Write a Java program to create a class called "Bank" with a collection of accounts and methods to add and remove accounts, and to deposit and withdraw money. Also define a class called "Account" to maintain account details of a particular customer.

Sample Solution:

Java Code:

```
// Account.java
// Define the Account class
public class Account {
    // Declare a private variable to store the name of the account holder
    private String name;
    // Declare a private variable to store the account number
    private String accountNumber;
    // Declare a private variable to store the balance of the account
    private double balance;

    // Constructor for the Account class that initializes the name, account
    public Account(String name, String accountNumber, double balance) {
        // Set the name variable to the provided name parameter
        this.name = name;
        // Set the accountNumber variable to the provided accountNumber par
        this.accountNumber = accountNumber;
        // Set the balance variable to the provided balance parameter
        this.balance = balance;
    }

    // Method to retrieve the name of the account holder
    public String getName() {
        // Return the value of the name variable
        return name;
    }

    // Method to set the name of the account holder
    public void setName(String name) {
        // Set the name variable to the provided name parameter
        this.name = name;
    }

    // Method to retrieve the account number
```

```
    public String getAccountNumber() {
        // Return the value of the accountNumber variable
        return accountNumber;
    }

    // Method to set the account number
    public void setAccountNumber(String accountNumber) {
        // Set the accountNumber variable to the provided accountNumber par
        this.accountNumber = accountNumber;
    }

    // Method to retrieve the balance of the account
    public double getBalance() {
        // Return the value of the balance variable
        return balance;
    }

    // Method to set the balance of the account
    public void setBalance(double balance) {
        // Set the balance variable to the provided balance parameter
        this.balance = balance;
    }

    // Method to deposit a specified amount into the account
    public void deposit(double amount) {
        // Increase the balance by the specified amount
        balance += amount;
    }

    // Method to withdraw a specified amount from the account
    public void withdraw(double amount) {
        // Decrease the balance by the specified amount
        balance -= amount;
    }

    // Method to retrieve the account information
    public String getAccountInfo() {
        // Return a string containing the name, account number, and balance
        return "Name: " + name + ", Account Number: " + accountNumber + ",
    }
```

```
    }
```

The above class has three private attributes: name, accountNumber and balance. There are several methods to deposit, withdraw, maintain balance in an individual account, print account details and more.

```
    // Bank.java
    // Import the ArrayList class from the Java Collections Framework
    import java.util.ArrayList;

    // Define the Bank class
    public class Bank {

      // Declare an ArrayList to store Account objects
      private ArrayList<Account> accounts;

      // Constructor for the Bank class
      public Bank() {
        // Initialize the accounts ArrayList
        accounts = new ArrayList<Account>();
      }

      // Method to add an Account to the accounts list
      public void addAccount(Account account) {
        // Add the given account to the accounts ArrayList
        accounts.add(account);
      }

      // Method to remove an Account from the accounts list
      public void removeAccount(Account account) {
        // Remove the given account from the accounts ArrayList
        accounts.remove(account);
      }

      // Method to deposit money into a specific Account
      public void depositMoney(Account account, double amount) {
        // Call the deposit method on the given account with the specified amou
        account.deposit(amount);
      }
```

```
    // Method to withdraw money from a specific Account
    public void withdrawMoney(Account account, double amount) {
      // Call the withdraw method on the given account with the specified amo
      account.withdraw(amount);
    }

    // Method to get the list of all accounts
    public ArrayList<Account> getAccounts() {
      // Return the accounts ArrayList
      return accounts;
    }
  }
```

The above class has a private accounts attribute, a constructor that initializes this attribute as an empty array list. It also has methods to add and remove accounts from the collection, and to deposit and withdraw money from an account.

Copy

```
// Main.java
// Import the ArrayList class from the Java Collections Framework
import java.util.ArrayList;

// Define the Main class
public class Main {

  // Main method, the entry point of the Java application
  public static void main(String[] args) {

    // Create a new Bank object
    Bank bank = new Bank();

    // Create three new Account objects with initial details
    Account account1 = new Account("Peter Irmgard", "C0011", 5000);
    Account account2 = new Account("Katja Ruedi", "C0121", 4500);
    Account account3 = new Account("Marcella Gebhard", "C0222", 20000);

    // Add the three accounts to the bank
    bank.addAccount(account1);
    bank.addAccount(account2);
    bank.addAccount(account3);

    // Retrieve the list of accounts from the bank
```

```
        ArrayList<Account> accounts = bank.getAccounts();

        // Loop through each account in the accounts list
        for (Account account: accounts) {
          // Print the account information for each account
          System.out.println(account.getAccountInfo());
        }

        // Print a message indicating the start of a deposit transaction
        System.out.println("\nAfter depositing 1000 into account1:");
        // Deposit 1000 into account1
        bank.depositMoney(account1, 1000);
        // Print the updated account information for account1
        System.out.println(account1.getAccountInfo());

        // Print a message indicating no transaction for account2
        System.out.println("No transaction in account2:");
        // Print the account information for account2
        System.out.println(account2.getAccountInfo());

        // Print a message indicating the start of a withdrawal transaction
        System.out.println("After withdrawing 5000 from account3:");
        // Withdraw 5000 from account3
        bank.withdrawMoney(account3, 5000);
        // Print the updated account information for account3
        System.out.println(account3.getAccountInfo());
    }
}
```

In the abc
the "Acco
print the a
1000 into
the 'withd
transactic

it

Sample C

```
Name:
Name:
Name:
```

```
After depositing 1000 into account1:
Name: Peter Irmgard, Account Number: C0011, Balance: 6000.0
No transaction in account2:
Name: Katja Ruedi, Account Number: C0121, Balance: 4500.0
After withdrawing 5000 from account3:
Name: Marcella Gebhard, Account Number: C0222, Balance: 15000.0
```

Flowchart: