Register | Login | Logout | Contact Us

Java-Success.com

Industrial strength Java Career Companion

search here … Go

Home | Java FAQs | 600+ Java Q&As | Career | Tutorials | Member | Why?

Can u Debug? | Java 8 ready? | Top X | Productivity Tools | Judging Experience?

Home › Interview › Testing & Profiling/Sampling Java Apps Q&A › jvisualvm profiling › 02: jvisualvm to detect memory leak – a quick tutorial style Java demo

02: jvisualvm to detect memory leak – a quick tutorial style Java demo

Posted on March 16, 2016 by Arulkumaran Kumaraswamipillai

This is a companion post to 8 Java Memory Management Interview Q&A demonstrating a memory leak scenario step by step with jvisualvm, which is a free profiling tool that gets shipped with JDK.

Step 1: Code that causes memory leak

Here is a sample code that causes memory leak. It is shown with a never ending while loop for the demo purpose, but in a real production application this could be a logic within a method that gets accessed very frequently.

```

import java.util.HashMap;
import java.util.Map;
import java.util.concurrent.TimeUnit;

```

600+ Full Stack Java/JEE Interview Q&As ♥Free ♦FAQs

open all | close all

⊞ Ice Breaker Interview
⊞ Core Java Interview (
⊞ JEE Interview Q&A (3
⊞ Pressed for time? Jav
⊞ SQL, XML, UML, JSC
⊞ Hadoop & BigData Int
⊞ Java Architecture Inte
⊞ Scala Interview Q&As
⊞ Spring, Hibernate, & I
⊟ Testing & Profiling/Sa
⊞ Automation Testing
⊞ Code Coverage (2)
⊞ Code Quality (2)
⊟ jvisualvm profiling (
01: jvisualvm to
02: jvisualvm to
03: jvisualvm to
⊞ Performance Testin
⊞ Unit Testing Q&A (2
⊞ Other Interview Q&A 1
⊞ ▶ Free Java Interview

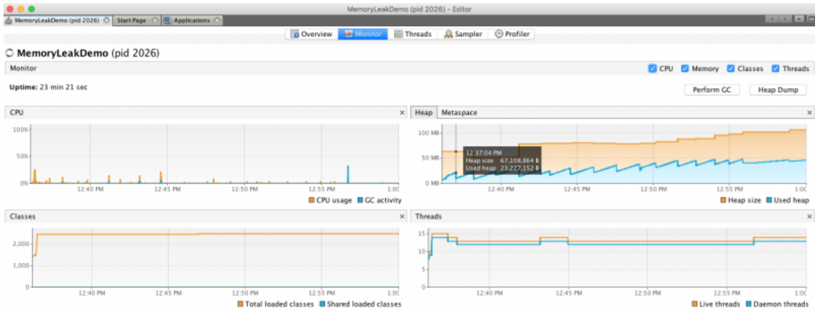
```
public class MemoryLeakDemo {
    public static void main(String[] args) throw
        Map<Key, String> map = new HashMap<Key,
        int counter = 0;
        while (true) {
            // creates duplicate objects due to
            map.put(new Key("dummyKey"), "value"
            counter++;
            if (counter % 1000 == 0) {
                System.out.println("map size: "
                TimeUnit.SECONDS.sleep(2);
            }
        }
    }

    // inner class key without hashcode() or equ
    static class Key {
        private String key;

        public Key(String key) {
            this.key = key;
        }
    }
}


```

In the above code the “Key” class that is used for storing values into a map is not properly implemented by overriding equals() & hashCode(), hence it will be using Object class’s implementation which uses the memory location of each new object created.

Step 2: Start jvisualvm

```

$ jvisualvm

```

Step 3: Run the code & monitor jvisualvm



Monitoring the heap – when started running the application

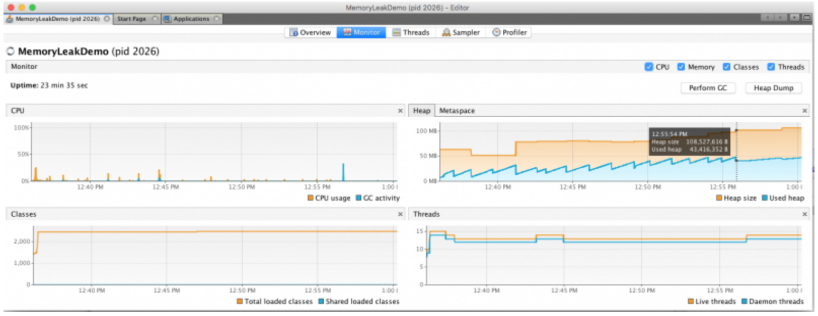
16 Technical Key Areas

open all | close all
⊞ Best Practice (6)
⊞ Coding (26)
⊞ Concurrency (6)
⊞ Design Concepts (7)
⊞ Design Patterns (11)
⊞ Exception Handling (3
⊞ Java Debugging (21)
⊞ Judging Experience I
⊞ Low Latency (7)
⊞ Memory Managemen
⊞ Performance (13)
⊞ QoS (8)
⊞ Scalability (4)
⊞ SDLC (6)
⊞ Security (13)
⊞ Transaction Managen

