

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: **264999**

Преподаватель:
Кулинич Ярослав Вадимович

Выполнил: Нодири Хисравхон
Группа: Р3231

1. Текст задания

Лабораторная работа #4

Вариант 264999

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, не 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 "попаданий" 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 наименование и версию JVM, поставщика виртуальной машины Java и номер её сборки.

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:

<https://github.com/Kuchizu/ItmoLabs>

2. Исходные коды разработанных MBean:

- Beans.AttemptStatsMBean.java

```
package io.github.Lab3.beans;

public interface AttemptStatsMBean {
    int getTotalAttempts();
    int getTotalHits();
}
```

- Beans.AttemptStats.java

```
package io.github.Lab3.beans;

import jakarta.enterprise.context.SessionScoped;
import jakarta.faces.application.FacesMessage;
import jakarta.faces.context.FacesContext;
import jakarta.inject.Named;

import javax.management.*;
import java.io.Serializable;
import java.util.concurrent.atomic.AtomicInteger;

@Named("attemptStats")
@SessionScoped
public class AttemptStats implements AttemptStatsMBean,
NotificationBroadcaster, Serializable {

    private final AtomicInteger totalAttempts = new AtomicInteger(0);
    private final AtomicInteger totalHits = new AtomicInteger(0);

    private final NotificationBroadcasterSupport broadcaster = new
NotificationBroadcasterSupport();

    @Override
    public int getTotalAttempts() {
        return totalAttempts.get();
    }

    @Override
    public int getTotalHits() {
        return totalHits.get();
    }

    public void updateAttempt(double x, double y, boolean hit) {
        System.out.println("HIT = " + hit);
        totalAttempts.incrementAndGet();
        if (hit) {
            totalHits.incrementAndGet();
        }

        if(Math.abs(x) > 7 || Math.abs(y) > 7){
            System.out.println("MESSAGE");
            FacesContext.getCurrentInstance().addMessage(null, new
FacesMessage(FacesMessage.SEVERITY_WARN, "Beyond of area", "Coordinates are
beyond of the displayed area."));
        }
    }
}
```

```

    }
}

public void clearAttempts(){
    totalAttempts.set(0);
    totalHits.set(0);
}

@Override
public void addNotificationListener(NotificationListener listener,
NotificationFilter filter, Object handback) throws IllegalArgumentException {
    broadcaster.addNotificationListener(listener, filter, handback);
}

@Override
public void removeNotificationListener(NotificationListener listener)
throws ListenerNotFoundException {
    broadcaster.removeNotificationListener(listener);
}

@Override
public MBeanNotificationInfo[] getNotificationInfo() {
    String[] types = new String[] { "coordinates.error" };
    String name = Notification.class.getName();
    String description = "Notification sent when x > 7 or y > 7";
    return new MBeanNotificationInfo[] { new MBeanNotificationInfo(types,
name, description) };
}
}

```

- Beans.HitRatioMBean.java

```

package io.github.Lab3.beans;

public interface HitRatioMBean {
    int getTotalAttempts();
    int getTotalHits();
    double getHitRatio();
}

```

- Beans.HitRatio.java

```

package io.github.Lab3.beans;

import jakarta.enterprise.context.SessionScoped;
import jakarta.inject.Named;

import java.io.Serializable;
import java.util.concurrent.atomic.AtomicInteger;

@Named("hitRatio")
@SessionScoped
public class HitRatio implements HitRatioMBean, Serializable {
    private final AtomicInteger totalAttempts = new AtomicInteger(0);
    private final AtomicInteger totalHits = new AtomicInteger(0);

    @Override
    public int getTotalAttempts() {
        return totalAttempts.get();
    }

    @Override
    public int getTotalHits() {
        return totalHits.get();
    }
}

```

```
@Override
public double getHitRatio() {
    if (totalAttempts.get() == 0) {
        return 0.0;
    }
    return (double) totalHits.get() / totalAttempts.get() * 100.0;
}

public void updateStats(boolean hit) {
    totalAttempts.incrementAndGet();
    if (hit) {
        totalHits.incrementAndGet();
    }
}

public void clearAttempts(){
    totalAttempts.set(0);
    totalHits.set(0);
}
}
```

3. JConsole

Lab-3

localhost:8080/Lab-4/main.xhtml

Lab-3

By Nodiri Khisrovkhon, P3231

Variant N. 51928588

Enter parameters:

Choose X:

-3

-2

-1

0

1

2

3

Enter Y:

Should be between -5 and 5

Choose R:

4

Total Ts: 20

Total Hits: 6

Hit percentage: 30.0%

Enable graph aiming

Clear table

Return to index page

Click

X	Y	R	Result	Executed at	Execution time
1.2561112060546868	-1.3253334045410163	3.0	Hit / Success	21/06/2024 21:34:09	7200
-1.2918887939453123	-0.457333404541016	3.0	Hit / Success	21/06/2024 21:34:09	1600
1.004111206054688	0.550666595458984	3.0	Hit / Success	21/06/2024 21:34:09	1600
2.964111206054687	-0.06533340454101566	3.0	Hit / Success	21/06/2024 21:34:55	2600
1.8161112060546873	0.6626665954589841	3.0	Hit / Success	21/06/2024 21:34:56	1400
3.020111206054688	2.622666595458984	3.0	Miss / Fail	21/06/2024 21:34:56	1400

Java Monitoring & Management Console - pid: 15124 jboss-modules.jar -mp C:\Users\Kuchizu\Downloads\WIL...

