МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики»

ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ

**ЛАБОРАТОРНАЯ РАБОТА №4**

по дисциплине

«Основы программной инженерии»

Вариант № 34294

***Выполнили:***

Студенты группы P3216

Векшин А. И.

Козодой А.С.

***Преподаватель:***

Письмак Алексей Евгеньевич

# 

[**Задание 3**](#_g7yvr5ee51ia)

[**build.xml 4**](#_49usfucba350)

[**build.properties 6**](#_tkdqlv5kdmvq)

[**Результаты работы 7**](#_nqi9yukr9qdx)

[**Вывод 8**](#_cyp0yd1eoiuk)

# 

# **Задание**

1. Для своей программы из [лабораторной работы #3](https://se.ifmo.ru/courses/web#lab3) по дисциплине "Веб-программирование" реализовать:

* MBean, считающий общее число установленных пользователем точек, а также число точек, не попадающих в область. В случае, если количество установленных пользователем точек стало кратно 15, разработанный MBean должен отправлять оповещение об этом событии.
* MBean, определяющий процентное отношение "попаданий" 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 имя и версию ОС, под управлением которой работает JVM.

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.

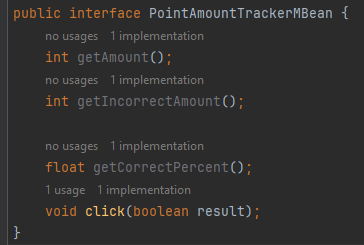
4. С помощью утилиты VisualVM и профилировщика IDE NetBeans, Eclipse или Idea локализовать и устранить проблемы с производительностью в [программе](https://se.ifmo.ru/documents/10180/189115/HttpUnit.tar.gz/7bf1032e-d16e-be85-c71b-dbe73c0178ba?t=1651168887037&download=true). По результатам локализации и устранения проблемы необходимо составить отчёт, в котором должна содержаться следующая информация:

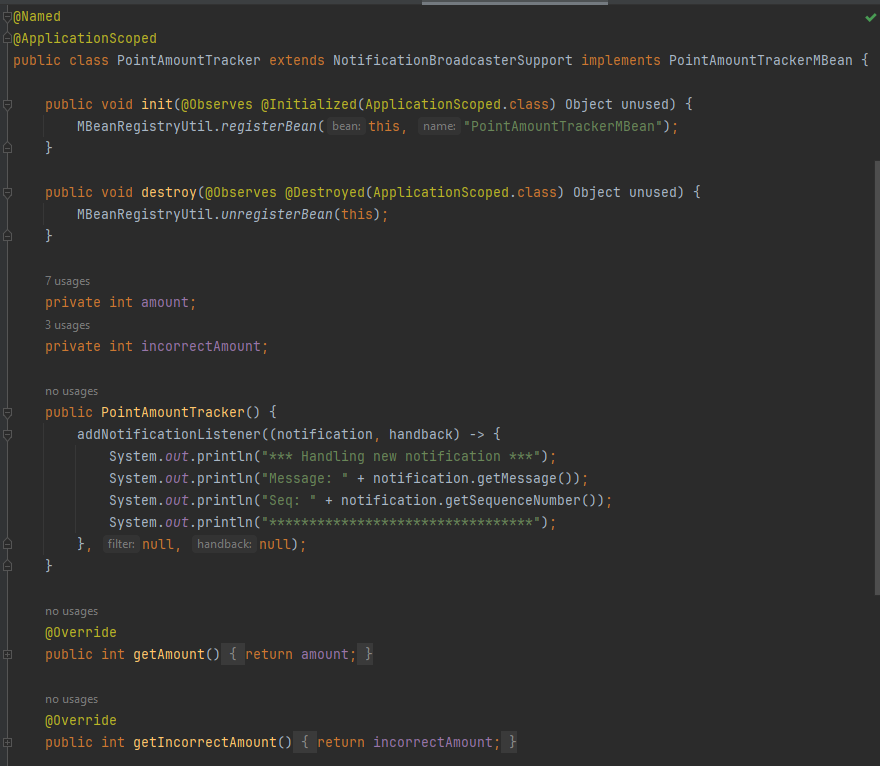
* Описание выявленной проблемы.
* Описание путей устранения выявленной проблемы.
* Подробное (со скриншотами) описание алгоритма действий, который позволил выявить и локализовать проблему.

Студент должен обеспечить возможность воспроизведения процесса поиска и локализации проблемы по требованию преподавателя.

