DSA - Assignment 1 (Deadline: the first slot of 6th week)
Sale Management System using linked list data structure

INTRODUCTION

Your **first assignment in this block** will be using linked list data structure 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. name (string): the name of the product.
3. maker (string): the maker of the product
4. unit (string): the unit of the product (box, carton, liter, ...)
5. category (string): the category of the product
6. quantity (integer): the number of products with the same code the shop has.
7. saled (integer): the number of products with the same code, which are saled. Condition: saled ≤ quantity.
8. 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. name (string): the name of the customer.
3. phone (string): The phone number of the customer (must be unique and contains digits only).

About order:

1. pcode (string): the code of the product to be ordered.
2. ccode (string): the code of the customer
3. odate (date): the date when customer order the product
4. sdate (date): the date when the product shipped to the customer
5. quantity (integer): takes number of ordered products (must be greater than 0).

YOUR TASKS: You should use 3 linked lists, each one is used to store data for products, customers or ordered 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 (4 marks):

1.1. Load data from file
 (load book list from products.txt)

1.2. Input & add to the end
(input and validate product data, then add the product to the end of the list)

1.3. Display data
(display info of all products in the product list)

1.4. Save product list to file
(save the product list to file products.txt)

1.5. Search by pcode
(input pcode to be searched, then return the found product data or not found)

1.6. Delete by pcode
(input pcode, then delete found product;
Remember to delete all related records in order list first)

1.7. Sort by pcode
(dislay products in ascending order of the pcode)

1.8. Input & add to beginning
(input and validate product data, then add the product to the begin of the list)
1.9. Add before position k
(input and validate product data, then add the product to the position k-1 of the list)
1.10. Delete position k
(delete the book to the position k of the list)
1.11. Search by name
(input name to be searced, then return the found products data or not found)
1.12. Search ordered by pcode
(input pcode to be searced, then return the found product data or not found;
Then list all customers who ordered the product)

Customer list (1.5 mark):
2.1. Load data from file
(load customer list from cumtomers.txt)
2.2. Input & add to the end
(input and validate customer data, then add the customer to the end of the list)
2.3. Display data
(display info of all customers in the customer list)
2.4. Save cusmtomer list to file
(save the customer list to file customers.txt)
2.5. Search by ccode
(input ccode to be searched, then return the found customer data or not found)
2.6. Delete by ccode
(input ccode, then delete found customer;
Remember to delete all related records in order list first)
2.7. Search by name
(input name to be searced, then return the found customers data or not found)
2.8. Search not shipped products by ccode
(input ccode to be searced, then return the found customer data or not found;
Then list all products that are still not shipped to customer)

Order list (1.5 mark):
3.1. Load data from file
(load order list from orders.txt)
3.2. Order book
(input pcode, ccode; check if product and customer exsist; check if order quantity is less than or equals to quantity of found product; odate to today, and sdate to null; add order to the begin of the lending list; substract order quantity from product quantity; add order quantity to product saled;)
3.3. Display data
(display info of all orders in the order list with total value, where total value equal price * quantity)
3.4. Save order list to file
(save the order list to file orders.txt)
3.5. Sort by pcode + ccode
(display orders in the descending of pcode first, then in the descending of the rcode)
3.6. Ship order by pcode + ccode
(input pcode and ccode to be searched order, if order is found and is not shipped, then set sdate to to today;)

Directly modify coding and answer questions at classroom (3 marks)

Submission Requirements

Create the directory with a name like **<class>_<roll number>_<name>_ASS1**,
e.g. **SE0508_HE180045_QuangTV_ASS1   (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 file (with the same name) and upload to Assignment1 in edunext.

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.