

Федеральное государственное автономное образовательное  
учреждение высшего образования  
«Национальный исследовательский университет ИТМО»  
Факультет Программной Инженерии и Компьютерной Техники

Лабораторная работа №4 по дисциплине  
«Основы программной инженерии»

Вариант №999

Выполнил:  
Состанов Тимур Айратович  
Группа: Р3214  
Преподаватель:  
Птицын Максим Евгеньевич

г. Санкт-Петербург  
2024

# Задание

Вариант 999

## Внимание! У разных вариантов разный текст задания!

1. Для своей программы из лабораторной работы #3 по дисциплине "Веб-программирование" реализовать:

- MBeap, считающий общее число установленных пользователем точек, а также число точек, попадающих в область. В случае, если координаты установленной пользователем точки вышли за пределы отображаемой области координатной плоскости, разработанный MBeap должен отправлять оповещение об этом событии.
- MBeap, определяющий процентное отношение "промахов" к общему числу кликов пользователя по координатной плоскости.

2. С помощью утилиты JConsole провести мониторинг программы:

- Снять показания MBeap-классов, разработанных в ходе выполнения задания 1.
- Определить версию Java Language Specification, реализуемую данной средой исполнения.

3. С помощью утилиты VisualVM провести мониторинг и профилирование программы:

- Снять график изменения показаний MBeap-классов, разработанных в ходе выполнения задания 1, с течением времени.
- Определить имя класса, объекты которого занимают наибольший объем памяти JVM; определить пользовательский класс, в экземплярах которого находятся эти объекты.

4. С помощью утилиты VisualVM и профилировщика IDE NetBeans, Eclipse или Idea локализовать и устранить проблемы с производительностью в программе. По результатам локализации и устранения проблемы необходимо составить отчет, в котором должна содержаться следующая информация:

- Описание выявленной проблемы.
- Описание путей устранения выявленной проблемы.
- Подробное (со скриншотами) описание алгоритма действий, который позволил выявить и локализовать проблему.

Студент должен обеспечить возможность воспроизведения процесса поиска и локализации проблемы по требованию преподавателя.

# Выполнение

## 1 Пункт

Реализованные MBean-ы:

[https://github.com/tsostanov/opi\\_lab4/tree/main/src/main/java/jmx](https://github.com/tsostanov/opi_lab4/tree/main/src/main/java/jmx)

Подключение и обработка:

[https://github.com/tsostanov/opi\\_lab4/blob/main/src/main/java/models/UserBean.java](https://github.com/tsostanov/opi_lab4/blob/main/src/main/java/models/UserBean.java)