Overview Memory Threads Classes VM Summary MBeans

Attribute values

Name	Value
HitRatio	30.0
TotalAttempts	20
TotalHits	6

Refresh

Lab-3

localhost:8080/Lab-4/main.xhtml

Lab-3

By Nodiri Khisrovkhon, P3231

Variant N. 51928588

Enter parameters:

Choose X:

-3

-2

-1

0

1

2

3

Enter Y:

Should be between -5 and 5

Choose R:

4

Total Ts: 20

Total Hits: 6

Hit percentage: 30.0%

Enable graph aiming

Clear table

Return to index page

Click

X	Y	R	Result	Executed at	Execution time
1.2561112060546868	-1.3253334045410163	3.0	Hit / Success	21/06/2024 21:34:09	7200
-1.2918887939453123	-0.457333404541016	3.0	Hit / Success	21/06/2024 21:34:09	1600
1.004111206054688	0.550666595458984	3.0	Hit / Success	21/06/2024 21:34:09	1600
2.964111206054687	-0.06533340454101566	3.0	Hit / Success	21/06/2024 21:34:55	2600
1.8161112060546873	0.6626665954589841	3.0	Hit / Success	21/06/2024 21:34:56	1400
3.020111206054688	2.622666595458984	3.0	Miss / Fail	21/06/2024 21:34:56	1400

Java Monitoring & Management Console - pid: 15124 jboss-modules.jar -mp C:\Users\Kuchizu\Downloads\WIL...

Overview Memory Threads Classes VM Summary MBeans

Attribute values

Name	Value
HitRatio	30.0
TotalAttempts	20
TotalHits	6

Refresh

Lab-3
By Nodiri Khisrovkhon, P3231
Variant N. 51928588

Enter parameters:
Choose X: ☐ -3 ☐ -2 ☐ -1 ☐ 0 ☐ 1 ☐ 2 ☐ 3
Enter Y: Should be between -5 and 5
Choose R:

Total Ts: 20
Total Hits: 6
Hit percentage: 30.0%

[Enable graph aiming](#)
[Clear table](#)
[Return to index page](#)

X	Y	R	Result	Executed at	Execution time
1.2561112060546868	-1.3253334045410163	3.0	Hit / Success	21/06/2024 21:34:09	7200
-1.2918887939453123	-0.457333404541016	3.0	Hit / Success	21/06/2024 21:34:09	1600
1.004111206054688	0.550666595458984	3.0	Hit / Success	21/06/2024 21:34:09	1600
2.964111206054687	-0.06533340454101566	3.0	Hit / Success	21/06/2024 21:34:55	2600
1.8161112060546873	0.6626665954589841	3.0	Hit / Success	21/06/2024 21:34:56	1400
3.020111206054688	2.622666595458984	3.0	Miss / Fail	21/06/2024 21:34:56	1400

Java Monitoring & Management Console - pid: 15124 jboss-modules.jar -mp C:\Users\Kuchizu\Downloads\WIL...

Overview Memory Threads Classes VM Summary MBeans

MBeanInfo

Name	Value
Info:	
ObjectName	io.github.Lab3.beans:type=HitRatio,name=hitRatio
ClassName	io.github.Lab3.beans.HitRatio
Description	Information on the management interface of the MBean
Constructor-0:	
Name	io.github.Lab3.beans.HitRatio
Description	Public constructor of the MBean

Descriptor

Name	Value
Info:	
immutableInfo	true
interfaceClassName	io.github.Lab3.beans.HitRatioMBean
mbean	false

Lab-3
By Nodiri Khisrovkhon, P3231
Variant N. 51928588

Enter parameters:
Choose X: ☐ -3 ☐ -2 ☐ -1 ☐ 0 ☐ 1 ☐ 2 ☐ 3
Enter Y: Should be between -5 and 5
Choose R:

Total Ts: 20
Total Hits: 6
Hit percentage: 30.0%

[Enable graph aiming](#)
[Clear table](#)
[Return to index page](#)

X	Y	R	Result	Executed at	Execution time
1.2561112060546868	-1.3253334045410163	3.0	Hit / Success	21/06/2024 21:34:09	7200
-1.2918887939453123	-0.457333404541016	3.0	Hit / Success	21/06/2024 21:34:09	1600
1.004111206054688	0.550666595458984	3.0	Hit / Success	21/06/2024 21:34:09	1600
2.964111206054687	-0.06533340454101566	3.0	Hit / Success	21/06/2024 21:34:55	2600
1.8161112060546873	0.6626665954589841	3.0	Hit / Success	21/06/2024 21:34:56	1400
3.020111206054688	2.622666595458984	3.0	Miss / Fail	21/06/2024 21:34:56	1400

Java Monitoring & Management Console - pid: 15124 jboss-modules.jar -mp C:\Users\Kuchizu\Downloads\WIL...

