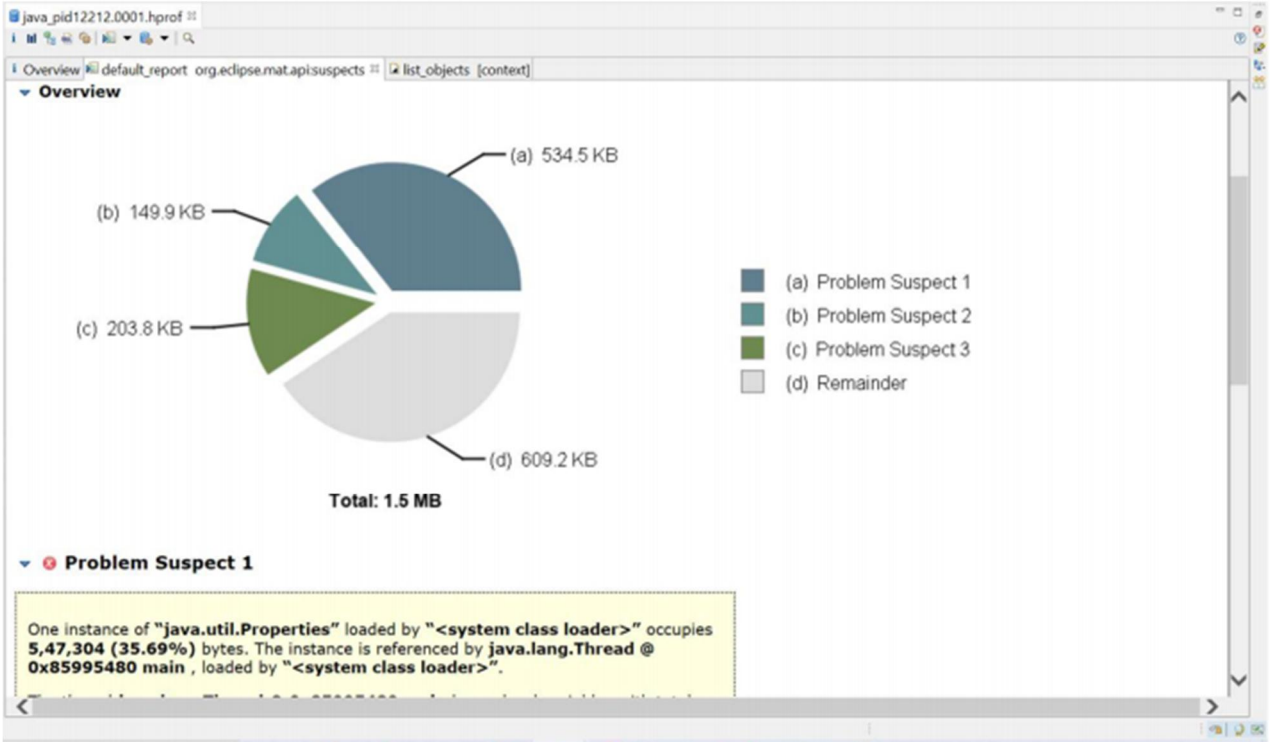
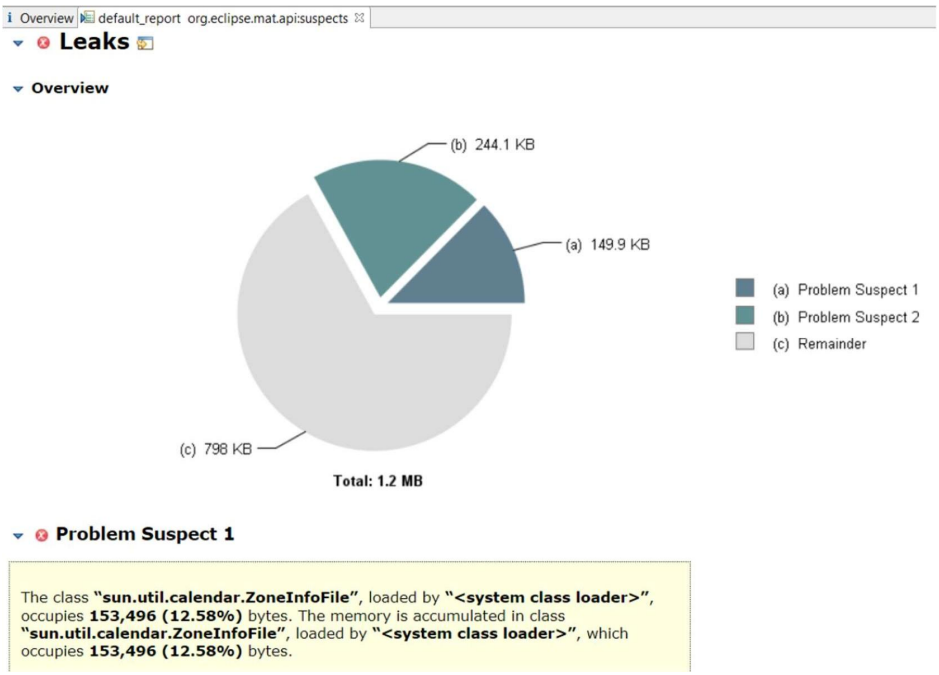
HashMap WithoutEqualAndHash pdf file

Problem: Inefficient memory consumption due to missing implementation of equals and hashcode function in Key.java class, hence using Object class for generating hashcode values for same entries.