# 

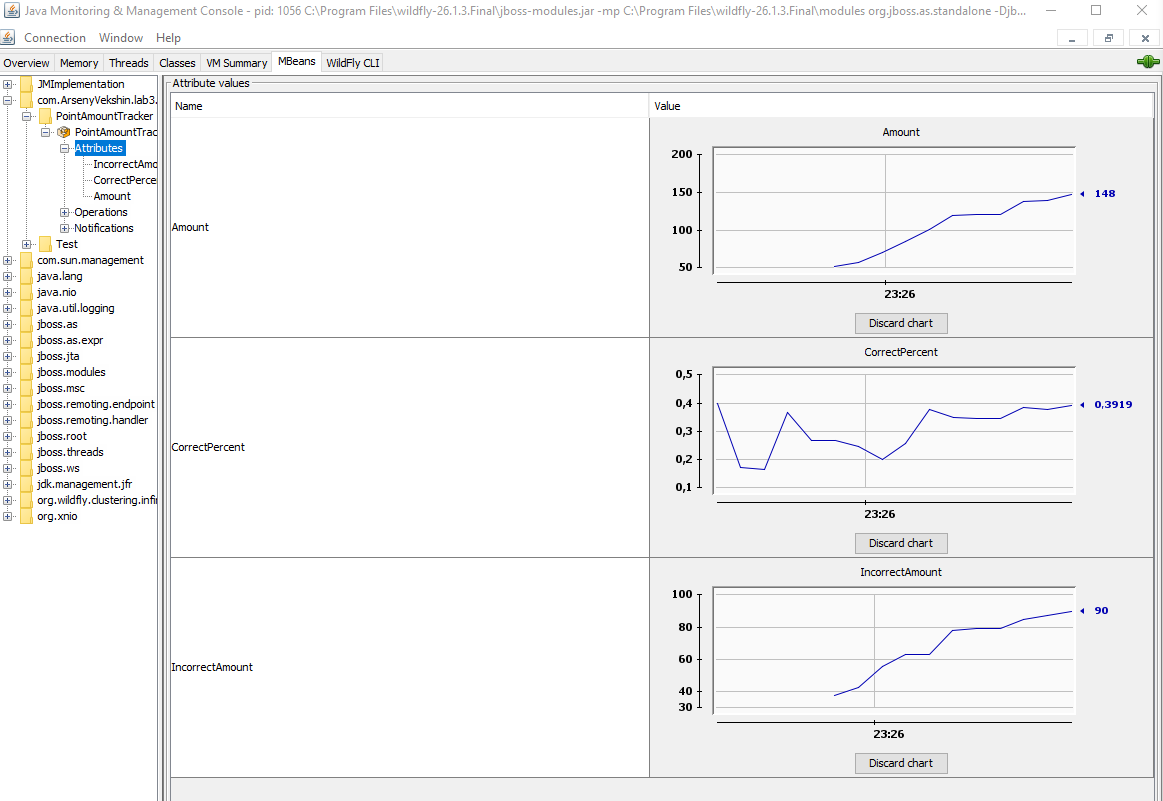
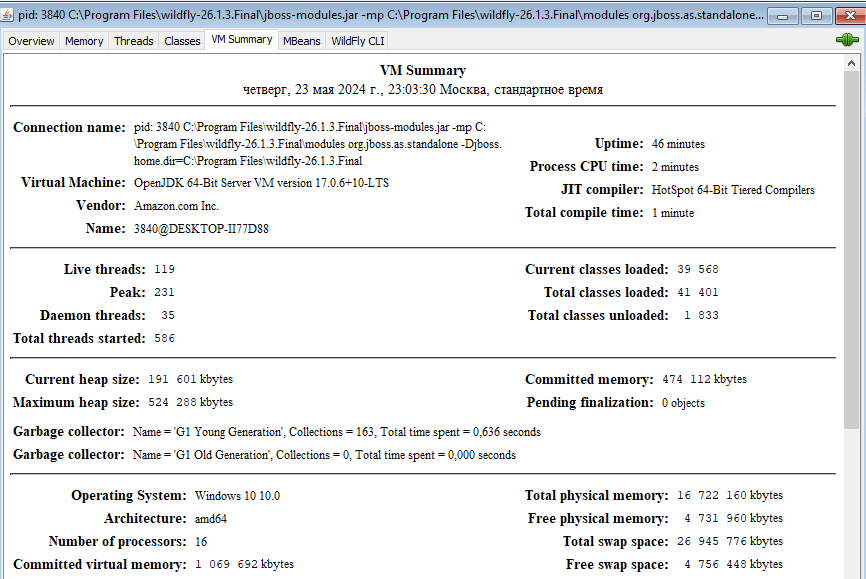
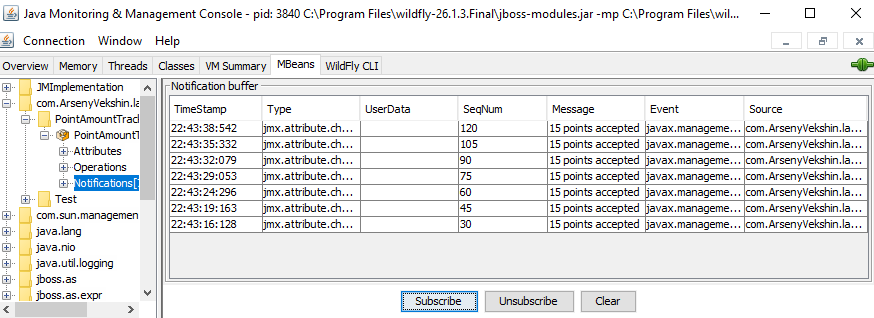
# **MBeans**



