

# Report: Deadlock Simulation & Solution

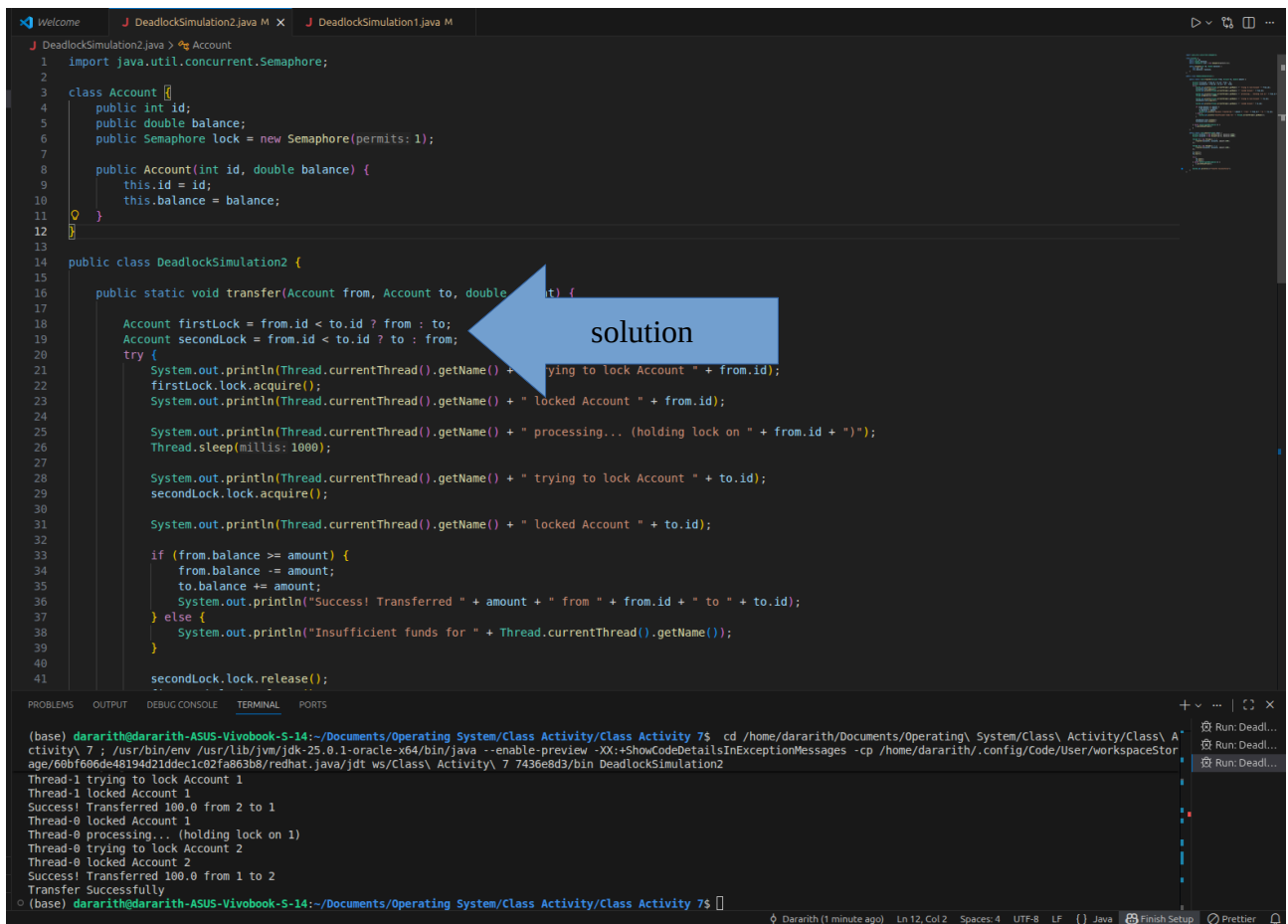
Github link: <https://github.com/Chumdararith-LOU/deadlock-simulation>

## Part 1: Deadlock Simulation

```
1 import java.util.concurrent.Semaphore;
2
3 class Account {
4     public int id;
5     public double balance;
6     public Semaphore lock = new Semaphore(permits: 1);
7
8     public Account(int id, double balance) {
9         this.id = id;
10        this.balance = balance;
11    }
12 }
13
14 public class DeadlockSimulation1 {
15
16     public static void transfer(Account from, Account to, double amount) {
17         try {
18             System.out.println(Thread.currentThread().getName() + " trying to lock Account " + from.id);
19             from.lock.acquire();
20             System.out.println(Thread.currentThread().getName() + " locked Account " + from.id);
21
22             System.out.println(Thread.currentThread().getName() + " processing... (holding lock on " + from.id + ")");
23             Thread.sleep(millis: 1000);
24
25             System.out.println(Thread.currentThread().getName() + " trying to lock Account " + to.id);
26             to.lock.acquire();
27
28             System.out.println(Thread.currentThread().getName() + " locked Account " + to.id);
29
30             if (from.balance >= amount) {
31                 from.balance -= amount;
32                 to.balance += amount;
33                 System.out.println("Success! Transferred " + amount + " from " + from.id + " to " + to.id);
34             } else {
35                 System.out.println("Insufficient funds for " + Thread.currentThread().getName());
36             }
37
38             to.lock.release();
39             from.lock.release();
40
41         } catch (InterruptedException e) {
42         }
```

