

Report: Deadlock Simulation & Solution

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(milliseconds: 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
43         }
44     }
45 }
46
47
48 public class DeadlockSimulation2 {
49
50     public static void transfer(Account from, Account to, double amount) {
51         try {
52             System.out.println(Thread.currentThread().getName() + " trying to lock Account " + from.id);
53             from.lock.acquire();
54             System.out.println(Thread.currentThread().getName() + " locked Account " + from.id);
55
56             System.out.println(Thread.currentThread().getName() + " processing... (holding lock on " + from.id + ")");
57             Thread.sleep(milliseconds: 1000);
58
59             System.out.println(Thread.currentThread().getName() + " trying to lock Account " + to.id);
60             to.lock.acquire();
61
62             System.out.println(Thread.currentThread().getName() + " locked Account " + to.id);
63
64             if (from.balance >= amount) {
65                 from.balance -= amount;
66                 to.balance += amount;
67                 System.out.println("Success! Transferred " + amount + " from " + from.id + " to " + to.id);
68             } else {
69                 System.out.println("Insufficient funds for " + Thread.currentThread().getName());
70             }
71
72             to.lock.release();
73             from.lock.release();
74
75         } catch (InterruptedException e) {
76
77         }
78     }
79 }
80
81
82 public class DeadlockSimulation3 {
83
84     public static void transfer(Account from, Account to, double amount) {
85         try {
86             System.out.println(Thread.currentThread().getName() + " trying to lock Account " + from.id);
87             from.lock.acquire();
88             System.out.println(Thread.currentThread().getName() + " locked Account " + from.id);
89
90             System.out.println(Thread.currentThread().getName() + " processing... (holding lock on " + from.id + ")");
91             Thread.sleep(milliseconds: 1000);
92
93             System.out.println(Thread.currentThread().getName() + " trying to lock Account " + to.id);
94             to.lock.acquire();
95
96             System.out.println(Thread.currentThread().getName() + " locked Account " + to.id);
97
98             if (from.balance >= amount) {
99                 from.balance -= amount;
100                to.balance += amount;
101                System.out.println("Success! Transferred " + amount + " from " + from.id + " to " + to.id);
102            } else {
103                System.out.println("Insufficient funds for " + Thread.currentThread().getName());
104            }
105
106            to.lock.release();
107            from.lock.release();
108
109        } catch (InterruptedException e) {
110
111        }
112    }
113 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

^ C(base) dararith@dararith-ASUS-Vivobook-S14:-/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/workspace\$ oracle/6bbf606de48194d21dedcc1c82fa863b8/redhat.java/jdt_ws/Class\ Activity\ 7_7436e8d3/bin DeadlockSimulation1

Starting Deadlock Simulation

Thread-1 trying to lock Account 2

Thread-1 locked Account 2

Thread-0 trying to lock Account 1

Thread-1 processing... (holding lock on 2)

Thread-0 locked Account 1

Thread-0 processing... (holding lock on 1)

Thread-1 trying to lock Account 1

Thread-0 trying to lock Account 2

Ln 38, Col 31 Spaces: 4 UTF-8 LF { } Java Finish Setup Prettier

Part 2: Deadlock Solution (Lock Ordering)

The screenshot shows a Java IDE interface with two tabs: "DeadlockSimulation2.java" and "DeadlockSimulation1.java". The "DeadlockSimulation2.java" tab contains the following code:

```
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
21         try {
22             System.out.println(Thread.currentThread().getName() + " trying to lock Account " + from.id);
23             firstLock.lock.acquire();
24             System.out.println(Thread.currentThread().getName() + " locked Account " + from.id);
25
26             System.out.println(Thread.currentThread().getName() + " processing... (holding lock on " + from.id + ")");
27             Thread.sleep(1000);
28
29             System.out.println(Thread.currentThread().getName() + " trying to lock Account " + to.id);
30             secondLock.lock.acquire();
31
32             System.out.println(Thread.currentThread().getName() + " locked Account " + to.id);
33
34             if (from.balance >= amount) {
35                 from.balance -= amount;
36                 to.balance += amount;
37                 System.out.println("Success! Transferred " + amount + " from " + from.id + " to " + to.id);
38             } else {
39                 System.out.println("Insufficient funds for " + Thread.currentThread().getName());
40             }
41
42             secondLock.lock.release();
43             firstLock.lock.release();
44
45         } catch (InterruptedException e) {
46             e.printStackTrace();
47         }
48     }
49 }
```

A blue arrow points from the word "solution" to the code. Below the code, the terminal window shows the execution of the code and its output:

```
(base) dararith@dararith-ASUS-Vivobook-S-14:~/Documents/Operating System/Class Activity/Class Activity 7$ git init
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint: git branch -m <name>
Initialized empty Git repository in /home/darith/Downloads/Operating System/Class Activity/Class Activity 7/.git/
(base) dararith@dararith-ASUS-Vivobook-S-14:~/Documents/Operating System/Class Activity/Class Activity 7$ git ^C
(base) dararith@dararith-ASUS-Vivobook-S-14:~/Documents/Operating System/Class Activity/Class Activity 7$ cd /home/darith/Downloads/Operating System/Class Activity/Class Activity 7
(base) dararith@dararith-ASUS-Vivobook-S-14:~/Documents/Operating System/Class Activity/Class Activity 7$ ./DeadlockSimulation2
Thread-0 trying to lock Account 2
Thread-0 trying to lock Account 1
Thread-1 locked Account 2
Thread-1 processing... (holding lock on 2)
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
Main finished.
```

The terminal also shows the current working directory as "/home/darith/Downloads/Operating System/Class Activity/Class Activity 7\$".