

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Факультет программной инженерии и компьютерных технологий

Основы программной инженерии

**Лабораторная работа № 4**

Выполнил студент

Шнейдерис Герардас

Группа № Р3220

Преподаватель:

Егошин Алексей Васильевич

г. Санкт-Петербург

2025

Лабораторная работа № 1	1
<b>Задание:</b>	<b>3</b>
Software Requirements Specification	4
(Спецификация требований к продукту)	4
1.1 Definitions, Acronyms and Abbreviations (Определения и аббревиатуры)	4
2. Specific Requirements (Спецификация требований)	4
2.1 Functionality (Функциональные требования)	4
2.2 Usability (Требования к удобству использования)	5
2.3 Reliability (Требования к надежности)	5
2.4 Performance (Требования к производительности)	6
2.5 Security Requirements (Требования к безопасности)	6
3. Requirements Attributes (Требования к атрибутам)	7
4. Risks (риски)	9
5. Use-case (Диаграмма системы)	10
6. Precedents (Прецеденты)	11

**Вариант: 13**

**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, попадающих в область. В случае, если пользователь совершил 3 "промаха" подряд, разработанный MBean должен отправлять оповещение об этом событии.
- MBean, определяющий процентное отношение "промахов" к общему числу кликов пользователя по координатной плоскости.

**2. С помощью утилиты JConsole провести мониторинг программы:**

- Снять показания MBean-классов, разработанных в ходе выполнения задания 1.
- Определить время (в мс), прошедшее с момента запуска виртуальной машины.

**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 потока, потребляющего наибольший процент времени CPU.

**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 разработанных MBeans: интерфейс CounterMBean:

```
package lt.shgg.weblab4.mbeans

import java.util.concurrent.atomic.AtomicInteger

interface CounterMBean {

    val totalAttempts: AtomicInteger

    val totalHits: AtomicInteger

    fun checkMissStreak()

}
```

## класс Counter:

```
package lt.shgg.weblab4.mbeans

import org.springframework.jmx.export.annotation.*
import org.springframework.stereotype.Component
import java.io.Serializable
import java.util.concurrent.atomic.AtomicInteger
import javax.management.*

@Component
@ManagedResource("lt.shgg.weblab4.services:type=Counter")

class Counter : CounterMBean, Serializable, NotificationEmitter,
NotificationListener {

    @get:ManagedAttribute
```

```
override val totalAttempts = AtomicInteger(0)

@get:ManagedAttribute
override val totalHits = AtomicInteger(0)

private val missStreak = AtomicInteger(0)

private val broadcaster = NotificationBroadcasterSupport()

override fun getNotificationInfo():
Array<MBeanNotificationInfo?> {

    return arrayOfNulls(0)

}

@ManagedOperation
fun updateAttempt(hit: Boolean) {

    totalAttempts.incrementAndGet()

    if (hit) {

        totalHits.incrementAndGet()

        missStreak.set(0)

    } else {

        missStreak.incrementAndGet()

    }

    checkMissStreak()

}
```

```

override fun checkMissStreak() {

    if (missStreak.get() == 3) {

        broadcaster.sendNotification(Notification(

            "misses.streak",

            this,

            System.currentTimeMillis(),

            "damn, 3 misses in a row"

        ))

        missStreak.set(0)

    }

}

@Throws(ListenerNotFoundException::class)

override fun removeNotificationListener(

    listener: NotificationListener?,

    filter: NotificationFilter?,

    handback: Any?

) {

    broadcaster.removeNotificationListener(listener, filter,

handback)

}

override fun handleNotification(notification: Notification?,

handback: Any?) {

    println(notification)

```

```
}

    override fun addNotificationListener(listener:
NotificationListener?, filter: NotificationFilter?, handback:
Any?) {

        broadcaster.addNotificationListener(listener, filter,
handback)

    }

    @Throws(ListenerNotFoundException::class)

    override fun removeNotificationListener(listener:
NotificationListener?) {

        broadcaster.removeNotificationListener(listener)

    }

}
```

### интерфейс RatioStatMBean:

```
package lt.shgg.weblab4.mbeans

import java.util.concurrent.atomic.AtomicInteger

interface RatioStatMBean {

    val totalAttempts: AtomicInteger

    val totalMisses: AtomicInteger

    val missesRatio: Double

}
```

### класс RatioStat:

```
package lt.shgg.weblab4.mbeans

import org.springframework.jmx.export.annotation.ManagedAttribute
import org.springframework.jmx.export.annotation.ManagedOperation
import org.springframework.jmx.export.annotation.ManagedResource
import org.springframework.stereotype.Service
import java.io.Serializable
import java.util.concurrent.atomic.AtomicInteger

@ManagedResource("lt.shgg.weblab4.services:type=RatioStat")
@Service
class RatioStat : RatioStatMBean, Serializable {

    @get:ManagedAttribute
    override val totalAttempts: AtomicInteger = AtomicInteger(0)
```



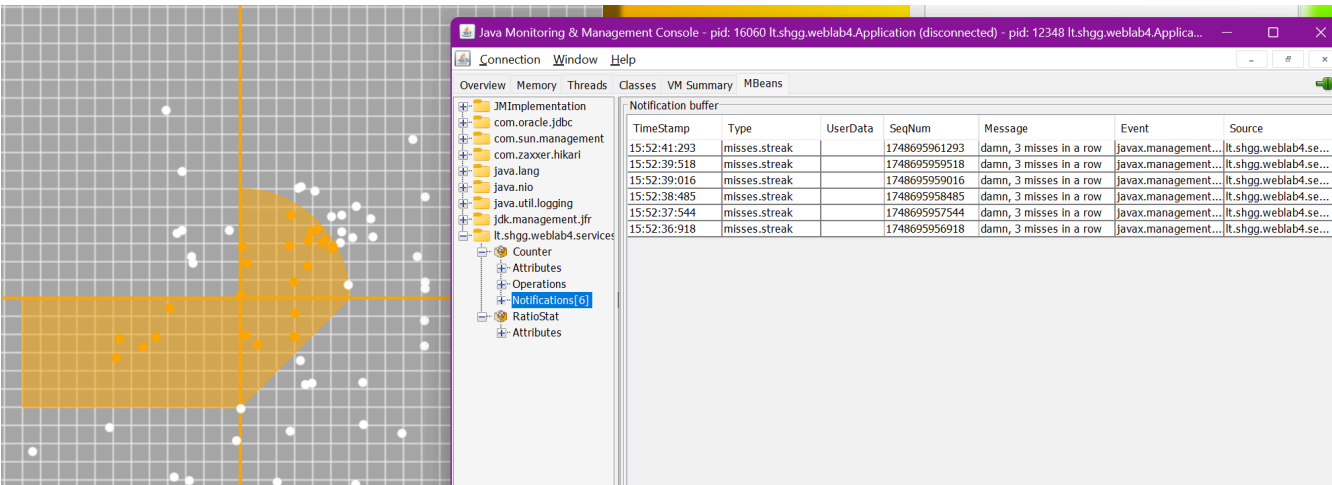
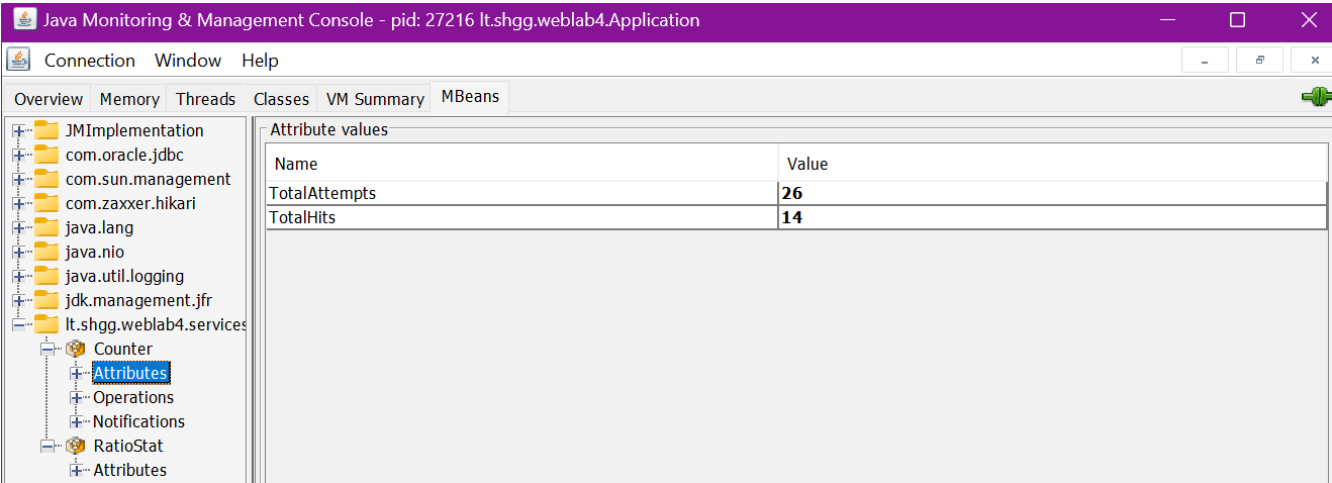
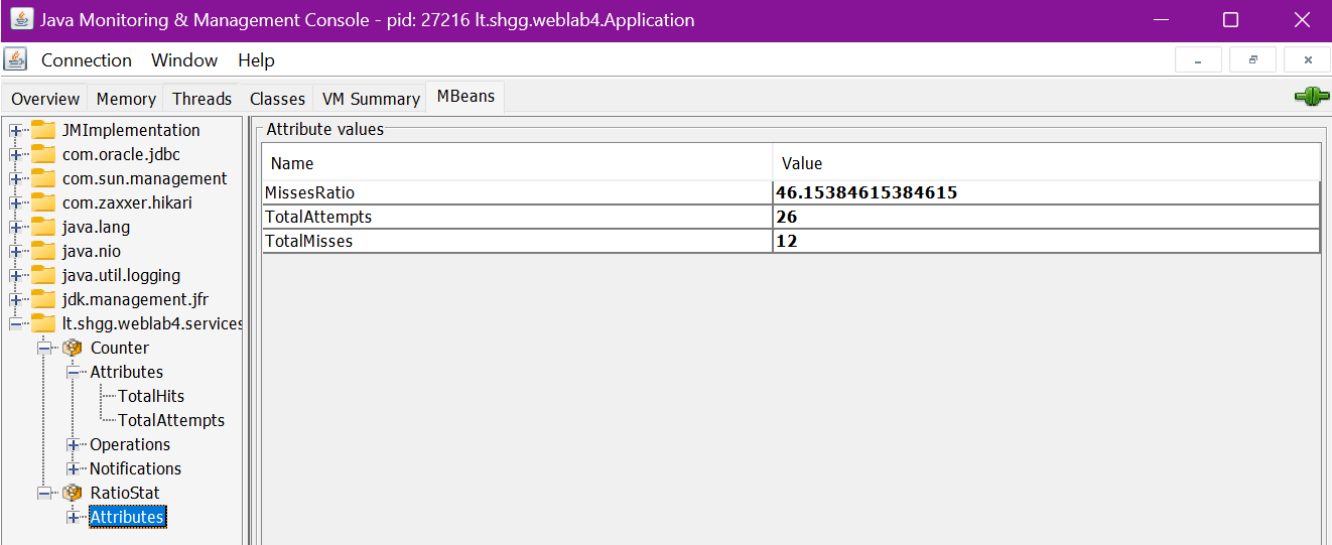
```
@get:ManagedAttribute
override val totalMisses: AtomicInteger = AtomicInteger(0)

@get:ManagedAttribute
override val missesRatio: Double
    get() = totalMisses.get().toDouble() / totalAttempts.get()
* 100

@ManagedOperation
fun updateStat(hit: Boolean) {
    totalAttempts.incrementAndGet()

    if (!hit) {
        totalMisses.incrementAndGet()
    }
}
}
```

Показания MBeans в JConsole:



Attribute value	
Name	Value
Uptime	<b>1091821</b>
<input type="button" value="Refresh"/>	

MBeanAttributeInfo	
Name	Value
Attribute:	
Name	Uptime
Description	Uptime
Readable	true
Writable	false
Is	false
Type	long

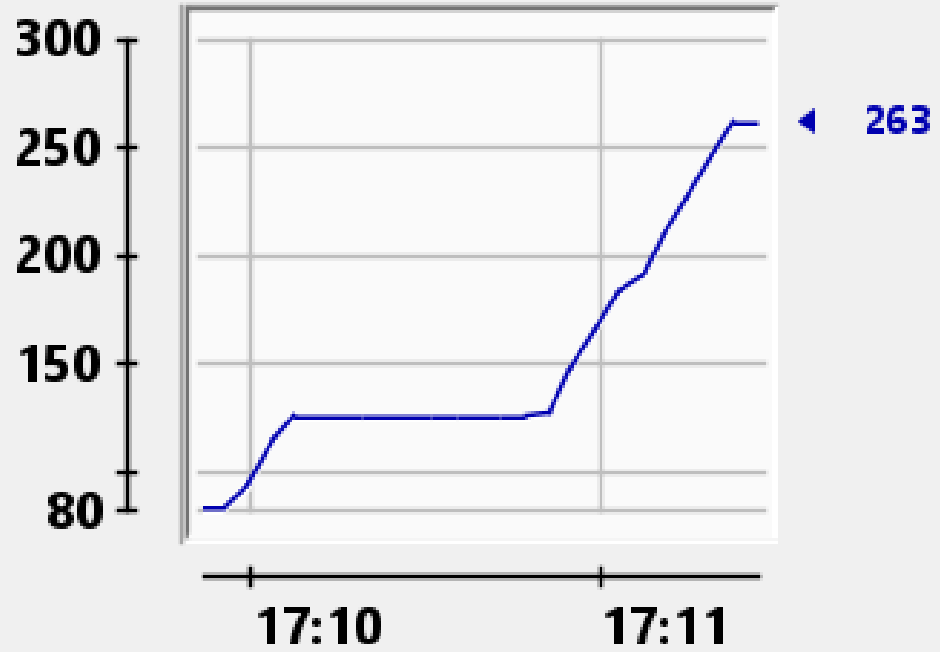
Descriptor	
Name	Value
Attribute:	
openType	javax.management.openmbean.SimpleType(name=java.lang.Long)
originalType	long

(аптайм тоже почему то попросили)

**Показания MBeans в VisualVM:**

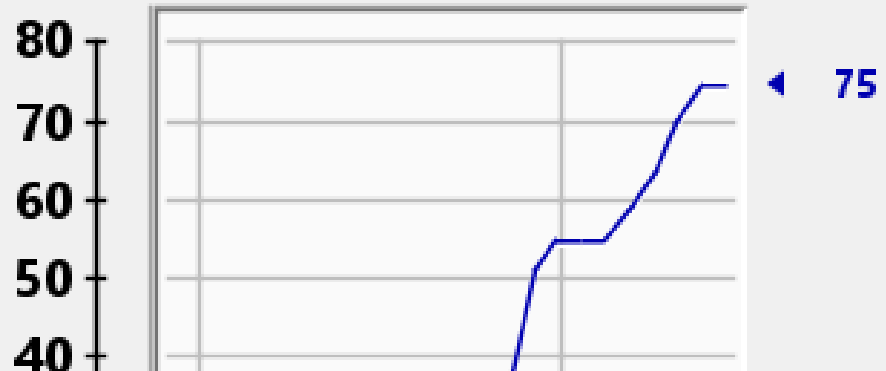
[38] | Metadata

Value



Discard chart

TotalHits



MBeans

+

 JMImplementation
 

+

 com.oracle.jdbc
 

+

 com.sun.management
 

+

 com.zaxxer.hikari
 

+

 java.lang
 

+

 java.nio
 

+

 java.util.logging
 

+

 jdk.management.jfr
 

+

 It.shgg.weblab4.services
 

+

 Counter
 

+

 RatioStat

Attributes | Operations | Notifications | Metadata

Attribute values

Name	Value
MissesRatio	<div> <div>MissesRatio</div> <div>69,8370</div> <div>Discard chart</div> </div>
TotalAttempts	<div> <div>TotalAttempts</div> <div>368</div> <div>Discard chart</div> </div>
TotalMisses	<div> <div>TotalMisses</div> <div>Attribute exposed for management</div> <div>257</div> <div>Discard chart</div> </div>

Refresh

Определить имя потока, потребляющего наибольший процент времени CPU.

It.shgg.weblab4.Application (pid 12348)

Sampler

Setting

Samp...

CPU

Memory

Stop

Status: CPU sampling in progress

CPU samples

Thread CPU time

Results:

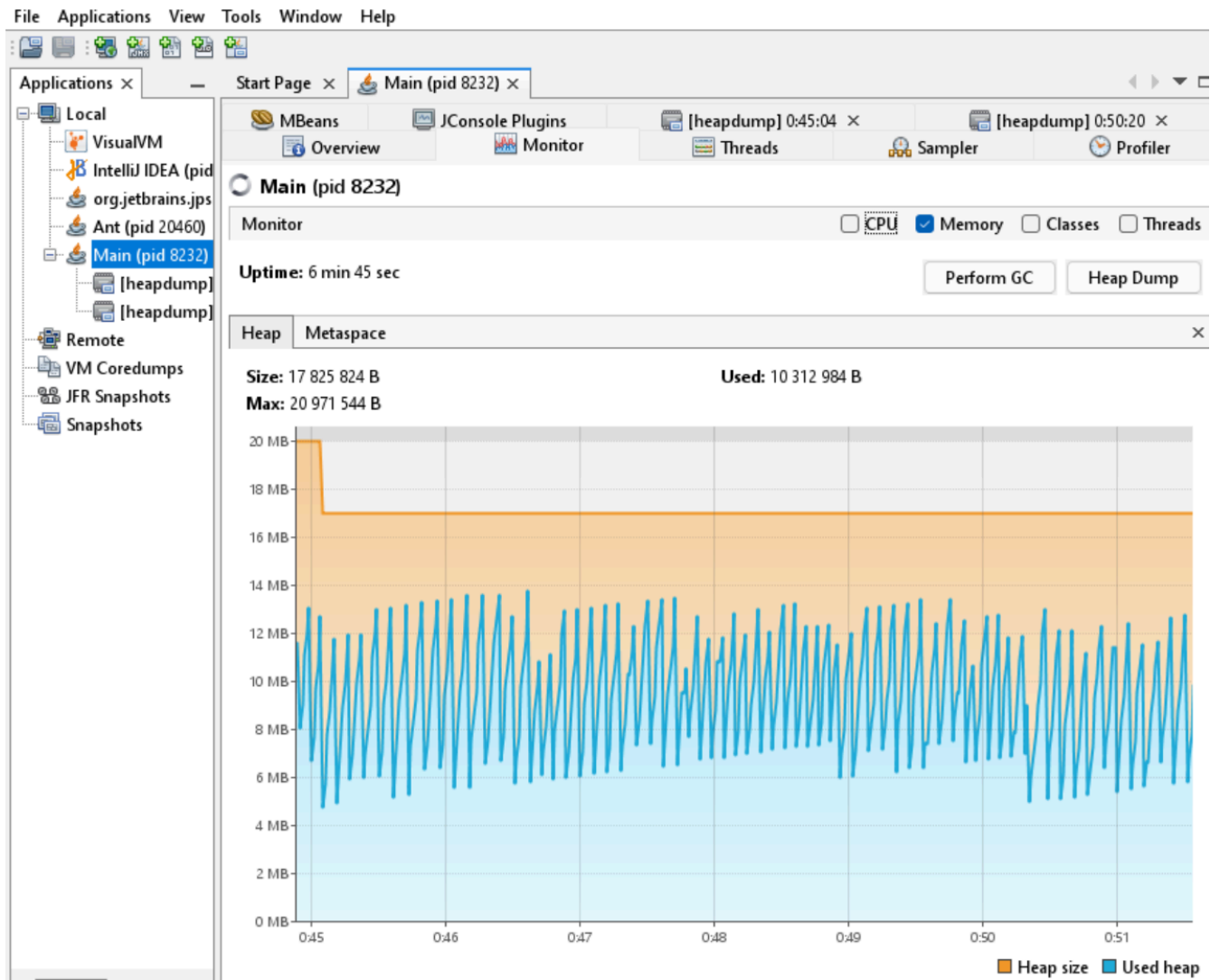
Statistics: Threads Count: 38 Total Time (CPU): 197 656 ms

