OOP Mini Project: ATM Machine

Code:

```
import java.io.*;
import java.util.*;


class Account {
    private String accountNumber;
    private String name;
    private String phoneNumber;
    private double balance;
    private int pin;


    public Account(String accountNumber, String name, String phoneNumber,
double balance, int pin) {
        this.accountNumber = accountNumber;
        this.name = name;
        this.phoneNumber = phoneNumber;
        this.balance = balance;
        this.pin = pin;
    }


    public Account(String accountNumber, String name, String phoneNumber, int
pin) {
        this(accountNumber, name, phoneNumber, 0.0, pin);
    }
```

```
    public String toTableFormat() {
        return String.format("%-15s %-20s %-15s %-10.2f %-4d", accountNumber,
name, phoneNumber, balance, pin);
    }

    public static Account fromTableFormat(String line) {
        String[] parts = line.trim().split("\\s+");
        return new Account(parts[0], parts[1], parts[2],
Double.parseDouble(parts[3]), Integer.parseInt(parts[4]));
    }

    public String getAccountNumber() {
        return accountNumber;
    }

    public String getName() {
        return name;
    }

    public String getPhoneNumber() {
        return phoneNumber;
    }

    public double getBalance() {
        return balance;
    }
```

```
    public int getPIN() {
        return pin;
    }

    public void deposit(double amount) {
        balance += amount;
    }

    public void withdraw(double amount) {
        if (balance >= amount) {
            balance -= amount;
        } else {
            System.out.println("Insufficient funds.");
        }
    }

    public void setPIN(int pin) {
        this.pin = pin;
    }
}

class BankDatabase {
    private static final String FILENAME = "accounts.txt";
    private Map<String, Account> accounts;

    public BankDatabase() {
```

```
        accounts = new HashMap<>();
        fetchAccountsFromFile();
    }

    public Account getAccount(String accountNumber) {
        return accounts.get(accountNumber);
    }

    public String createAccount(String name, String phoneNumber, int pin) {
        String accountNumber = generateAccountNumber();
        Account account = new Account(accountNumber, name, phoneNumber,
pin);
        accounts.put(accountNumber, account);
        saveAccountsToFile();
        return accountNumber;
    }

    private String generateAccountNumber() {
        StringBuilder sb = new StringBuilder();
        sb.append("AC");
        for (int i = 0; i < 3; i++) {
            sb.append(new Random().nextInt(10));
        }
        return sb.toString();
    }

    public void saveAccountsToFile() {
```

```
        try (PrintWriter writer = new PrintWriter(new FileWriter(FILENAME))) {
            writer.println(String.format("%-15s %-20s %-15s %-10s %-4s", "Account Number", "Name", "Phone Number", "Balance", "PIN"));
            for (Account account : accounts.values()) {
                writer.println(account.toTableFormat());
            }
        } catch (IOException e) {
            System.err.println("Error saving accounts data: " + e.getMessage());
        }
    }

    public void fetchAccountsFromFile() {
        try (BufferedReader reader = new BufferedReader(new FileReader(FILENAME))) {
            String line = reader.readLine(); // Skip the header line
            while ((line = reader.readLine()) != null) {
                Account account = Account.fromTableFormat(line);
                accounts.put(account.getAccountNumber(), account);
            }
        } catch (FileNotFoundException e) {
            System.out.println("No existing accounts file found. Starting with an empty database.");
        } catch (IOException e) {
            System.err.println("Error reading accounts data: " + e.getMessage());
        }
    }
```

```
    public boolean isValidAccountNumber(String accountNumber) {
        return accounts.containsKey(accountNumber);
    }
}

public class ATM {
    private Scanner scanner;
    private BankDatabase bankDatabase;

    public ATM() {
        scanner = new Scanner(System.in);
        bankDatabase = new BankDatabase();
    }

    public void start() {
        System.out.println("Welcome to ATM");
        while (true) {
            displayMainMenu();
        }
    }

    private void displayMainMenu() {
        System.out.println("Welcome! Are you a new user or an existing user?");
        System.out.println("1. New User");
        System.out.println("2. Existing User");
```

```
    int userChoice = scanner.nextInt();
    scanner.nextLine(); // Consume newline

    switch (userChoice) {
        case 1:
            createNewAccount();
            break;
        case 2:
            System.out.println("Enter your account number:");
            String accountNumber = scanner.nextLine();
            if (bankDatabase.isValidAccountNumber(accountNumber)) {
                Account account = bankDatabase.getAccount(accountNumber);
                displayTransactionMenu(account);
            } else {
                System.out.println("Invalid account number. Please try again.");
            }
            break;
        default:
            System.out.println("Invalid choice");
    }
}

private void createNewAccount() {
    System.out.println("Enter your name:");
    String name = scanner.nextLine();
```

```
        System.out.println("Enter phone number:");
        String phoneNumber = scanner.nextLine();

        System.out.println("Enter your desired PIN:");
        int pin = scanner.nextInt();
        scanner.nextLine(); // Consume newline

        String accountNumber = bankDatabase.createAccount(name,
phoneNumber, pin);
        if (accountNumber != null) {
            System.out.println("Account created successfully. Your account number
is: " + accountNumber);
        } else {
            System.out.println("Failed to create account. Please try again later.");
        }
    }

    private void displayTransactionMenu(Account account) {
        while (true) {
            System.out.println("Main Menu:");
            System.out.println("1. Deposit");
            System.out.println("2. PIN Change");
            System.out.println("3. Balance Enquiry");
            System.out.println("4. Cash Withdrawal");
            System.out.println("5. Exit");

            int choice = scanner.nextInt();
```

```
            scanner.nextLine(); // Consume newline

            switch (choice) {
                case 1:
                    performDeposit(account);
                    break;
                case 2:
                    performPINChange(account);
                    break;
                case 3:
                    performBalanceEnquiry(account);
                    break;
                case 4:
                    performCashWithdrawal(account);
                    break;
                case 5:
                    System.out.println("Thank you for banking with us.");
                    bankDatabase.saveAccountsToFile(); // Save changes to file before
exiting
                    return;
                default:
                    System.out.println("Invalid choice");
            }
        }
    }

    private void performCashWithdrawal(Account account) {
```

```
        System.out.println("Enter your PIN:");
        int enteredPIN = scanner.nextInt();
        scanner.nextLine(); // Consume newline

        if (enteredPIN == account.getPIN()) {
            System.out.println("Enter withdrawal amount:");
            double amount = scanner.nextDouble();
            scanner.nextLine(); // Consume newline

            if (amount > 0 && amount <= account.getBalance()) {
                account.withdraw(amount);
                System.out.println("Withdrawal successful. Remaining balance: " +
account.getBalance());
            } else {
                System.out.println("Invalid withdrawal amount or insufficient
balance.");
            }
        } else {
            System.out.println("Incorrect PIN.");
        }
    }

    private void performDeposit(Account account) {
        System.out.println("Enter your PIN:");
        int enteredPIN = scanner.nextInt();
        scanner.nextLine(); // Consume newline
```

```
        if (enteredPIN == account.getPIN()) {
            System.out.println("Enter deposit amount:");
            double amount = scanner.nextDouble();
            scanner.nextLine(); // Consume newline

            if (amount > 0) {
                account.deposit(amount);
                System.out.println("Deposit successful. New balance: " +
account.getBalance());
            } else {
                System.out.println("Invalid deposit amount.");
            }
        } else {
            System.out.println("Incorrect PIN.");
        }
    }

    private void performPINChange(Account account) {
        System.out.println("Enter your current PIN:");
        int currentPIN = scanner.nextInt();
        scanner.nextLine(); // Consume newline

        if (currentPIN == account.getPIN()) {
            System.out.println("Enter your new PIN:");
            int newPIN = scanner.nextInt();
            scanner.nextLine(); // Consume newline
```

```
            account.setPIN(newPIN);
            System.out.println("PIN changed successfully.");
        } else {
            System.out.println("Incorrect PIN.");
        }
    }

    private void performBalanceEnquiry(Account account) {
        System.out.println("Your current balance is: " + account.getBalance());
    }

    public static void main(String[] args) {
        ATM atm = new ATM();
        atm.start();
    }
}
```