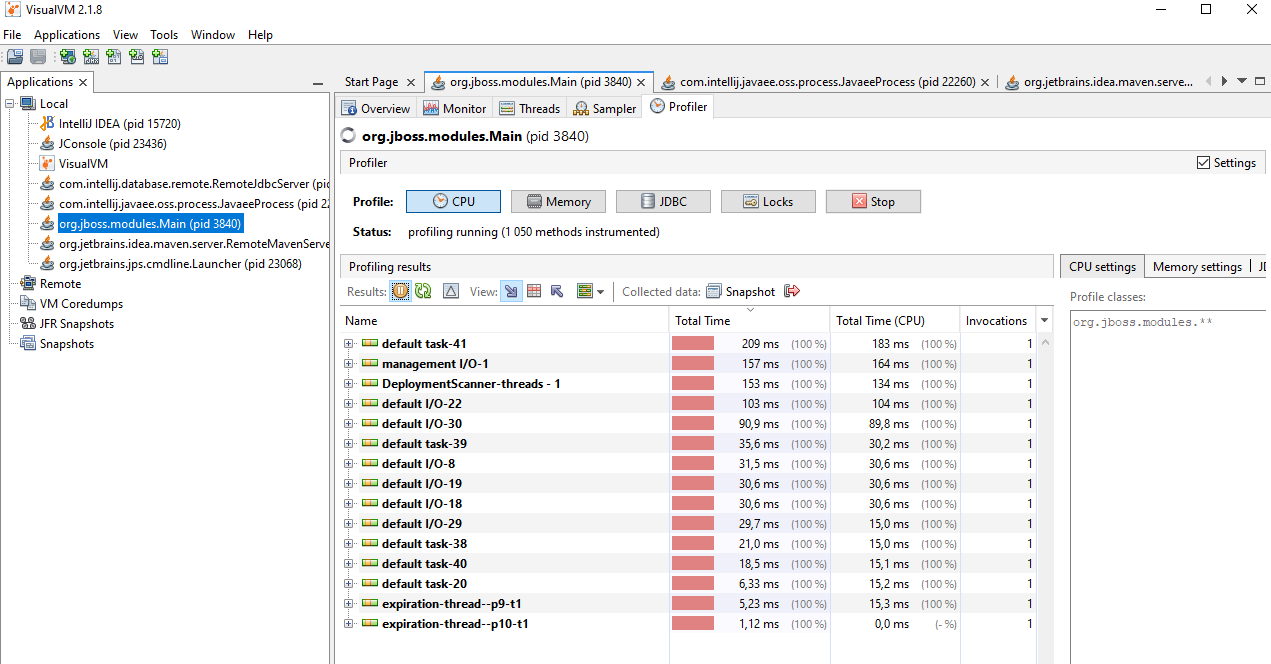


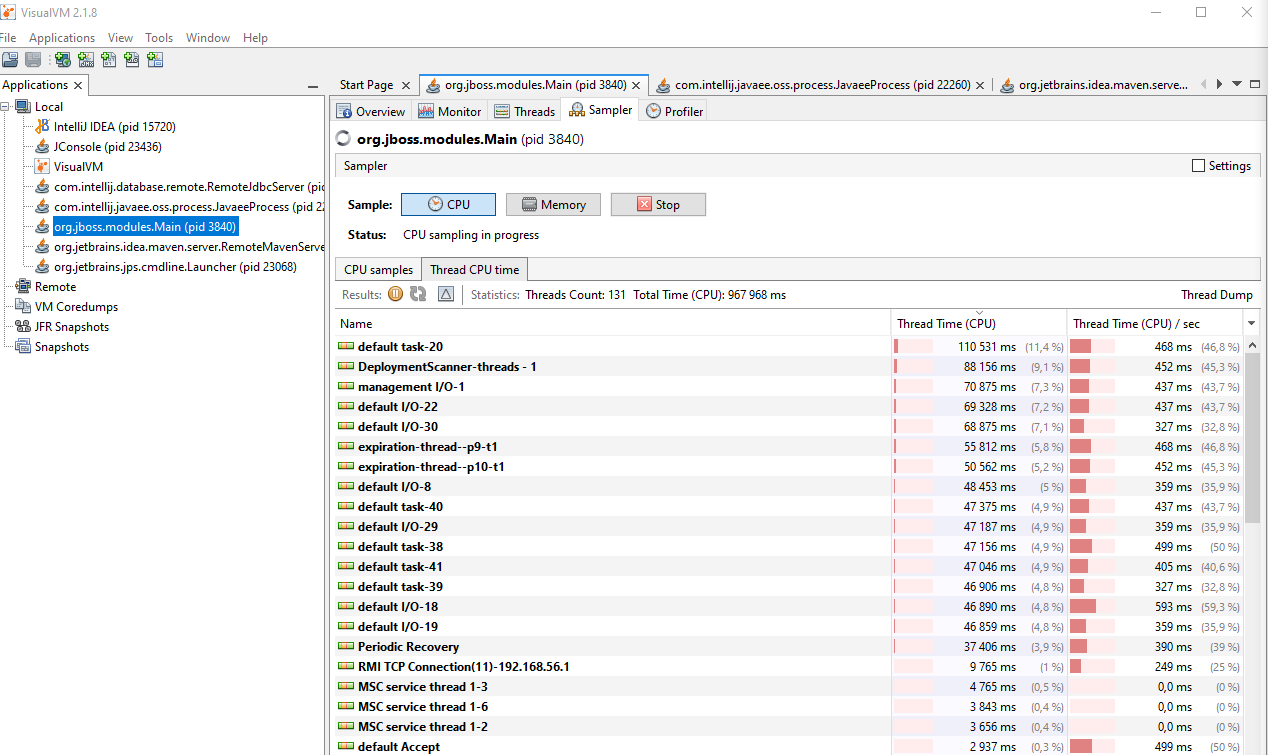


# **JConsole**



# **VisualVM**

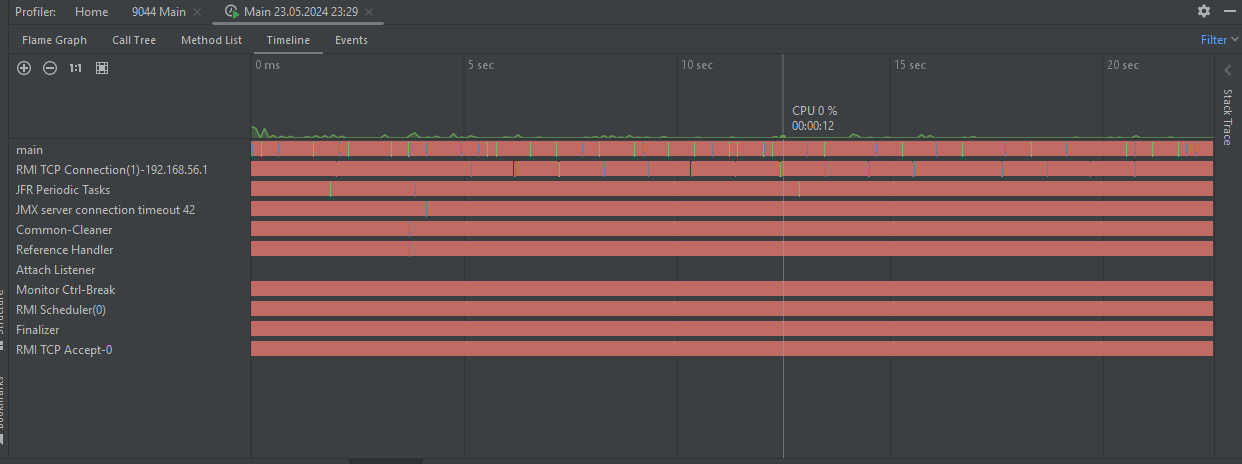
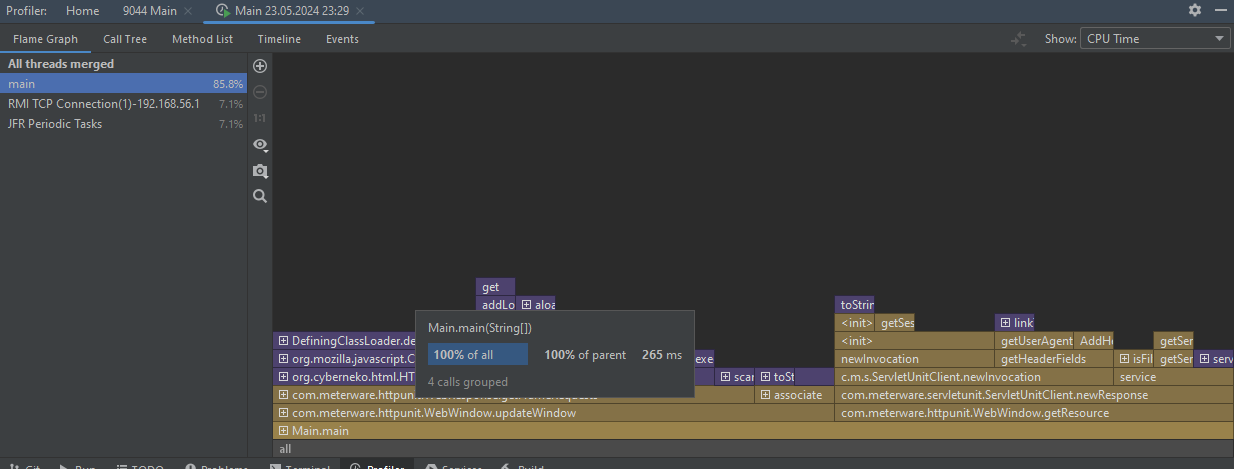
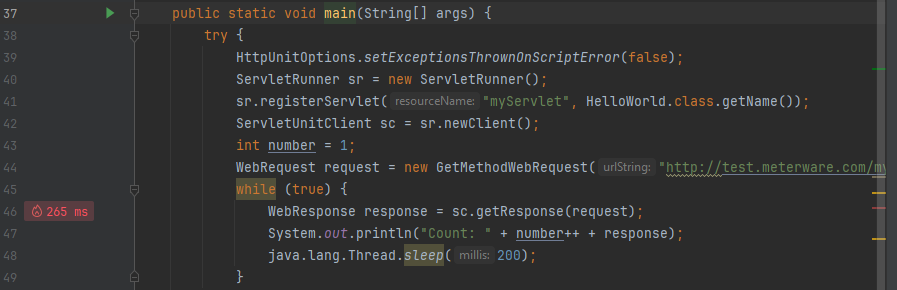


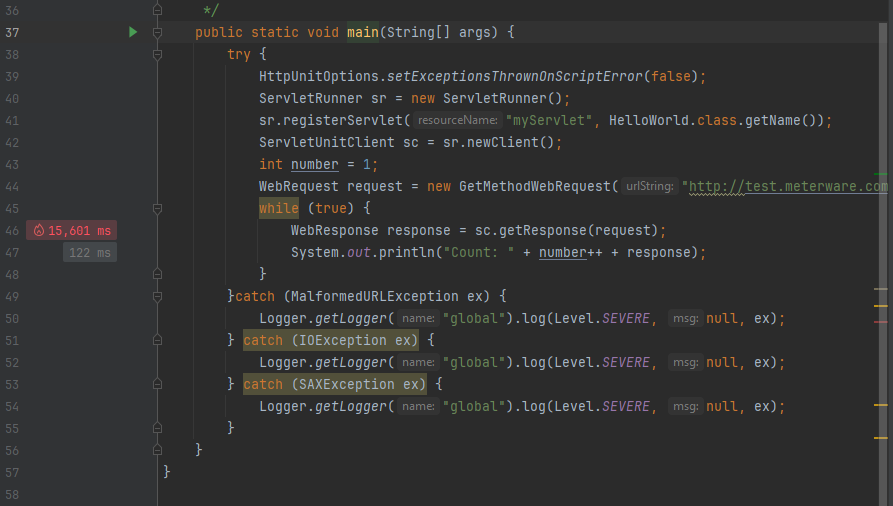
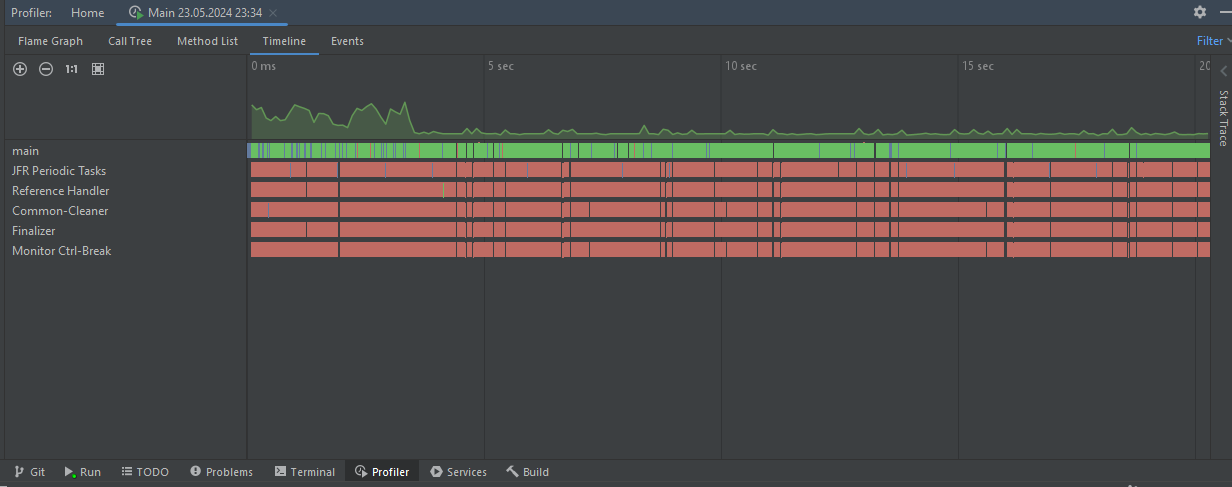


# 

# **Профилировщик**

## Производительность

1. Запустим профилировщик  
     
   
2. 

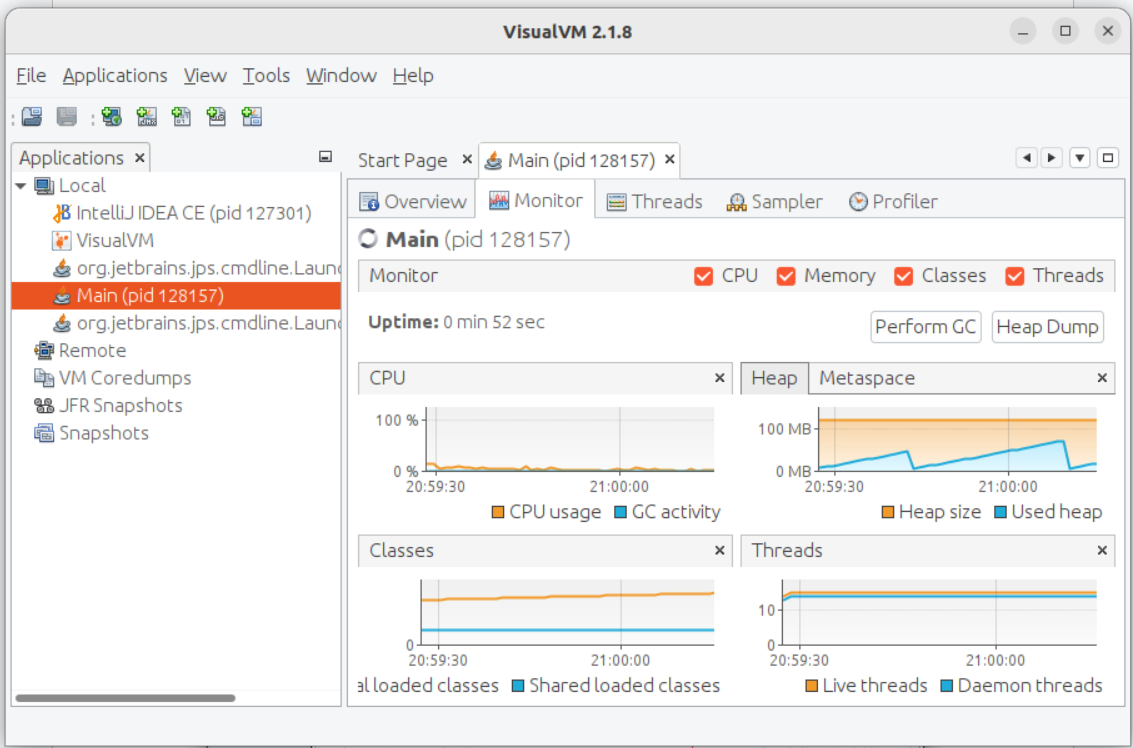


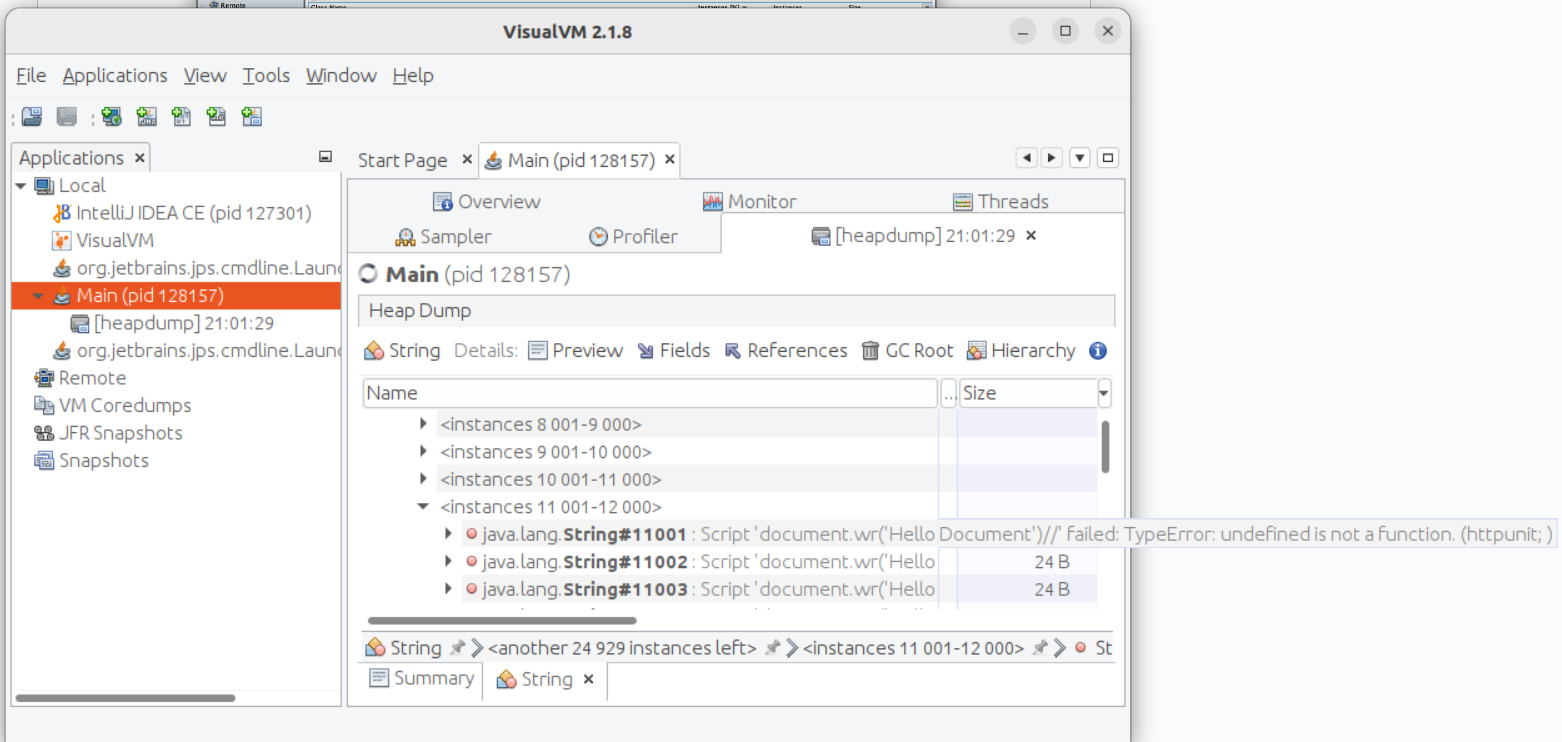
1. Profit