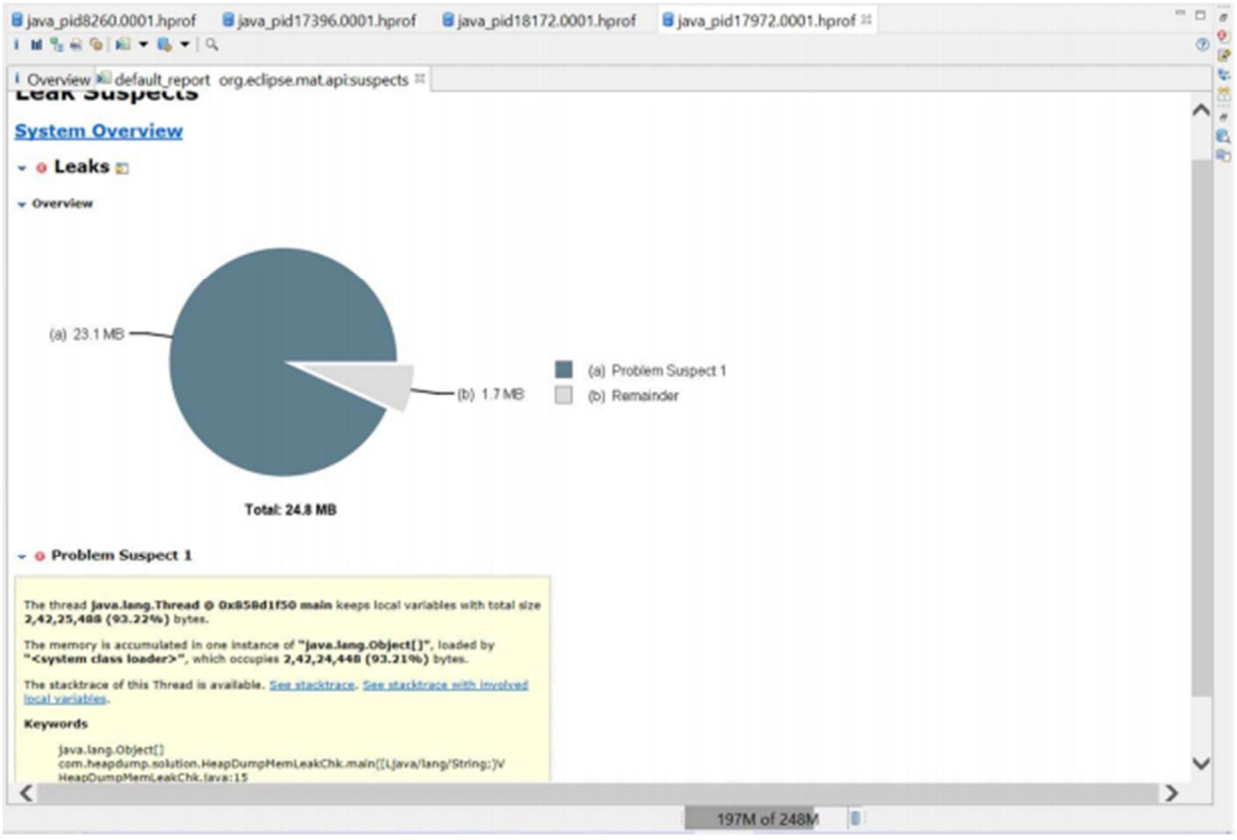


Solution: HashMap WithoutEqualAndHashcode Optimised Version



HeapDumpMemLeakChk pdf file

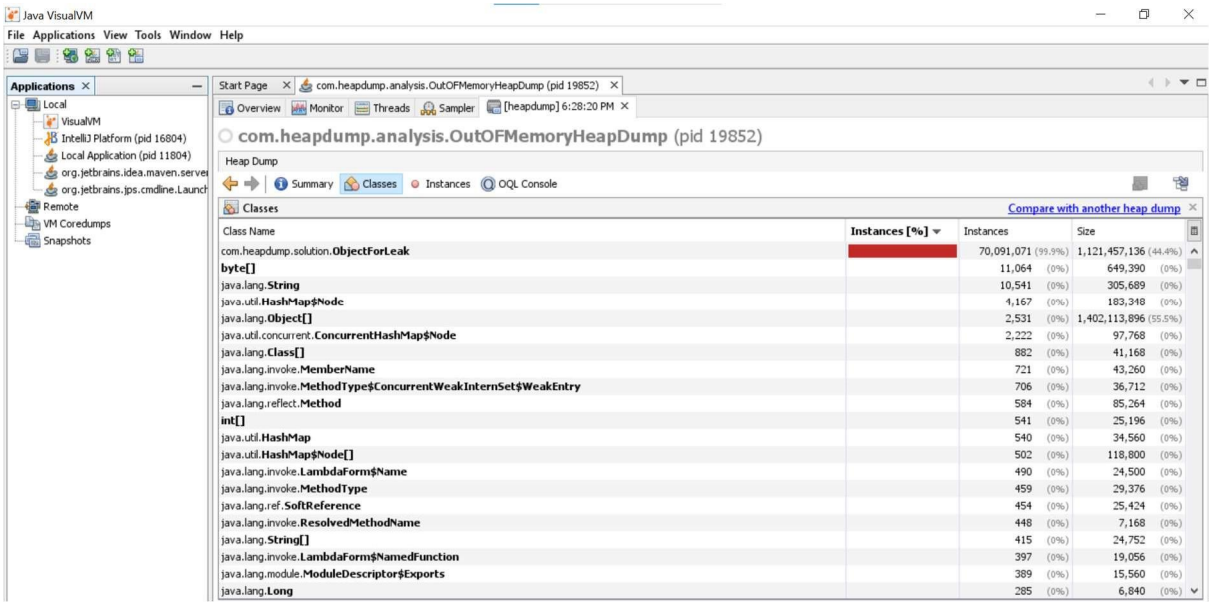
Problem: Unused ArrayList containing duplicate instances of Leak.java class lying in JVM heap memory.