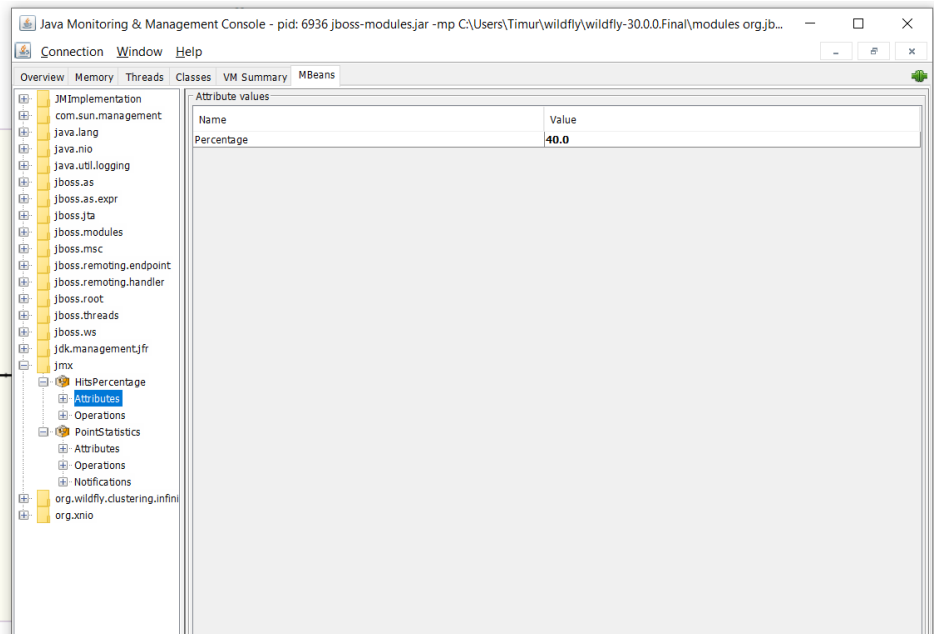
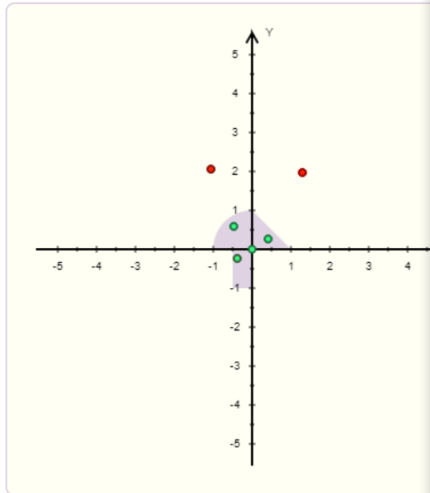
```
public UserBean(){ no usages new *
    hitsPercentage = new HitsPercentage(requests, clickCounter);
    pointStatisticsMBean = new PointStatistics(requests);
    try {
        JMXMonitor.configure(hitsPercentage, pointStatisticsMBean);
    } catch (MalformedObjectNameException e) {
        throw new RuntimeException(e);
    } catch (NotCompliantMBeanException e) {
        throw new RuntimeException(e);
    } catch (InstanceAlreadyExistsException e) {
        throw new RuntimeException(e);
    } catch (MBeanRegistrationException e) {
        throw new RuntimeException(e);
    }
}
```

```
public void addPoint(Point point) { 2 usages tsostanov *
    hitsPercentage.updatePoints(requests, clickCounter);
    hitsPercentage.getPercentage();
    pointStatisticsMBean.updatePoints(requests);
    pointStatisticsMBean.getTotalPoints();|
    pointStatisticsMBean.getPointsInArea();
    DatabaseHandler.getDatabaseManager().addPoint(point);
    this.requests.add(index: 0, point);
}
```

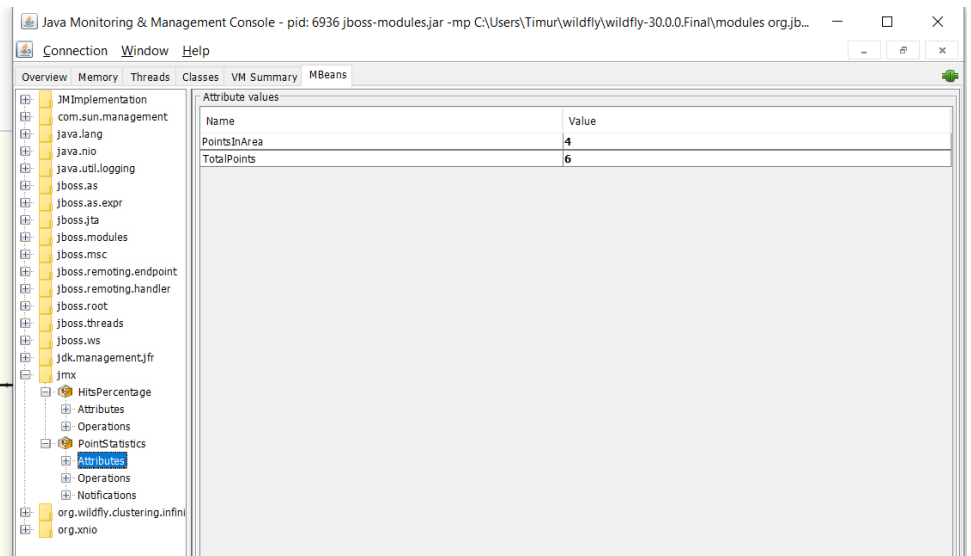
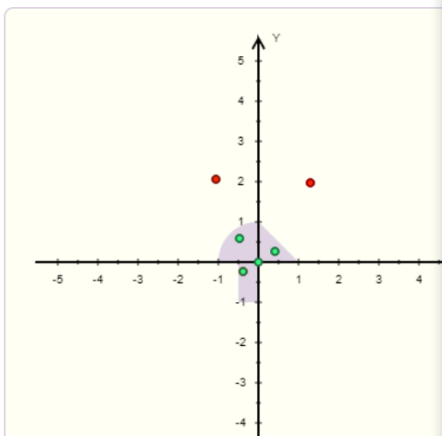
## 2 Пункт

Показания разработанных MBean-ов:

Start



Start





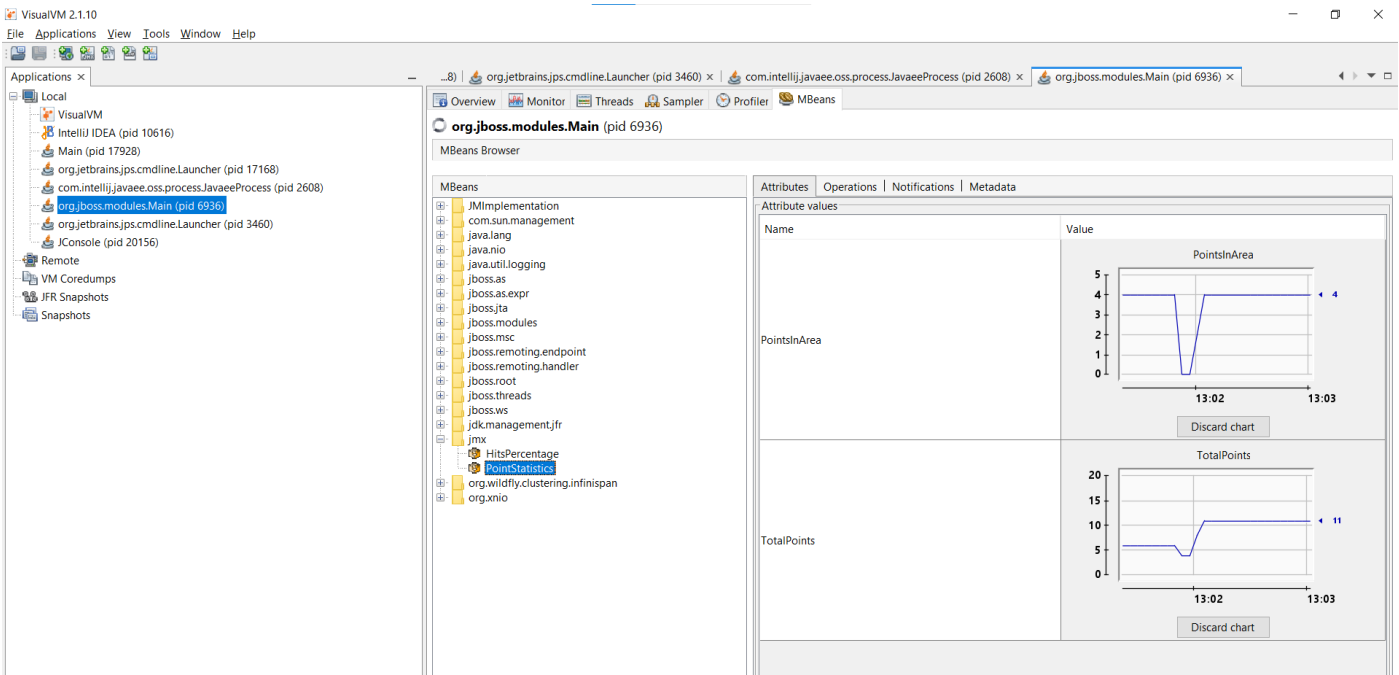
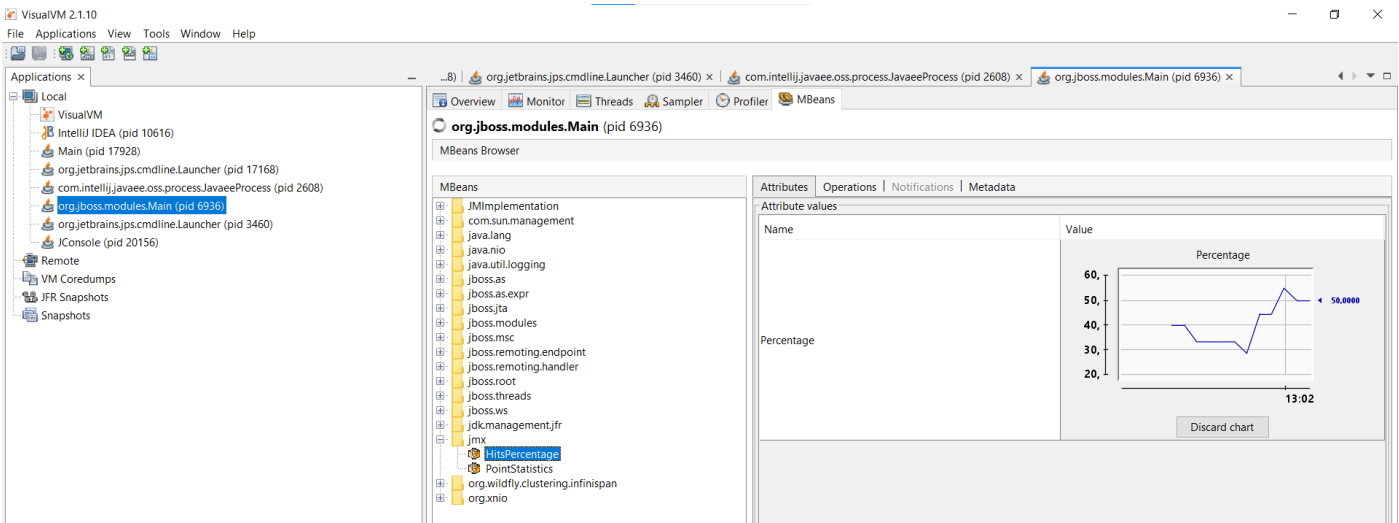
**VM arguments:** -Dprogram.name=standalone.bat -Dvisualvm.id=13782762528500 --add-exports=java.desktop/sun.awt=ALL-UNNAMED --add-exports=java.naming/com.sun.jndi.ldap=ALL-UNNAMED --add-exports=java.naming/com.sun.jndi.url.ldap=ALL-UNNAMED --add-exports=java.naming/com.sun.jndi.url.ldap=ALL-UNNAMED --add-exports=jdk.naming.dns/com.sun.jndi.dns=ALL-UNNAMED --add-opens=java.base/java.lang=ALL-UNNAMED --add-opens=java.base/java.lang.invoke=ALL-UNNAMED --add-opens=java.base/java.lang.reflect=ALL-UNNAMED --add-opens=java.base/java.io=ALL-UNNAMED --add-opens=java.base/java.net=ALL-UNNAMED --add-opens=java.base/java.security=ALL-UNNAMED --add-opens=java.base/java.util=ALL-UNNAMED --add-opens=java.base/java.util.concurrent=ALL-UNNAMED --add-opens=java.management/javax.management=ALL-UNNAMED --add-opens=java.naming/javax.naming=ALL-UNNAMED -Djava.security.manager=allow -Dorg.jboss.boot.log.file=C:\Users\Timur\wildfly\wildfly-30.0.0.Final\standalone\log\server.log -Dlogging.configuration=file:C:\Users\Timur\wildfly\wildfly-30.0.0.Final\standalone\configuration\logging.properties

