

Национальный исследовательский университет информационных технологий, механики и  
оптики  
Факультет ПИиКТ

## Методы и средства программной инженерии

### Лабораторная работа №4

Работу выполнили:  
Конаныхина Антонина  
Руденко Мария

Группа:  
Р3215  
Р3232

Вариант:  
5564

Преподаватель:  
Покид А.В.

Санкт-Петербург  
2022

## **Задание:**

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

- MBean, считающий общее число установленных пользователем точек, а также число точек, не попадающих в область. В случае, если координаты установленной пользователем точки вышли за пределы отображаемой области координатной плоскости, разработанный MBean должен отправлять оповещение об этом событии.
- MBean, определяющий средний интервал между кликами пользователя по координатной плоскости.

2. С помощью утилиты JConsole провести мониторинг программы:

- Снять показания MBean-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 MBean-классов, разработанных в ходе выполнения задания 1, с течением времени.
- Определить имя класса, объекты которого занимают наибольший объём 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 отчёт, 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.

## **Код:**

<https://github.com/tchn11/ITMO-labs/tree/main/2nd%20year/mispi/Lab4>

# JConsole:

Java Monitoring & Management Console - pid: 13240 msplab4.Main

Connection Window Help

Overview Memory Threads Classes VM Summary MBeans

+

 JMImplementation

+

 com.sun.management

+

 java.lang

+

 java.nio

+

 java.util.logging

+

 jdk.management.jfr

+

 msplab4.mbeans

+

 MidInterval

+

 Attributes

+

 Operations

+

 Notifications

+

 PointsCounter

Attribute values

Name	Value
MidInterval	8

Refresh

Java Monitoring & Management Console - pid: 13240 msplab4.Main

Connection Window Help

Overview Memory Threads Classes VM Summary MBeans

+

 JMImplementation

+

 com.sun.management

+

 java.lang

+

 java.nio

+

 java.util.logging

+

 jdk.management.jfr

+

 msplab4.mbeans

+

 MidInterval

+

 Attributes

+

 Operations

+

 Notifications

+

 PointsCounter

+

 Attributes

+

 Operations

+

 Notifications

Attribute values

Name	Value
SumAllPoint	6
SumMissedPoint	2

Refresh

Java Monitoring & Management Console - pid: 13240 mslab4.Main

Connection Window Help

Overview Memory Threads **Classes** VM Summary MBeans

Notification buffer

TimeStamp	Type	UserData	SeqNum	Message	Event	Source
19:23:56.767	false_point		3	you have false_point	Javafx.management.Notification...	mslab4.mbeans.type=PointsCou...