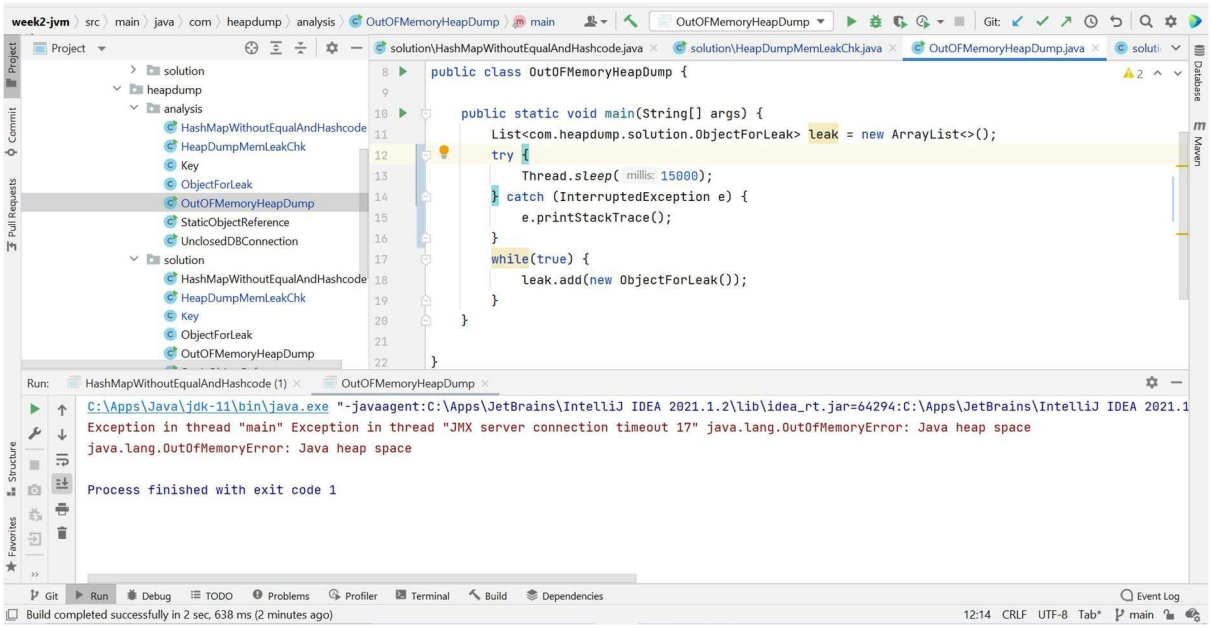


Solution: Remove unused leak ArrayList instance or Make ObjectForLeak.java class singleton

OutOFMemoryHeapDump pdf file

Problem: Adding new instances of ObjectForLeak.java class in an infinite loop.





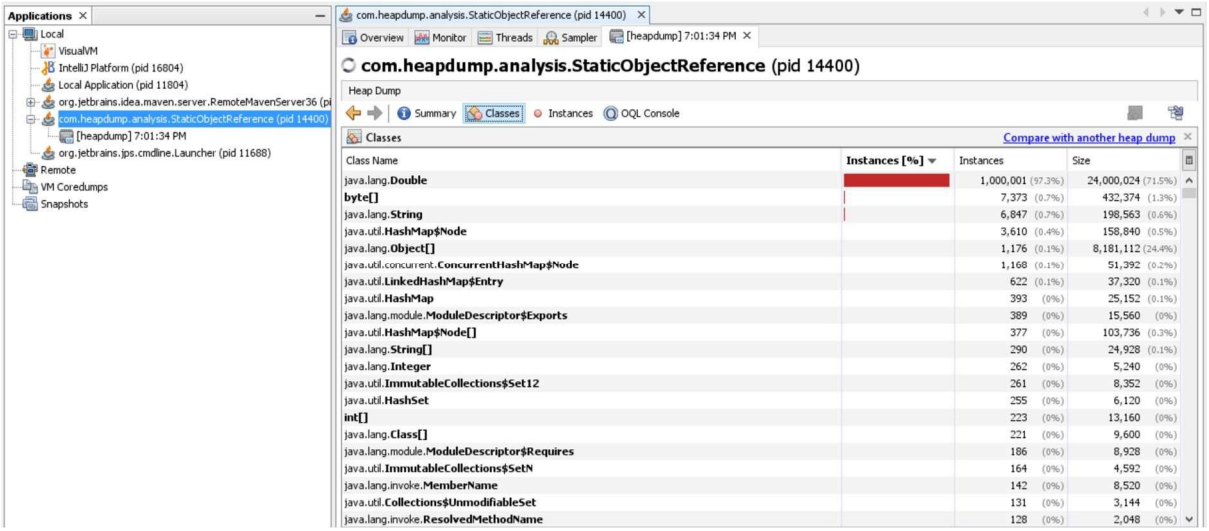
Solution : Put the break in while loop

```java
package com.heapdump.solution;

import ...

public class OutOFMemoryHeapDump {

    public static void main(String[] args) {
        List<com.heapdump.solution.ObjectForLeak> leak = new ArrayList<>();

        while(true) {
            leak.add(new ObjectForLeak());
            break;
        }
    }

}
```