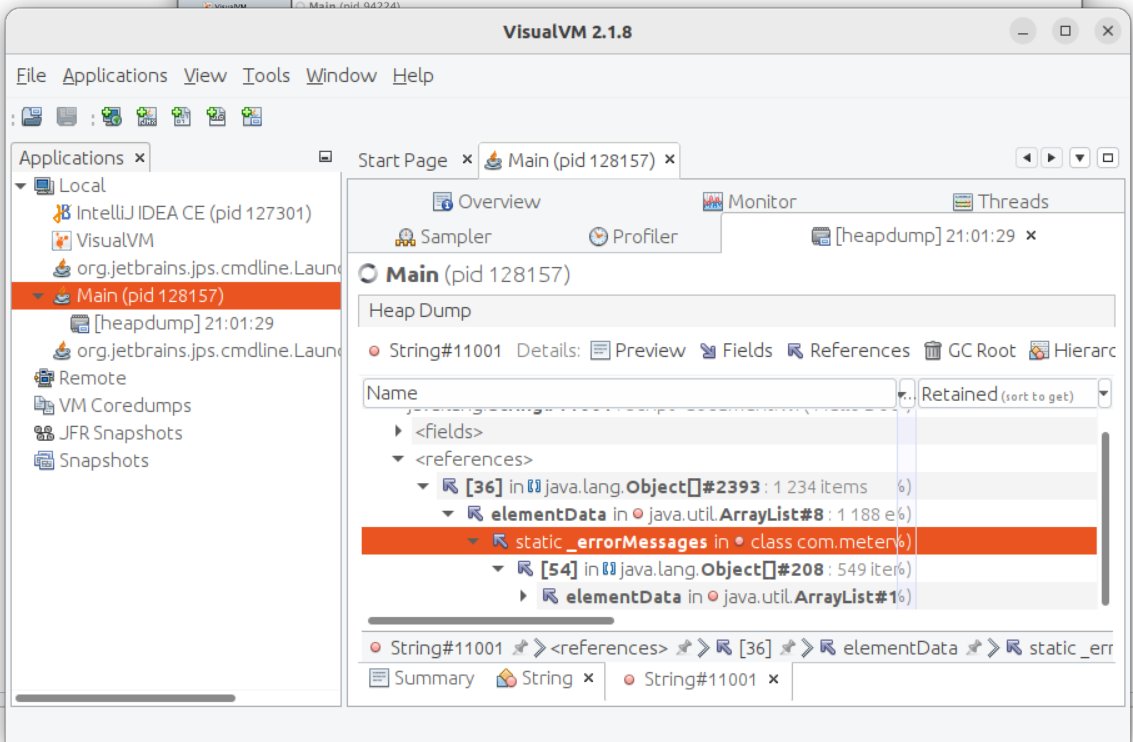
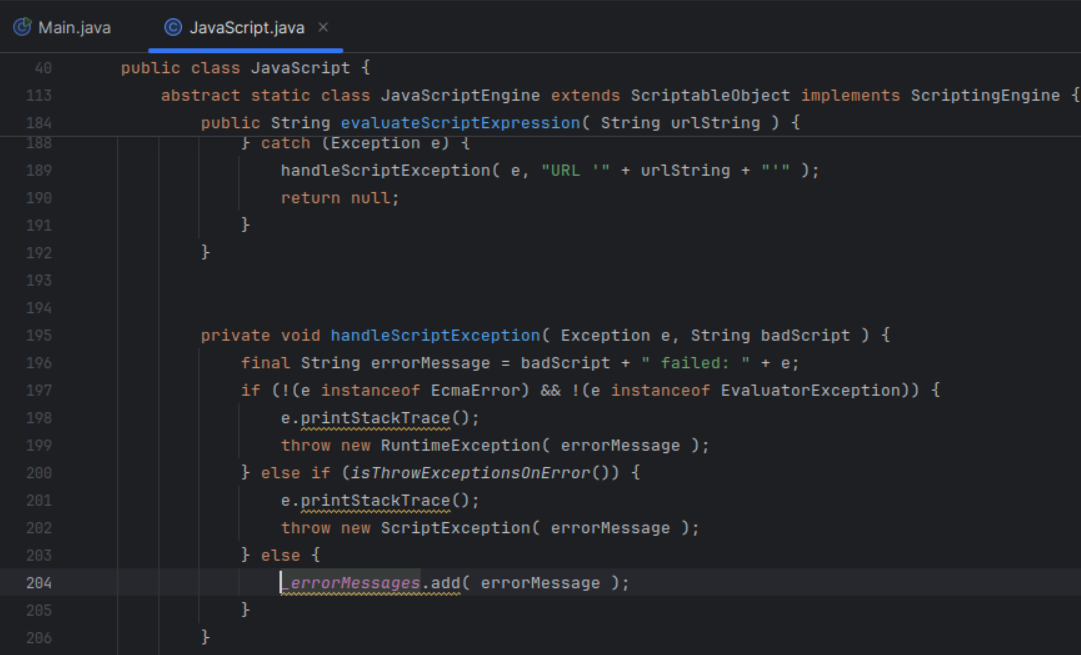
## 