Thread Dum

Name	Thread Time (CPU)	Thread Time (CPU) / sec
RMI TCP Connection(31)-192.168.0.105	<div></div> 192 859 ms (97,6 %)	<div></div> 109 ms (10,9 %)
DestroyJavaVM	<div></div> 1 734 ms (0,9 %)	<div></div> 0,0 ms (0 %)
RMI TCP Connection(3)-192.168.0.105	<div></div> 546 ms (0,3 %)	<div></div> 0,0 ms (0 %)
RMI TCP Connection(2)-192.168.0.105	<div></div> 500 ms (0,3 %)	<div></div> 0,0 ms (0 %)
JFR Periodic Tasks	<div></div> 234 ms (0,1 %)	<div></div> 0,0 ms (0 %)
RMI TCP Connection(36)-192.168.0.105	<div></div> 203 ms (0,1 %)	<div></div> 15,6 ms (1,6 %)
http-nio-8080-exec-1	<div></div> 171 ms (0,1 %)	<div></div> 0,0 ms (0 %)
http-nio-8080-exec-10	<div></div> 171 ms (0,1 %)	<div></div> 0,0 ms (0 %)

## Локализовать и устранить проблемы с производительностью в HttpUnit

При запуске VisualVM видим проблемы с памятью:



падает(



делаем heap dump и видим большое количество объектов String:

Applications x — Start Page x Main (pid 8232) x

Overview Monitor Threads Sampler  
Profiler MBeans JConsole Plugins [heapdump] 0:50:20 x

Local  
VisualVM  
IntelliJ IDEA (pid  
org.jetbrains.jps  
Ant (pid 20460)  
Main (pid 8232)  
[heapdump]  
[heapdump]  
[heapdump]  
Remote  
VM CoreDumps  
JFR Snapshots  
Snapshots

Main (pid 8232)

Heap Dump

Objects Preset: All Objects Aggregation: Details: Preview Fields References GC Root

Name	Count	Size	Retained (sort to get)
byte[]	22 769 (19,4%)	1 513 816 B (31,6%)	n/a
java.lang.String	21 882 (18,6%)	525 168 B (11%)	n/a
java.util.TreeMap\$Entry	6 343 (5,4%)	253 720 B (5,3%)	n/a
java.util.concurrent.ConcurrentHashMap\$Node	5 514 (4,7%)	176 448 B (3,7%)	n/a
java.lang.Object[]	5 241 (4,5%)	280 368 B (5,9%)	n/a
java.util.HashMap\$Node	3 662 (3,1%)	117 184 B (2,4%)	n/a
java.lang.Long	3 472 (3%)	83 328 B (1,7%)	n/a
java.util.ImmutableCollections\$List12	2 201 (1,9%)	52 824 B (1,1%)	n/a
java.util.TreeMap	1 985 (1,7%)	95 280 B (2%)	n/a
java.util.LinkedHashMap\$Entry	1 977 (1,7%)	79 080 B (1,7%)	n/a
java.util.TreeMap\$KeySet	1 950 (1,7%)	31 200 B (0,7%)	n/a
javax.management.openmbean.CompositeData	1 938 (1,6%)	46 512 B (1%)	n/a
jdk.jfr.AnnotationElement	1 460 (1,2%)	35 040 B (0,7%)	n/a
jdk.jfr.internal.AnnotationConstruct	1 411 (1,2%)	33 864 B (0,7%)	n/a
java.lang.Object	1 384 (1,2%)	22 144 B (0,5%)	n/a
java.lang.String[]	1 324 (1,1%)	43 560 B (0,9%)	n/a
java.lang.Class[]	1 288 (1,1%)	35 168 B (0,7%)	n/a
java.lang.reflect.Method	1 274 (1,1%)	112 112 B (2,3%)	n/a
java.util.Collections\$UnmodifiableRandomAccess	1 179 (1%)	28 296 B (0,6%)	n/a
java.lang.invoke.MemberName	1 090 (0,9%)	43 600 B (0,9%)	n/a

