

Методы и средства программной инженерии

Лабораторная работа №4

Вариант №3357

Преподаватель: Покид Александр Владимирович

Выполнили: Груздева Евгения Николаевна,

Липонина Екатерина Алексеевна

N3248

Санкт-Петербург

2022

1. Для своей программы изпредыдущей лабораторной реализовать:

* MBean, считающий количество последовательных ненайденных слов в тексте. В случае, если не было найдено три слова подряд MBean посылает уведомление об этом.
* MBean, определяющий количество найденных, ненайденных слов и процент найденных слов.

2. С помощью утилиты JConsole провести мониторинг программы:

* Снять показания MBean-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 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](https://se.ifmo.ru/documents/10180/189115/HttpUnit.tar.gz/7bf1032e-d16e-be85-c71b-dbe73c0178ba?t=1651168887037&download=true). По результатам локализации и устранения проблемы необходимо составить отчёт.

1. **Main**

package mispi;

import java.util.Scanner;

import java.io.File;

import java.io.IOException;

import javax.management.\*;

import java.lang.management.ManagementFactory;

import java.util.\*;

/\*\* В классе Main производится поиск слов в заданном заранее тексте \*\*/

public class Main {

public static void main(String[] args) throws MalformedObjectNameException, InstanceAlreadyExistsException, MBeanRegistrationException, NotCompliantMBeanException, IOException {

//MBean

MBeanServer mbs = ManagementFactory.getPlatformMBeanServer();

WordsCounter wordsBean = new WordsCounter();

ObjectName nameW = new ObjectName("mispi.Main:name=WordsCounter");

mbs.registerMBean(wordsBean, nameW);

MissedNot missBean = new MissedNot();

ObjectName nameM = new ObjectName("mispi.Main:name=MissedNot");

mbs.registerMBean(missBean, nameM);

//search

int entry\_number = 0;

Scanner word\_scanner = new Scanner(System.in);

File document = new File("/home/master/test/mispi\_lab4/poem.txt");

Scanner doc\_scanner = new Scanner(document);

while(word\_scanner.hasNext()) {

SearchedString substring = new SearchedString(word\_scanner.next());

int number = 0;

while (doc\_scanner.hasNextLine()) {

String searchIn = doc\_scanner.nextLine();

int result = substring.NumberOfEntry(searchIn);

number += result;

}

doc\_scanner.close();

doc\_scanner = new Scanner(document);

System.out.println(number);

entry\_number += number;

wordsBean.updateNumber(number != 0);

missBean.updateCounter(number != 0);

}

word\_scanner.close();

doc\_scanner.close();

}

}

**WordsCounterMBean**

package mispi;

/\*

Подсчет количества введенных слов, а так же числа ненайденных слов и возврат процента успеха.

\*/

public interface WordsCounterMBean {

double SuccessPct();

int getWordsNumber();

int getMissedNumber();

}

**WordsCounter**

package mispi;

public class WordsCounter implements WordsCounterMBean {

private int wordsNumber = 0;

private int missedNumber = 0;

public void updateNumber(boolean success) {

wordsNumber++;

if (!success) missedNumber++;

}

@Override

public double SuccessPct() {

return (1 - (double)missedNumber / wordsNumber);

}

@Override

public int getWordsNumber() {

return wordsNumber;

}

@Override

public int getMissedNumber() {

return missedNumber;

}

}

**MissedNotMBean**

package mispi;

import javax.management.NotificationBroadcaster;

/\*

Если не введенные слова не присутсвовали в тексте три раза подряд - посылается уведомление

\*/

public interface MissedNotMBean extends NotificationBroadcaster {

boolean isThreeInRow();

int getMissedInRow();

}

**MissedNot**

package mispi;

import javax.management.Notification;

import javax.management.NotificationBroadcasterSupport;

import javax.management.AttributeChangeNotification;

import javax.management.MBeanNotificationInfo;

public class MissedNot extends NotificationBroadcasterSupport implements MissedNotMBean {

private boolean ThreeInRow = false;

private int InRow = 0;

private int sequenceNumber = 0;

public void updateCounter(boolean success) {

if (success) {

InRow = 0;

ThreeInRow = false;

}

else {

InRow++;

if (InRow >= 3) {

ThreeInRow = true;

}

}

if(ThreeInRow) {

Notification notification = new Notification("threeAbsendInRow", this, ++sequenceNumber, System.currentTimeMillis(), "Three absent words in a row happend");

sendNotification(notification);

}

}

@Override

public boolean isThreeInRow() {

return ThreeInRow;

}

@Override

public int getMissedInRow() {

return InRow;

}

@Override

public MBeanNotificationInfo[] getNotificationInfo(){

String[] types = new String[]{ AttributeChangeNotification.ATTRIBUTE\_CHANGE};

String name = AttributeChangeNotification.class.getName();

String description = "Three absent words in a row happend";

MBeanNotificationInfo info = new MBeanNotificationInfo(types, name, description);

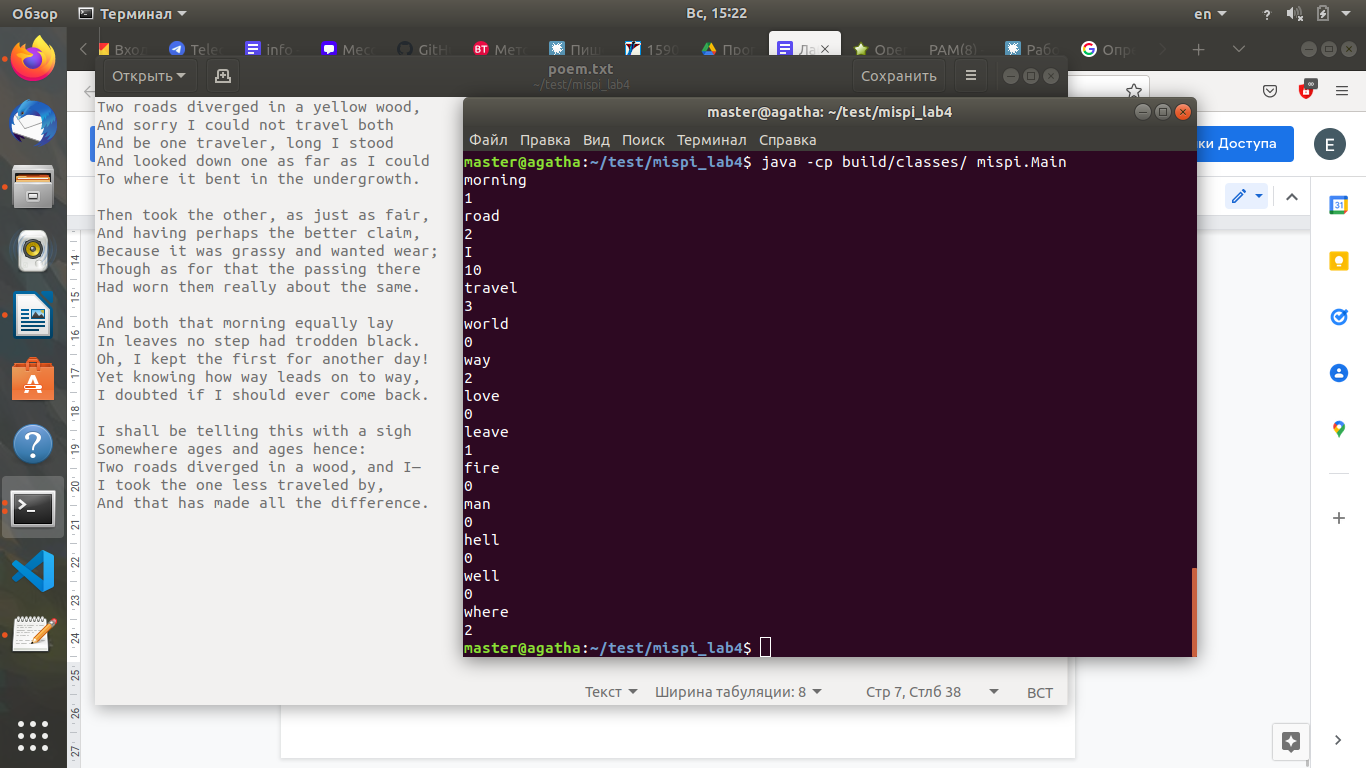
return new MBeanNotificationInfo[]{info};

}

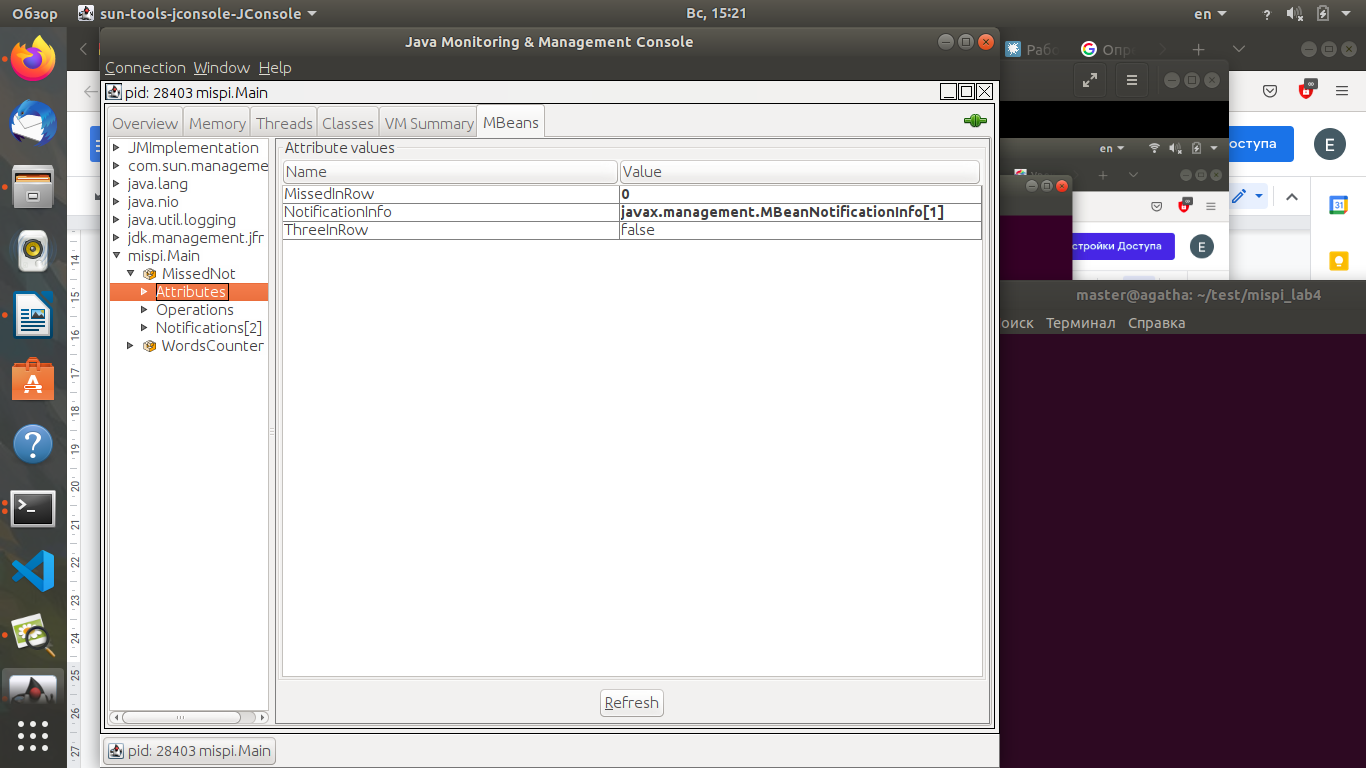
}

2. jconsole $(pgrep java)

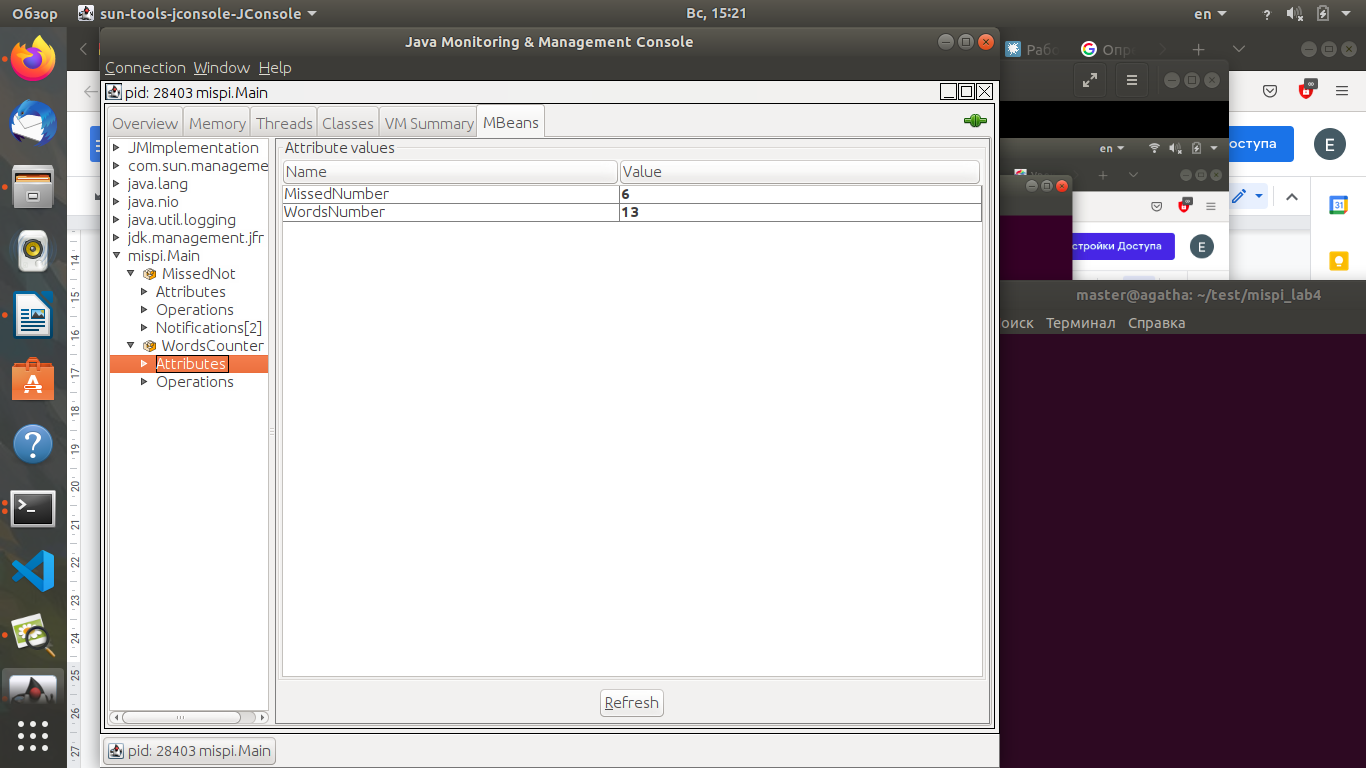
Входные данные

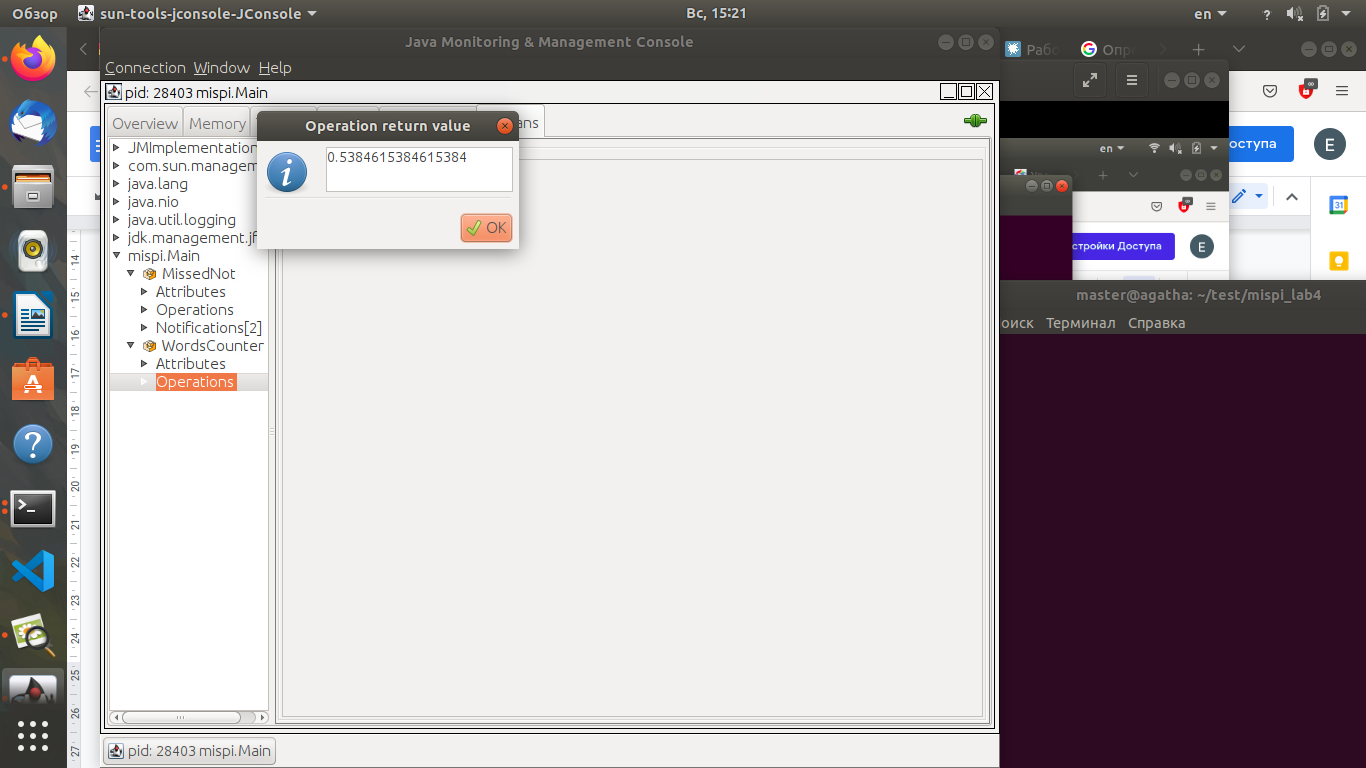


MissedNotMBean

