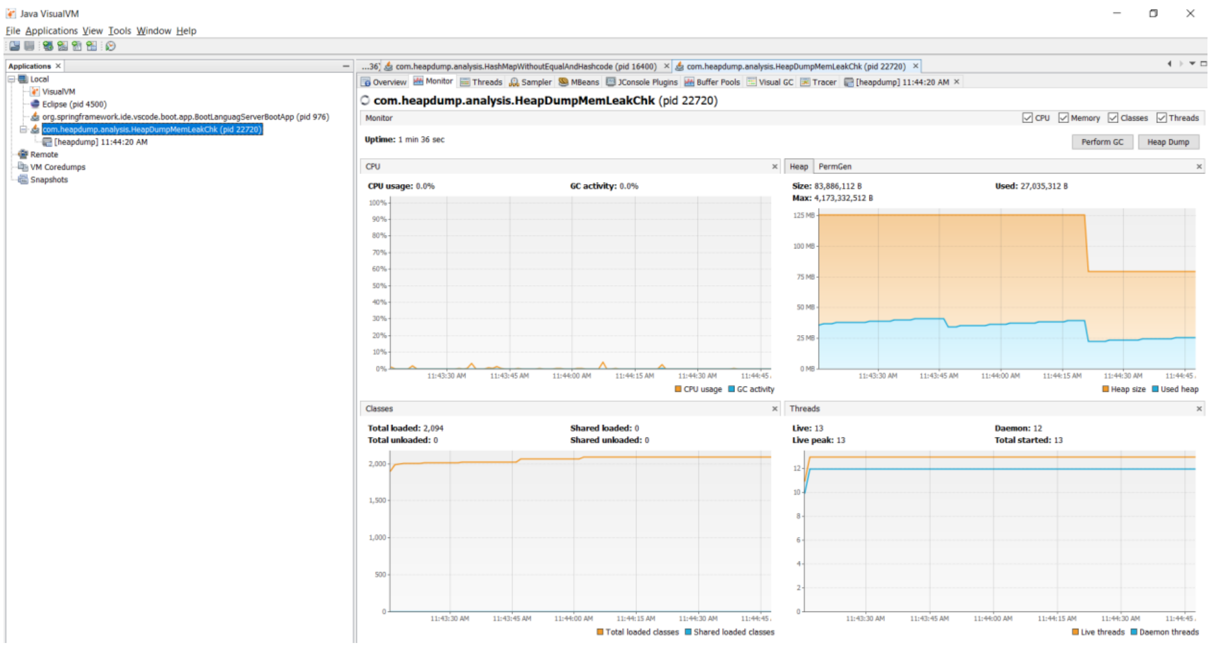
HeapDumpMemLeakChk

HEAP MONITOR:


Java VisualVM
File Applications View Tools Window Help
Applications
Local
VisualVM
Eclipse (pid 4500)
org.springframework.ide.vscode.boot.app.BootLanguagServerBootApp (pid 976)
com.heapdump.analysis.HeapDumpMemLeakChk (pid 22720)
[heapdump] 11:44:20 AM
Remote
VM Coredumps
Snapshots
...36 com.heapdump.analysis.HashMapWithoutEqualAndHashcode (pid 16400)
com.heapdump.analysis.HeapDumpMemLeakChk (pid 22720)
Overview Monitor Threads Sampler MBeans JConsole Plugins Buffer Pools Visual GC Tracer [heapdump] 11:44:20 AM
Monitor
CPU Memory Classes Threads
Perform GC
Heap Dump
CPU
Heap PermGen
Classes
Threads
CPU usage
GC activity
Heap size
Used heap
Total loaded classes
Shared loaded classes
Live threads
Daemon threads

com.heapdump.analysis.HeapDumpMemLeakChk (pid 22720)