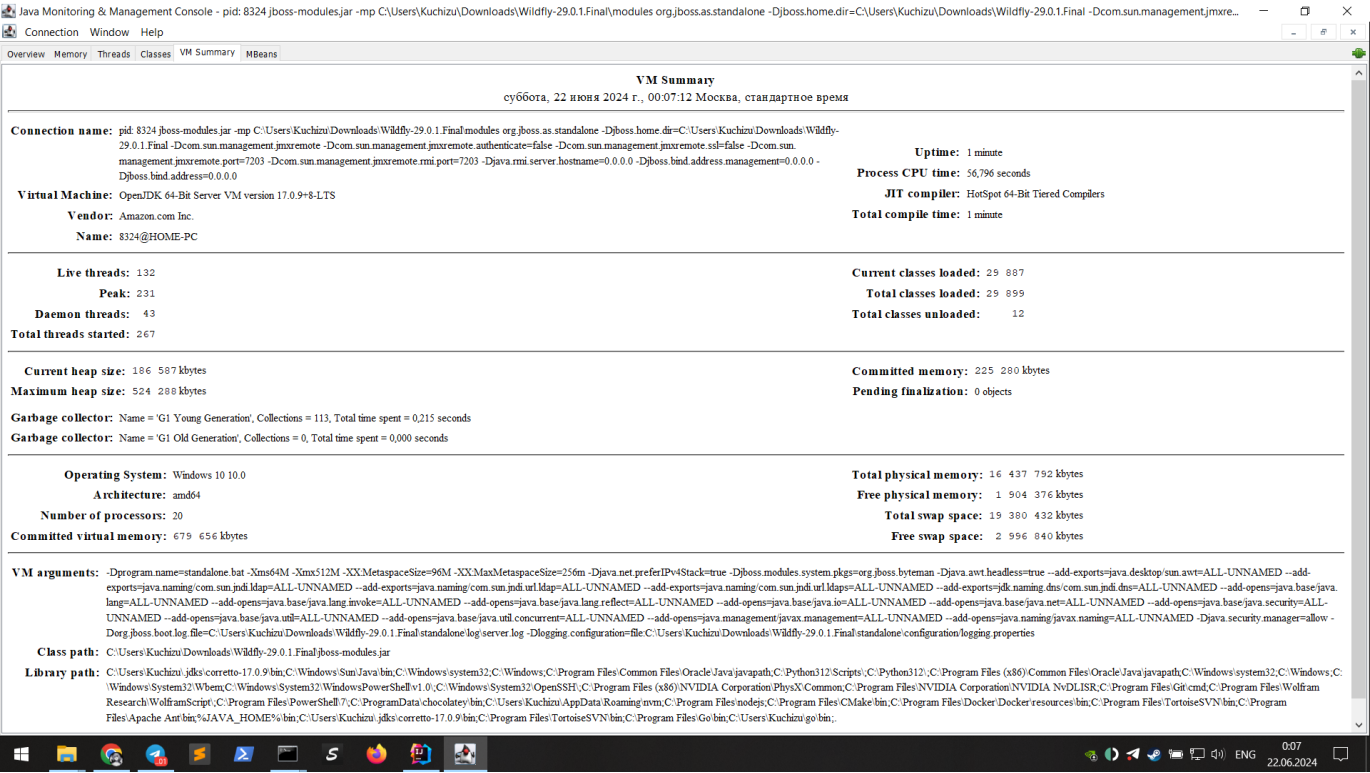
Overview Memory Threads Classes VM Summary MBeans

MBeanInfo

Name	Value
Info:	
ObjectName	io.github.Lab3.beans:type=AttemptStats,name=attemptStats
ClassName	io.github.Lab3.beans.AttemptStats
Description	Information on the management interface of the MBean
Constructor-0:	
Name	io.github.Lab3.beans.AttemptStats
Description	Public constructor of the MBean

Descriptor

Name	Value
Info:	
immutableInfo	false
interfaceClassName	io.github.Lab3.beans.AttemptStatsMBean
mbean	false