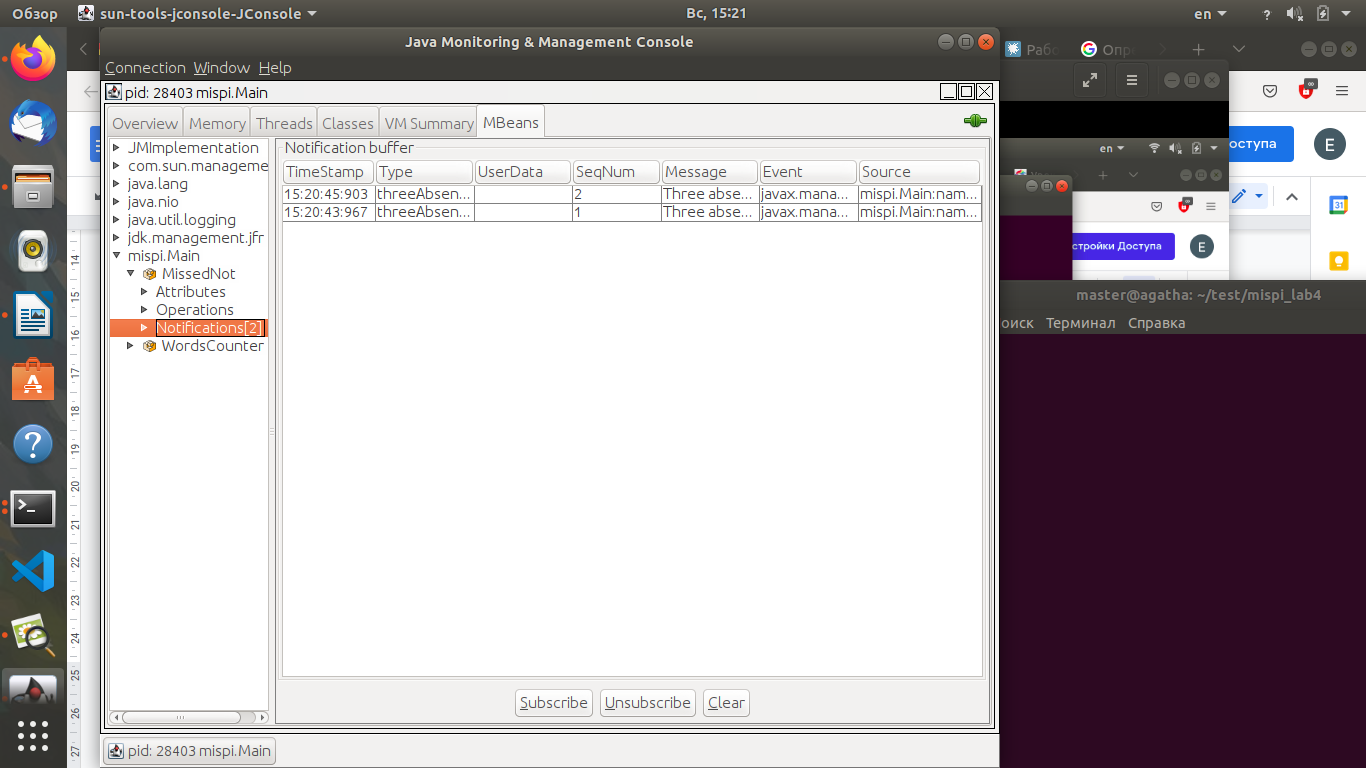


WordsCounterMBean

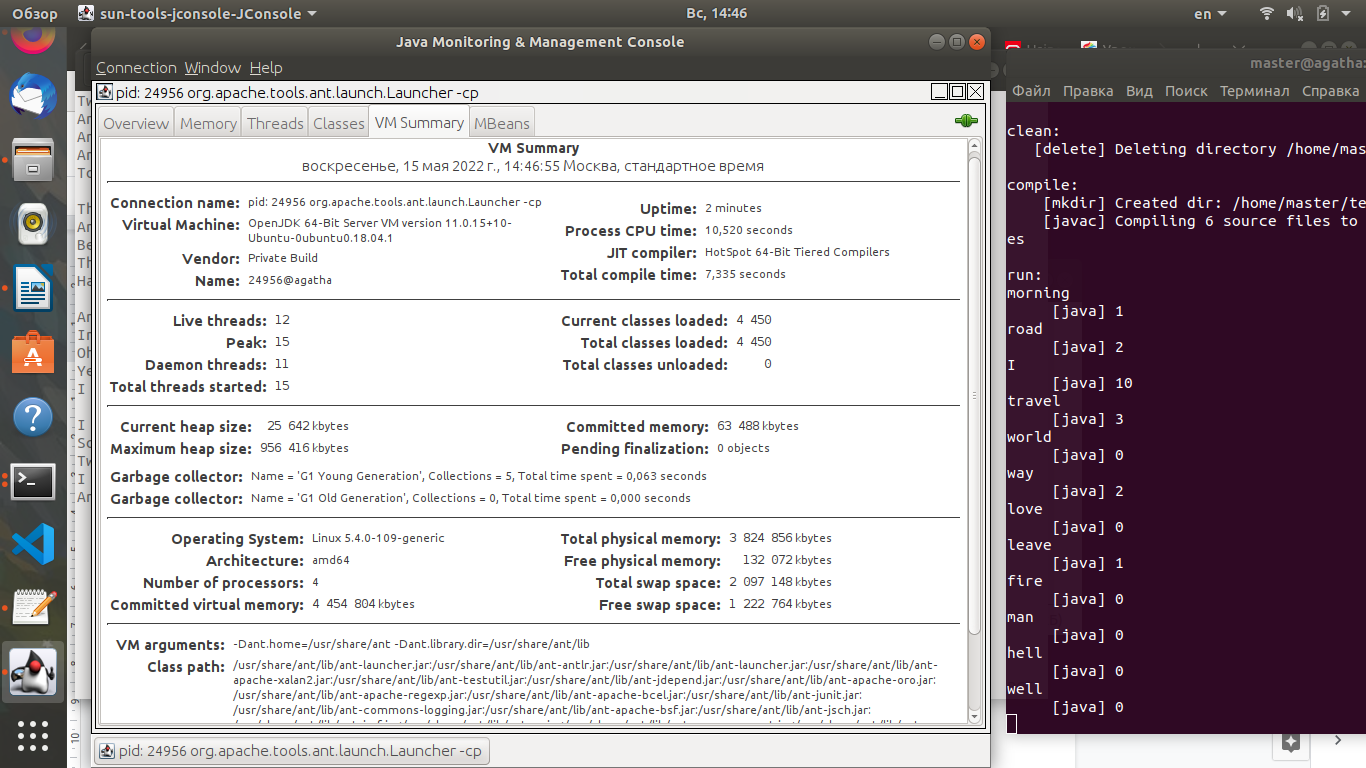