Uptime: 1 min 36 sec

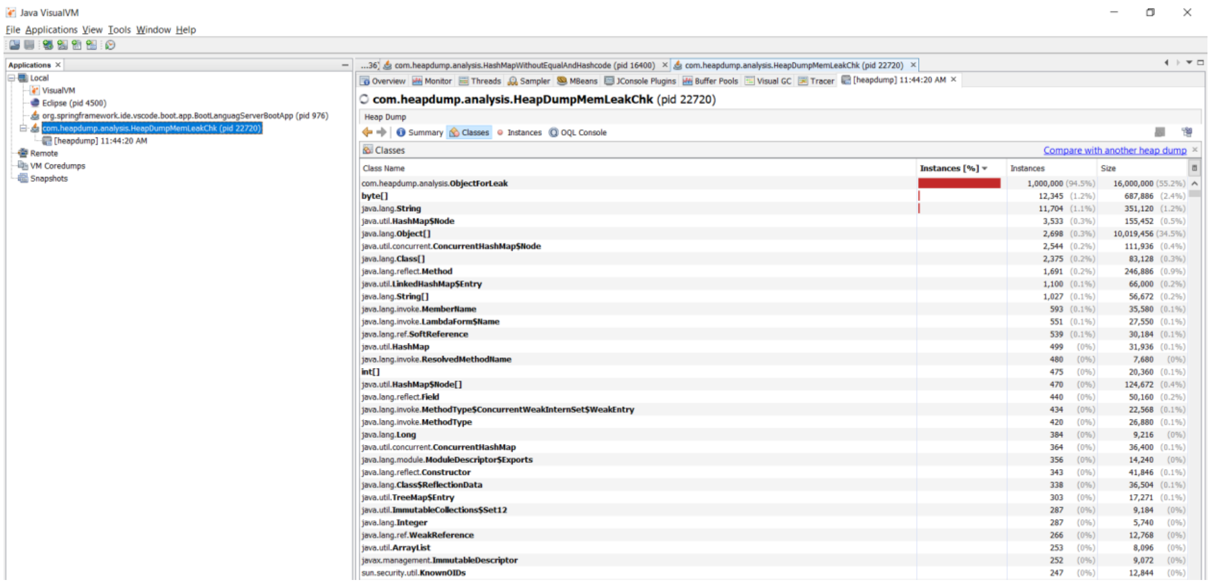
CPU usage: 0.0% GC activity: 0.0%

Size: 83,886,112 B Used: 27,035,312 B
Max: 4,173,332,512 B

Total loaded: 2,094 Shared loaded: 0
Total unloaded: 0 Shared unloaded: 0

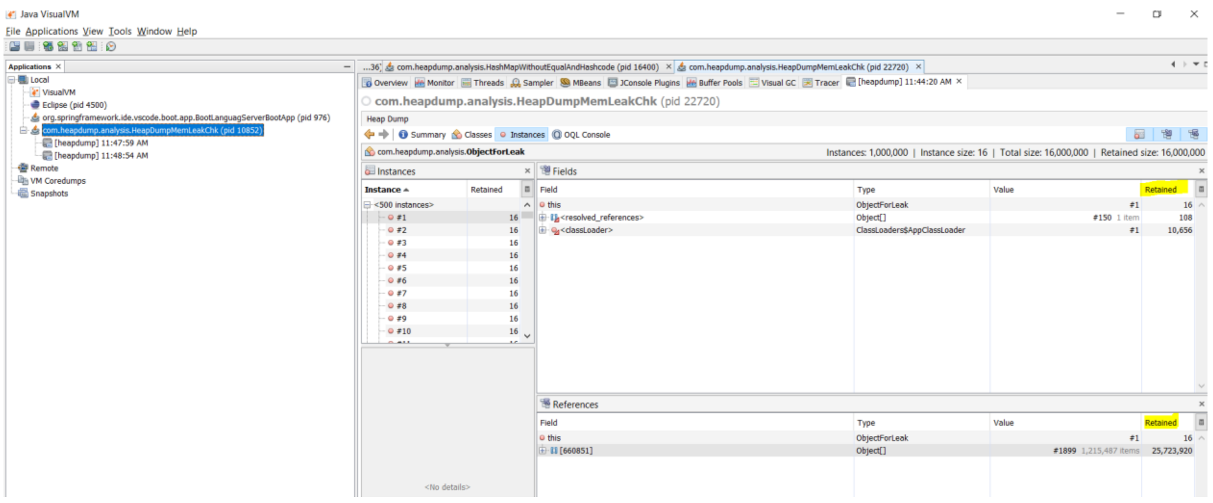
Live: 13 Daemon: 12
Live peak: 13 Total started: 13

HEAP DUMP:


Java VisualVM
File Applications View Tools Window Help
Applications
Local
VisualVM
Eclipse (pid 4500)
org.springframework.ide.vscode.boot.app.BootLanguagServerBootApp (pid 976)
com.heapdump.analysis.HeapDumpMemLeakChk (pid 22720)
[heapdump] 11:44:20 AM
Remote
VM Coredumps
Snapshots
...36 com.heapdump.analysis.HashMapWithoutEqualAndHashcode (pid 16400)
com.heapdump.analysis.HeapDumpMemLeakChk (pid 22720)
Overview Monitor Threads Sampler MBeans JConsole Plugins Buffer Pools Visual GC Tracer [heapdump] 11:44:20 AM
Heap Dump
Summary Classes Instances OQL Console
Classes
Compare with another heap dump

com.heapdump.analysis.HeapDumpMemLeakChk (pid 22720)