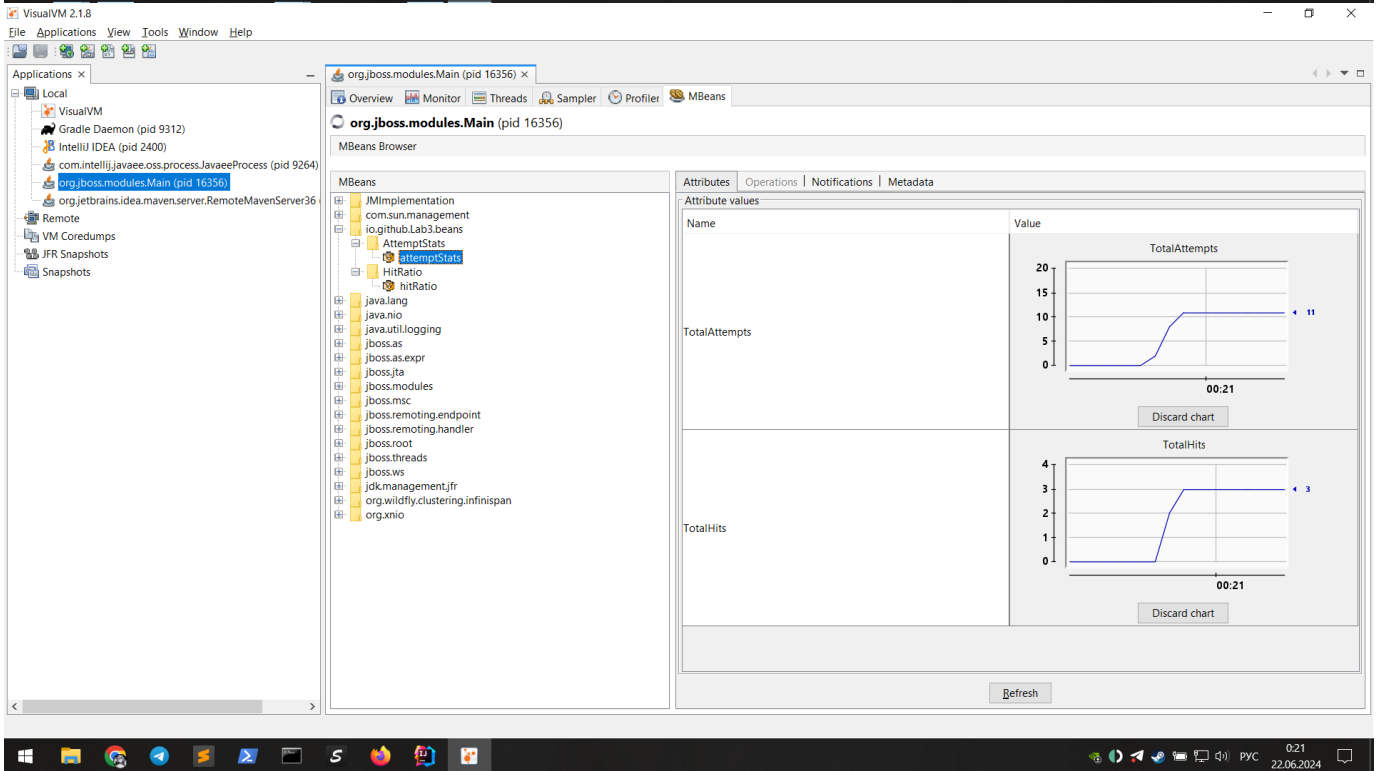
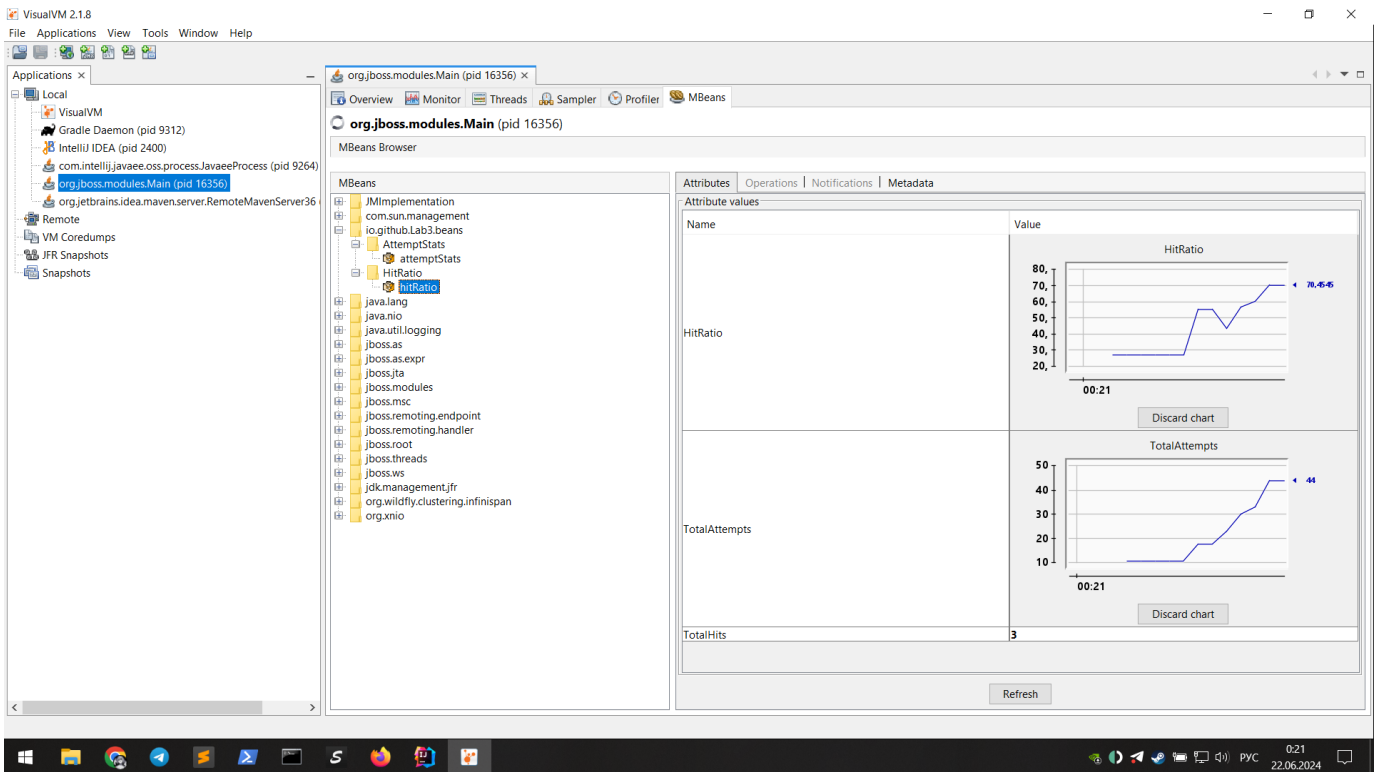