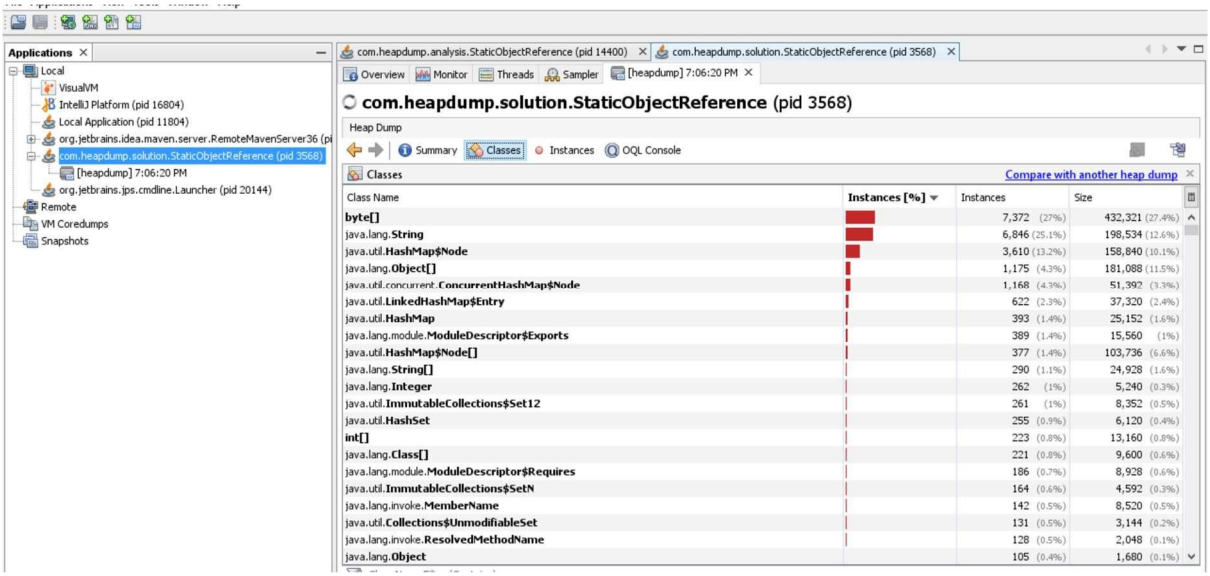
```
FMemoryHeapDump (1)
vaagent:C:\Apps\JetBrains\IntelliJ IDEA 2021.1.2\lib\idea_rt.jar=61163:C:\Apps\JetBrains\IntelliJ IDEA 2021.1
```

StaticObjectReference pdf

Problem : Unused double object :



