```java
package com;

import java.util.*;

public class Main {

    public static void main(String[] args) {

        new Main();

    }//main loop

    Main() {
        Scanner keyVal = new Scanner(System.in);
        BankAccount commBA = null;
        ArrayList<double[]> userTL = new ArrayList<>();


        double[] transaction_1 = {50, 10, -20, 10, -20, 20, 10, 50, -10, 10, -10, 50};
        userTL.add(transaction_1);

        double[] transaction_2 = {20, 20, -20, 50, -20, 10, 50, 50, -20, 10, 10};
        userTL.add(transaction_2);

        double[] transaction_3 = {50, 10, 10, -10, -10, 50, 20, -10, -20};
        userTL.add(transaction_3);

        double[] transaction_4 = {50, 10, -20, 20, 10, -20};
        userTL.add(transaction_4);

        ArrayList<User> userAL = new ArrayList<>();

        while (true) {
            System.out.println("\n1. Create Bank Account\n2. Create User\n3. Run SyncSim\n4. Run UnSyncSim\n5. Reset bank account to 1980\n6. Exit");
            switch (keyVal.nextInt()) {
                case 1:
                    //create new bank account
                    commBA = new BankAccount(9876543210L, 1980);
                    System.out.println("Created new bank account, Acc No: " + commBA.getAccountNo() + " with initial balance of: " + commBA.getAccountBalance());
                    break;

                case 2:

                    try {
                        userAL.add(new User("Saul", "Goodman", commBA, userTL.get(0)));
                        userAL.add(new User("Walter", "White", commBA, userTL.get(1)));
                        userAL.add(new User("Jessie", "Pinkman", commBA, userTL.get(2)));
                        userAL.add(new User("Hank", "Schrader", commBA, userTL.get(3)));

                        for (int i = 0; i < userAL.size(); i++) {
                            System.out.println("User: " + userAL.get(i).getUName() +
                                    " Bank Account: " + userAL.get(i).getBA().getAccountNo() +
                                    " Available Money: " + userAL.get(i).getBA().getAccountBalance());
                        }

                    } catch (NullPointerException e) {
                        System.out.println("No Bank account Created!!");
                    }
                    break;

                case 3:
                    try {
                        for (int i = 0; i < userTL.size(); i++) {
                            SyncSim syncUser = new SyncSim(userAL.get(i));
                            syncUser.start();
                            //uncommenting this line may jeopardize the result.
                            //syncUser.join();
                        }
                    } catch (Exception e) {
                        System.out.println("No bank account or user accounts created!");
                        e.printStackTrace();
                    }
                    break;

                case 4:
                    try {
                        for (int i = 0; i < userTL.size(); i++) {
                            UnSyncSim unsyncUser = new UnSyncSim(userAL.get(i));
                            unsyncUser.start();
                            //uncomment this line prints menu after threads finish
```

```
                                //but doesn't show the problem of wrong acc value.
                                //unsyncUser.join();
                            }
                        } catch (Exception e) {
                            System.out.println("No bank account or user accounts created!");
                            e.printStackTrace();
                        }
                        break;

                    case 5:
                        try {
                            commBA.setAccountBalance(1980);
                        }catch (NullPointerException e){
                            System.out.println("No Bank account Created!!");
                        }

                        break;
                    case 6:
                        System.exit(0);
                        break;
                    default:
                        System.out.println("Wrong input value!");
                        break;
                }
            }
        }
}
```

```java
package com;

public class User
{
    private String name;
    private String surname;
    private BankAccount bankAccount;
    private double[] transactionList;

    public User(String name, String surname, BankAccount bA, double[] tL)
    {
        this.name = name;
        this.surname = surname;
        this.bankAccount = bA;
        this.transactionList = tL;
    }

    public String getUName(){
        return this.name;
    }

    public BankAccount getBA(){
        return this.bankAccount;
    }

    //get user TranList
    public double[] getTransactionList(){
        return this.transactionList;
    }

    //set user TranList
    public void setTransactionList(double[] list){
        this.transactionList = list;
    }

    public double getTransaction(int index){
        return this.transactionList[index];
    }

    public int lengthTransactionList(){
        return this.transactionList.length;
    }
}
```

```java
package com;

public class SyncSim extends Thread{

    private User user;
    private double[] transactions;
    private double value;
    private BankAccount ba;

    public SyncSim(User anyUser)
    {
        super("name");

        this.user = anyUser;
        this.transactions = anyUser.getTransactionList();
        this.ba = user.getBA();
    }

    public void run(){
        try{
            for(int i = 0; i < transactions.length; i++) {
                value = transactions[i];
                if (value > 0) {
                    ba.sync_deposit(value, user);
                }
                if (value < 0) {
                    ba.sync_withdraw(value, user);
                }
                if (value == 0) {
                    System.out.println("No Transaction!\n");
                }
                sleep(5);
            }
        }
        catch(Exception e) {
            System.out.println("No account has been created!\n");
        }
    }

}
```

```java
package com;

public class UnSyncSim extends Thread{

    private User user;
    private double[] transactions;
    private double value;
    private BankAccount ba;

    public UnSyncSim(User anyUser)
    {
        super("name");

        this.user = anyUser;
        this.transactions = anyUser.getTransactionList();
        this.ba = user.getBA();
    }

    public void run(){
        try{
            for(int i = 0; i < transactions.length; i++) {
                value = transactions[i];
                if (value > 0) {
                    ba.deposit(value, user);
                }
                if (value < 0) {
                    ba.withdraw(value, user);
                }
                if (value == 0) {
                    System.out.println("No Transaction!\n");
                }
                sleep(5);
            }
        }
        catch(Exception e) {
            System.out.println("No account has been created!\n");
        }
    }

}

```

```java
package com;

public class BankAccount
{
    private long accountNo;
    private double accountBalance;


    public BankAccount(long accountNo, double accountBalance)
    {
        this.accountNo = accountNo;
        this.accountBalance = accountBalance;
    }

    public long getAccountNo()
    {
        return this.accountNo;
    }

    public double getAccountBalance()
    {
        return this.accountBalance;
    }

    public void setAccountBalance(double newBalance)
    {
        this.accountBalance = newBalance;
        System.out.println("Account balance value: " + this.accountBalance);
    }

    public synchronized void sync_deposit(double value, User u)
    {
        accountBalance += value;
        notifyAll();
        System.out.println("Transaction value:  " + value + "\tBalance after transaction: " + accountBalance + "\tClient: " + u.getUName());
    }

    public synchronized void sync_withdraw(double value, User u)
    {
        while(Math.abs(value) > accountBalance){
            try{
                wait();
            }catch (InterruptedException e){
                System.out.println("Thread interrupted");
            }

        }
        accountBalance += value;
        System.out.println("Transaction value: " + value + "\tBalance after transaction: " + accountBalance + "\tClient: " + u.getUName());

    }

    public void deposit(double value, User u)
    {
        accountBalance += value;
        System.out.println("Transaction value:  " + value + "\tBalance after transaction: " + accountBalance + "\tClient: " + u.getUName());
    }

    public void withdraw(double value, User u)
    {
        accountBalance += value;
        System.out.println("Transaction value: " + value + "\tBalance after transaction: " + accountBalance + "\tClient: " + u.getUName());

    }

}
```