**Class path:** C:\Users\Timur\wildfly\wildfly-30.0.0.Final\jboss-modules.jar

**Library path:** C:\Users\Timur\jdk\corretto-17.0.9\bin;C:\Windows\Sun\Java\bin;C:\Windows\system32;C:\Windows;C:\Program Files (x86)\Common Files\Oracle\Java\javapath;C:\Windows\system32;C:\Windows;C:\Windows\System32\Wbem;C:\Windows\System32\WindowsPowerShell\v1.0\;C:\Windows\System32\OpenSSH\;C:\Program Files (x86)\NVIDIA Corporation\PhysX\Common;C:\Program Files\NVIDIA Corporation\NVIDIA NvDLISR;C:\Program Files\dotnet\;C:\Program Files\nodejs\;C:\Program Files\TortoiseSVN\bin;C:\Program Files\Apache Subversion\bin;C:\Program Files\Git\cmd;C:\Users\Timur\AppData\Local\Programs\Python\Python311\Scripts\;C:\Users\Timur\AppData\Local\Programs\Python\Python311\;C:\Users\Timur\AppData\Roaming\Python\Scripts\;C:\Users\Timur\AppData\Local\Programs\Python\Python310\Scripts\;C:\Users\Timur\AppData\Local\Programs\Python\Python310\;C:\Users\Timur\AppData\Local\Microsoft\WindowsApps\;C:\Program Files\JetBrains\PyCharm Community Edition 2024.1.1\bin;;C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.1.2\bin;;C:\Users\Timur\AppData\Local\Programs\MiKTeX\miktex\bin\x64\;C:\Program Files\JetBrains\IntelliJ IDEA 2024.2.3\bin;;C:\Program Files\JetBrains\PyCharm 2024.1.1\bin;;C:\Users\Timur\AppData\Local\Programs\Microsoft VS Code\bin;C:\Users\Timur\AppData\Local\GitHubDesktop\bin;C:\Users\Timur\AppData\Roaming\npm;C:\Program Files\JetBrains\CLion 2024.1.1\bin;;C:\apache-ant-1.10.15\bin;C:\Program Files\Java\jdk-23\bin;;.

3 Пункт

График изменений показаний MBean-ов:



Имя класса, который занимает наибольший объем памяти:

Start Page

org.jetbrains.jps.cmdline.Launcher (pid 22872)

org.jetbrains.jps.cmdline.Launcher (pid 4136)

org.jboss.modules.Main (pid 23052)

Overview

Monitor

Threads

Sampler

Profiler

MBeans

org.jboss.modules.Main (pid 23052)

Sampler

Settings

Sample:

CPU

Memory

Stop

Status: memory sampling in progress

Heap histogram

Per thread allocations

Results: Collected data: Snapshot

Perform GC

Heap Dump

Name	Live Bytes	Live Objects
byte[]	36 777 248 B (16,3 %)	461 618 (8,4 %)
java.lang.Object[]	16 222 216 B (7,2 %)	400 096 (7,3 %)
java.util.HashMap\$Node[]	15 769 312 B (7 %)	182 355 (3,3 %)
java.util.LinkedHashMap\$Entry	13 821 520 B (6,1 %)	345 538 (6,3 %)
java.util.HashMap\$Node	13 062 784 B (5,8 %)	408 212 (7,4 %)
int[]	10 811 816 B (4,8 %)	24 600 (0,4 %)
org.jboss.dmr.ModelNode	9 909 504 B (4,4 %)	412 896 (7,5 %)

Start Page

org.jetbrains.jps.cmdline.Launcher (pid 22872)

org.jetbrains.jps.cmdline.Launcher (pid 4136)

org.jboss.modules.Main (pid 23052)

Overview

Monitor

Threads

Sampler

Profiler

MBeans

[heapdump] 13:18:20

org.jboss.modules.Main (pid 23052)

Heap Dump

Summary

Heap

Environment

Size: 87 141 776 B

Classes: 29 095

Instances: 1 897 349

Classloaders: 605

GC Roots: 6 911

Objects Pending for Finalization: 0

System

Architecture: amd64 64bit

Java Home: C:\Users\Timur\jdk\corretto-17.0.9

Java Version: 17.0.9 2023-10-17 LTS

Java Name: OpenJDK 64-Bit Server VM (17.0.9+8-LTS, mixed mode, sharing)

Java Vendor: Amazon.com Inc.

JVM Uptime: 1 min 42 sec

JVM Arguments [ show ]

Enabled Modules [ show ]

System Properties [ show ]

Classes by Number of Instances [ view all ]

Classes by Size of Instances [ view all ]

byte[] 315 847 (16,6 %)

java.lang.String 277 917 (14,6 %)

java.util.HashMap\$Node 235 102 (12,4 %)

java.lang.Object[] 69 674 (3,7 %)

java.util.ArrayList 43 047 (2,3 %)

byte[] 29 421 440 B (33,8 %)

java.util.HashMap\$Node 7 523 264 B (8,6 %)

java.lang.String 6 670 008 B (7,7 %)

java.lang.Object[] 3 941 664 B (4,5 %)

java.util.HashMap\$Node 3 272 672 B (3,8 %)

Instances by Size [ view all ]

Dominators by Retained Size [ view all ]

byte[]#64971: 736 776 B (0,8 %)

byte[]#64948: 601 940 B (0,7 %)

byte[]#65314: 354 839 B (0,4 %)

byte[]#42195: 332 200 B (0,4 %)

byte[]#41206: 295 207 B (0,3 %)

java.util.ArrayList#3180: 7 elements 5 689 816 B (6,5 %)

org.jboss.weld.resources.ClassTransformer#1 3 086 840 B (3,5 %)

org.jboss.as.controller.registry.ConcreteNotificationHandlerRegistr 1 547 832 B (1,8 %)

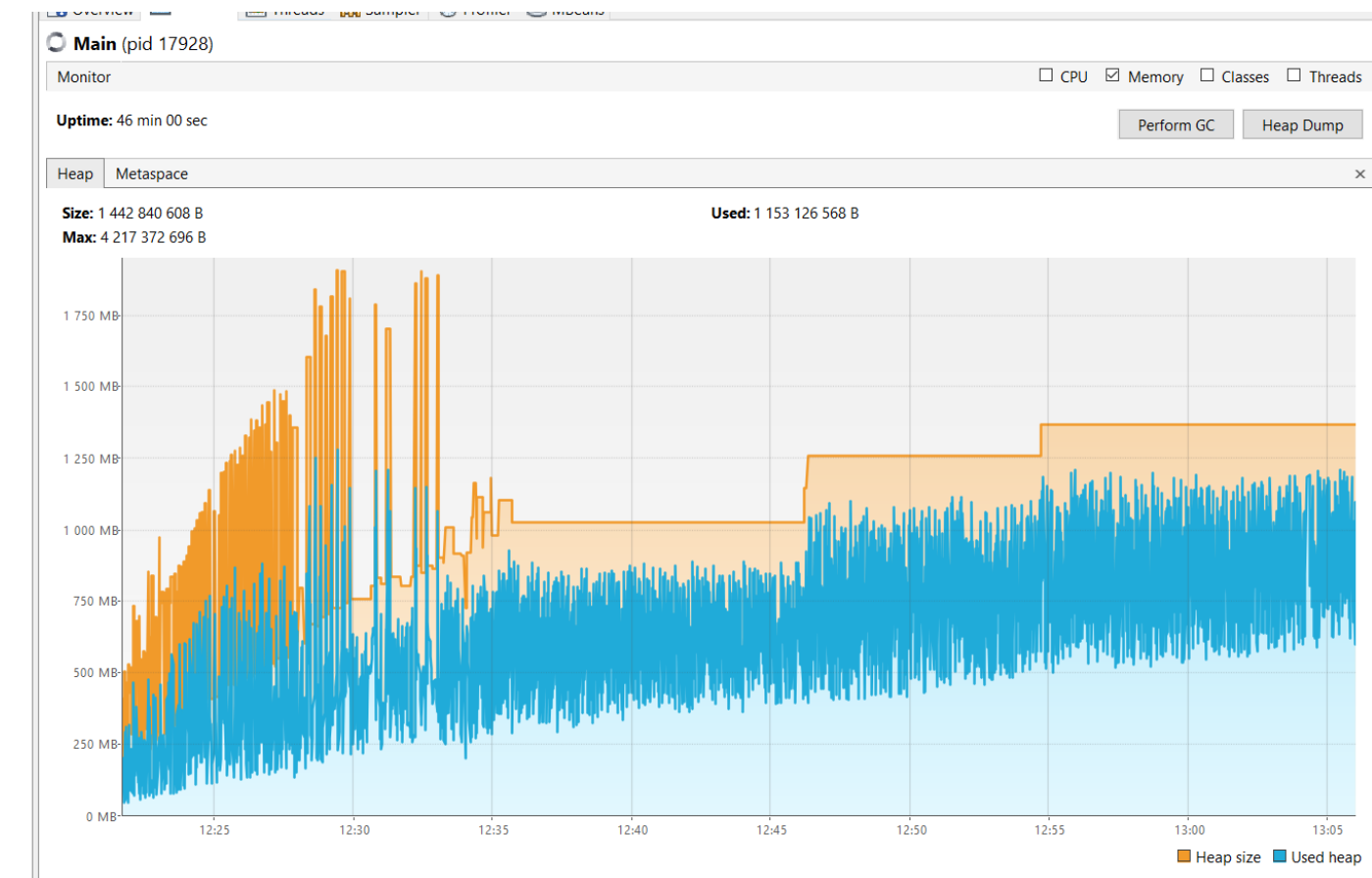
org.jboss.as.controller.registry.NodeSubregistry#213 1 083 848 B (1,2 %)