80+ step by step Java Tutorials

open all | close all
⊞ Setting up Tutorial (6)
⊞ Tutorial - Diagnosis (2
⊞ Akka Tutorial (9)
⊞ Core Java Tutorials (2
⊞ Hadoop & Spark Tuto
⊞ JEE Tutorials (19)
⊞ Scala Tutorials (1)
⊞ Spring & HIbernate Tı
⊞ Tools Tutorials (19)
⊞ Other Tutorials (45)

After 35 minutes:



Uptime: 23 min 35 sec

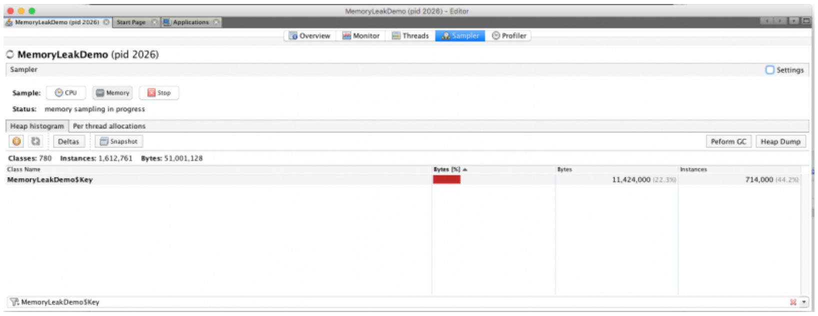
12:55:54 PM
Heap size 108,527,616 B
Used heap 43,416,352 B

Monitoring the JVM heap after 35 minutes

As, you can see the JVM heap memory usage is keep going up, The "saw tooth" like diagram shown above indicates memory leak. The memory used has gone up from 23MB to 43MB within 35 minutes.

Step 4: Uncontrolled creation of the instances of Key class is the culprit

As you can see 714K instances created at this point.



Sample: CPU Memory Stop

Status: memory sampling in progress

Classes: 780 Instances: 1,612,761 Bytes: 51,001,128

Class Name	Bytes [%] ▲	Bytes	Instances
MemoryLeakDemo$Key	🟥	11,424,000 (22.3%)	714,000 (44.2%)

MemoryLeakDemo$Key

Sampling the JVM heap

Step 5: How to fix the code?

Implement the hashCode() & equals() method to the "Key" class and run the code and profile with jvisualvm.

100+ Java pre-interview coding tests

open all | close all

- Can you write code? (
- ♦ Complete the given
- Converting from A to I
- Designing your classe
- Java Data Structures
- Passing the unit tests
- What is wrong with th
- Writing Code Home A
- Written Test Core Jav
- Written Test JEE (1)

How good are your .....?

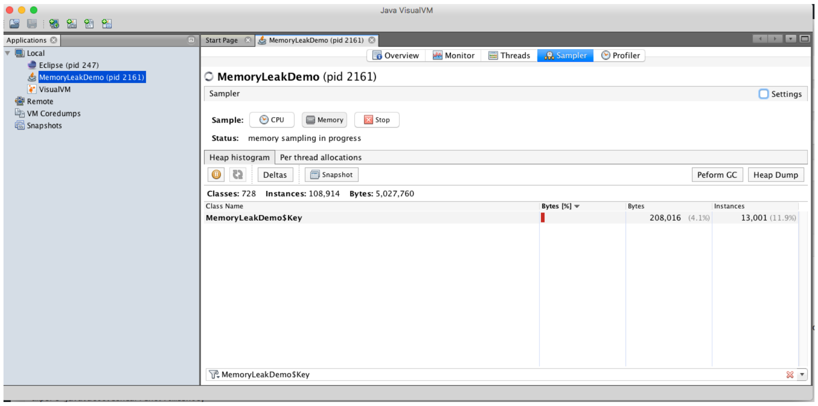
open all | close all

- Career Making Know-
- Job Hunting & Resum

```java

import java.util.HashMap;
import java.util.Map;
import java.util.Objects;
import java.util.concurrent.TimeUnit;

public class MemoryLeakDemo {
    public static void main(String[] args) throw
        Map<Key, String> map = new HashMap<Key,
        int counter = 0;
        while (true) {
            // creates duplicate objects due to
            map.put(new Key("dummyKey"), "value"
            counter++;
            if (counter % 1000 == 0) {
                System.out.println("map size: "
                TimeUnit.SECONDS.sleep(2);
            }
        }
    }

    // inner class key without hashcode() or equ
    static class Key {
        private String key;

        public Key(String key) {
            this.key = key;
        }

        @Override
        public int hashCode() {
            return Objects.hash(key); // Java 8
        }

        @Override
        public boolean equals(Object obj) {
            if (obj == null) {
                return false;
            }
            if (getClass() != obj.getClass()) {
                return false;
            }

            Key other = (Key) obj;

            return Objects.equals(this.key, othe
        }
    }
}


```

Step 6: jvisualvm sampling after fixing the code

Even though the instances count shows 13,000 instances, it was because the GC has not been kicked in yet. Click on the “Perform GC” button a few times and you will see the count go down to 1.