Уведомления

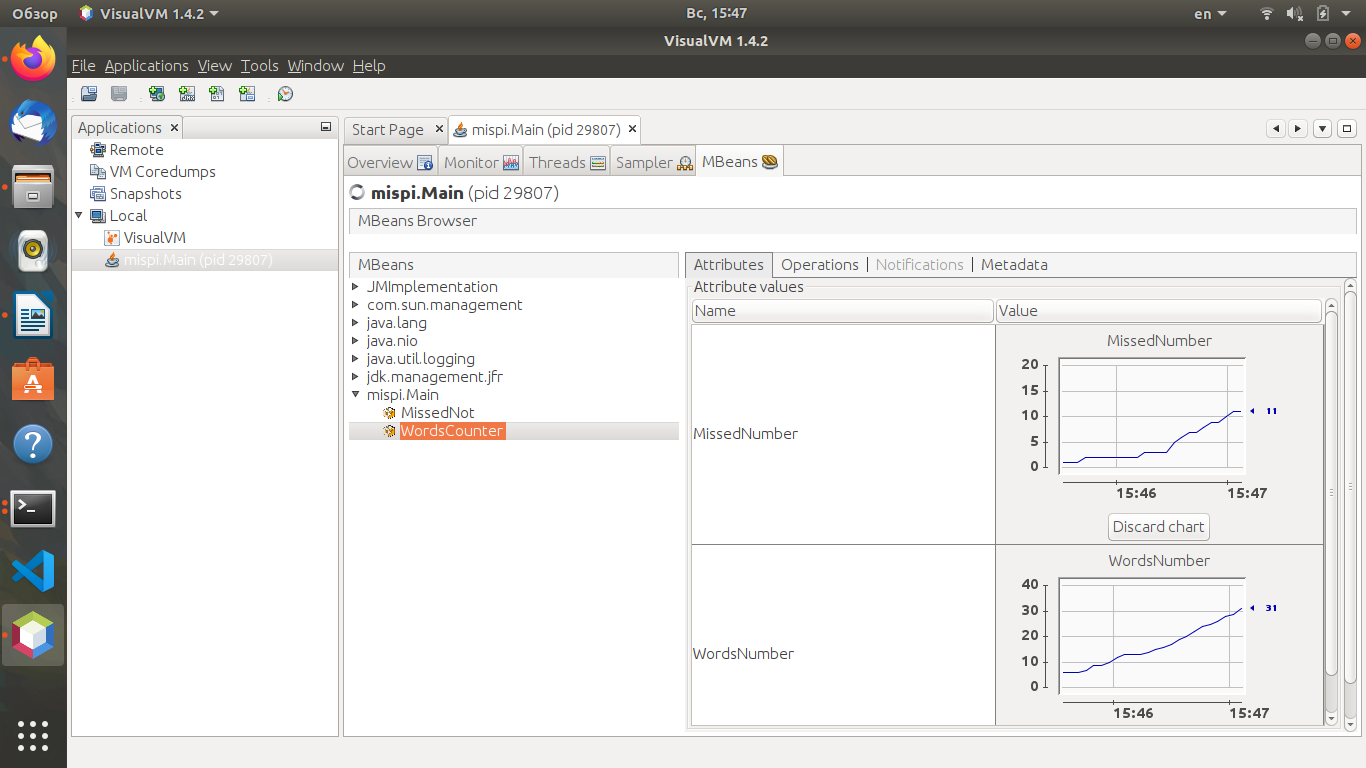


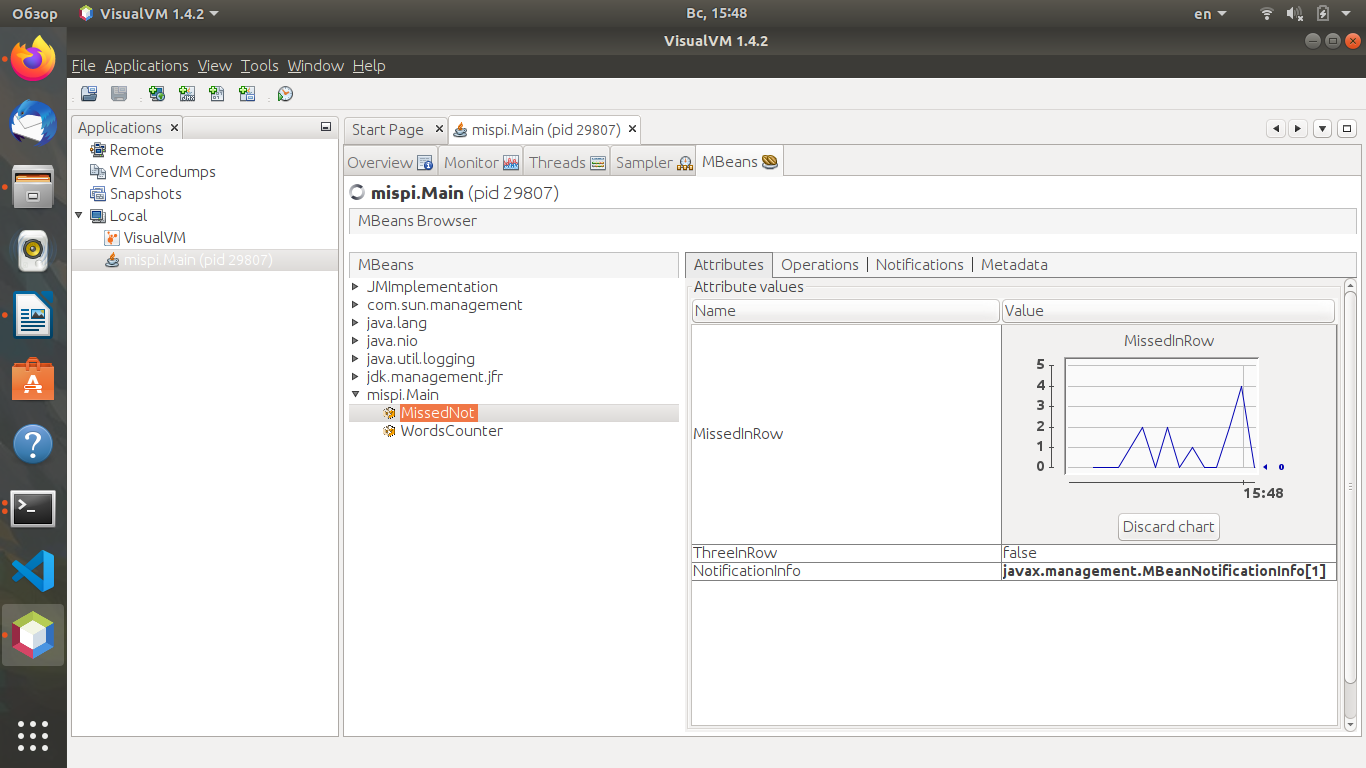
Наименование и версия JVM, поставщик виртуальной машины Java и номер её сборки.

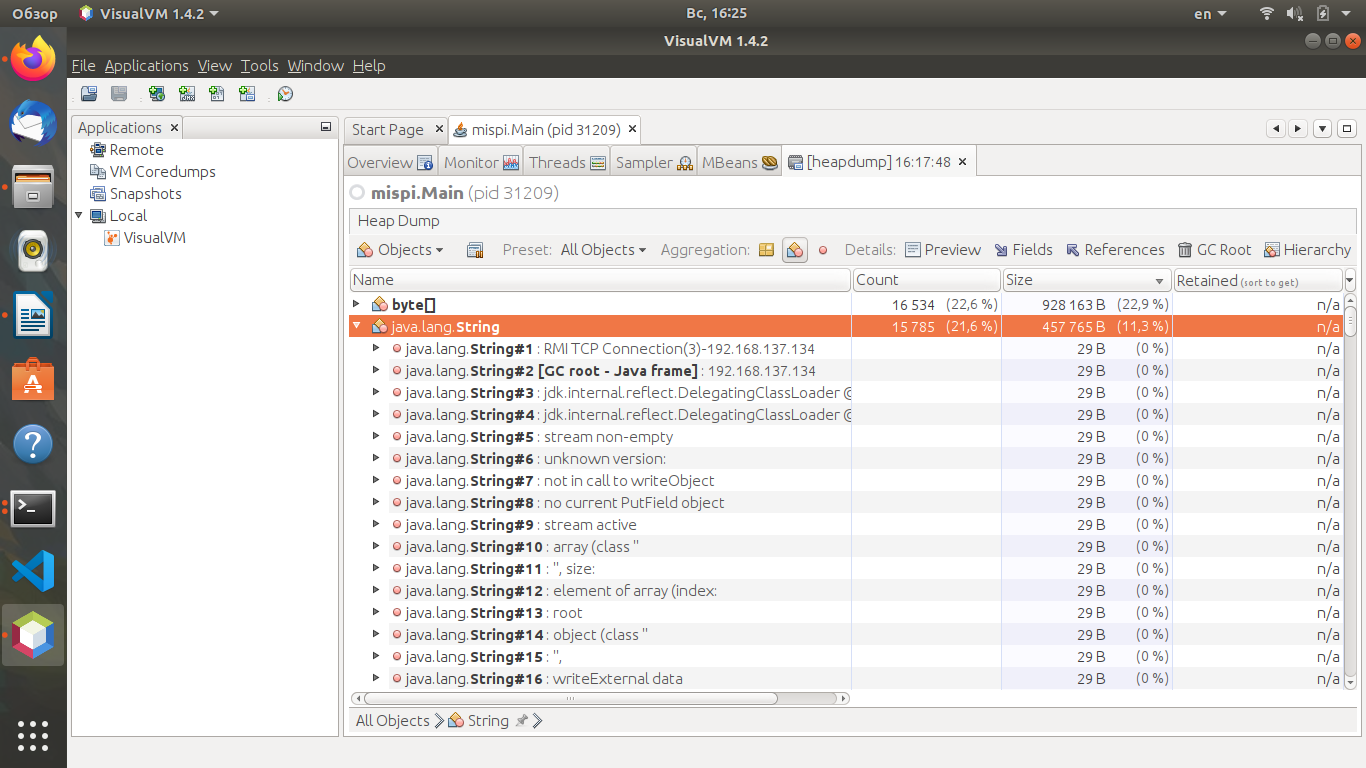


3.

Изменения показаний MBean классов со временем (для числовых значений)

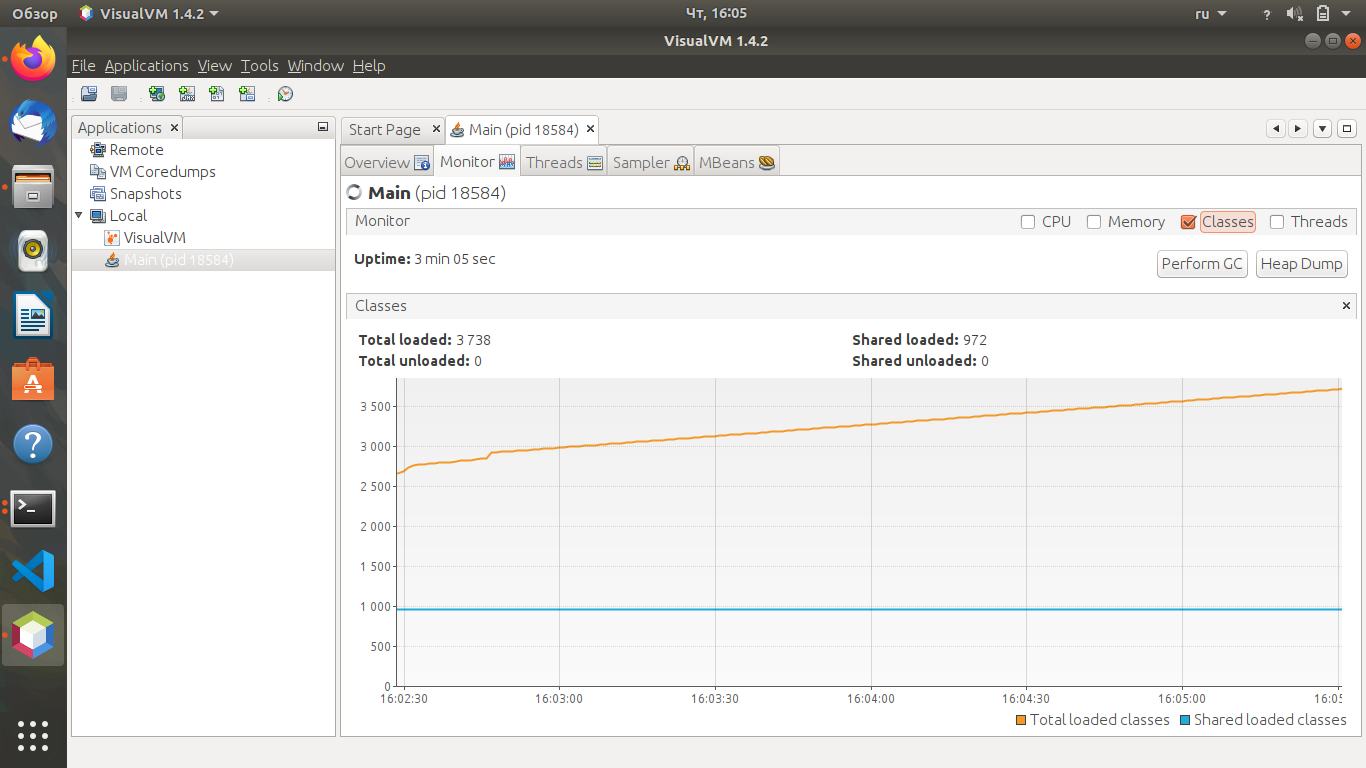
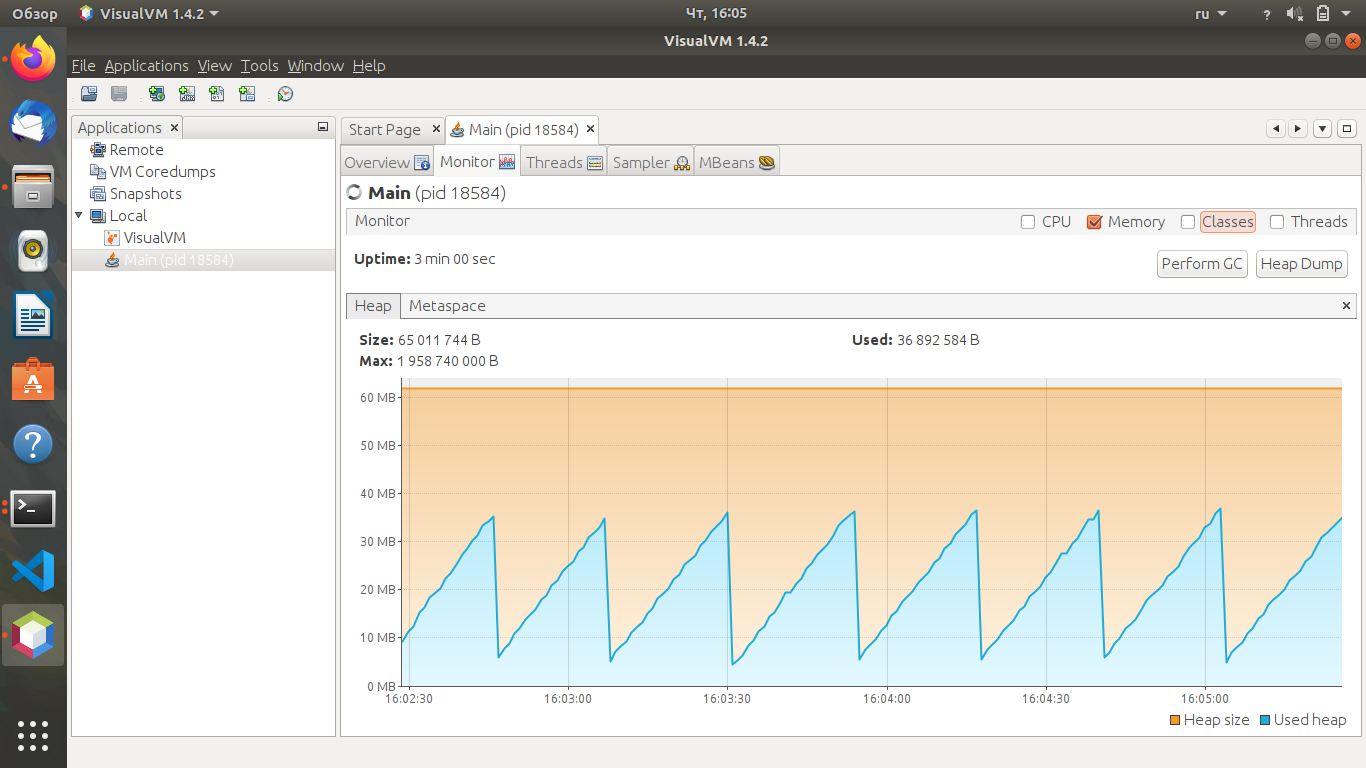






