Scenario: Shop Management System

You are tasked with developing a Shop Management System for a local grocery store. The store owner wants a system that can handle inventory management, sales tracking, and customer information. As first-year programmers, you need to implement the system using Java and ensure it incorporates the following concepts:

1. **File I/O**: The system should be able to read and write data to files for storing inventory information and sales records. This includes functionalities such as reading product details from a file, updating inventory (adding to inventory, adding info after each sale), and storing transaction information from customers. **[20 points]**

2. **Inheritance**: Implement inheritance to represent different types of products in the inventory. For example, you might have a base class `Product`, with subclasses like `FoodProduct`, `ElectronicsProduct`, etc. Each subclass should inherit common attributes and methods from the `Product` class while being able to define their specific properties. **[20 points]**

3. **Polymorphism**: Utilize polymorphism to handle different types of transactions from customers. For instance, you might have a `Transaction` class with methods for processing cash transactions and card transactions. Each transaction type should override a common method to calculate the total cost based on specific rules. **[20 points]**

4. **Abstraction**: Design the system with abstraction to hide unnecessary implementation details from the user interface. For example, you could abstract away the intricacies of inventory management by providing simple methods for adding/removing products or updating stock levels, without exposing the underlying data structures. **[20 points]**

5. **Encapsulation**: Ensure that each class encapsulates its data and behavior effectively. Use access modifiers to restrict access to certain data members and methods, allowing for better control over data manipulation and ensuring data integrity. **[20 points]**

Project Structure:

The project should consist of at least three classes (not inclusive of subclasses):

1. `Product`: Base class representing a product with common attributes like name, price, and quantity.

a. Subclasses of `Product` (e.g., `FoodProduct`, `ElectronicsProduct`, etc.) representing different types of products with specific properties.

Ensure to use Inheritance and have at least 3 sub classes.

2. `InventoryManager`: A class responsible for managing the store's inventory.
   Fields include:
   - Name/Title: The name or title of the item.
   - ID/Code: A unique identifier for the item, which can be alphanumeric.
   - Quantity: The quantity of the item available in stock.
   - Price: The price of the item, either per unit or total.
   - Category/Type: The category or type of the item, which helps in organizing and searching.

   Methods include:
   - addItem: Adds a new item to the inventory.
   - removeItem: Removes an item from the inventory.
   - displayAllItems: Displays all details of all items in the inventory.
   - itemExists: Checks if an item exists in the inventory.
   - getItemDetails: Displays details of a specific item.

Ensure to use Encapsulation on fields

3. `Transaction`: An abstract class handling simple transactions, with one method for processing payments and one method to write sales records to a text file.
   Fields include:
   - Payment method: The customer can either use CASH or CARD
   - Order Code: A code given to the order created.
   - Name(s)/Title(s): The name or title of the items chosen
   - Quantity: The quantity of the item(s) selected.
   - Total Price: The total price of the item (s)

   Methods include:
   - calculateOrder: Calculate total price for the item(s) and add 15% V.A.T if paying by CASH or 18% V.A.T if paying by CARD.
   - writeToFile: Append all fields and calculations to the text file

Ensure to use Polymorphism
Ensure to use File I/O

Research Component:

The research component of the project involves gathering data from local shopkeepers regarding their inventory, including real-time prices and quantities whenever possible. This process entails reaching out to various local businesses and establishing communication channels to collect relevant information about the items they stock.

Ensure to create a menu that calls each function represented in the project.

GOOD LUCK! 😊