Solution: unreferenced the list object

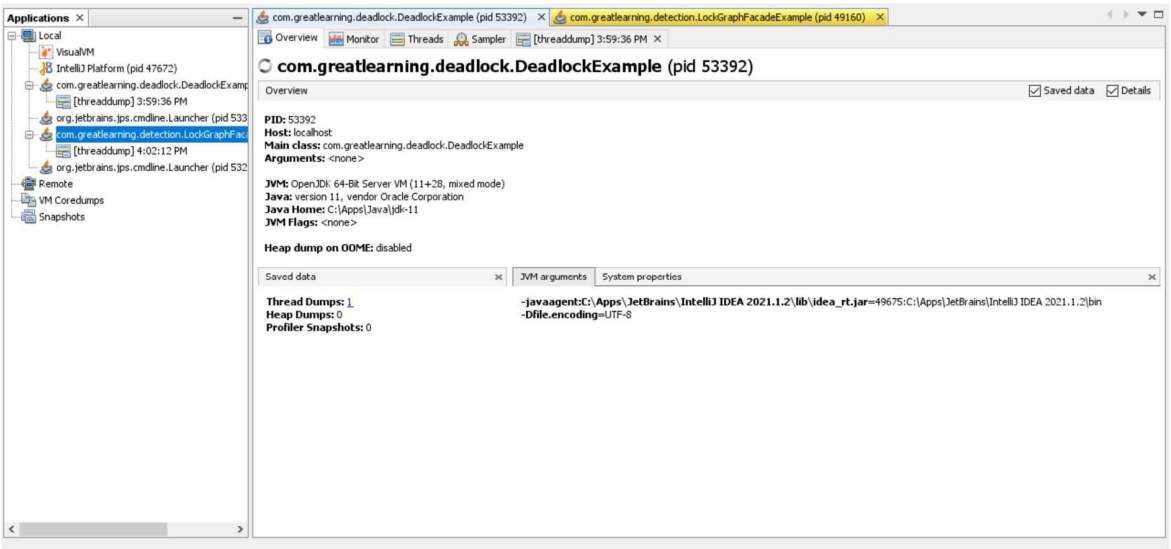


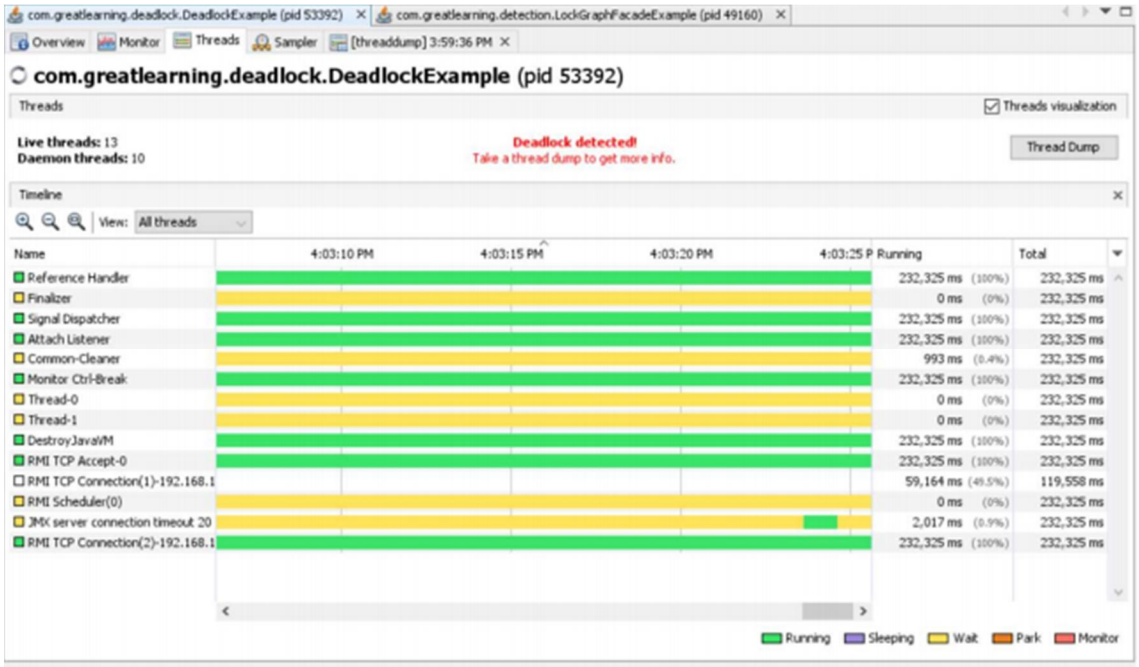
Unclosed DB Connection:

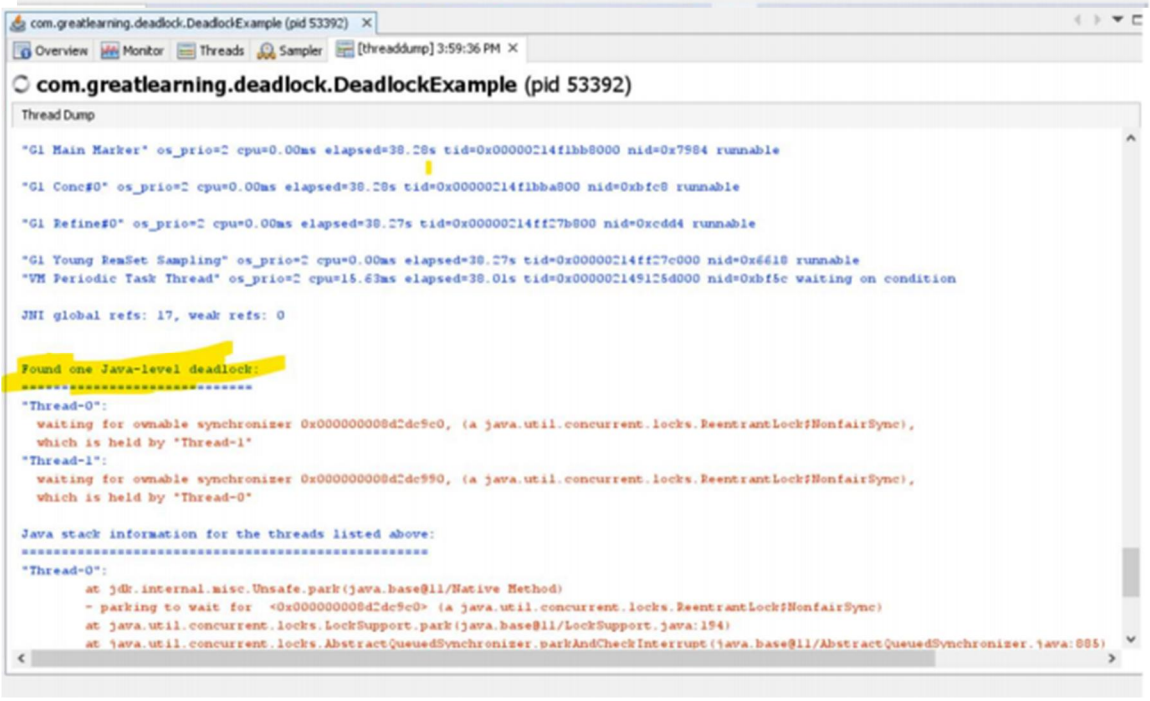
Problem: Connection not closed after opening Solution: Close connection after opening

1. You are required, to generate the deadlock report using JVisualVM and share the proper Screenshots.
2. Refactor the code in order to resolve the Deadlock.