Subscribe Unsubscribe Clear

## Версия Java Language Specification:

Java Monitoring & Management Console - pid: 13240 mslab4.Main

Connection Window Help

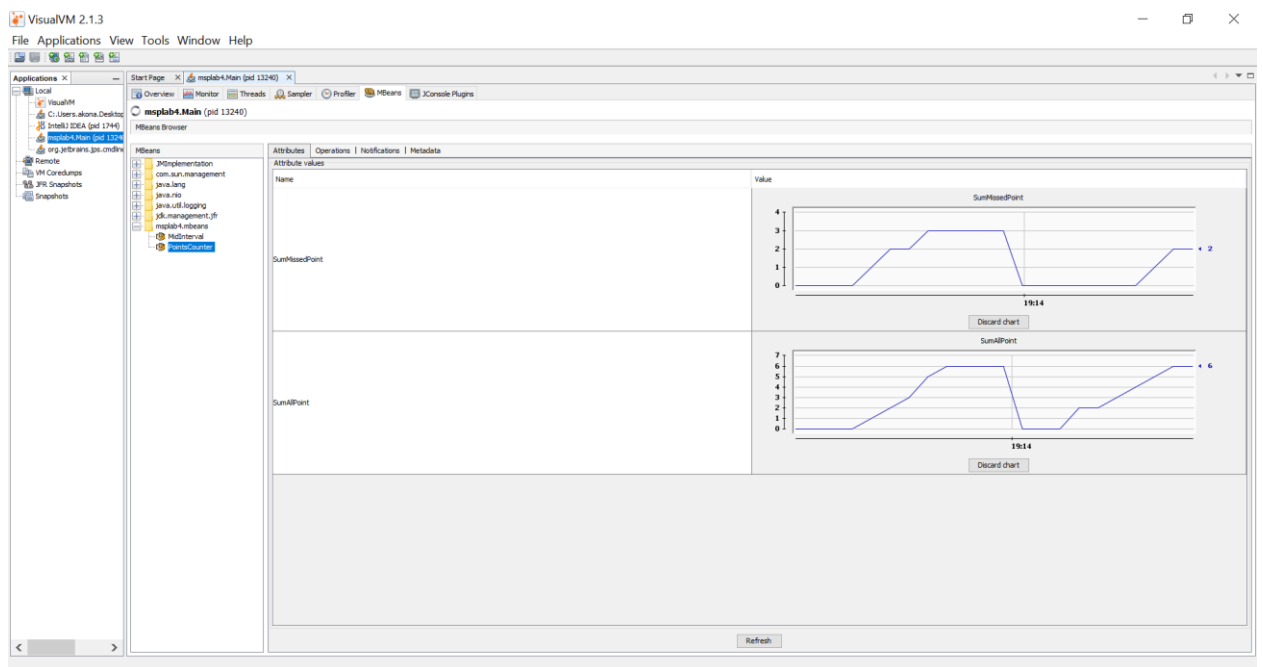
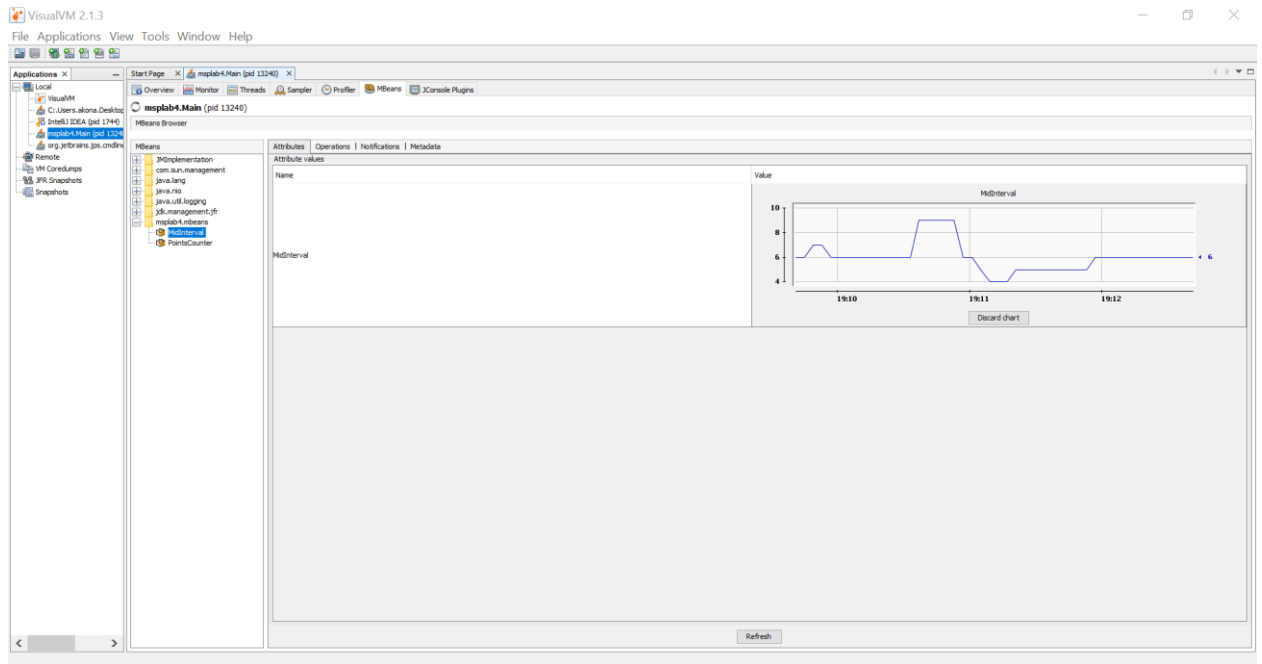
Overview Memory Threads Classes **VM Summary** MBeans