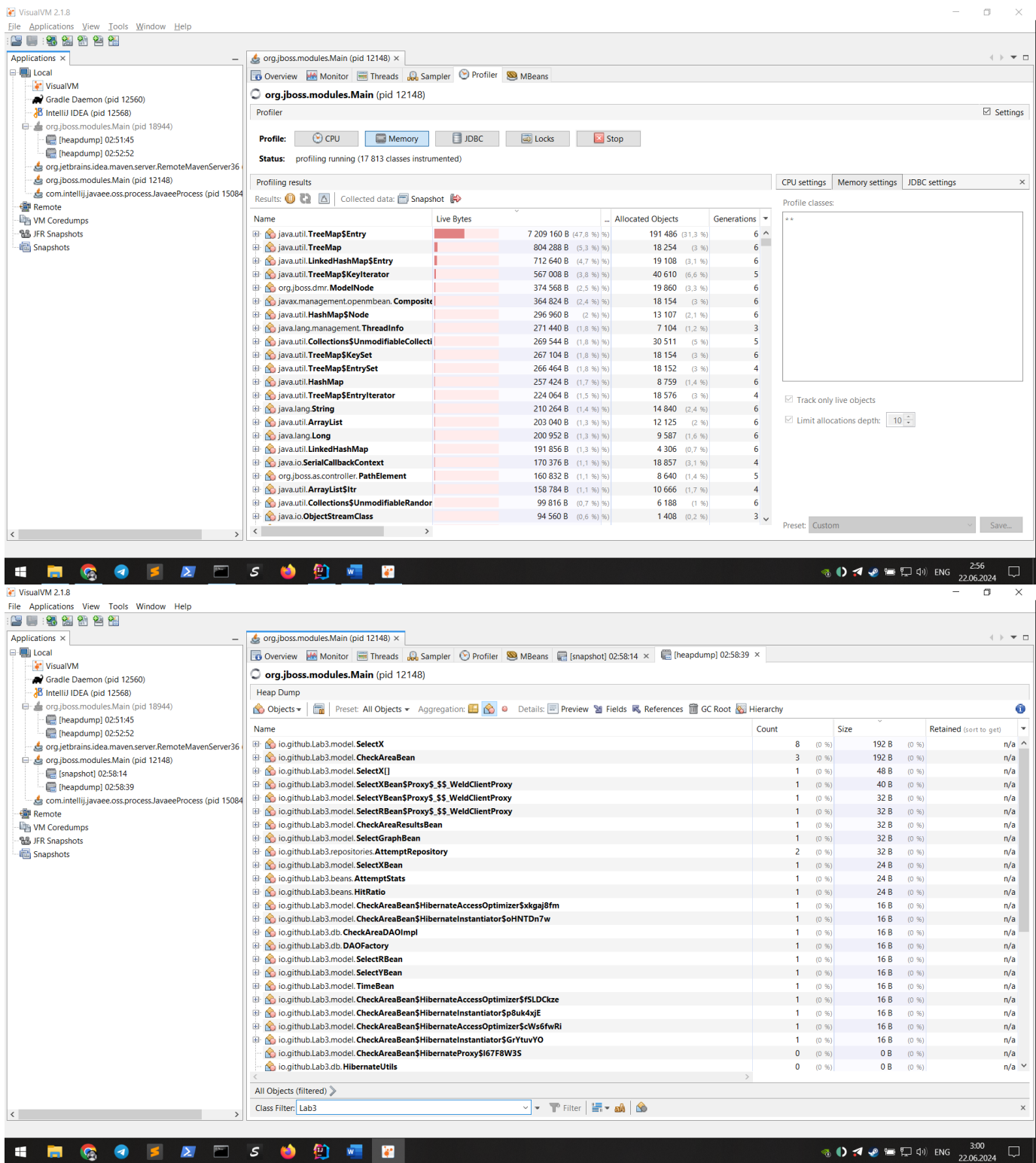
Выводы по результатам мониторинга:

В ходе мониторинга с использованием утилиты Jconsole было выявлено, что:

- AttemptStats и HitRatio были успешно разработаны и зарегистрированы. В случае, если координаты установленной пользователем точки вышла за пределы отображаемой области координатной плоскости, разработанный MBean отправит оповещение об этом событии.
- Из раздела VM Summary можно унать интересующую нас информацию такие как наименование и версию JVM, поставщика виртуальной машины Java и тд.

4. VisualVM



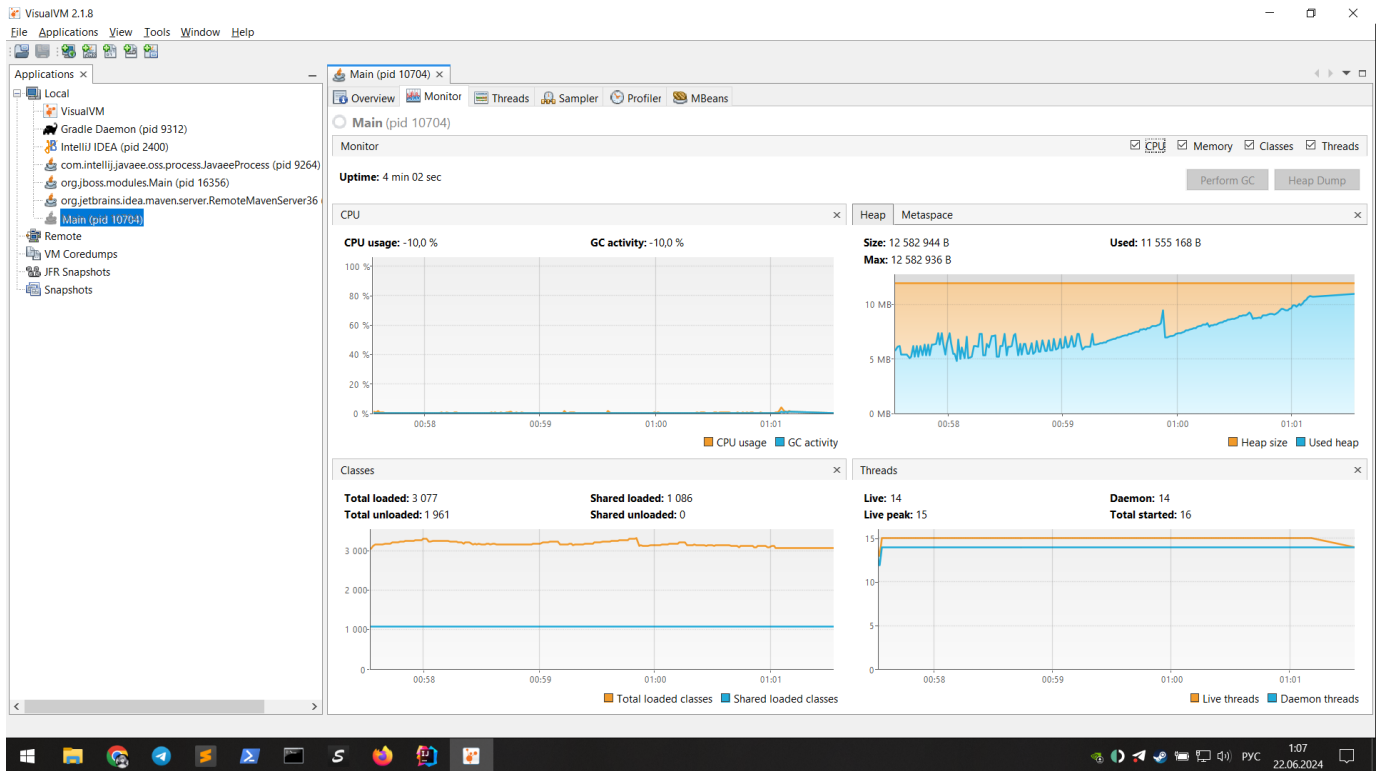


Выводы по результатам мониторинга и профилирования:

- AttemptStats MBean: График показывает общее число установленных пользователем точек, а также общее число точек, попадающих в область и также меняется со временем.
- HitRatio MBean: На графике HitRatio видны изменения количества точек в базе данных приложения, количество попаданий и процент попаданий. График показывает динамику добавления попавших и не попавших точек. Видно, что процент "попаданий" варьируется с течением времени, что отражает взаимодействие пользователя с системой (создание точек на графике).

