Practical No.: 13

Name: Bhairavi Narendra Rewatkar

Roll No.: DMET1221006

Subject: Blockchain Technology Laboratory

Title: Multi-Signature (multi-sig) Transactions.

Aim: Write a program to implement multi-signature (multi-sig) transactions in Java.

Source Code:

```
import java.util.ArrayList;
import java.util.List;
class Account {
    private String accountId;
    private int balance;
    public Account(String accountId, int balance) {
        this.accountId = accountId;
        this.balance = balance;
    }
    public String getAccountId() {
        return accountId;
    }
    public int getBalance() {
        return balance;
    }
    public boolean debit(int amount) {
        if (balance >= amount) {
            balance -= amount;
            return true;
        }
        return false;
    }
    public void credit(int amount) {
        balance += amount;
    }
```

```
}
class MultiSigTransaction {
    private Account sender;
    private Account receiver;
    private int amount;
    private List<String> signers;
    private int requiredSignatures;
    private List<String> approvals;
    public MultiSigTransaction(Account sender, Account receiver, int amount, List<String> signers, int requiredSignatures) {
        this.sender = sender;
        this.receiver = receiver;
        this.amount = amount;
        this.signers = signers;
        this.requiredSignatures = requiredSignatures;
        this.approvals = new ArrayList<>();
    }
    // Method to approve the transaction by a signer
    public void approveTransaction(String signerId) {
        if (signers.contains(signerId) && !approvals.contains(signerId)) {
            approvals.add(signerId);
            System.out.println(signerId + " approved the transaction.");
        } else {
            System.out.println(signerId + " cannot approve the transaction.");
        }
    }
    // Method to execute the transaction if enough approvals are present
    public String executeTransaction() {
        if (approvals.size() >= requiredSignatures) {
            if (sender.debit(amount)) {
                receiver.credit(amount);
                return "Transaction executed successfully: " + amount + " transferred from " + sender.getAccountId() + " to " + receiver.getAccountId();
```

```
            } else {
                return "Transaction failed: Insufficient balance in sender's account.";
            }
        } else {
            return "Transaction failed: Not enough approvals.";
        }
    }
}

public class MultiSigDemo {
    public static void main(String[] args) {
        // Creating two accounts: sender and receiver
        Account sender = new Account("Sender", 1000);
        Account receiver = new Account("Receiver", 200);

        // List of signers (multi-sig participants)
        List<String> signers = new ArrayList<>();
        signers.add("Alice");
        signers.add("Bob");
        signers.add("Charlie");

        // Create a multi-signature transaction requiring 2 out of 3 signatures
        MultiSigTransaction multiSigTransaction = new MultiSigTransaction(sender, receiver, 300,
signers, 2);

        // Display initial balances
        System.out.println("Initial Balances:");
        System.out.println("Sender: " + sender.getBalance() + " tokens");
        System.out.println("Receiver: " + receiver.getBalance() + " tokens");

        // Approve the transaction by Alice and Bob
        multiSigTransaction.approveTransaction("Alice");
        multiSigTransaction.approveTransaction("Bob");
```

```
        // Attempt to execute the transaction
        String result = multiSigTransaction.executeTransaction();
        System.out.println(result);

        // Display updated balances after the transaction
        System.out.println("Updated Balances:");
        System.out.println("Sender: " + sender.getBalance() + " tokens");
        System.out.println("Receiver: " + receiver.getBalance() + " tokens");
    }
}
```

Output: