Lab Exercise #2: Inventory Manager

In this laboratory exercise you are to create a simple inventory system that will manage products from different brands. An item in the inventory for a given brand is either a product packed as a single item, or several products boxed together. To represent these two kinds of product in the inventory you will implement the following classes:

SingleProduct Class

Method	Description
`SingleProduct (String brand)`	Creates a new single product object for the given brand
`String getBrand()`	Returns the brand of the product

BoxedProduct Class

Method	Description
`BoxedProduct (String brand, int quantity)`	Creates a new boxed product object for the given brand containing the specified number of items
`String getBrand()`	Returns the brand of the boxed product
`int getQuantity()`	Returns the number of items in the box

As before, you will separate the UI functionalities from the needed inventory functionalities. To achieve this, you are to create another class for inventory management:

InventoryManager Class

Method	Description
`void add(SingleProduct p)`	Add one individually packed product to the inventory
`void add(SingleProduct p, int quantity)`	Add the specified number of individually packed product to the inventory
`void add(BoxedProduct p)`	Add one box to the inventory
`void add(BoxedProduct p, int quantity)`	Add the specified number of boxes to the inventory
`String[] getBrands()`	Return all brands in the inventory
`BoxedProduct[] getBoxes(String brand)`	Return the array containing all the boxes for the given brand

`SingleProduct[] getSingles(String brand)`	Return the array containing all individually packaged item for the given brand

The main client app will perform all the user interactions. When the app executes, it will present a menu containing 3 options – (1) add a single product, (2) add a box product, or (3) exit. When option 1 is selected, the app will ask for the brand of the single product and the quantity to add in the inventory. When option 2 is selected, the app will ask for the brand, the number of items in a box, and the number of boxes to add in the inventory. After inputting the needed data when option 1 or 2 is selected, the app will present the menu again. Selecting the third option will present a report showing, for each brand, the number of single items and boxes, along with the total product pieces (total single items + total pieces in all boxes) and then end the program.

Your client app should use all the three classes and all the methods specified above. Your submission must contain 4 Java files – one file for the main client app and one file for each class above.

Sample Execution of the Inventory System

```
Options:
      [1] Add Single Product
      [2] Add Box Product
      [3] Exit
Choice: 1
Brand: Brand X
Quantity: 1

Options:
      [1] Add Single Product
      [2] Add Box Product
      [3] Exit
Choice: 1
Brand: Brand Y
Quantity: 5

Options:
      [1] Add Single Product
      [2] Add Box Product
      [3] Exit
Choice: 2
Brand: Brand Y
Items in Box: 6
Quantity: 1

Options:
      [1] Add Single Product
      [2] Add Box Product
      [3] Exit
Choice: 2
Brand: Brand Y
Items in Box: 4
Quantity: 3
```

```
Options:
      [1] Add Single Product
      [2] Add Box Product
      [3] Exit
Choice: 2
Brand: Brand X
Items in Box: 4
Quantity: 5

Options:
      [1] Add Single Product
      [2] Add Box Product
      [3] Exit
Choice: 3

-Inventory Report-

Brand X
      Singles: 1
      Boxes: 5
      Total Pieces: 21

Brand Y
      Singles: 5
      Boxes: 2
      Total Pieces: 23
```

Rubric

Your grade will be based on the following scoring system:

Criteria	Score
SingleProduct Class	**5pts** for every correctly implemented method
BoxedProduct Class	**5pts** for every correctly implemented method
InventoryManager Class	**5pts** for every correctly implemented method
Client Input	**0pts** Does not process the user input **5pts** Process user input but with errors **10pts** Correctly process all user input but not using all three classes **15pts** Correctly process all user input using all three classes
Client Output	**0pts** No report **5pts** Has incorrect detail in the report **10pts** Correctly outputs the report but not using all three classes **15pts** Correctly outputs the report using all three classes