- На основе профилирования памяти видно, что больше всего памяти занимают значения коллекции TreeMap, на втором месте итераторы – всего объекты класса TreeMap занимают ~45%.
- Больше всего памяти из пользовательских классов занимают объекты SelectX и CheckAreaBean – 56% от пользовательских классов. Однако в масштабе всего приложения эти объекты занимают примерно 0.1%.

4. Исследование производительности программы



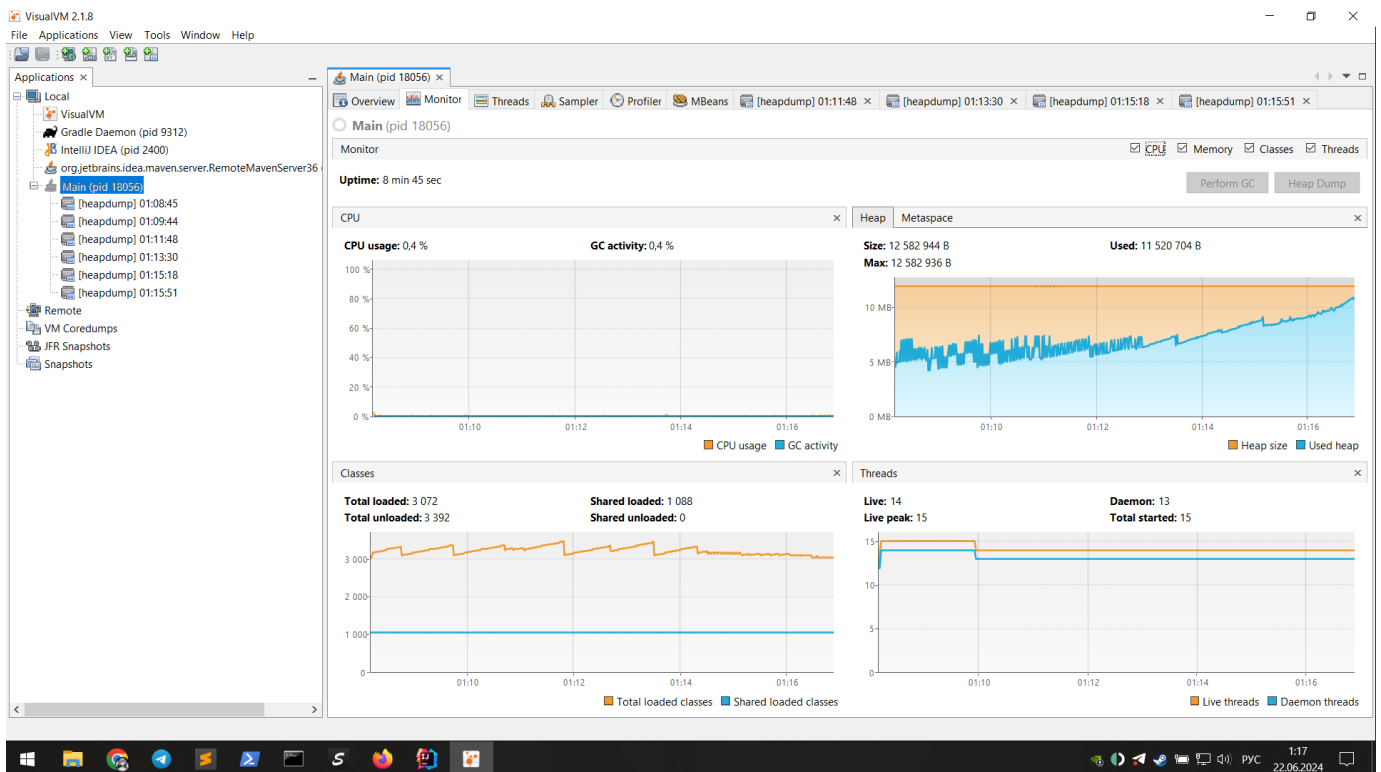
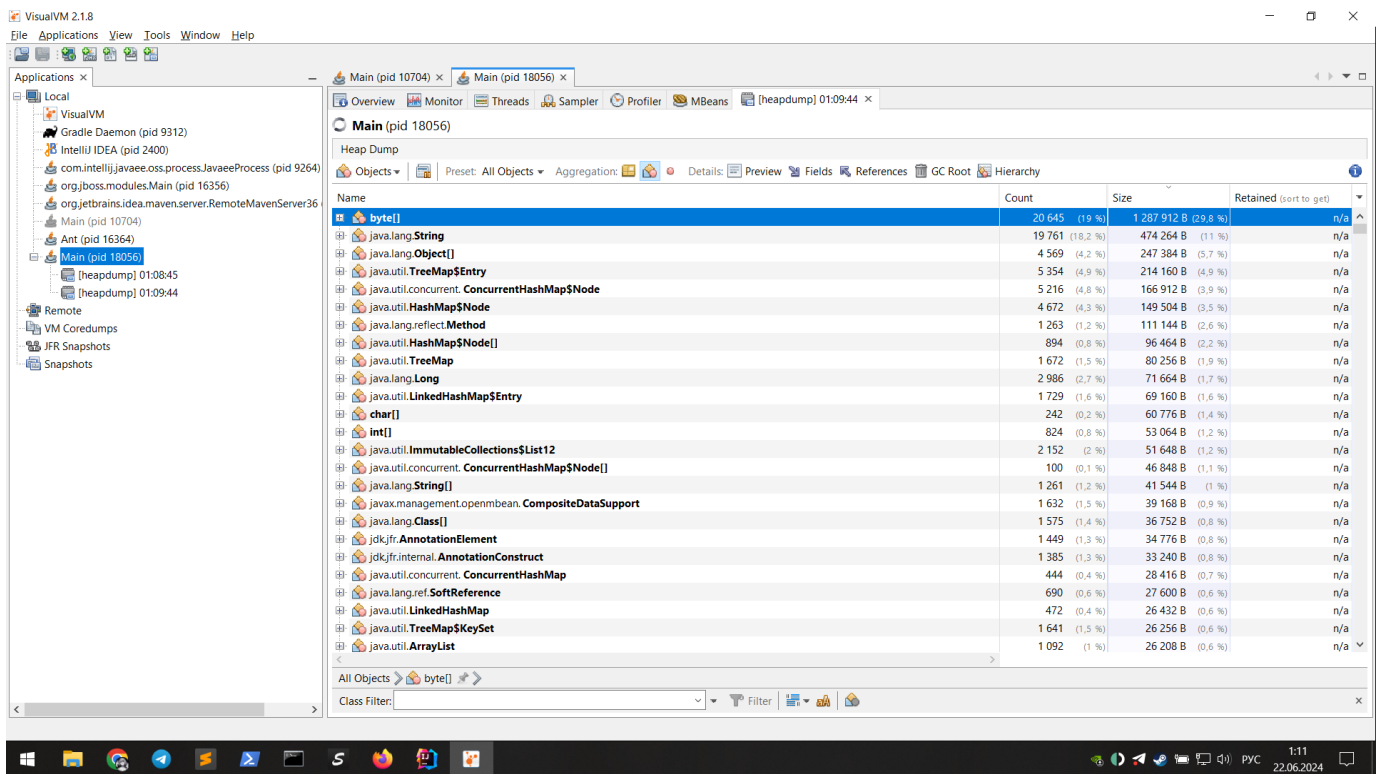
Программа падает через пару из-за нехватки памяти.

```
[java] Count: 1097[ _response = com.meterware.servletunit.ServletUnitHttpResponse@67507df]
[java] Count: 1098[ _response = com.meterware.servletunit.ServletUnitHttpResponse@462abec3]
[java] Count: 1099[ _response = com.meterware.servletunit.ServletUnitHttpResponse@49d30c6f]
[java] Count: 1100[ _response = com.meterware.servletunit.ServletUnitHttpResponse@2b0dc227]
[java]
[java] Exception: java.lang.OutOfMemoryError thrown from the UncaughtExceptionHandler in thread "main"
[java]
[java] Exception: java.lang.OutOfMemoryError thrown from the UncaughtExceptionHandler in thread "RMI TCP Connection(idle)"
[java]
[java] Exception: java.lang.OutOfMemoryError thrown from the UncaughtExceptionHandler in thread "RMI TCP Connection(idle)"
[java]

BUILD FAILED
C:\Users\Kuchizu\Desktop\HttpUnit\nbproject\build-impl.xml:1320: The following error occurred while executing this line:
C:\Users\Kuchizu\Desktop\HttpUnit\nbproject\build-impl.xml:958: Java returned: 1

total time: 4 minutes 2 seconds
C:\Users\Kuchizu\Desktop\HttpUnit>
```