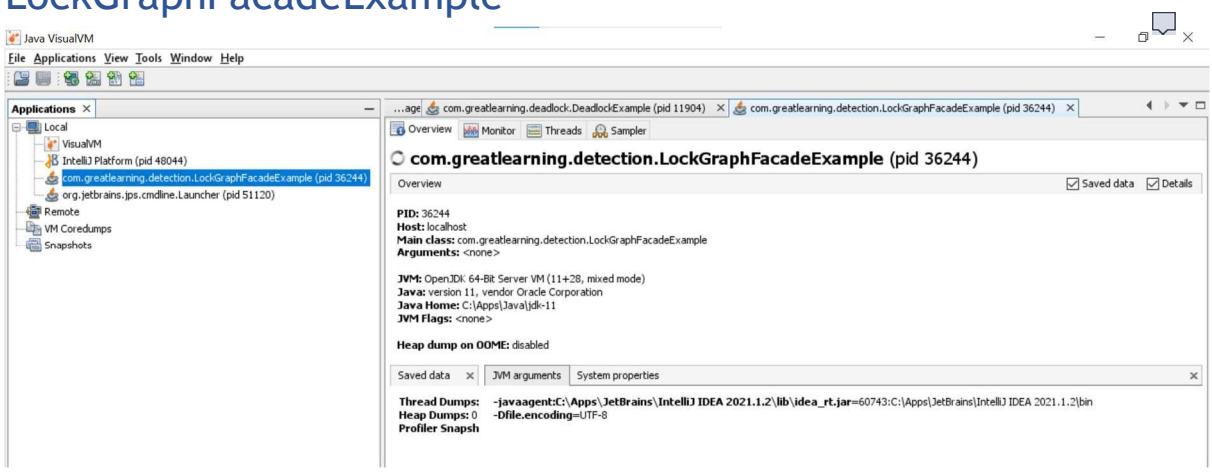
DeadLock screenshot using jvisualVm

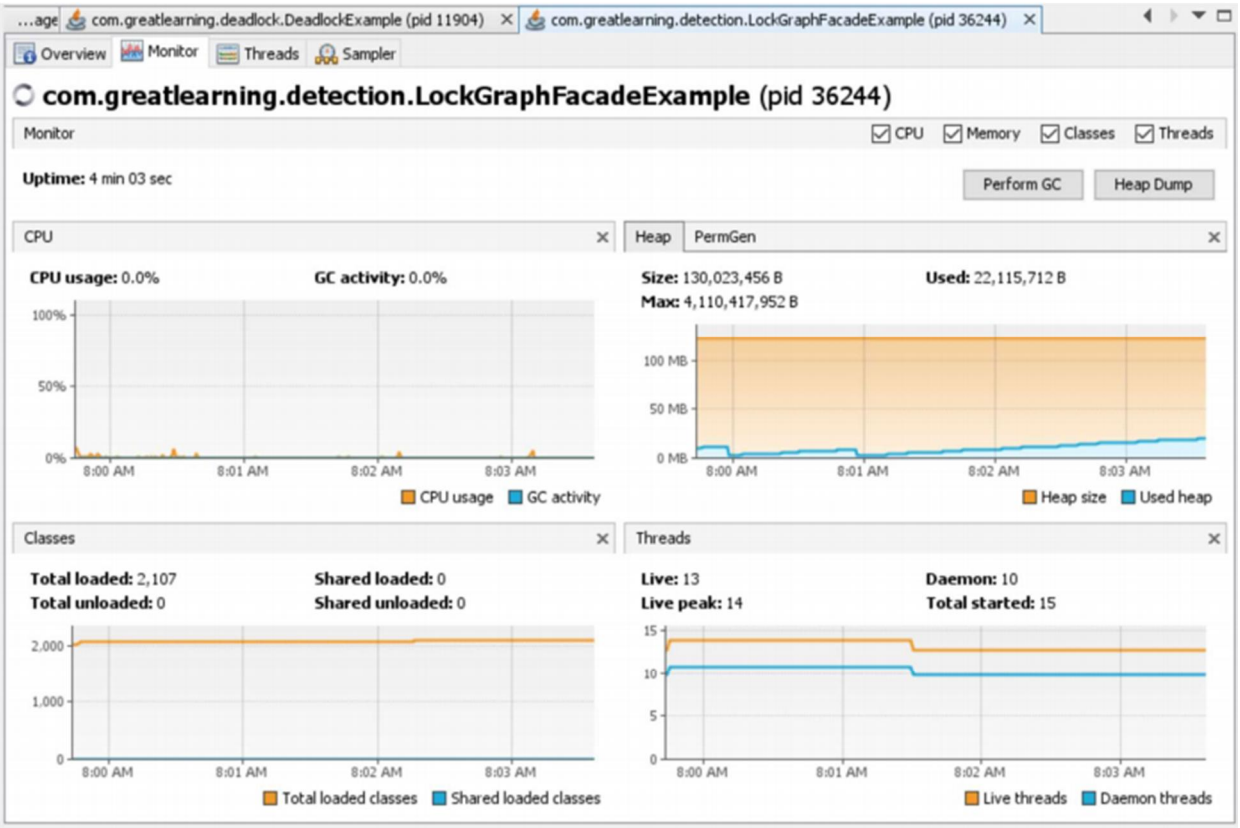


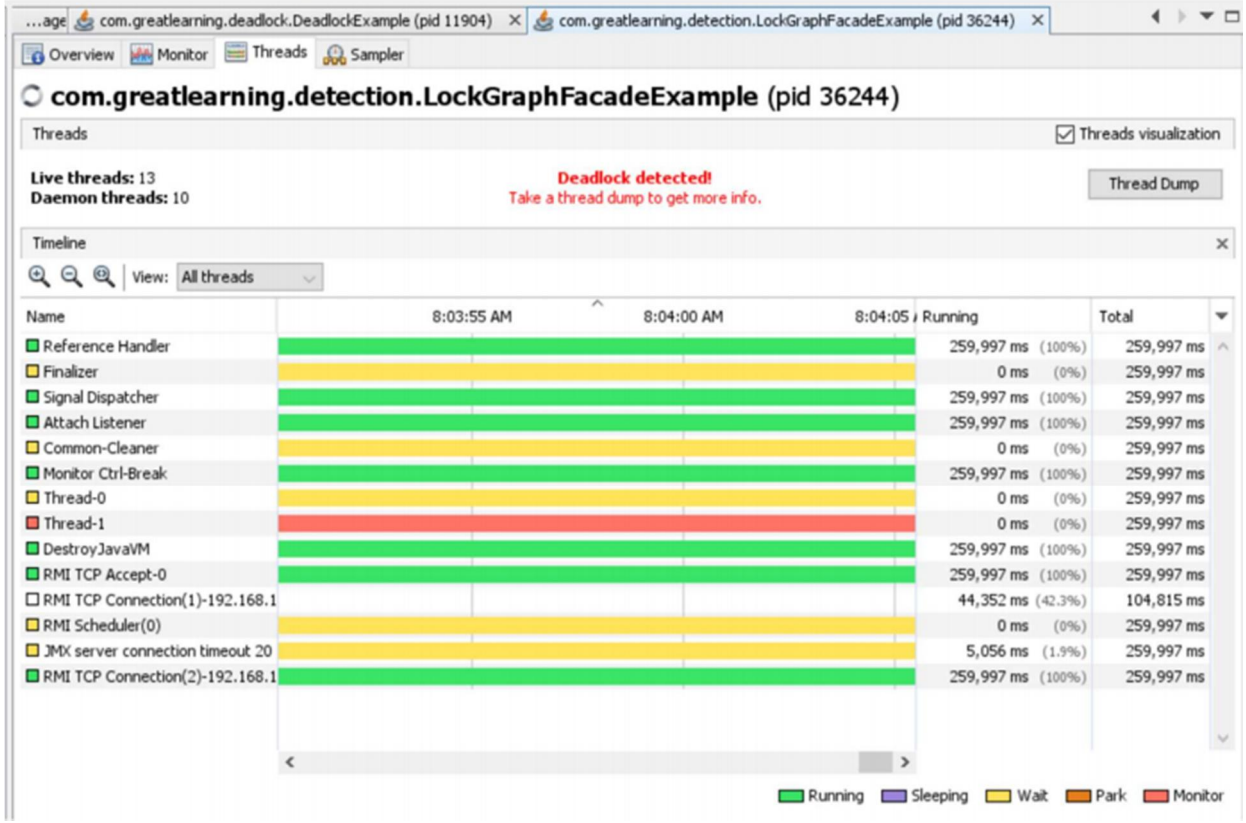




LockGraphFacadeExample



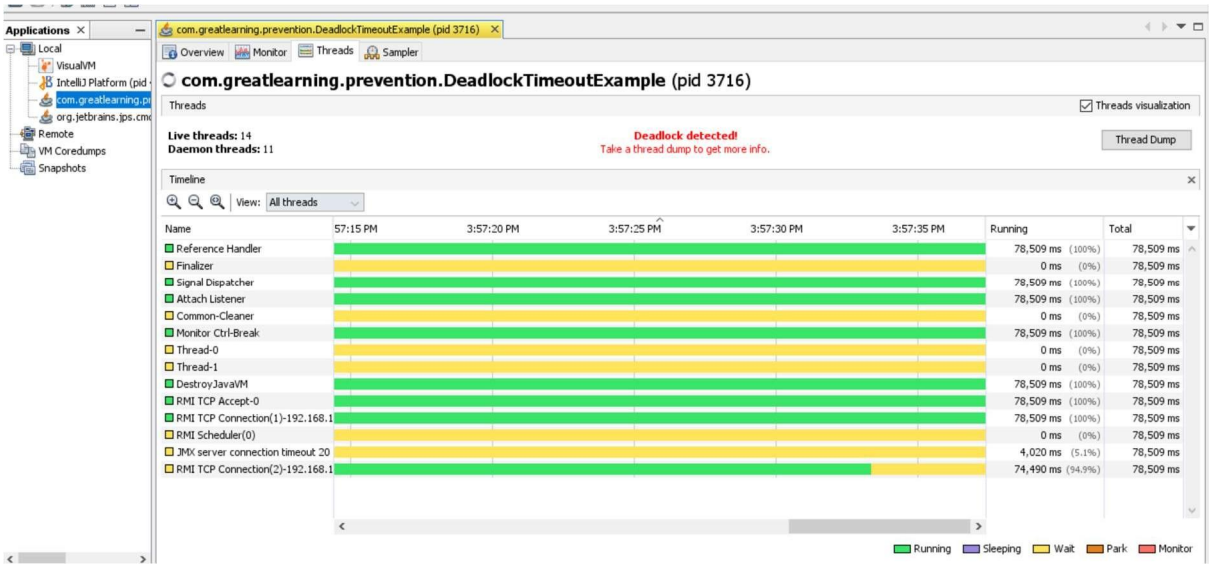

...age
com.greatlearning.deadlock.DeadlockExample (pid 11904)
com.greatlearning.detection.LockGraphFacadeExample (pid 36244)
Overview
Monitor
Threads
Sampler
com.greatlearning.detection.LockGraphFacadeExample (pid 36244)
Monitor
CPU
Memory
Classes
Threads
Uptime: 4 min 03 sec
Perform GC
Heap Dump
CPU
CPU usage: 0.0%
GC activity: 0.0%
100%
50%
0%
8:00 AM
8:01 AM
8:02 AM
8:03 AM
CPU usage
GC activity
Heap
PermGen
Size: 130,023,456 B
Used: 22,115,712 B
Max: 4,110,417,952 B
100 MB
50 MB
0 MB
Heap size
Used heap
Classes
Total loaded: 2,107
Shared loaded: 0
Total unloaded: 0
Shared unloaded: 0
2,000
1,000
0
Total loaded classes
Shared loaded classes
Threads
Live: 13
Daemon: 10
Live peak: 14
Total started: 15
15
10
5
Live threads
Daemon threads