Activity\ 7 ; /usr/bin/env /usr/lib/jvm/jdk-25.0.1-oracle-x64/bin/java --enable-preview -XX:+ShowCodeDetailsInExceptionMessages -cp /home/dararith/.config/Code/User/workspaceSt  
orage/60bf606de48194d21ddc1c82fa863b8/redhat.java/jdt\_ws/Class\ Activity\ 7\_7436e8d3/bin DeadlockSimulation1  
Dararith@dararith-ASUS-Vivobook-S-14:~/Documents/Operating System/Class Activity/Class Activity 7\$ cd /home/dararith/Documents/Operating System/Class Activity/Class A  
ctivity\ 7 ; /usr/bin/env /usr/lib/jvm/jdk-25.0.1-oracle-x64/bin/java --enable-preview -XX:+ShowCodeDetailsInExceptionMessages -cp /home/dararith/.config/Code/User/workspaceSto  
rage/60bf606de48194d21ddc1c82fa863b8/redhat.java/jdt\_ws/Class\ Activity\ 7\_7436e8d3/bin DeadlockSimulation1  
Thread-1 trying to lock Account 2  
Thread-0 trying to lock Account 1  
Thread-0 locked Account 1  
Thread-1 locked Account 2  
Thread-0 processing... (holding lock on 1)  
Thread-1 processing... (holding lock on 2)  
Thread-1 trying to lock Account 1  
Thread-0 trying to lock Account 2

## Part 2: Deadlock Solution (Lock Ordering)



```
1 import java.util.concurrent.Semaphore;
2
3 class Account {
4     public int id;
5     public double balance;
6     public Semaphore lock = new Semaphore(permits: 1);
7
8     public Account(int id, double balance) {
9         this.id = id;
10        this.balance = balance;
11    }
12 }
13
14 public class DeadlockSimulation2 {
15
16     public static void transfer(Account from, Account to, double amount) {
17
18         Account firstLock = from.id < to.id ? from : to;
19         Account secondLock = from.id < to.id ? to : from;
20         try {
21             System.out.println(Thread.currentThread().getName() + " trying to lock Account " + from.id);
22             firstLock.lock.acquire();
23             System.out.println(Thread.currentThread().getName() + " locked Account " + from.id);
24
25             System.out.println(Thread.currentThread().getName() + " processing... (holding lock on " + from.id + ")");
26             Thread.sleep(millis: 1000);
27
28             System.out.println(Thread.currentThread().getName() + " trying to lock Account " + to.id);
29             secondLock.lock.acquire();
30             System.out.println(Thread.currentThread().getName() + " locked Account " + to.id);
31
32             if (from.balance >= amount) {
33                 from.balance -= amount;
34                 to.balance += amount;
35                 System.out.println("Success! Transferred " + amount + " from " + from.id + " to " + to.id);
36             } else {
37                 System.out.println("Insufficient funds for " + Thread.currentThread().getName());
38             }
39
40             secondLock.lock.release();
41         } catch (InterruptedException e) {
42             e.printStackTrace();
43         }
44     }
45 }
```

terminal

```
(base) dararith@dararith-ASUS-Vivobook-S-14:~/Documents/Operating System/Class Activity/Class Activity 7$ cd /home/dararith/Documents/Operating System/Class Activity/Class Activity 7 ; /usr/bin/env /usr/lib/jvm/jdk-25.0.1-oracle-x64/bin/java --enable-preview -XX:+ShowCodeDetailsInExceptionMessages -cp /home/dararith/.config/Code/User/workspaceStorage/69bf696de48194d21ddec1c02fa863b8/redhat.java/jdt ws/class Activity 7 7436e8d3/bin DeadlockSimulation2
Thread-1 trying to lock Account 1
Thread-1 locked Account 1
Success! Transferred 100.0 from 2 to 1
Thread-0 locked Account 1
Thread-0 processing... (holding lock on 1)
Thread-0 trying to lock Account 2
Thread-0 locked Account 2
Success! Transferred 100.0 from 1 to 2
Transfer Successfully
(base) dararith@dararith-ASUS-Vivobook-S-14:~/Documents/Operating System/Class Activity/Class Activity 7$
```