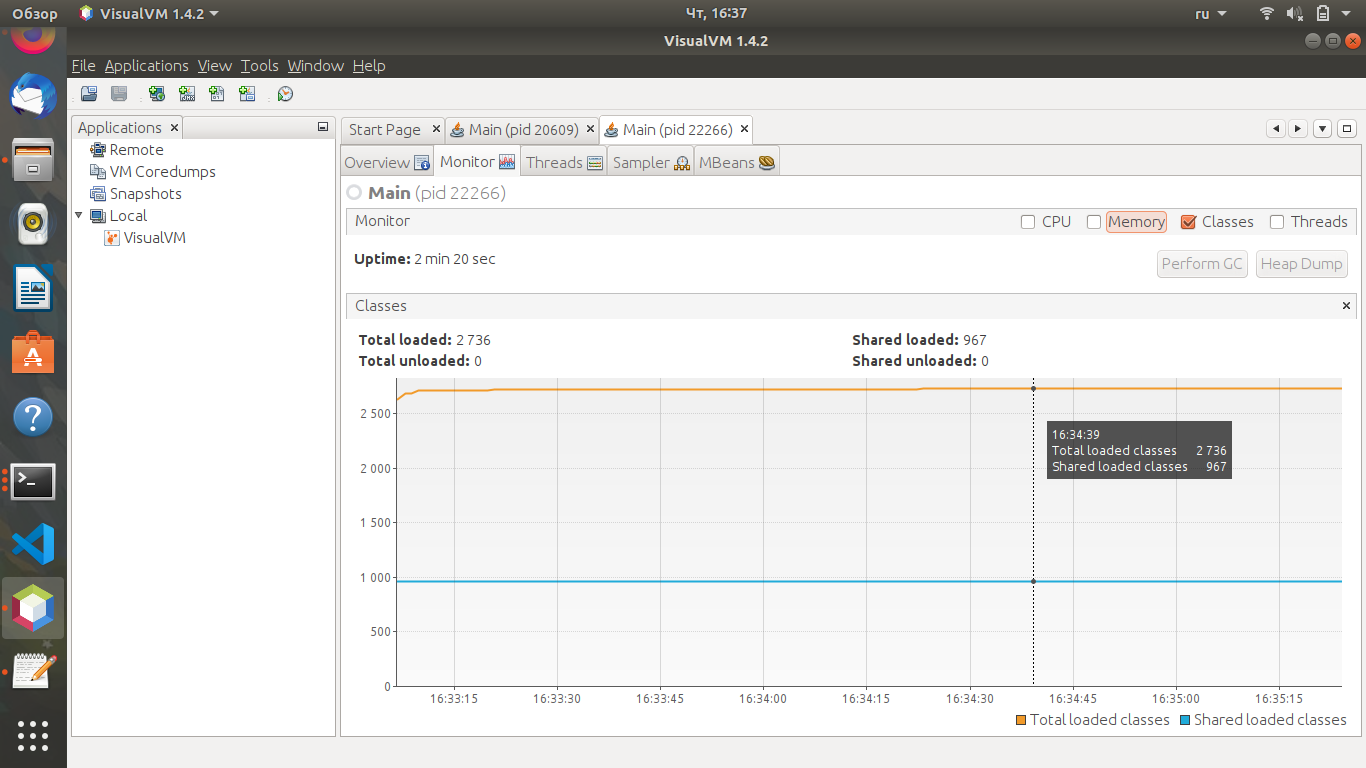
4.

Работа программы изначально



Если мы вынесем строку за границы цикла, чтобы объект был создан лишь единожды, количество классов перестанет расти

WebResponse response = sc.getResponse(request);



Также существует идея в том, что при каждом вызове заполняется массив ошибок, так как в запросе в HelloWorld.java находится ошибка, а в главной функции устанавливается в ложь флаг выбрасывания исключения при неправильном запросе. Если добавить

if (number % 100 == 0) {

HttpUnitOptions.clearScriptErrorMessages();

}

Тогда ошибочные сообщения будут удаляться.