Если пролистать их в конец видим очень много одинаковых:

Name	Size	Retained (sort to get)
java.lang.String#21094 : Script 'document.wr('Hello Document')/' failed: TypeError: undefined is not a function. (httpunit; )	24 B (0%)	n/a
java.lang.String#21095 : Script 'document.wr('Hello Document')/' failed: TypeError: undefined is not a function. (httpunit; )	24 B (0%)	n/a
java.lang.String#21096 : Script 'document.wr('Hello Document')/' failed: TypeError: undefined is not a function. (httpunit; )	24 B (0%)	n/a
java.lang.String#21097 : Script 'document.wr('Hello Document')/' failed: TypeError: undefined is not a function. (httpunit; )	24 B (0%)	n/a
java.lang.String#21098 : Script 'document.wr('Hello Document')/' failed: TypeError: undefined is not a function. (httpunit; )	24 B (0%)	n/a
java.lang.String#21099 : Script 'document.wr('Hello Document')/' failed: TypeError: undefined is not a function. (httpunit; )	24 B (0%)	n/a
java.lang.String#21100 : Script 'document.wr('Hello Document')/' failed: TypeError: undefined is not a function. (httpunit; )	24 B (0%)	n/a
java.lang.String#21101 : Script 'document.wr('Hello Document')/' failed: TypeError: undefined is not a function. (httpunit; )	24 B (0%)	n/a
java.lang.String#21102 : Script 'document.wr('Hello Document')/' failed: TypeError: undefined is not a function. (httpunit; )	24 B (0%)	n/a
java.lang.String#21103 : Script 'document.wr('Hello Document')/' failed: TypeError: undefined is not a function. (httpunit; )	24 B (0%)	n/a
java.lang.String#21104 : Script 'document.wr('Hello Document')/' failed: TypeError: undefined is not a function. (httpunit; )	24 B (0%)	n/a
java.lang.String#21105 : Script 'document.wr('Hello Document')/' failed: TypeError: undefined is not a function. (httpunit; )	24 B (0%)	n/a
java.lang.String#21106 : Script 'document.wr('Hello Document')/' failed: TypeError: undefined is not a function. (httpunit; )	24 B (0%)	n/a
java.lang.String#21107 : Script 'document.wr('Hello Document')/' failed: TypeError: undefined is not a function. (httpunit; )	24 B (0%)	n/a
java.lang.String#21108 : Script 'document.wr('Hello Document')/' failed: TypeError: undefined is not a function. (httpunit; )	24 B (0%)	n/a
java.lang.String#21109 : Script 'document.wr('Hello Document')/' failed: TypeError: undefined is not a function. (httpunit; )	24 B (0%)	n/a
java.lang.String#21110 : Script 'document.wr('Hello Document')/' failed: TypeError: undefined is not a function. (httpunit; )	24 B (0%)	n/a
java.lang.String#21111 : Script 'document.wr('Hello Document')/' failed: TypeError: undefined is not a function. (httpunit; )	24 B (0%)	n/a
java.lang.String#21112 : Script 'document.wr('Hello Document')/' failed: TypeError: undefined is not a function. (httpunit; )	24 B (0%)	n/a
java.lang.String#21113 : Script 'document.wr('Hello Document')/' failed: TypeError: undefined is not a function. (httpunit; )	24 B (0%)	n/a
java.lang.String#21114 : Script 'document.wr('Hello Document')/' failed: TypeError: undefined is not a function. (httpunit; )	24 B (0%)	n/a
java.lang.String#21115 : Script 'document.wr('Hello Document')/' failed: TypeError: undefined is not a function. (httpunit; )	24 B (0%)	n/a
java.lang.String#21116 : Script 'document.wr('Hello Document')/' failed: TypeError: undefined is not a function. (httpunit; )	24 B (0%)	n/a

