import java.util.ArrayList;

import java.util.List;

import java.util.Scanner;

interface AtmOperationInterface {

void viewBalance();

void withdrawAmount(double withdrawAmount);

void depositAmount(double depositAmount);

void viewMiniStatement();

}

class ATM {

private double balance;

private double depositAmount;

private double withdrawAmount;

private List<String> miniStatement;

public ATM() {

this.balance = 0.0;

this.depositAmount = 0.0;

this.withdrawAmount = 0.0;

this.miniStatement = new ArrayList<>();

}

public double getBalance() {

return balance;

}

public void setBalance(double balance) {

this.balance = balance;

}

public double getDepositAmount() {

return depositAmount;

}

public void setDepositAmount(double depositAmount) {

this.depositAmount = depositAmount;

}

public double getWithdrawAmount() {

return withdrawAmount;

}

public void setWithdrawAmount(double withdrawAmount) {

this.withdrawAmount = withdrawAmount;

}

public List<String> getMiniStatement() {

return miniStatement;

}

public void addToMiniStatement(String transaction) {

miniStatement.add(transaction);

}

}

class AtmOperation implements AtmOperationInterface {

private final ATM atm;

public AtmOperation(ATM atm) {

this.atm = atm;

}

public void viewBalance() {

System.out.println("Available Balance is: " + atm.getBalance());

}

public void withdrawAmount(double withdrawAmount) {

if (withdrawAmount <= atm.getBalance()) {

atm.setBalance(atm.getBalance() - withdrawAmount);

atm.addToMiniStatement(withdrawAmount + " Amount Withdrawn");

System.out.println("Collect the Cash " + withdrawAmount);

System.out.println("Available Balance is : " + atm.getBalance());

} else {

System.out.println("Insufficient Balance !!");

}

}

public void depositAmount(double depositAmount) {

atm.setBalance(atm.getBalance() + depositAmount);

atm.addToMiniStatement(depositAmount + " Amount Deposited");

System.out.println(depositAmount + " Deposited Successfully !!");

System.out.println("Available Balance is: " + atm.getBalance());

}

public void viewMiniStatement() {

System.out.println("Mini Statement:");

for (String transaction : atm.getMiniStatement()) {

System.out.println(transaction);

}

}

}

public class MainClass {

public static void main(String[] args) {

ATM atm = new ATM();

AtmOperation atmOperation = new AtmOperation(atm);

Scanner scanner = new Scanner(System.in);

System.out.println("Welcome to ATM Machine !!!");

System.out.print("Enter Atm Number: ");

String atmNumber = scanner.nextLine();

System.out.print("Enter Pin: ");

String pin = scanner.nextLine();

System.out.println("1. View Available Balance");

System.out.println("2. Withdraw Amount");

System.out.println("3. Deposit Amount");

System.out.println("4. View Mini Statement");

System.out.println("5. Exit");

boolean exit = false;

while (!exit) {

System.out.print("\nEnter Choice: ");

int choice = scanner.nextInt();

switch (choice) {

case 1:

atmOperation.viewBalance();

break;

case 2:

System.out.print("Enter amount to withdraw: ");

double withdrawAmount = scanner.nextDouble();

atmOperation.withdrawAmount(withdrawAmount);

break;

case 3:

System.out.print("Enter Amount to Deposit: ");

double depositAmount = scanner.nextDouble();

atmOperation.depositAmount(depositAmount);

break;

case 4:

atmOperation.viewMiniStatement();

break;

case 5:

exit = true;

break;

default:

System.out.println("Invalid choice. Please try again.");

}

}

System.out.println("Collect your ATM Card");

System.out.println("Thank you for using ATM Machine!!");

}

}