Output:

```
Welcome to ATM
Welcome! Are you a new user or an existing user?
1. New User
2. Existing User
1
Enter your name:
Rahul Das
Enter phone number:
```

1234567890

Enter your desired PIN:

1234

Account created successfully. Your account number is: AC174

Welcome! Are you a new user or an existing user?

1. New User

2. Existing User

2

Enter your account number:

AC174

Main Menu:

1. Deposit

2. PIN Change

3. Balance Enquiry

4. Cash Withdrawal

5. Exit

1

Enter your PIN:

1234

Enter deposit amount:

1000

Deposit successful. New balance: 1000.0

Main Menu:

1. Deposit

2. PIN Change

3. Balance Enquiry

4. Cash Withdrawal

5. Exit

2

Enter your current PIN:

1234

Enter your new PIN:

1212

PIN changed successfully.

Main Menu:

1. Deposit

2. PIN Change

3. Balance Enquiry

4. Cash Withdrawal

5. Exit

3

Your current balance is: 1000.0

Main Menu:

1. Deposit

2. PIN Change

3. Balance Enquiry

4. Cash Withdrawal

5. Exit

4

Enter your PIN:

1212

Enter withdrawal amount:

100

Withdrawal successful. Remaining balance: 900.0

Main Menu:

1. Deposit

2. PIN Change

3. Balance Enquiry

4. Cash Withdrawal

5. Exit

5

Thank you for banking with us.