of the product.
3. quantity (integer): the number of products with the same code in a shop at beginning of a day.
4. saled (integer): the number of products with the same code, which are saled in the day. Condition: saled ≤ quantity.
5. price (double): The price of the product.

About a customer:

1. ccode (string): the code of the customer (this should be unique for the customer).
2. cus_name (string): the name of the customer.
3. phone (string): The phone number of the customer (must contain digits only).

About ordering:

1. pcode (string): the code of the product to be ordered.
2. ccode (string): the code of the customer.
3. quantity (integer): the number of ordered products.

YOUR TASKS
You should use 3 linked lists, each one is used to store data for products, customers or product ordering items. You should create linked lists from scratch, do not use list structures available in java like ArrayList, Vector or LinkedList classes.
On running, your program displays the menu as below:
Product list (8 marks):
1.1. Load data from file
1.2. Input & add to the end
1.3. Display data
1.4. Save product list to file
1.5. Search by pcode

1.6.	Delete by pcode
1.7.	Sort by pcode
1.8.	Add after position k
1.9.	Delete the node after the node having code = xCode

Customer list (1 mark):
2.1.	Load data from file
2.2.	Input & add to the end
2.3.	Display data
2.4.	Save customer list to file
2.5.	Search by ccode
2.6.	Delete by ccode

Order list (1 mark):
3.1.	Input data
3.2.	Display data with total value
3.3.	Sort by pcode + ccode

tasks explanation
Product list (8 marks):
1.1.	Load data from file
Allow a user to input the file name that contains information of products. The content of the file may be

```
P03  |  Sugar      |  12   |  3   |  25.1
P01  |  Miliket    |  10  |  5   |  5.2
P02  |  Apple      |    5  |  2   |  4.3
P05  |  Rose       |    7  |  6   | 15.4
P07  |  Beer       |  11  |   3  | 12.2
P04  |  Book       |    9  |   5  |  5.2
```

The first line means that: pcode = P03, pro_name = Sugar, quantity = 12, saled = 3 and price = 25.1

1.2.	Input & add to the end
Allow a user to add new information about a product. After checking validation of data (including that the pcode could not be duplicated), the product is added to the end of the list.

1.3.	Display data
Display data in format:
pcode | pro_name | quantity | saled | price | value
where value = price*saled
For example after loading the above file, this option give the output below:

```
code |   Pro_name  |  Quantity  |  saled |  Price   |   Value
-----------------------------------------------------------------
P03  |  Sugar        |     12       |     3   |   25.1    |    75.3
P01  |  Miliket      |     10       |     5   |     5.2    |   26.0
```

```
P02  |  Apple          |       5        |    2  |     4.3   |     8.6
P05  |  Rose           |       7        |    6  |   15.4   |    92.4
P07  |  Beer           |     11         |    3  |    12.2  |    36.6
P04  |  Book           |       9        |    5  |      5.2  |   26.0
```

1.4.	Save product list to file
Allow a user to input the file name and save the product list to the file. The information and format like the option 1.3.

1.5.	Search by pcode
Write the function:
Node search(String xCode) {}
which return reference to the node whose info contains the product with pcode = xcode.
Allow a user to input the pcode to be searched and display the result: found or not found.

1.6.	Delete by pcode
Write the function:
void dele(String xCode) {}
which deletes the node whose info contains the product with pcode = xCode.
Allow a user to input the pcode to be deleted and then delete the product having that pcode.

1.7.	Sort by pcode

1.8.	Add after position k
The position of the first element is 0, the second's is 1

1.9.	Delete the node after the node having code = xCode

Customer list (1 mark):
2.1.	Load data from file
Allow a user to input the file name that contains information of customers. The content of the file may be

```
C03  |  Hoa   | 1902
C01  |  La      | 1901
C02  |  Canh  | 1903
C05  |  Cay    | 1910
```

The first line means that: ccode = C03, name = Hoa, phone = 1902

2.2.	Input & add to the end
2.3.	Display data
2.4.	Save customer list to file
2.5.	Search by ccode
2.6.	Delete by ccode

Ordering list (1 mark):

3.1. Input data

Allow a user to input saleing item.

When running, the screen looks like:

```
Enter product code:
Enter customer code:
Enter quantity:
```

After the user enter pcode and ccode, the program check and acts as follows:

- If pcode not found in the products list or ccode not found in the customers list then data is not accepted.
- If both pcode and ccode are found in the order list then data is not accepted.
- If pcode and ccode found in products and customers lists but saled = quantity then inform the user that the product is exhausted.
- If pcode or ccode found and in the products list saled < quantity and k is the entered quantity then if k <= quantity - saled then data is accepted and added to the end of the Order list.

3.2. Display odering data

3.3. Sort by pcode + ccode

Submission Requirements

Create the directory with a name like **<class>-<name><roll number>-ASS1**, e.g.

SE0508-QuangTV00456-AS1 **(1)**

The (1) directory contains the following files:

1. The run.bat file to run your program.
2. Your source code files.
3. Your input test files.

The statements in run.bat file may be:

```
cls
javac Main.java
java Main
pause
del *.class
```

Compress the folder (1) to .zip (or .rar) file (with the same name) and upload to cms.

Assignment assessment

You will be asked to modify immediately and to explain your assignment in lab room to be sure that you are really the author of the assignment you submitted.