CS1040

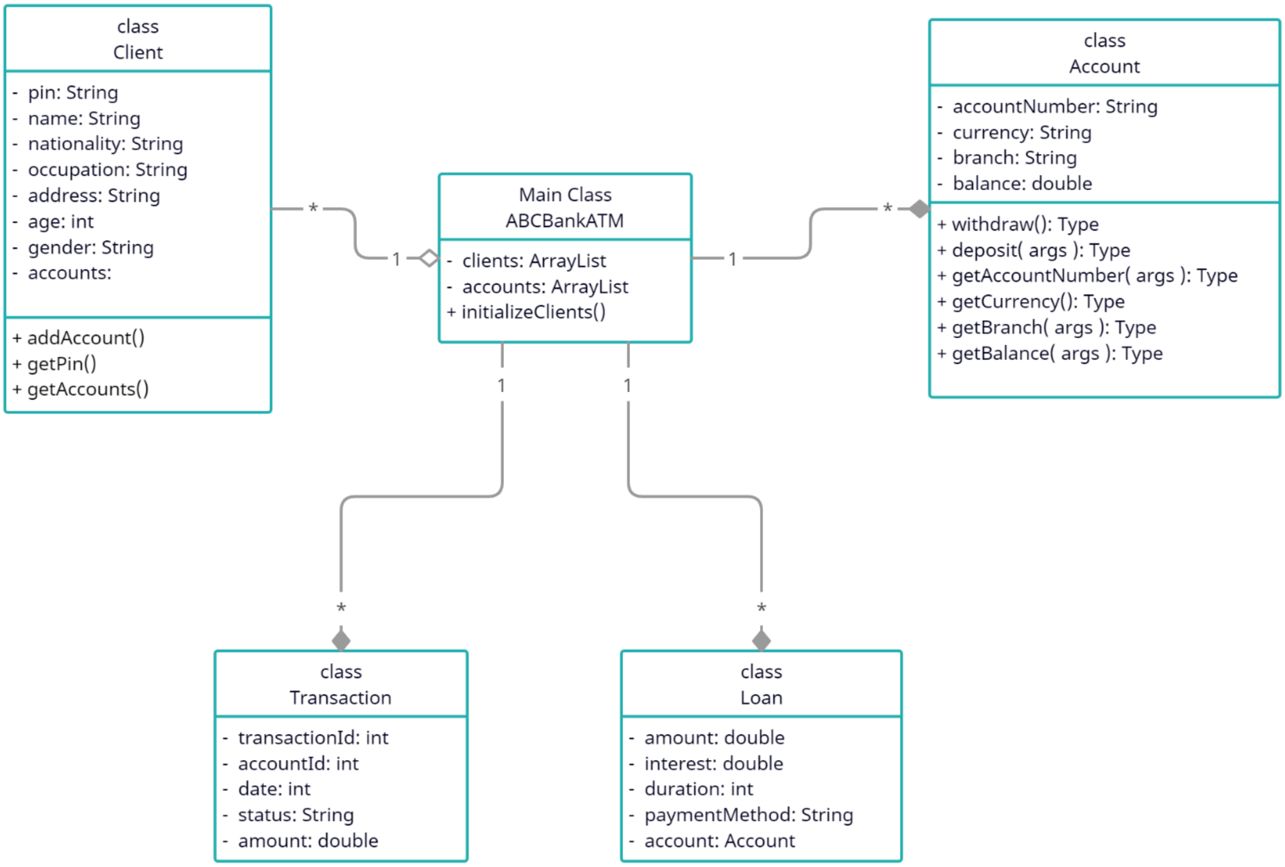
Program Construction

Lab Exercise 6

NAME : SANDEEPA H.N.A

INDEX NO : 210571L

Class Diagram:



Java source code:

```
/*Name: SANDEEPA H.N.A
Index No: 210571L */

/* In this code, we have three classes `ABCBankATM`, `Client`, and `Account`. The `ABCBankATM` class is the main class that implements the ATM simulation. It starts by asking for the client's PIN and then finding the corresponding client based on the entered PIN. Once the client is found, it shows the list of the client's accounts and asks the client to select one. Then it shows the main menu and performs the selected transaction.
The `Client` class represents a bank client with their personal information such as name, nationality, occupation, address, age, and gender. It also has a list of accounts associated with the client. The `Account` class represents a bank account with its account number, currency, branch, and balance. It has methods for withdrawing and depositing money. */

import java.util.ArrayList; // To store data of clients and accounts
import java.util.Scanner; // Getting user input to perform actions

public class ABCBankATM {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);
        System.out.println("Welcome!");
        System.out.print("Please enter your PIN: ");
        String pin = scanner.nextLine(); // Get the PIN

        // Assume we have a list of clients with their information
        ArrayList<Client> clients = initializeClients();

        // Find the user with given PIN number
        Client client = null;
        for (Client c : clients) {
            if (c.getPin().equals(pin)) {
                client = c;
                break;
            }
        }

        // If no client is found within the given PIN number, then prints "Invalid PIN. Exiting"
        if (client == null) {
            System.out.println("Invalid PIN. Exiting.");
            return;
        }

        // Show the accounts of the client and ask to select one
        System.out.println("Accounts:");
        ArrayList<Account> accounts = client.getAccounts();
        for (int i = 0; i < accounts.size(); i++) {
            Account account = accounts.get(i);
```

```java
    System.out.println((i+1) + ". " + account.getAccountNumber() + " (" + account.getCurrency() + ")");
}
System.out.print("Select an account: ");
int accountIndex = scanner.nextInt();
scanner.nextLine(); // consume the newline character

// Get the selected account
Account account = accounts.get(accountIndex-1);
if (account == null) {
    System.out.println("Invalid selection. Exiting.");
    return;
}

// Show the main menu
while (true) {
    System.out.println("\nMain Menu:\n1. View Balance\n2. Withdraw Money\n3. Deposit Money\n4. Exit");
    /*"Main Menu:" - Display the main menu
    "1. View Balance" - Display the balance
    "2. Withdraw Money" - Option that can withdraw money
    "3. Deposit Money" - Option that can deposit money
    "4. Exit" - Option that can exit*/

    System.out.print("Enter your choice: ");
    int choice = scanner.nextInt();
    scanner.nextLine(); // consume the newline character

    switch (choice) {
        case 1: // View Balance
            System.out.println("Balance: " + account.getAccountBalance());
            break;
        case 2: // Withdraw Money
            System.out.print("Enter amount to withdraw: ");
            double withdrawAmount = scanner.nextDouble();
            scanner.nextLine(); // consume the newline character
            if (account.withdraw(withdrawAmount)) {
                System.out.println("Withdrawal successful");
                System.out.println("Remaining balance: " + account.getAccountBalance());
            } else {
                System.out.println("Withdrawal failed. Insufficient balance.");
            }
            break;
        case 3: // Deposit Money
            System.out.print("Enter amount to deposit: ");
            double depositAmount = scanner.nextDouble();
            scanner.nextLine(); // consume the newline character
            account.deposit(depositAmount); // deposit the amount
            System.out.println("Deposit successful");
            System.out.println("Remaining balance: " + account.getAccountBalance());
            break;
```

```
                case 4: // Exit
                    System.out.println("Exiting...");
                    return;
                default: // The scenario if user enter a wrong choice.
                    System.out.println("Invalid choice. Please try again.");
                    break;
            }
        }
    }

    private static ArrayList<Client> initializeClients() {
        // Initialize a sample list of customers and accounts
        ArrayList<Client> clients = new ArrayList<>();

        // I created two objects of clients
        Client client1 = new Client("1234", "Nimal", 30, "123, abcd Road, Colombo 07", true, "Sri Lankan", "Teacher");