Sample:

Status: memory sampling in progress

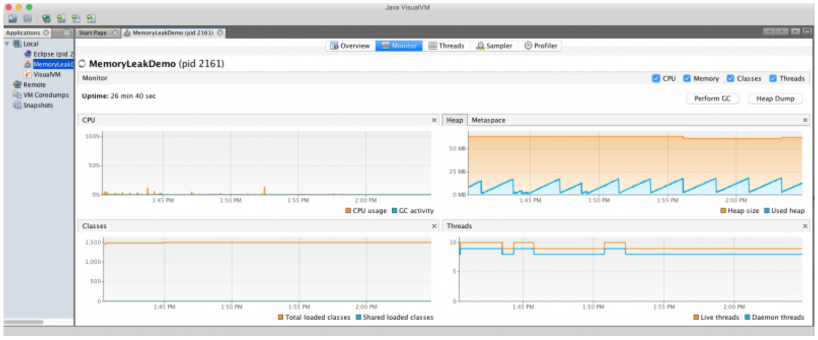
Classes: 728 **Instances:** 108,914 **Bytes:** 5,027,760

Class Name	Bytes [%]	Bytes	Instances
MemoryLeakDemo$Key		208,016 (4.1%)	13,001 (11.9%)

MemoryLeakDemo$Key

JVM heap sampling

Step 7: jvisualvm heap memory monitoring after fixing the code



Uptime: 26 min 40 sec

Heap Memory Monitoring

As you can see, the memory usage is fully under control without any leaks. The key objects created in the while loop periodically gets garbage collected.

Popular Member Posts

- ♦ 11 Spring boot interview questions & answers

  905 views

- ♦ Q11-Q23: Top 50+ Core on Java OOP Interview Questions & Answers

  816 views

- 001A: ♦ 7+ Java integration styles & patterns interview questions & answers

427 views

18 Java scenarios based interview Questions and Answers

409 views

♦ 7 Java debugging interview questions & answers

324 views

01b: ♦ 13 Spring basics Q8 – Q13 interview questions & answers

311 views

01: ♦ 15 Ice breaker questions asked 90% of the time in Java job interviews with hints

304 views

♦ 10 ERD (Entity-Relationship Diagrams) Interview Questions and Answers

301 views

♦ Q24-Q36: Top 50+ Core on Java classes, interfaces and generics interview questions & answers

251 views

♦ Object equals Vs == and pass by reference Vs value

234 views

0 Like Share Tweet submit reddit 0 G+1 Share

Bio **Latest Posts**

Arulkumaran Kumaraswamipillai

Mechanical Eng to freelance Java developer in 3 yrs. Contracting since 2003, and attended 150+ Java job interviews, and often got 4 - 7 job offers to choose from. It pays to prepare. So, published Java interview Q&A books via Amazon.com in 2005, and sold 35,000+ copies. Books are outdated and replaced with this subscription based site.**945+** paid members. join my LinkedIn Group. **Reviews**

About Arulkumaran Kumaraswamipillai

Mechanical Eng to freelance Java developer in 3 yrs. Contracting since 2003, and attended 150+ Java job interviews, and often got 4 - 7 job offers to choose from. It pays to prepare. So, published Java interview Q&A books via Amazon.com in 2005, and sold 35,000+ copies. Books are outdated and replaced with this subscription based site.**945+** paid members. join my LinkedIn Group. **Reviews**

‹ 01: jvisualvm to sample Java heap memory

11: Threads performing tasks by talking to each other ›

Posted in jvisualvm profiling**,** JVM**,** Memory Management

Empowers you to open more doors, and fast-track

Technical Know Hows

☀ Java generics in no time ☀ Top 6 tips to transforming your thinking from OOP to FP ☀ How does a HashMap internally work? What is a hashing function?
☀ 10+ Java String class interview Q&As ☀ Java auto un/boxing benefits & caveats ☀ Top 11 slacknesses that can come back and bite you as an experienced Java developer or architect

Non-Technical Know Hows

☀ 6 Aspects that can motivate you to fast-track your career & go places ☀ Are you reinventing yourself as a Java developer? ☀ 8 tips to safeguard your Java career against offshoring ☀ My top 5 career mistakes

Prepare to succeed

☀ Turn readers of your Java CV go from “Blah blah” to “Wow”? ☀ How to prepare for Java job interviews? ☀ 16 Technical Key Areas ☀ How to choose from multiple Java job offers?

Select Category ▼

© Disclaimer

The contents in this Java-Success are copy righted. The author has the right to correct or enhance the current content without any prior notice.

These are general advice only, and one needs to take his/her own circumstances into consideration. The author will not be held liable for any damages caused or alleged to be caused either directly or indirectly by these materials and resources. Any trademarked names or labels used in this blog remain the property of their respective trademark owners. No guarantees are made regarding the accuracy or usefulness of content, though I do make an effort to be accurate. Links to external sites do not imply endorsement of the linked-to sites.

↑