Узнаем откуда они:

```
[1142] in java.lang.Object[]#5035 : 1 851 items
elementData in java.util.ArrayList#7 : 1 626 elements
static_errorMessages in class com.meterware.httpunit.javascript.JavaScript : JavaScript
```

Находим в соответствующем файле:

```
static void clearErrorMessages() {  
    _errorMessages.clear();  
}
```

Переходим несколько раз по **usage** и находим в HttpUnitOptions

```
public static void clearScriptErrorMessages() {  
    getScriptingEngine().clearErrorMessages();  
}
```

И у этого метода нет **usages** а значит строки нигде не удаляются и это нехорошо

## Пути решения:

- отказаться от хранения ошибок (ну а зачем)
- хранить последние n ошибок (например n = 100)

для этого изменим объявление `_errorMessages`

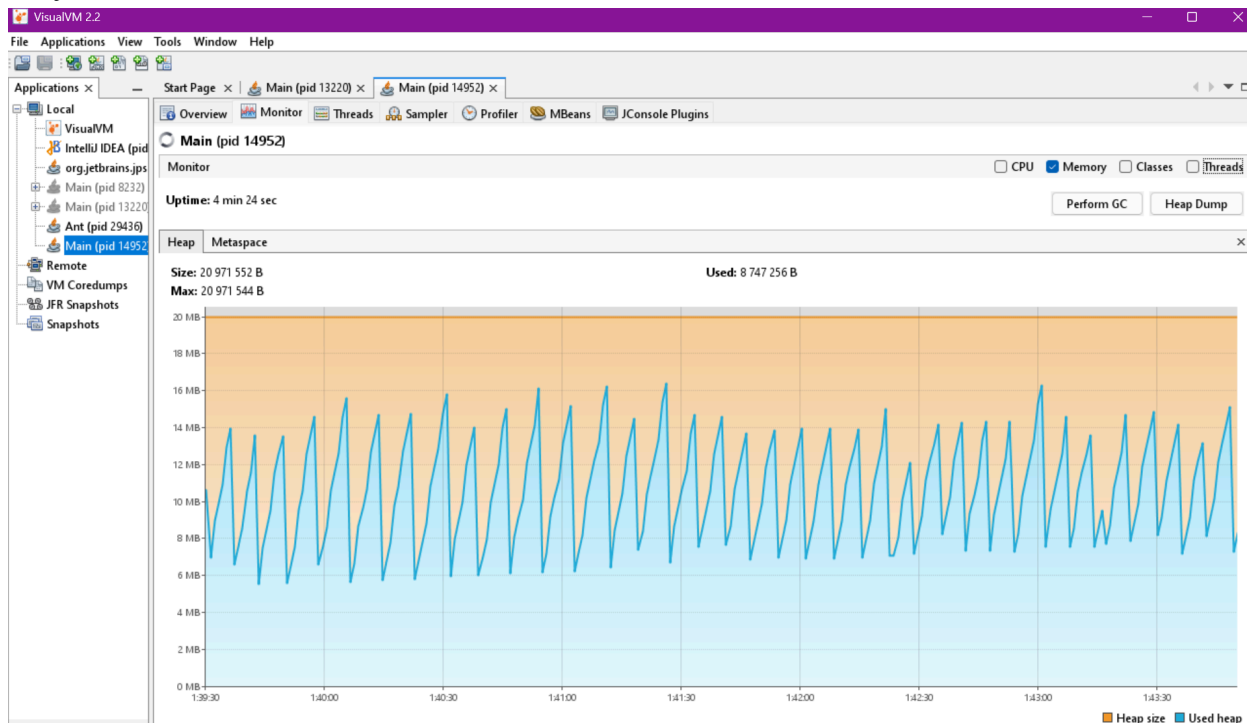
```
private static ArrayList _errorMessages = new ArrayList(100);
```

- очищать ошибки после обработки каждого запроса

для этого надо добавить одну строчку в Main

```
while (true) {  
  
    WebResponse response = sc.getResponse(request);  
  
    System.out.println("Count: " + number++ + response);  
  
    HttpUnitOptions.clearScriptErrorMessages();  
  
    java.lang.Thread.sleep(200);  
}
```

получим:



в heap dump строк тоже стало меньше ура победа