**VM Summary**  
воскресенье, 29 мая 2022 г., 19:31:25 Москва, стандартное время

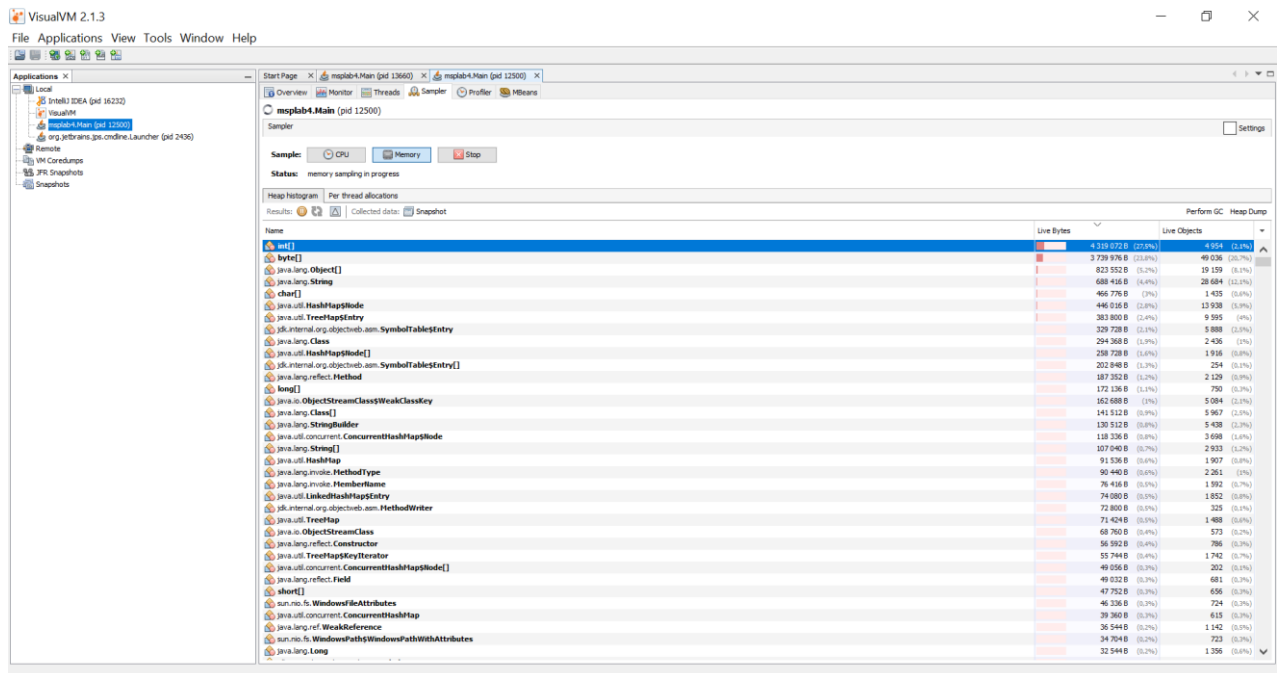
<b>Connection name:</b> pid: 13240 mslab4.Main		<b>Uptime:</b> 22 minutes	
<b>Virtual Machine:</b> OpenJDK 64-Bit Server VM version 16+36-2231		<b>Process CPU time:</b> 19,468 seconds	
<b>Vendor:</b> Oracle Corporation		<b>JIT compiler:</b> HotSpot 64-Bit Tiered Compilers	
<b>Name:</b> 13240@LAPTOP-INCM5JKU		<b>Total compile time:</b> 11,548 seconds	
<b>Live threads:</b> 17		<b>Current classes loaded:</b> 2 329	
<b>Peak:</b> 20		<b>Total classes loaded:</b> 2 329	
<b>Daemon threads:</b> 16		<b>Total classes unloaded:</b> 0	
<b>Total threads started:</b> 24			
<b>Current heap size:</b> 43 270 kbytes		<b>Committed memory:</b> 98 304 kbytes	
<b>Maximum heap size:</b> 1 558 528 kbytes		<b>Pending finalization:</b> 0 objects	
<b>Garbage collector:</b> Name = 'G1 Young Generation', Collections = 4, Total time spent = 0,026 seconds			
<b>Garbage collector:</b> Name = 'G1 Old Generation', Collections = 0, Total time spent = 0,000 seconds			
<b>Operating System:</b> Windows 10 10.0		<b>Total physical memory:</b> 6 228 524 kbytes	
<b>Architecture:</b> amd64		<b>Free physical memory:</b> 1 047 784 kbytes	
<b>Number of processors:</b> 8		<b>Total swap space:</b> 17 762 860 kbytes	
<b>Committed virtual memory:</b> 195 972 kbytes		<b>Free swap space:</b> 4 481 456 kbytes	
<b>VM arguments:</b> -javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2020.3.3\lib\idea_rt.jar=53527:C:\Program Files\JetBrains\IntelliJ IDEA 2020.3.3\bin -Dfile.encoding=UTF-8			
<b>Class path:</b> C:\Users\akona\Desktop\arropoli-sys\MrCIT\H.Msp4-console-main\out\production\Msp4-console-main			
<b>Library path:</b> C:\Users\akona\jdk\openjdk-16\bin;C:\WINDOWS\Sun\Java\bin;C:\WINDOWS\system32;C:\WINDOWS;C:\Program Files (x86)\Common Files\Oracle\Java\javapath;C:\Python310\Scripts;C:\Python310;C:\Program Files\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\Git\cmd;C:\Program Files\nodejs;C:\ProgramData\chocolatey\bin;C:\Program Files\Microsoft SQL Server\120\Tools\Binn;C:\Program Files\Microsoft SQL Server\130\Tools\Binn;C:\Program Files\TortoiseSVN\bin;C:			