С помощью Heap Dump найдём объекты, занимающие большую часть памяти.



- Видно что при работе программы на каждом запросе создаются экземпляры строк и массивы byte[], затем всё ещё остаются по всей программе и не очищаются.
- Размер кучи постоянно увеличивается и из-за этого через некоторое время получаем ошибку OutOfMemoryError.

```
<items>
[0] = java.lang.String#4633 : Script 'document.wr("Hello Document");//' failed: TypeError: undefined is not a function. (httpunit; )
[1] = java.lang.String#4632 : Script 'document.wr("Hello Document");//' failed: TypeError: undefined is not a function. (httpunit; )
[2] = java.lang.String#4631 : Script 'document.wr("Hello Document");//' failed: TypeError: undefined is not a function. (httpunit; )
[3] = java.lang.String#4630 : Script 'document.wr("Hello Document");//' failed: TypeError: undefined is not a function. (httpunit; )
[4] = java.lang.String#4629 : Script 'document.wr("Hello Document");//' failed: TypeError: undefined is not a function. (httpunit; )
```

Можем видеть что программа не хранит повторяющиеся ненужные сообщения об ошибке.

```
190         return null;
191     }
192 }
193
194
195 private void handleScriptException( Exception e, String badScript ) {
196     final String errorMessage = badScript + " failed: " + e;
197     if (!(e instanceof EcmaError) && !(e instanceof EvaluatorException)) {
198         e.printStackTrace();
199         throw new RuntimeException( errorMessage );
200     } else if (isThrowExceptionsOnError()) {
201         e.printStackTrace();
202         throw new ScriptException( errorMessage );
203     } else {
204         _errorMessages.add( errorMessage );
205     }
206 }
207
208
209 void initialize( JavaScriptEngine parent, ScriptableDelegate scriptable )
210     throws SAXException, PropertyException, JavaScriptException, NotAFunctionException {
211     _scriptable = scriptable;
212     _scriptable.setScriptEngine( this );
213     _parent = parent;
214     if (parent != null) setParentScope( parent );
215 }
216
217
218 String getName() {
219     return _scriptable instanceof NamedDelegate ? ((NamedDelegate) _scriptable).getName() : "";
220 }
```

