ASSINGEMENT # 2

Read the instructions below carefully before start coding.

Students are ONLY allowed to use:

- Materials on his/her computer (including JDK, NetBeans).
- For distance learning: Google Meet, Hangout (for Exam Ming Purpose).

Follow the steps below to complete PE:

1. Create a folder to save given projects, e.g. CSD_given (1). Download the given materials to (1).
2. Complete the requirements of the test.
3. Before submission:
    - Clean and Build Project (Shift+F11)
    - Then rename the folder dist to run. (If the folder run already exists, delete it before renaming).
4. Submission:
    - To submit the project Q1, at first you must select the Question No: 1
    - Browse and select the project folder (e.g. Q1, or Q1A, or Q2, …)
    - Then click the Submit button.
5. Do not use accented Vietnamese when writing comments in programs.
6. Do not add new import statement(s) to given files.
7. Software tools must be used: NetBeans IDE 8.x and Java JDK 1.8.

If at least one of the above requirements is not followed, the exam will get ZERO.

Notes:

- The input and expected output below are only used to test your codes.
- The input and expected output in the real testcases (for marking) are different with in the examples below.
- Do not hardcode with the given expected results.

Troubleshooting:

If the given project (e.g. Q1) runs with error, you need to run "Clean and Build Project" (Shift+F11). If still error, try to rename or copy the project to other one, e.g. from Q1 to Q1X or Q1Y.

Problem Description

The given files contain statements to implement a program for managing a supply chain inventory system. The structure of the main classes is as follows:

- **Class Product**: contains information about a product, including id, name, and quantity in stock.
- **Class Node**: includes a Product object and a next pointer for linking.

- **Class InventoryList**: is a singly linked list that manages Nodes with complete product information. This class plays the role of managing all products in the warehouse.
- **Class ShippingList**: is a stack structure (implemented as a linked list), where the component data of the Nodes contains the id and shipping quantity. This class represents a list of products scheduled for shipping.
- **Class SupplyChainManager**: is the main class of the program, containing an InventoryList to store data about all products in the warehouse and a ShippingList corresponding to the list of products scheduled to be shipped.

Students are required to carefully read the provided code segments to fully understand the relationships between the classes and the functions within each class. The specific tasks of the test are to execute the following requirements:

a. f1() (2.5 marks):

To complete the requirement f1, students need to fulfill two specific tasks: implement the function addLast() in the InventoryList structure and the function push() in the ShippingList structure. The expected output used to test your code is as follows:

Warehouse Inventory: (P101,Laptop,50) (P102,Smartphone,120) (P103,Headphones,200) (P104,Charger,150) (P105,Monitor,80) (P106,Keyboard,100)

Shipping List: (P102,35) (P106,40) (P101,15) (P105,25) (P103,30)

b. f2() (2.5 marks):

To complete the requirement f2, students need to perform two specific tasks: implement the pop() function of the ShippingList structure and the updateInventory function in the SupplyChainManager class, then use the returned result from pop() to execute the updateInventory action for the f2 function.

The updateInventory function should:

- Search for the product (id) to update in the inventory. If found, proceed to step 2.
- If the product is found in the InventoryList, update its stock by subtracting the shipping quantity.

The expected output used to test your code are as follows:

Warehouse Inventory: (P101,Laptop,50) (P102,Smartphone,120) (P103,Headphones,200) (P104,Charger,150) (P105,Monitor,80) (P106,Keyboard,100)

Shipping List: (P102,35) (P106,40) (P101,15) (P105,25) (P103,30)

Warehouse Inventory: (P101,Laptop,50) (P102,Smartphone,85) (P103,Headphones,200) (P104,Charger,150) (P105,Monitor,80) (P106,Keyboard,100)

Shipping List: (P106,40) (P101,15) (P105,25) (P103,30)

Explanation: Initially, the ShippingList contains 5 elements corresponding to 5 products scheduled for shipping. After performing pop() to remove the highest priority element P102 from the list, the ShippingList will only have 4 elements. When executing the updateInventory action, we find P102 in the InventoryList and update its stock by subtracting the shipping quantity (120 - 35 = 85).

c. f3() (2.5 marks):

Perform the pair of operations pop() and updateInventory for all items in the ShippingList. The expected output used to test your code is as follows:

Warehouse Inventory: (P101,Laptop,50) (P102,Smartphone,120) (P103,Headphones,200) (P104,Charger,150) (P105,Monitor,80) (P106,Keyboard,100)

Shipping List: (P102,35) (P106,40) (P101,15) (P105,25) (P103,30)

Warehouse Inventory: (P101,Laptop,35) (P102,Smartphone,85) (P103,Headphones,170) (P104,Charger,150) (P105,Monitor,55) (P106,Keyboard,60)

Shipping List: Empty

Explanation: Since the updateInventory action is executed for all products in the ShippingList, in the end, the ShippingList will be empty. All products with corresponding IDs in the InventoryList will have their quantities updated by subtracting the shipping quantity. Note that the Charger (P104) still retains its original quantity because it is not in the shipping list.

d. f4() (2.5 marks):

Perform the pair of actions, pop() and updateInventory, for all the items in the ShippingList and calculate the total quantity of products that have been shipped. The expected output used to test your code is as follows:

Warehouse Inventory: (P101,Laptop,50) (P102,Smartphone,120) (P103,Headphones,200) (P104,Charger,150) (P105,Monitor,80) (P106,Keyboard,100)

Shipping List: (P102,35) (P106,40) (P101,15) (P105,25) (P103,30)

Total Shipped Quantity: 145

Explanation: There are 5 products in the ShippingList. The total shipping quantity is calculated as follows: Value = 30 + 25 + 15 + 40 + 35 = 145 units