# VisualVM:

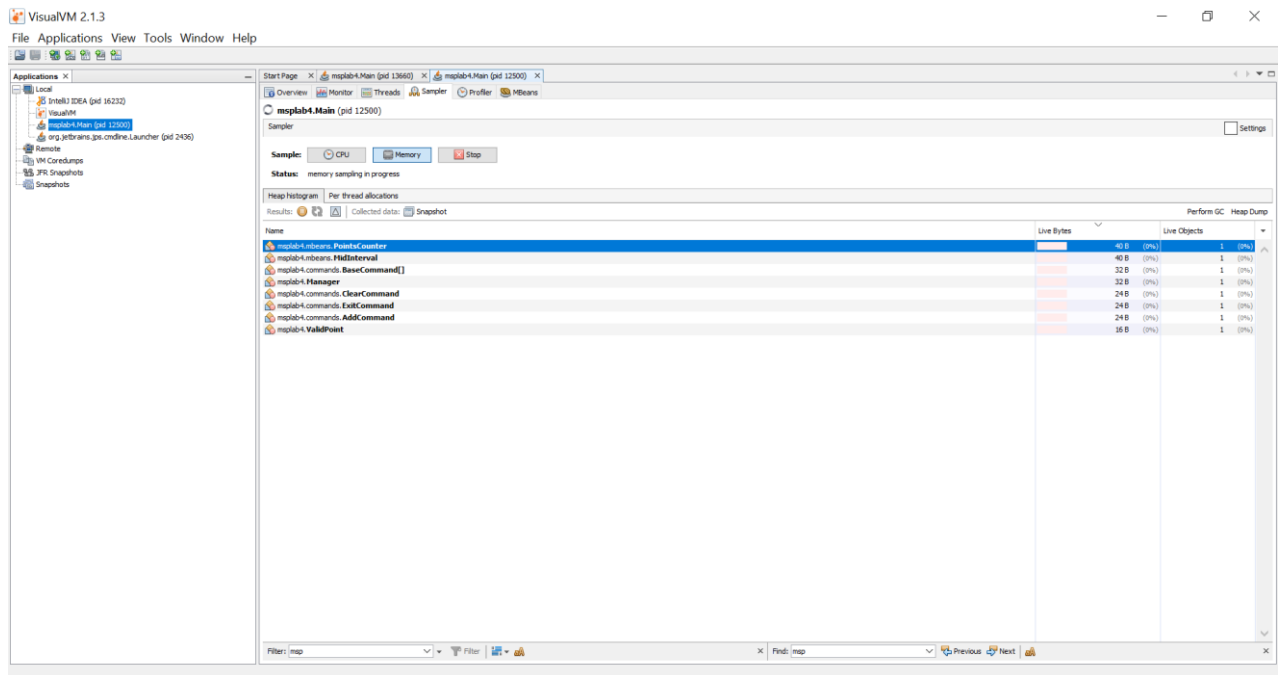
Графики изменения показаний МВеап-классов с течением времени:



Имя класса, объекты которого занимают наибольший объём памяти JVM:

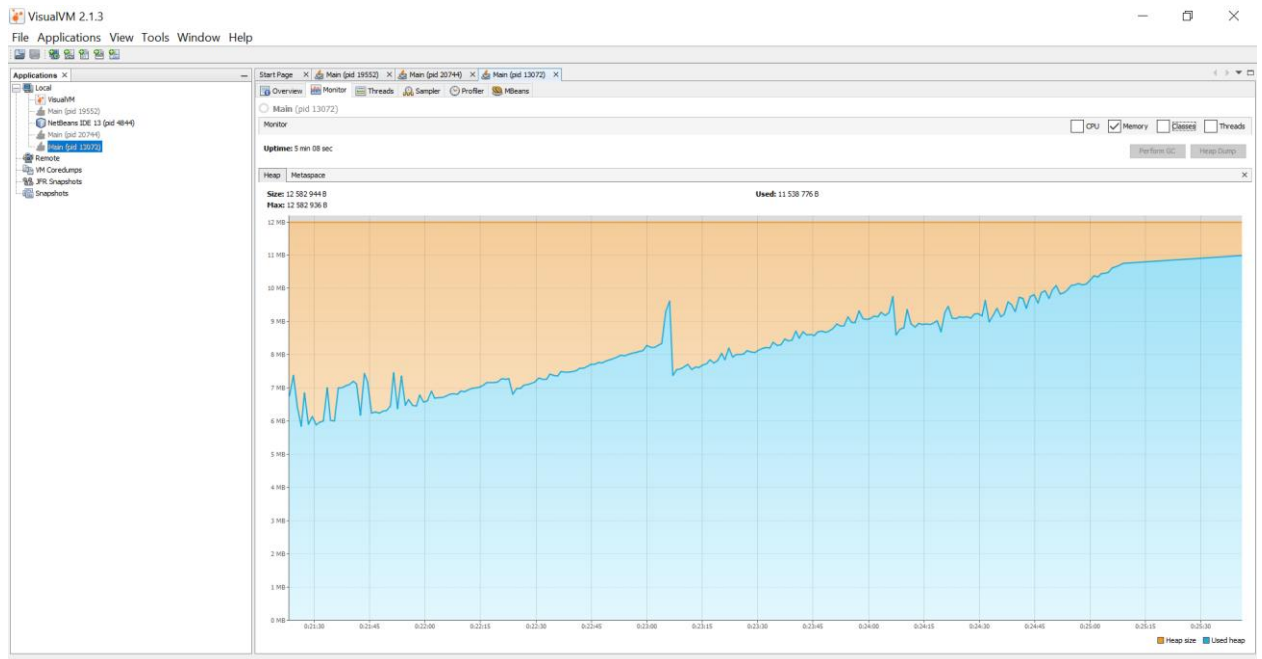


Пользовательский класс, в экземплярах которого находятся эти объекты:



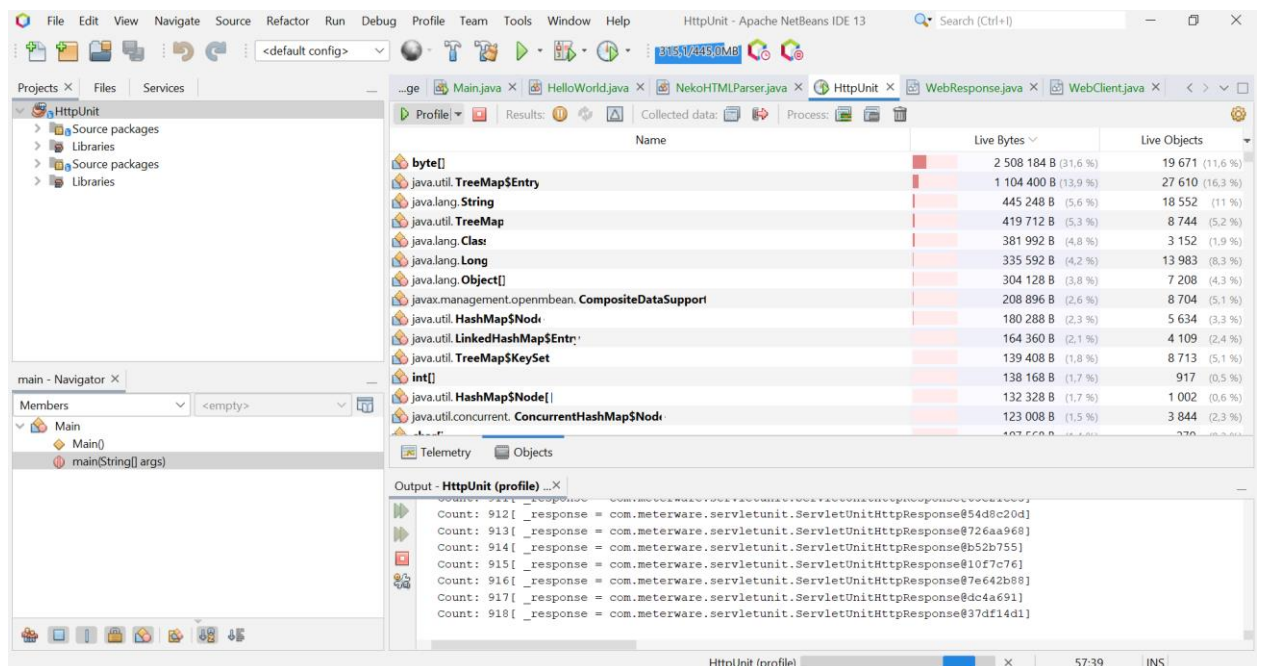
# Поиск проблем с производительностью:

## 1) Ищем проблему

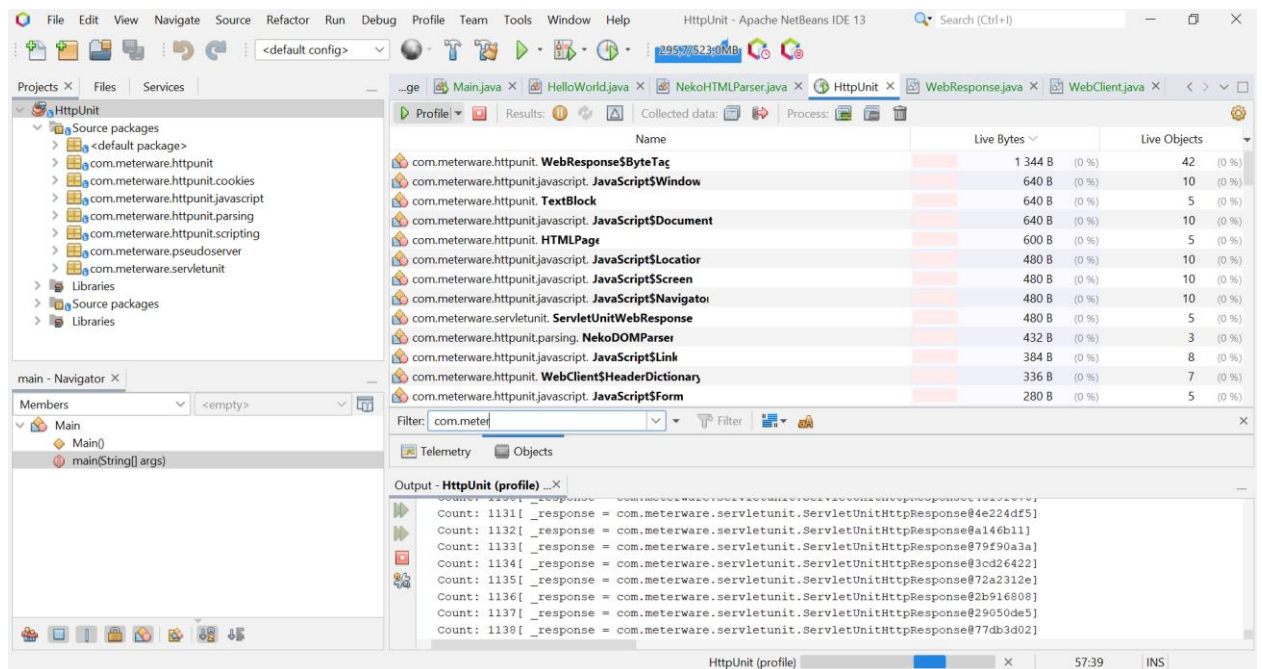


С помощью вкладки Monitor наблюдаем за использованием ресурсов. Как видим – размер кучи постоянно растет, и место под кучу почти кончилось. Размер кучи достаточно часто меняется.

## 2) Поиск источника проблемы



Видим, что больше всего места занимает примитив byte, поэтому будем искать класс с похожим поведением.



Отфильтруем по com.meterware., т.е. по нашему пакету. Видим, что объём памяти класса WebResponse постоянно растёт и удаляется.

### 3) Исправление ошибки.

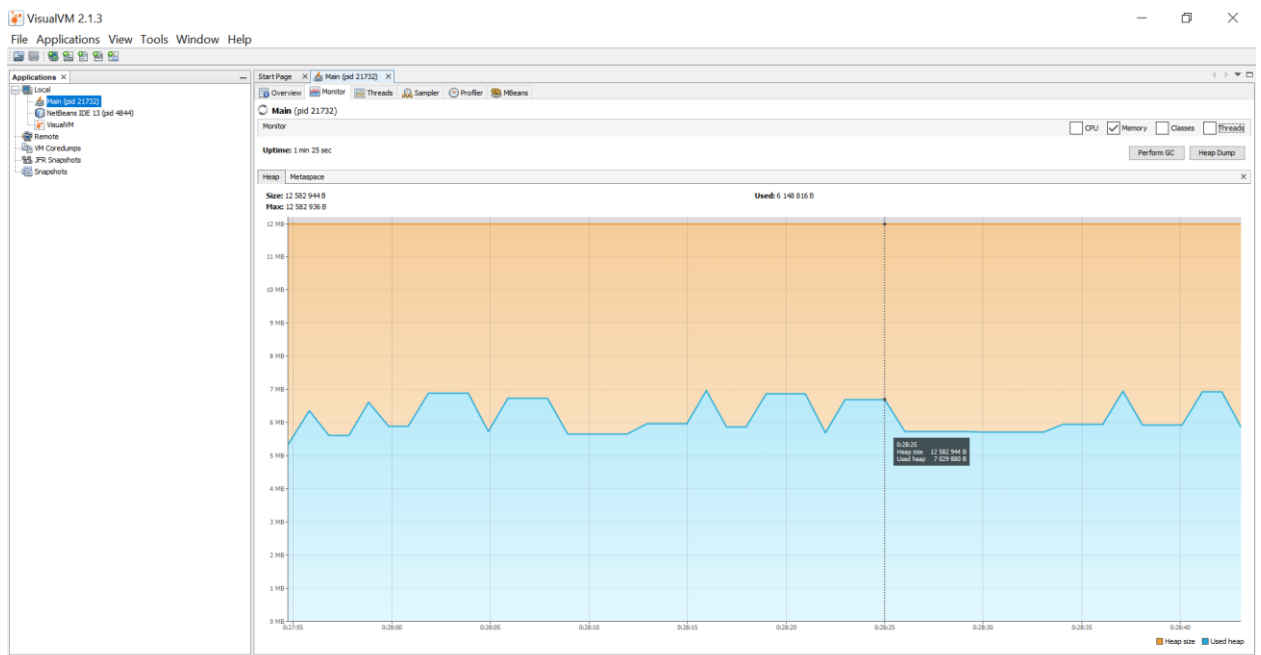
```
int number = 1;
WebRequest request = new GetMethodWebRequest("http://test.meterware.com/myServlet");
while (true) {
    WebResponse response = sc.getResponse(request);
    System.out.println("Count: " + number++ + response);
    java.lang.Thread.sleep(200);
}
```