        Account account1a = new Account("1", "LKR", "Main", 8000);
        Account account1b = new Account("2", "LKR", "Main", 10000);
        client1.addingAccount (account1a);
        client1.addingAccount (account1b);
        clients.add(client1);

        Client client2 = new Client("5678", "Amali", 40, "Engineer", false, "Sri Lankan", "Engineer");
        Account account2a = new Account("3", "USD", "Main", 2000);
        Account account2b = new Account("4", "USD", "Main", 5000);
        client2.addingAccount (account2a);
        client2.addingAccount (account2b);
        clients.add(client2);

        return clients;
    }
}

// Class representing a client of ABC Bank
class Client {
    private String pin; // pin number
    private String clientName; // client name
    private int age; // age of the user
    private String address; // address of the user
    private boolean gender; // True if gender is selected male, false if gender is selected female
    private String nationality; // nationality of the user
    private String occupation; // Occupation of the user
    private ArrayList<Account> accounts;

    // Constructor
    public Client(String pin, String clientName, int age,  String address, Boolean gender, String nationality, String
occupation) {
        this.pin = pin;
        this.clientName = clientName;
        this.age = age;
        this.address = address;
        this.gender = gender;
        this.nationality = nationality;
        this.occupation = occupation;
        this.accounts = new ArrayList<>();
    }

    // Add Account
    public void addingAccount (Account account) {
        accounts.add(account);
    }

    // Getter of pin number
    public String getPin() {
        return pin;
    }

    // Getter of accounts
    public ArrayList<Account> getAccounts() {
        return accounts;
    }
}

// Class representing an account of ABC Bank
class Account {
    private String accountNumber; // The account number
    private String currency; // The type of currency to be used
    private String accounntBranch; // The branch to be used for the account
    private double accountBalance; // The balance of the account

    // Constructor of Account class
    public Account(String accountNumber, String currency, String accounntBranch, double balance) {
        this.accountNumber = accountNumber;
        this.currency = currency;
        this.accounntBranch = accounntBranch;
        this.accountBalance = balance;
    }

    // Withdraw Money
    public boolean withdraw(double amount) {
        if (accountBalance >= amount) {
            accountBalance -= amount;
            return true;
        } else {
            return false;
        }
    }
    // Deposit Money
    public void deposit(double amount) {
        accountBalance += amount;
    }

    // Getter of account number
    public String getAccountNumber() {
        return accountNumber;
    }

    // Getter of currency
    public String getCurrency() {
        return currency;
    }

    // Getter of accounntBranch
    public String getAccountBranch() {
        return accounntBranch;
    }

    // Getter of balance
    public double getAccountBalance() {
        return accountBalance;
    }
}

// Class representing transactions of ABC Bank
class Transactions {
    int transactionId; // Transaction ID number
    int TransactionAccountId; // Transaction Account ID number
    int transactionDate; // Transaction Date
    String transactionStatus; // Transaction Status
    double transactionAmount; // Transaction Amount

    public Transactions(int transactionId, int TransactionAccountId, int transactionDate, String transactionStatus,
double transactionAmount) {
        this.transactionId = transactionId;
        this.TransactionAccountId = TransactionAccountId;
        this.transactionDate = transactionDate;
        this.transactionStatus = transactionStatus;
        this.transactionAmount = transactionAmount;
    }
}

// Class representing a loan of ABC Bank
// Since the loan is not available in the ATM machine, It is not need to implement methods for loan in ATM machine
class Loans {
    double loanAmount; // Amount of the loan
    double loanInterest; // Amount of interest
    int loanDuration; // Duration of the loan
    String loanPaymentMethod; // Payment method
    Account account;

    // Constructor
    public Loans(double loanAmount, double loanInterest, int loanDuration, String loanPaymentMethod, Account
account) {
        this.loanAmount = loanAmount;
        this.loanInterest = loanInterest;
        this.loanDuration = loanDuration;
        this.loanPaymentMethod = loanPaymentMethod;
        this.account = account;
    }
}
```