Class Name	Instances [%]	Instances	Size
com.heapdump.analysis.**ObjectForLeak**		1,000,000 (94.5%)	16,000,000 (55.2%)
byte[]		12,345 (1.2%)	687,886 (2.4%)
java.lang.**String**		11,704 (1.1%)	351,120 (1.2%)
java.util.**HashMap$Node**		3,533 (0.3%)	155,452 (0.5%)
java.lang.**Object[]**		2,698 (0.3%)	10,019,456 (34.5%)
java.util.concurrent.**ConcurrentHashMap$Node**		2,544 (0.2%)	111,936 (0.4%)
java.lang.**Class[]**		2,375 (0.2%)	83,128 (0.3%)
java.lang.reflect.**Method**		1,691 (0.2%)	246,886 (0.9%)
java.util.**LinkedHashMap$Entry**		1,100 (0.1%)	66,000 (0.2%)
java.lang.**String[]**		1,027 (0.1%)	56,672 (0.2%)
java.lang.invoke.**MemberName**		593 (0.1%)	35,580 (0.1%)
java.lang.invoke.**LambdaForm$Name**		551 (0.1%)	27,550 (0.1%)
java.lang.ref.**SoftReference**		539 (0.1%)	30,184 (0.1%)
java.util.**HashMap**		499 (0%)	31,936 (0.1%)
java.lang.invoke.**ResolvedMethodName**		480 (0%)	7,680 (0%)
int[]		475 (0%)	20,360 (0.1%)
java.util.**HashMap$Node[]**		470 (0%)	124,672 (0.4%)
java.lang.reflect.**Field**		440 (0%)	50,160 (0.2%)
java.lang.invoke.**MethodType$ConcurrentWeakInternSet$WeakEntry**		434 (0%)	22,568 (0.1%)
java.lang.invoke.**MethodType**		420 (0%)	26,880 (0.1%)
java.lang.**Long**		384 (0%)	9,216 (0%)
java.util.concurrent.**ConcurrentHashMap**		364 (0%)	36,400 (0.1%)
java.lang.module.**ModuleDescriptor$Exports**		356 (0%)	14,240 (0%)
java.lang.reflect.**Constructor**		343 (0%)	41,846 (0.1%)
java.lang.**Class$ReflectionData**		338 (0%)	36,504 (0.1%)
java.util.**TreeMap$Entry**		303 (0%)	17,271 (0.1%)
java.util.**ImmutableCollections$Set12**		287 (0%)	9,184 (0%)
java.lang.**Integer**		287 (0%)	5,740 (0%)
java.lang.ref.**WeakReference**		266 (0%)	12,768 (0%)
java.util.**ArrayList**		253 (0%)	8,096 (0%)
javax.management.**ImmutableDescriptor**		252 (0%)	9,072 (0%)
sun.security.util.**KnownOIDs**		247 (0%)	12,844 (0%)


Java VisualVM
File Applications View Tools Window Help
Applications
Local
VisualVM
Eclipse (pid 4500)
org.springframework.ide.vscode.boot.app.BootLanguagServerBootApp (pid 976)
com.heapdump.analysis.HeapDumpMemLeakChk (pid 10852)
[heapdump] 11:47:59 AM
[heapdump] 11:48:54 AM
Remote
VM Coredumps
Snapshots
...36 com.heapdump.analysis.HashMapWithoutEqualAndHashcode (pid 16400)
com.heapdump.analysis.HeapDumpMemLeakChk (pid 22720)
Overview Monitor Threads Sampler MBeans JConsole Plugins Buffer Pools Visual GC Tracer [heapdump] 11:44:20 AM
Heap Dump
Summary Classes Instances OQL Console
<No details>

com.heapdump.analysis.HeapDumpMemLeakChk (pid 22720)

com.heapdump.analysis.**ObjectForLeak**

Instances: 1,000,000 | Instance size: 16 | Total size: 16,000,000 | Retained size: 16,000,000

Instances

Instance	Retained
<500 instances>	
#1	16
#2	16
#3	16
#4	16
#5	16
#6	16
#7	16
#8	16
#9	16
#10	16

Fields

Field	Type	Value	Retained
this	ObjectForLeak	#1	16
<resolved_references>	Object[]	#150 1 item	108
<classLoader>	ClassLoaders$AppClassLoader	#1	10,656

References

Field	Type	Value	Retained
this	ObjectForLeak	#1	16
[660851]	Object[]	#1899 1,215,487 items	25,723,920

Reason for leak:

In above diagrams we see that new Objects of ObjectForLeak are being created, with the new keyword being used, continuously which is NOT CLEARED by the GC and stays in the memory causing retainment of the objects for this code.