

МИНИСТЕРСТВО НАУКИ И ОБРАЗОВАНИЯ РФ

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 №-324567

Выполнили:

Студенты группы Р3232

Чмурова Мария Владиславовна

Комягин Дмитрий Анатольевич

Проверила:

Бострикова Дарья

Константиновна

г. Санкт-Петербург

2024 год

Оглавление

Задание	3
Код Apache Ant (build.xml)	Error! Bookmark not defined.
Код build.properties	Error! Bookmark not defined.
Вывод	6

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 пользователь совершил 3 "промаха" подряд, разработанный MBean должен отправлять оповещение об этом событии.
- MBean, определяющий площадь получившейся фигуры.

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 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.

Выполнение

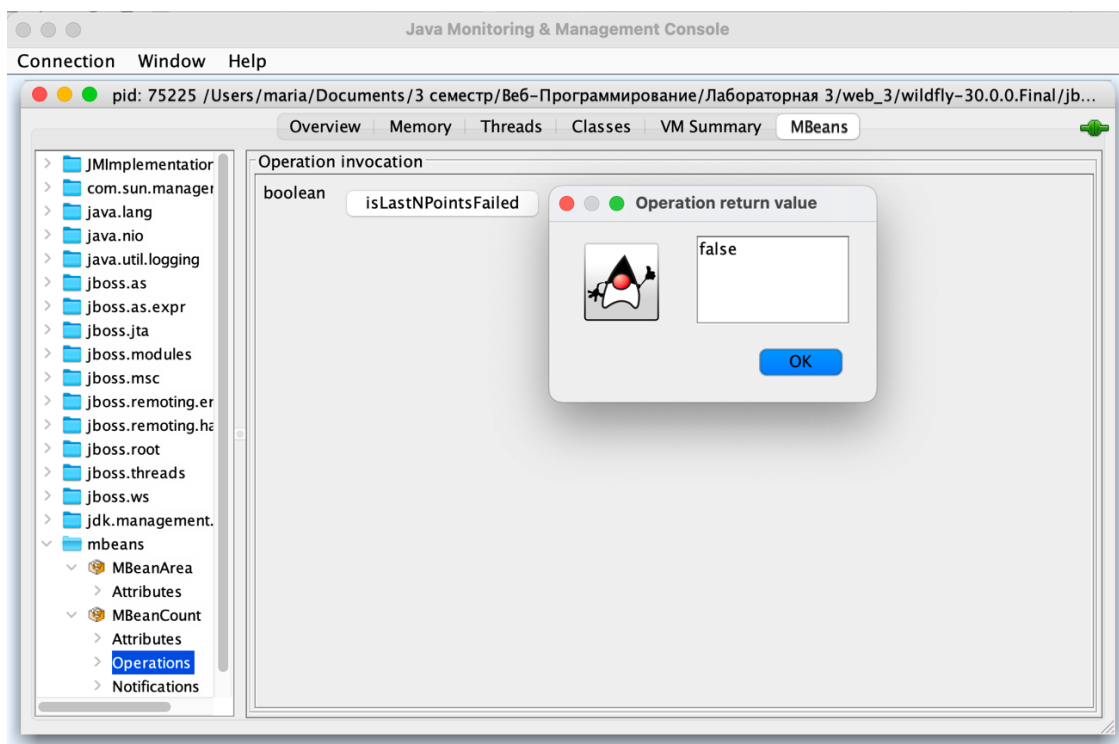
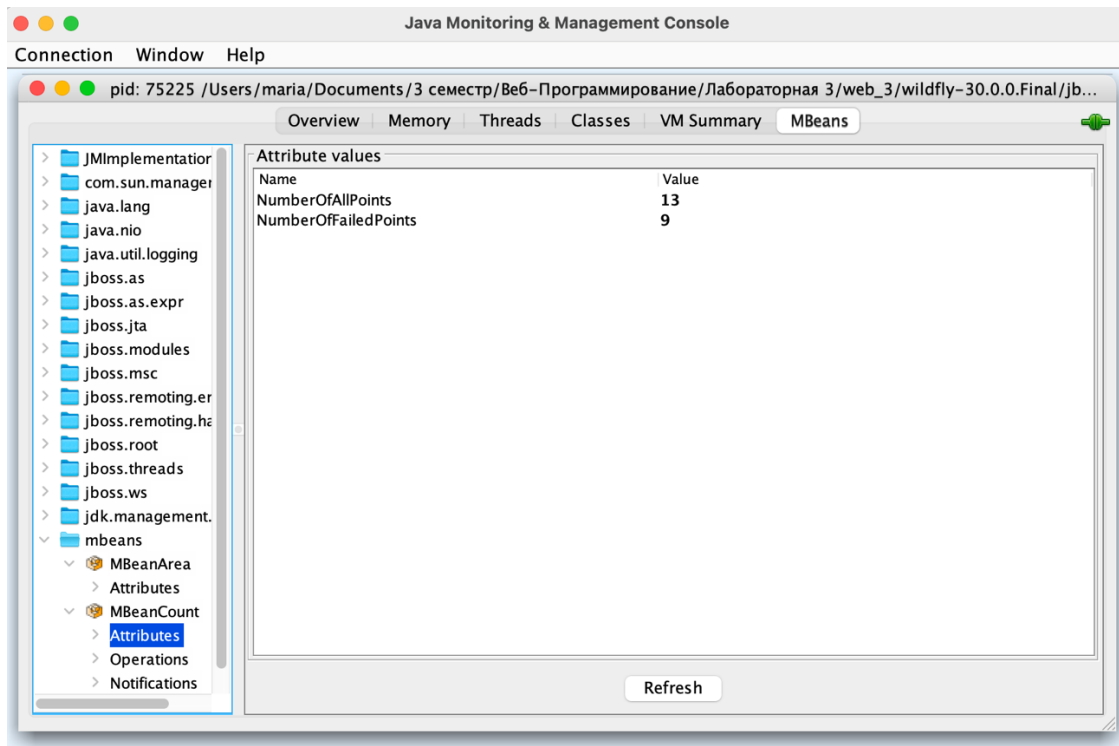
1. Проект и реализация MBean

