import java.util.ArrayList;

import java.util.Scanner;

// Transaction class representing a bank transaction

class Transaction {

private int transactionId;

private String transactionType;

protected double amount;

// Constructor

public Transaction(int transactionId, String transactionType, double amount) {

this.transactionId = transactionId;

this.transactionType = transactionType;

this.amount = amount;

}

// Getter and Setter methods

public int getTransactionId() {

return transactionId;

}

public void setTransactionId(int transactionId) {

this.transactionId = transactionId;

}

public String getTransactionType() {

return transactionType;

}

public void setTransactionType(String transactionType) {

this.transactionType = transactionType;

}

public double getAmount() {

return amount;

}

public void setAmount(double amount) {

this.amount = amount;

}

// Method to perform transaction

public void performTransaction() {

System.out.println("Transaction ID " + transactionId + ": " + transactionType + " of $" + amount);

}

}

// DepositTransaction class inheriting from Transaction

class DepositTransaction extends Transaction {

private String depositorName;

// Constructor

public DepositTransaction(int transactionId, double amount, String depositorName) {

super(transactionId, "Deposit", amount);

this.depositorName = depositorName;

}

// Getter and Setter for additional attribute

public String getDepositorName() {

return depositorName;

}

public void setDepositorName(String depositorName) {

this.depositorName = depositorName;

}

// Overriding performTransaction method

@Override

public void performTransaction() {

System.out.println("Deposit by " + depositorName + ": $" + amount);

}

}

// WithdrawalTransaction class inheriting from Transaction

class WithdrawalTransaction extends Transaction {

// Constructor

public WithdrawalTransaction(int transactionId, double amount) {

super(transactionId, "Withdrawal", amount);

}

// Method overloading within class

public void performTransaction(String accountType) {

System.out.println("Withdrawal from " + accountType + " account: $" + amount);

}

}

// BankApplication class for user interaction

public class BankApplication {

private static ArrayList<Transaction> transactions = new ArrayList<>();

private static int transactionCounter = 1;

private static Scanner scanner = new Scanner(System.in);

public static void main(String[] args) {

int choice;

do {

System.out.println("=== Bank Transaction Menu ===");

System.out.println("1. Deposit");

System.out.println("2. Withdrawal");

System.out.println("3. Display Transactions");

System.out.println("4. Exit");

System.out.print("Enter your choice: ");

choice = scanner.nextInt();

scanner.nextLine(); // Consume newline character

switch (choice) {

case 1:

performDeposit();

break;

case 2:

performWithdrawal();

break;

case 3:

displayTransactions();

break;

case 4:

System.out.println("Exiting...");

break;

default:

System.out.println("Invalid choice. Please enter a number from 1 to 4.");

}

} while (choice != 4);

scanner.close();

}

// Method to create and perform a deposit transaction

private static void performDeposit() {

System.out.print("Enter depositor's name: ");

String depositorName = scanner.nextLine();

System.out.print("Enter deposit amount: ");

double amount = scanner.nextDouble();

transactions.add(new DepositTransaction(transactionCounter++, amount, depositorName));

System.out.println("Deposit successful.");

}

// Method to create and perform a withdrawal transaction

private static void performWithdrawal() {

System.out.print("Enter withdrawal amount: ");

double amount = scanner.nextDouble();

transactions.add(new WithdrawalTransaction(transactionCounter++, amount));

System.out.println("Withdrawal successful.");

}

// Method to display all transactions

private static void displayTransactions() {

if (transactions.isEmpty()) {

System.out.println("No transactions yet.");

} else {

System.out.println("=== List of Transactions ===");

for (Transaction transaction : transactions) {

transaction.performTransaction();

}

}