Exercise: Sales Management System

Create a Java program that models a simple Sales Management System. The system should have the following classes:

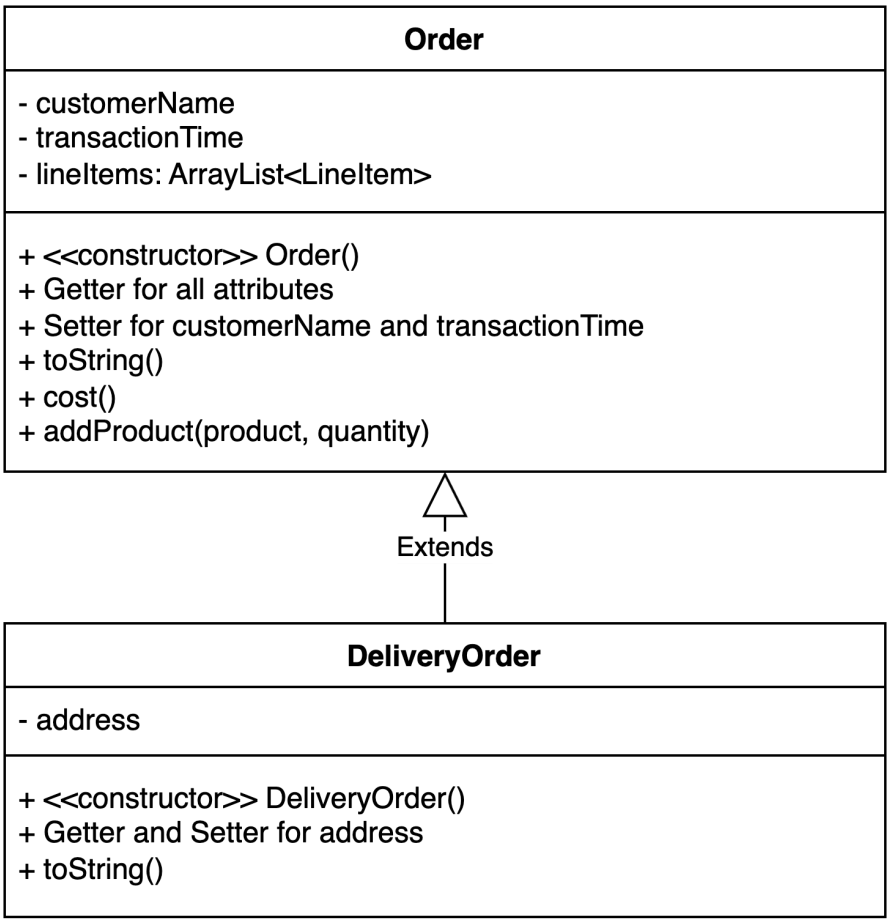
1. **Product**:

Product
- sku - name - category - price - quantity
+ <<constructor>> Product(sku, name, category, price, quantity) + Getter and Setter for all attributes + toString()

2. **LineItem**:

LineItem
- product: Product - quantity
+ <<constructor>> LineItem(product, quantity) + Getter and Setter for all attributes + toString() + cost()

- Method **cost()** to calculate and return the cost of the products in the line item (product price x quantity).



3. **Order**:

- Field **lineItems** is assigned to an empty **ArrayList** by default
- Method **cost()** to calculate and return the total cost of the order (sum of total costs of all line items in the order).
- Method **addProduct(product, quantity)**:
    - If quantity to purchase ≤ available quantity: Create a new instance of **LineItem** and add to the order's line item list. Then update product availability and return true.
    - If quantity to purchase > available quantity: do nothing and return false.

4. **DeliveryOrder**: A subclass or **Order**

Requirements:

1. Write separate Java files for each class (Product.java, LineItem.java, Order.java and DeliveryOrder.java). Use appropriate access modifiers for attributes and methods (e.g., private, public, protected) and use appropriate data types for attributes and methods (e.g., int, double, String, void)

2. Write Main class with a main() method to do the following:

a. Initialize products as described in table below:

#	SKU	Name	Category	Price	Quantity
1	F523	Tomato	Food	1.5	589
2	A763	Zara shirt	Appearance	12.0	90
3	H320	Kitchen table	Household	299.0	13
4	E092	IPhone	Electronic	1000.0	4
5	S108	Football	Sport	19.9	2

b. Create a new **Order** for “Mike Tyson” at “2023-08-08 11:30:00”, then:
- Add 5 “Tomato” to the order then print the result (success or fail) and total cost of order
- Add 2 “IPhone” to the order then print the result (success or fail) and total cost of order
- Add 4 “Football” to the order then print the result (success or fail) and total cost of order
- Print all the information of the order

c. Create a new **Delivery order** for “Chris Evans” at “2023-08-09 13:14:00”, then:
- Set address to “123 Cau Giay”
- Add 3 “Zara shirt” to the order then print the result (success or fail) and total cost of order
- Add 3 “IPhone” to the order then print the result (success or fail) and total cost of order
- Print all the information of the order