Ссылка на GitHub с MBean:

https://github.com/kkettch/lab_3

2. Мониторинг программы с помощью JConsole

Показания MBean-классов, разработанных в ходе выполнения задания 1.



Java Monitoring & Management Console

Connection Window Help

pid: 75502 /Users/maria/Documents/3 семестр/Веб-Программирование/Лабораторная 3/web_3/wildfly-30.0.0.Final/jb...

Overview Memory Threads Classes VM Summary **MBeans**

Notification buffer

TimeStamp	Type	UserData	SeqNum	Message	Event	Source
16:33:09:9...	Три промах...		3	Было совер...	javax.manag...	mbeans.Count
16:32:59:5...	Три промах...		2	Было совер...	javax.manag...	mbeans.Count

Subscribe Unsubscribe Clear

Left sidebar tree:

- JMImplementation
- com.sun.management
- java.lang
- java.nio
- java.util.logging
- jboss.as
- jboss.as.expr
- jboss.jta
- jboss.modules
- jboss.msc
- jboss.remoting.endpoint
- jboss.remoting.handler
- jboss.root
- jboss.threads
- jboss.ws
- jdk.management.jfr
- mbeans
 - MBeanArea
 - MBeanCount
 - Attributes
 - Operations
 - Notifications[2]
- org.xnio

Java Monitoring & Management Console

Connection Window Help

pid: 75502 /Users/maria/Documents/3 семестр/Веб-Программирование/Лабораторная 3/web_3/wildfly-30.0.0.Final/jb...

