Self-Checkout Software

In the previous phase, you may have seen that your code contains many variables, some of which are interconnected. To elevate the quality of your programs in this phase, you'll be leveraging object-oriented concepts. A crucial move in this direction is the introduction of a new class called "StoreItem". This class is tailored to encapsulate various aspects of items within your store and is governed by the following UML specification:

StoreItem
- number : int
- quantity: int
- name: String
- price double
+ <u>TotalProfit</u> :double
- reserved : int
+ sold :int
+ isAvaliable(int soldQuatity): boolean
+ reserve(int soldQuantity):boolean
+ sell() :void
+ cancel():void
+ getTotalCost ():double
+printInfo():void
+ printReserved():void
+ Setters & getters (as needed)
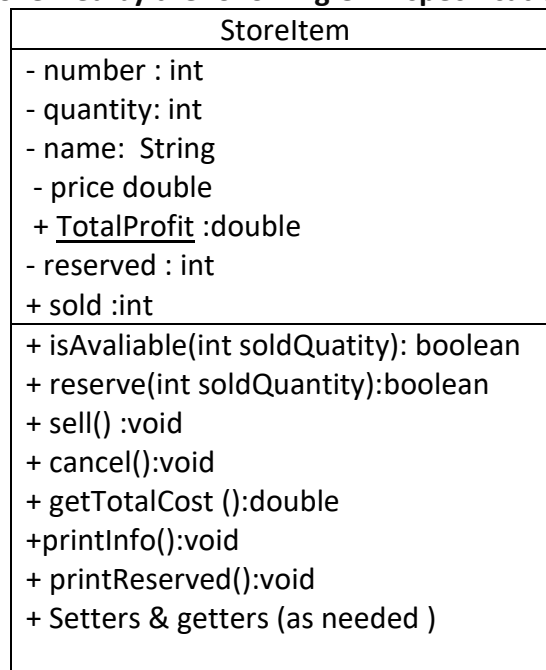
Attributes:
- Number: Stores the unique identification number of the item.
- Name: Stores the name of the item.
- Quantity: Tracks the available quantity of the item in the store.
- Price: Records the price of a single unit of the item.
- Reserved: Keeps count of the items selected by users but not yet checked out.
- Sold: Records the total number of items sold.
- Total profit: Accumulates the total profit generated from all items.

Methods:
- **isAvailable(int soldQuantity):** Checks if the specified quantity of items is available by evaluating both the available quantity and reserved items.
- **reserve(int soldQuantity):** If the specified quantity is available, updates the reserved attribute. Returns true if the reservation is successful, otherwise returns false.
- **sell():** Upon checkout, updates the quantity by deducting the reserved items, adjusts the total profit, and resets the count of reserved items to zero.

- **cancel():** Resets the count of reserved items to zero.
- getTotalCost(): Computes the total cost for the item by multiplying the reserved items with the price.
- **printInfo():** Displays comprehensive information about the item, akin to the details presented on the menu.
- **printReserved():** Prints item information on the bill if the number of reserved items exceeds zero.

Update your program to leverage this class effectively. By implementing the StoreItem class according to this specification, you can streamline the management of your store's inventory, facilitate sales transactions, and provide clearer and more organized information about your items.
   -

   -