В коде создаётся объект ответа с данными от сервера в бесконечном цикле while (true). Метод public void doGet(HttpServletRequest request, HttpServletResponse response) возвращает каждый раз одни и те же данные, которые можно "вернуть" всего один раз и не занимать лишнюю память, выделяя каждый раз место в куче для нового объекта ответа. Решение: можно переместить создание нового объекта из бесконечного цикла выше, чтобы объект создавался только один раз, и дальше обращение было к этой области памяти, а не каждый раз к вновь определённой.

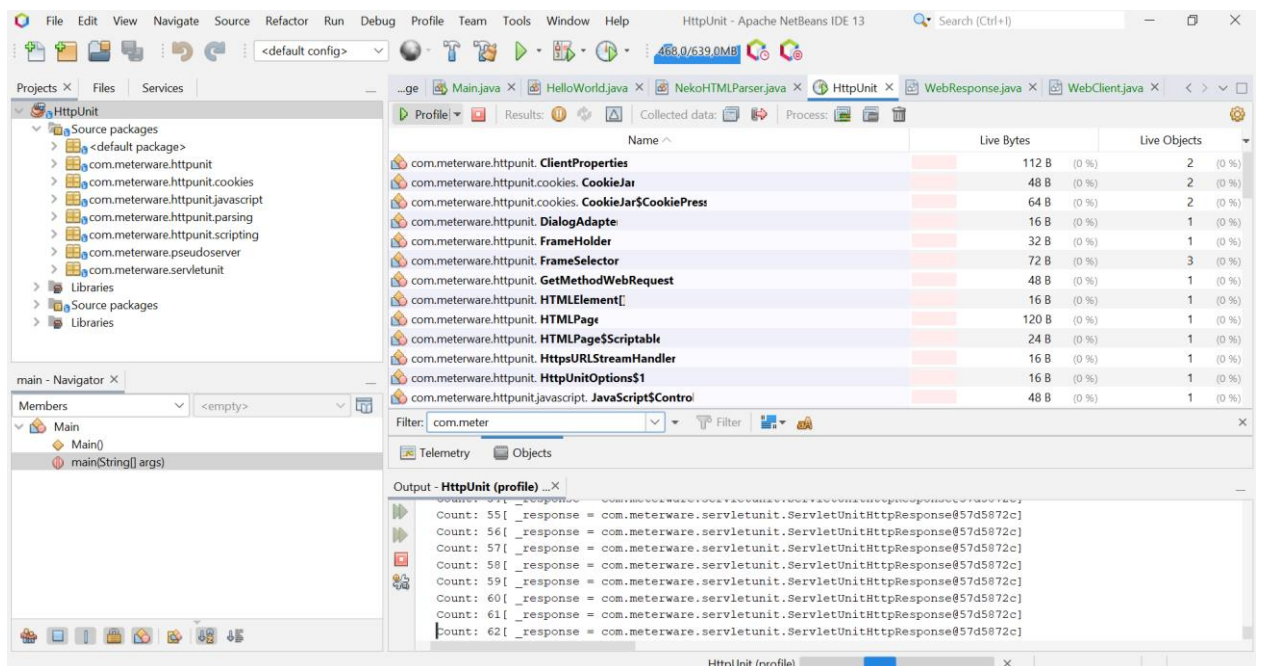
```
WebRequest request = new GetMethodWebRequest("http://test.meterware.com/myServlet");
WebResponse response = sc.getResponse(request);
while (true) {
    System.out.println("Count: " + number++ + response);
    java.lang.Thread.sleep(200);
}
```



#### 4) Проверка, что проблема устранена



Теперь все в порядке, размер кучи не растет.



Здесь количество объектов стало постоянным.

## Вывод:

В результате работы была изучена технология JMX, созданы MBeans для получения информации в JMX Agent (JConsole). В JConsole были получены метрики и информация о JVM. С помощью VisualVM и профилировщика NetBeans были выявлены и устранены проблемы в выданной по варианту программе.