class BankAccount {

    private double balance;

    // Synchronizing the methods to ensure thread safety

    public synchronized void deposit(double amount) {

        balance += amount;

        System.out.println("Deposited " + amount + ". New balance: " + balance);

    }

    public synchronized void withdraw(double amount) {

        if (amount > balance) {

            System.out.println("Insufficient funds for withdrawal of " + amount + ". Current balance: " + balance);

        } else {

            balance -= amount;

            System.out.println("Withdrew " + amount + ". New balance: " + balance);

        }

    }

    public double getBalance() {

        return balance;

    }

}

class UserTransaction implements Runnable {

    private BankAccount account;

    private double depositAmount;

    private double withdrawAmount;

    public UserTransaction(BankAccount account, double depositAmount, double withdrawAmount) {

        this.account = account;

        this.depositAmount = depositAmount;

        this.withdrawAmount = withdrawAmount;

    }

    @Override

    public void run() {

        account.deposit(depositAmount);

        account.withdraw(withdrawAmount);

    }

}

public class Task2 {

    public static void main(String[] args) throws InterruptedException {

        // Create a BankAccount object with an initial balance of 1000

        BankAccount account = new BankAccount();

        // Create multiple threads to simulate concurrent access

        Thread thread1 = new Thread(new UserTransaction(account, 200, 100));

        Thread thread2 = new Thread(new UserTransaction(account, 150, 200));

        Thread thread3 = new Thread(new UserTransaction(account, 300, 400));

        // Start the threads

        thread1.start();

        thread2.start();

        thread3.start();

        // Wait for all threads to complete

        thread1.join();

        thread2.join();

        thread3.join();

        // Final balance

        System.out.println("Final balance: " + account.getBalance());

    }

}