Overview Memory Threads Classes VM Summary **MBeans**

Attribute values

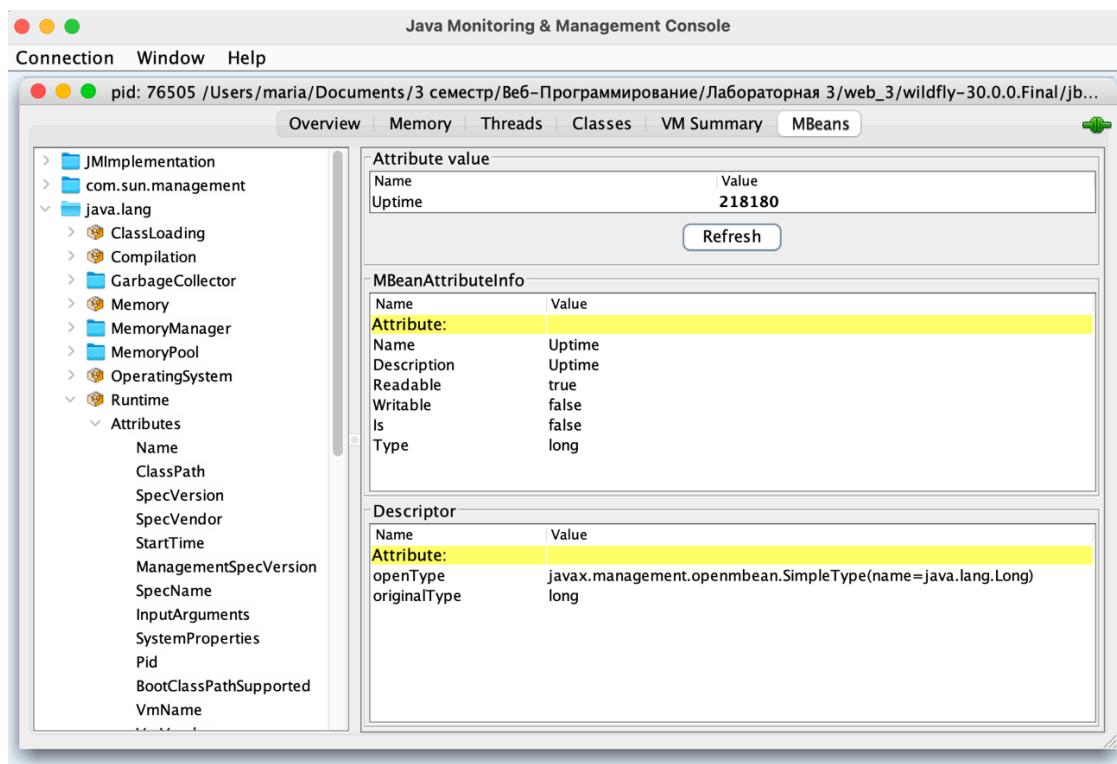
Name	Value
AreaSize	11.892145867644258

Refresh

Left sidebar tree:

- JMImplementation
- com.sun.management
- java.lang
- java.nio
- java.util.logging
- jboss.as
- jboss.as.expr
- jboss.jta
- jboss.modules
- jboss.msc
- jboss.remoting.endpoint
- jboss.remoting.handler
- jboss.root
- jboss.threads
- jboss.ws
- jdk.management.jfr
- mbeans
 - MBeanArea
 - Attributes
 - MBeanCount
 - Attributes
 - Operations
 - Notifications[2]
- org.xnio

Время (в мс), прошедшее с момента запуска виртуальной машины:



3. VisualVM. Мониторинг и профилирование программы

График изменения 1-го MBean (Count):

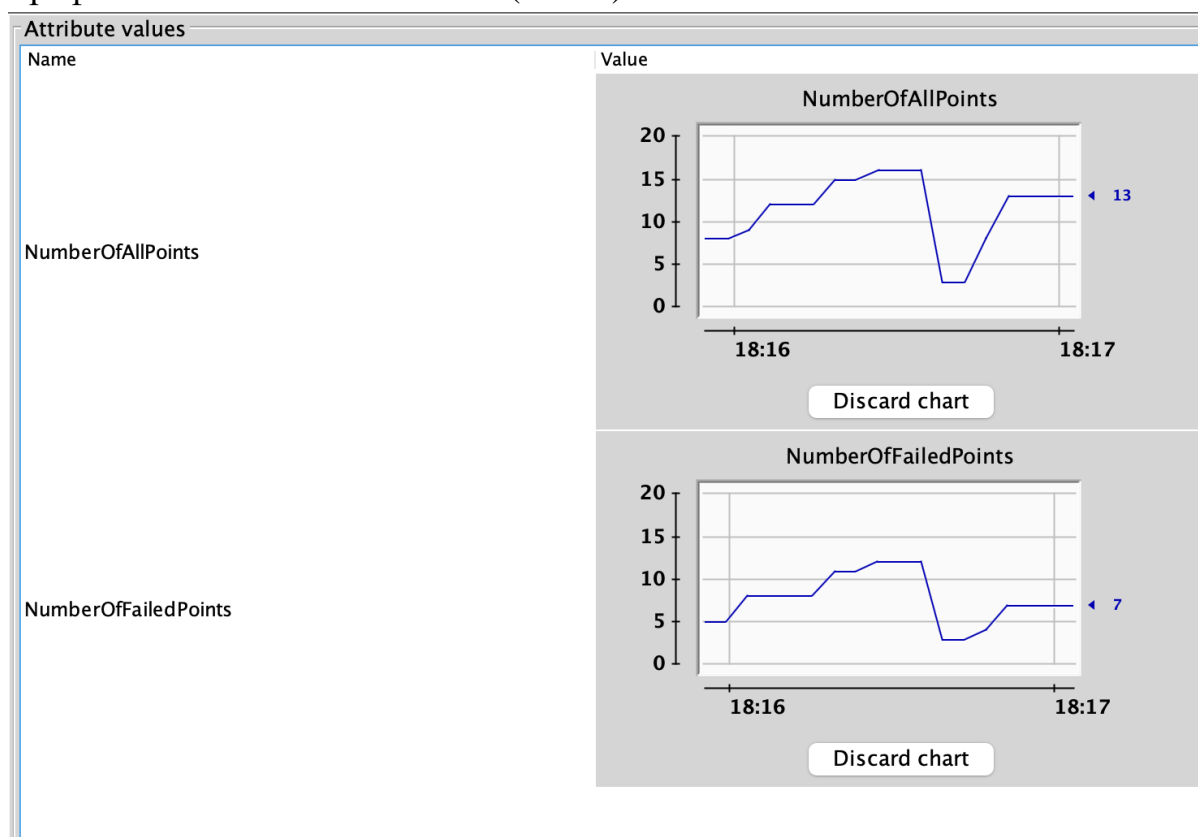
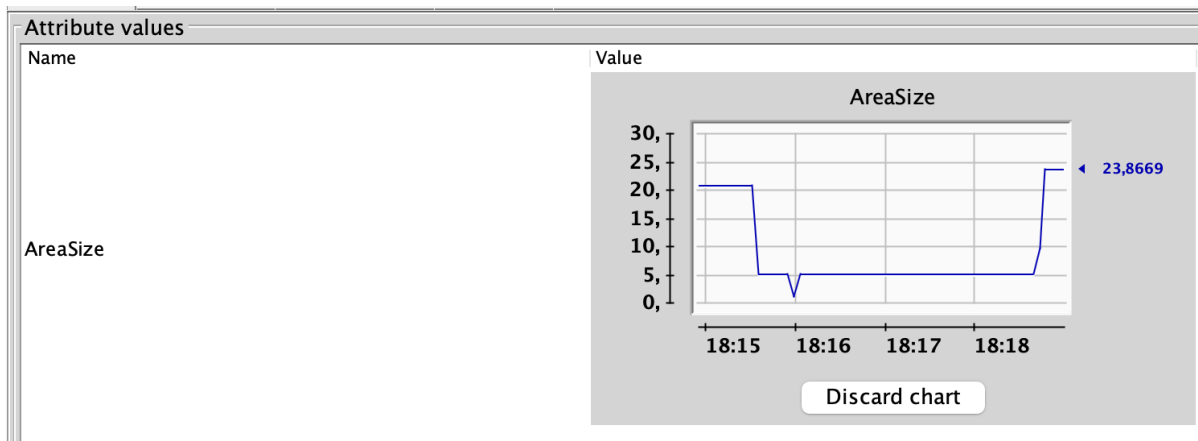


График изменения 2-го MBeans (Area):



Классы, объекты которых занимают наибольший объем памяти JVM:

Classes by Size of Instances [view all]

byte[]	28 939 640 B	(34,5 %)
java.util.HashMap\$Node	6 968 768 B	(8,3 %)
java.lang.String	5 492 160 B	(6,6 %)
java.lang.Object[]	3 983 696 B	(4,8 %)
java.util.HashMap\$Node[]	2 770 880 B	(3,3 %)

Наибольший объем памяти занимает: **java.util.HashMap\$Node**

Пользовательские классы, объекты которого содержат этот класс:

mbeans.Area	1	(0 %)	16 B	(0 %)	18 376 B	(0 %)
mbeans.Area#1			16 B	(0 %)	18 376 B	(0 %)
<fields>						
<references>						
value in java.util.HashMap\$Node#198118			32 B	(0 %)	32 B	(0 %)
instance in org.jboss.weld.contexts.SerializableContextualInstanceImpl#3			24 B	(0 %)	392 B	(0 %)
value in org.jboss.weld.bean.ContextualInstanceStrategy\$ApplicationScoped			16 B	(0 %)	16 B	(0 %)
point.PointBean	1	(0 %)	40 B	(0 %)	43 992 B	(0,1 %)
point.PointBean#1			40 B	(0 %)	43 992 B	(0,1 %)
<fields>						
<references>						
value in java.util.HashMap\$Node#190606			32 B	(0 %)	32 B	(0 %)
instance in org.jboss.weld.contexts.SerializableContextualInstanceImpl#5			24 B	(0 %)	496 B	(0 %)
value in org.jboss.weld.bean.ContextualInstanceStrategy\$ApplicationScoped			16 B	(0 %)	16 B	(0 %)
mbeans.Count	1	(0 %)	40 B	(0 %)	36 528 B	(0 %)
mbeans.Count#1			40 B	(0 %)	36 528 B	(0 %)
<fields>						
<references>						
value in java.util.HashMap\$Node#181114			32 B	(0 %)	32 B	(0 %)
instance in org.jboss.weld.contexts.SerializableContextualInstanceImpl#2			24 B	(0 %)	392 B	(0 %)
value in org.jboss.weld.bean.ContextualInstanceStrategy\$ApplicationScoped			16 B	(0 %)	16 B	(0 %)

4. Локализация и устранение проблемы с производительностью в программе

Вывод

В ходе лабораторной работы мы составили собственный сценарий сборки проекта java при помощи утилиты Apache Ant. Также написали собственные цели различных направленностей. Узнали и о других утилитах сборки, их отличиях между собой.