java.util.zip.ZipFile\$Source#140 833 744 B (1 %)



## 4 Пункт

Утечка данных:



String-и занимают очень много памяти из-за ошибок парсинга Js кода, который написан в HelloWorld.java:

[illegible]

Проблему можно решать по-разному: основная ошибка в том, что команды `wr` нет, у `document` можно вызвать только `write`. Исправим это:

```

    */
    public class HelloWorld extends HttpServlet {

        @Override no usages
        public void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
            PrintWriter out = response.getWriter();
            response.setContentType("text/html");
            out.println("<!DOCTYPE HTML PUBLIC \"-//W3C//DTD HTML 4.0 \" +
                \"Transitional//EN\"><\\n\" +
                \"<HTML>\\n\" +
                \"<HEAD><TITLE>Hello World</TITLE></HEAD>\\n\" +
                \"<BODY>\\n\" +
                \"<H1>Hello World</H1>\\n\");
            out.println("<script language=\"JavaScript\" type=\"text/javascript\">");
            out.println("<!--");
            out.println("document.write('Hello Document')"); // wr --> write
            out.println("//-->");
            out.println("</script>");
            out.println("</BODY></HTML>");
        }
    }
}
```

Ошибки больше не появляются:

Classes String								
Item		Shallow	Retained	Shortest Paths	Incoming References	Dominators	Retained Objects	Dominator Tree
> 'jdk.internal.reflect.DelegatingClassLoader @6a2a2a7b'	java.lang.String	24 B	0 B	v 'Point in time to dump data from, e.g. 09:00, 21:35:00, 2018-06-03T18:12:56.827Z, 2018-06-03T20:13:46.832Z, ...'				
> 'jdk.internal.reflect.DelegatingClassLoader @7603f8e1'	java.lang.String	24 B	0 B					
> 'jdk.internal.reflect.DelegatingClassLoader @16d633ea'	java.lang.String	24 B	0 B	> java.lang.Object[38]			4,63 kB	
> 'jdk.internal.reflect.GeneratedMethodAccessor10'	java.lang.String	24 B	0 B					
> 'objectInputFilter'	java.lang.String	24 B	0 B					
> 'jdk.internal.reflect.DelegatingClassLoader @7797c616'	java.lang.String	24 B	0 B					
> '()TT;'	java.lang.String	24 B	0 B					
> 'com.sun.jmx.remote.util.OrderClassLoaders'	java.lang.String	24 B	0 B					
> '()Ljava.lang.Object;'	java.lang.String	24 B	0 B					
> '()Ljava/lang/Class<Ljava/lang/Boolean;>'	java.lang.String	24 B	0 B					
> 'jdk.internal.reflect.DelegatingClassLoader @7f5551f4'	java.lang.String	24 B	0 B					
> '()Ljava/util/Optional<Ljava/lang/constant/DynamicConstantDesc<Ljava/lang/E		24 B	0 B					
> 'parseBoolean'	java.lang.String	24 B	0 B					
> 'logicalAnd'	java.lang.String	24 B	0 B					
> 'logicalOr'	java.lang.String	24 B	0 B					
> 'logicalXor'	java.lang.String	24 B	0 B					
> 'org.mozilla.javascript.SecurityController'	java.lang.String	24 B	0 B					
> 'com.meterware.servletunit.ServletUnitHttpSession'	java.lang.String	24 B	0 B					
> 'java.lang.reflect.Method'	java.lang.String	24 B	0 B					
> 'com.sun.management.internal.HotSpotDiagnostic\$\$Lambda\$116/0x00000188		24 B	0 B					