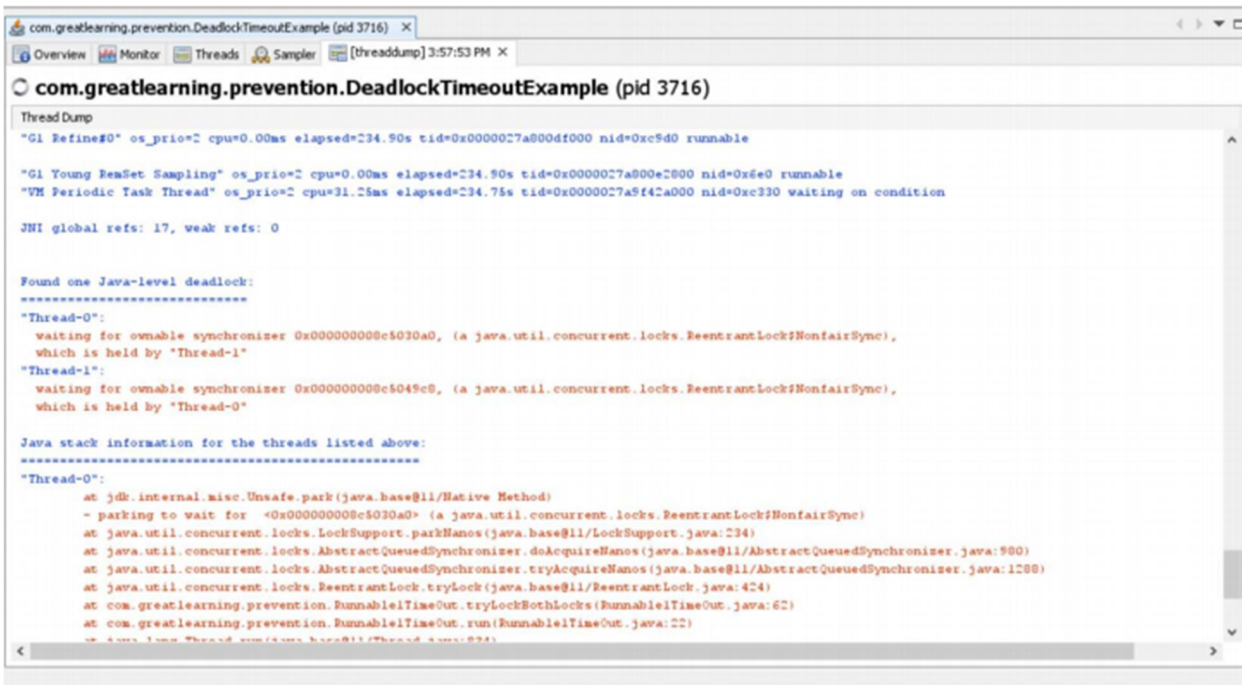

...age
com.greatlearning.deadlock.DeadlockExample (pid 11904)
com.greatlearning.detection.LockGraphFacadeExample (pid 36244)
Overview
Monitor
Threads
Sampler
com.greatlearning.detection.LockGraphFacadeExample (pid 36244)
Threads
Threads visualization
Live threads: 13
Daemon threads: 10
Deadlock detected!
Take a thread dump to get more info.
Thread Dump
Timeline
View: All threads
Name
8:03:55 AM
8:04:00 AM
8:04:05
Running
Total
Reference Handler 259,997 ms (100%) 259,997 ms
Finalizer 0 ms (0%) 259,997 ms
Signal Dispatcher 259,997 ms (100%) 259,997 ms
Attach Listener 259,997 ms (100%) 259,997 ms
Common-Cleaner 0 ms (0%) 259,997 ms
Monitor Ctrl-Break 259,997 ms (100%) 259,997 ms
Thread-0 0 ms (0%) 259,997 ms
Thread-1 0 ms (0%) 259,997 ms
DestroyJavaVM 259,997 ms (100%) 259,997 ms
RMI TCP Accept-0 259,997 ms (100%) 259,997 ms
RMI TCP Connection(1)-192.168.1 44,352 ms (42.3%) 104,815 ms
RMI Scheduler(0) 0 ms (0%) 259,997 ms
JMX server connection timeout 20 5,056 ms (1.9%) 259,997 ms
RMI TCP Connection(2)-192.168.1 259,997 ms (100%) 259,997 ms
Running
Sleeping
Wait
Park
Monitor

Thread Dump

```
"G1 Conc#0" os_prio=2 cpu=0.00ms elapsed=16.27s tid=0x000001db8dc4c800 nid=0x4048 runnable

"G1 Refine#0" os_prio=2 cpu=0.00ms elapsed=16.27s tid=0x000001db8dcd0000 nid=0x9b70 runnable

"G1 Young RemSet Sampling" os_prio=2 cpu=0.00ms elapsed=16.27s tid=0x000001db8dcd1000 nid=0xd68 runnable
"VM Periodic Task Thread" os_prio=2 cpu=0.00ms elapsed=16.09s tid=0x000001dbad022000 nid=0x2f90 waiting on condition

JNI global refs: 17, weak refs: 0


Found one Java-level deadlock:
=============================
"Thread-0":
  waiting to lock monitor 0x000001dbac416c00 (object 0x000000008d2dfb70, a com.greatlearning.detection.LockGraphFacade),
  which is held by "Thread-1"
"Thread-1":
  waiting for ownable synchronizer 0x000000008d2ele38, (a java.util.concurrent.locks.ReentrantLock$NonfairSync),
  which is held by "Thread-0"

Java stack information for the threads listed above:
===================================================
"Thread-0":
        at com.greatlearning.detection.LockGraphFacade.tryLock(LockGraphFacade.java:15)
        - waiting to lock <0x000000008d2dfb70> (a com.greatlearning.detection.LockGraphFacade)
        at com.greatlearning.detection.RunnableIDeadlockDetection.tryLockBothLocks(RunnableIDeadlockDetection.java:58)
        at com.greatlearning.detection.RunnableIDeadlockDetection.run(RunnableIDeadlockDetection.java:24)
        at java.lang.Thread.run(java.base@11/Thread.java:834)
"Thread-1":
```

DeadlockTimeoutExample





HashMapWithoutEqualandHashCode

HeapDumpMemLeakChk pdf file