Видим что строки об ошибках хранятся в _errorMessages.

```
/**
 * Clears the accumulated script error messages.
 */
no usages
public static void clearScriptErrorMessages() {
    getScriptingEngine().clearErrorMessages();
}
```

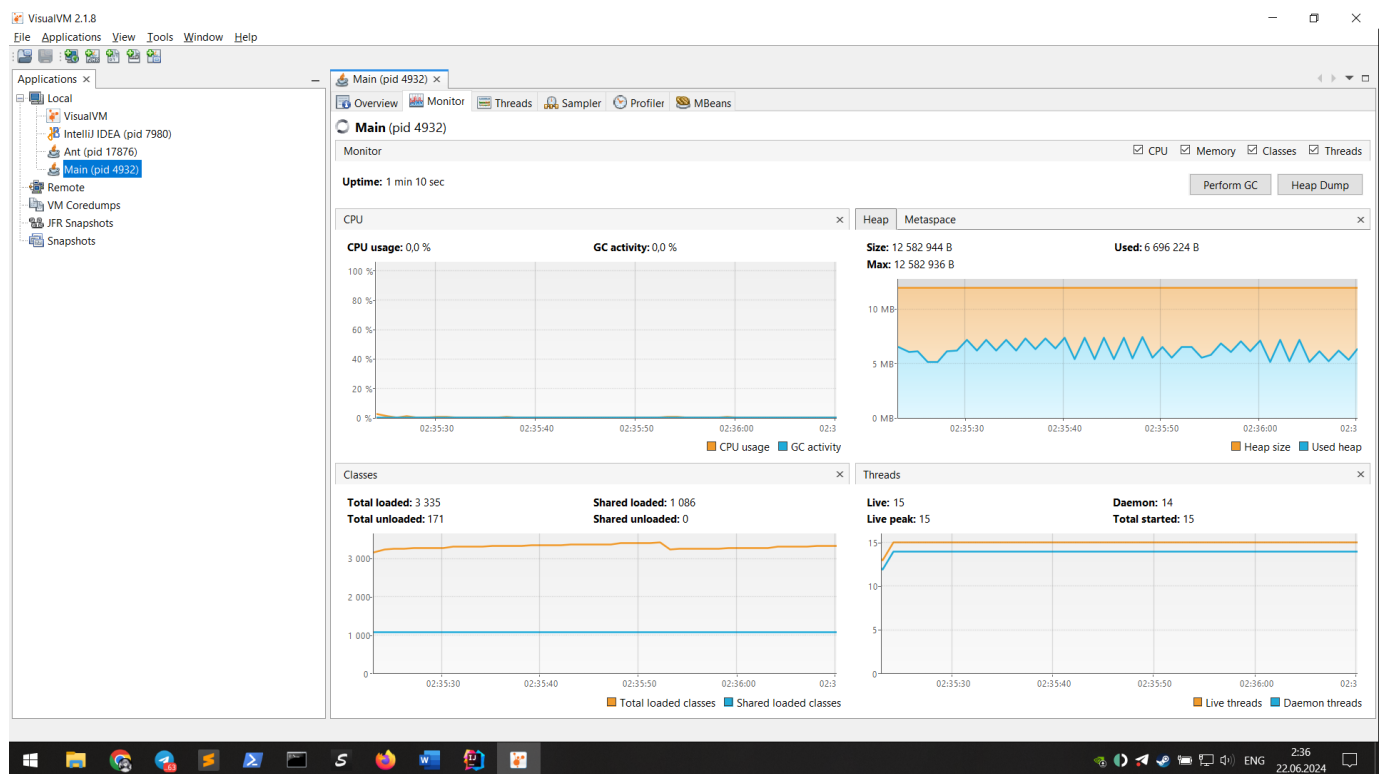
Видим что в программе уже есть функция для нашей задачи но она не используется. Нам достаточно после каждого запроса очищать список ошибок и ошибка с утечкой памяти пропадёт.

```

30  /** Creates a new instance of Main */
31  public Main() {
32  }
33
34  /**
35   * @param args the command line arguments
36   */
37  public static void main(String[] args) {
38      try {
39          HttpUnitOptions.setExceptionsThrownOnScriptError(false);
40          ServletRunner sr = new ServletRunner();
41          sr.registerServlet("myServlet", HelloWorld.class.getName());
42          ServletUnitClient sc = sr.newClient();
43          int number = 1;
44          WebRequest request = new GetMethodWebRequest("http://test.meterware.com/myServlet");
45          while (true) {
46              WebResponse response = sc.getResponse(request);
47              System.out.println("Count: " + number++ + response);
48              java.lang.Thread.sleep(200);
49              HttpUnitOptions.clearScriptErrorMessages();
50          }
51      } catch (InterruptedException ex) {
52          Logger.getLogger("global").log(Level.SEVERE, null, ex);
53      } catch (MalformedURLException ex) {
54          Logger.getLogger("global").log(Level.SEVERE, null, ex);
55      } catch (IOException ex) {
56          Logger.getLogger("global").log(Level.SEVERE, null, ex);
57      } catch (SAXException ex) {
58          Logger.getLogger("global").log(Level.SEVERE, null, ex);
59      }
60  }
61  }

```

Запускаем программу с внесёнными изменениями и видим что программа стабильно работает.



6. Вывод

В ходе лабораторной работы я освоил практику создания MBeans для веб-приложений, а также изучил инструменты мониторинга и профилирования программ, такие как JConsole и VisualVM. Также приобрёл опыт в анализе для выявления утечек памяти и их устранения.