**Example: Bank Account Management System**

**Requirements**

**1. Create a Java program that manages bank account data.**

**2. Use OOPs concepts to design and implement the program.**

**Step 1: Create Classes and Objects**

**1. Account.java: Create an Account class with attributes like accountNumber, accountHolderName, and balance.**

**2. AccountManager.java: Create an AccountManager class that manages an array of Account objects.**

**a. array of object**

**b. List of account object.**

**Step 2: Encapsulation**

**1. Encapsulate attributes: Use private access modifiers to encapsulate attributes in the Account class.**

**2. Provide getter and setter methods: Create public getter and setter methods to access and modify the attributes.**

**Step 3: Abstraction**

**1. Abstract class: Create an abstract class BankAccount with attributes like accountNumber and accountHolderName(optional).**

**2. Abstract method: Declare an abstract method calculateInterest() in the BankAccount class.**

**Step 4: Inheritance**

**1. Inherit from BankAccount: Make the Account class inherit from the BankAccount class.**

**2. Override abstract method: Override the calculateInterest() method in the Account class.**

**Step 5: Final Keyword**

**1. Final variable: Declare a final variable BANK\_NAME in the Account class.**

**2. Final method: Declare a final method getBankName() in the Account class.**

**Step 6: Static**

**1. Static variable: Declare a static variable accountCount in the Account class.**

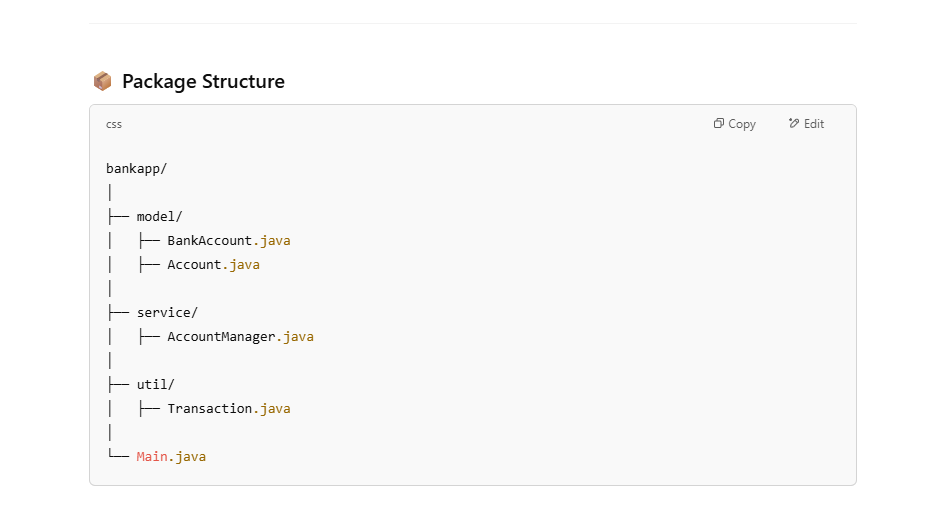
**2. Static method: Declare a static method getAccountCount() in the Account class.**

**Step 7: Interface**

**1. Interface: Create an interface Transaction with methods deposit() and withdraw().**

**2. Implement interface: Make the Account class implement the Transaction interface.**

**Package structure**

****

**Main method Sample code**

**Here's an example of a main method that demonstrates the usage of the Account and AccountManager classes:**

**public class Main {**

**public static void main(String[] args) {**

**AccountManager accountManager = new AccountManager();**

**do {**

**1: addAccount(), 2 : deposit() 3 : withdraw() 4 : displayAccountDetails()**

**5: calculateInterest() 6 : getAccountCount() , 7 :getBalance()**

**switch() {**

**}**

**}while()**

**}**

**}**