## Утечки памяти

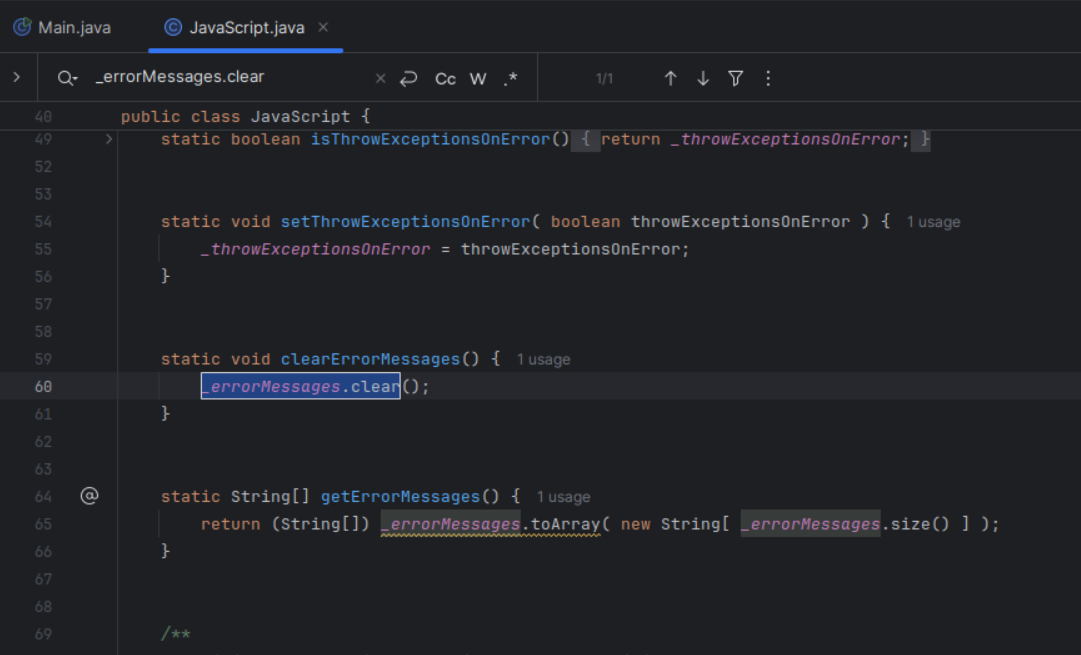
Уменьшим максимальный размер кучи до 100 МБ.

Память постоянно растет, а затем сборщик удаляет большое количество неиспользуемых объектов. Найдем их в Heap Dump

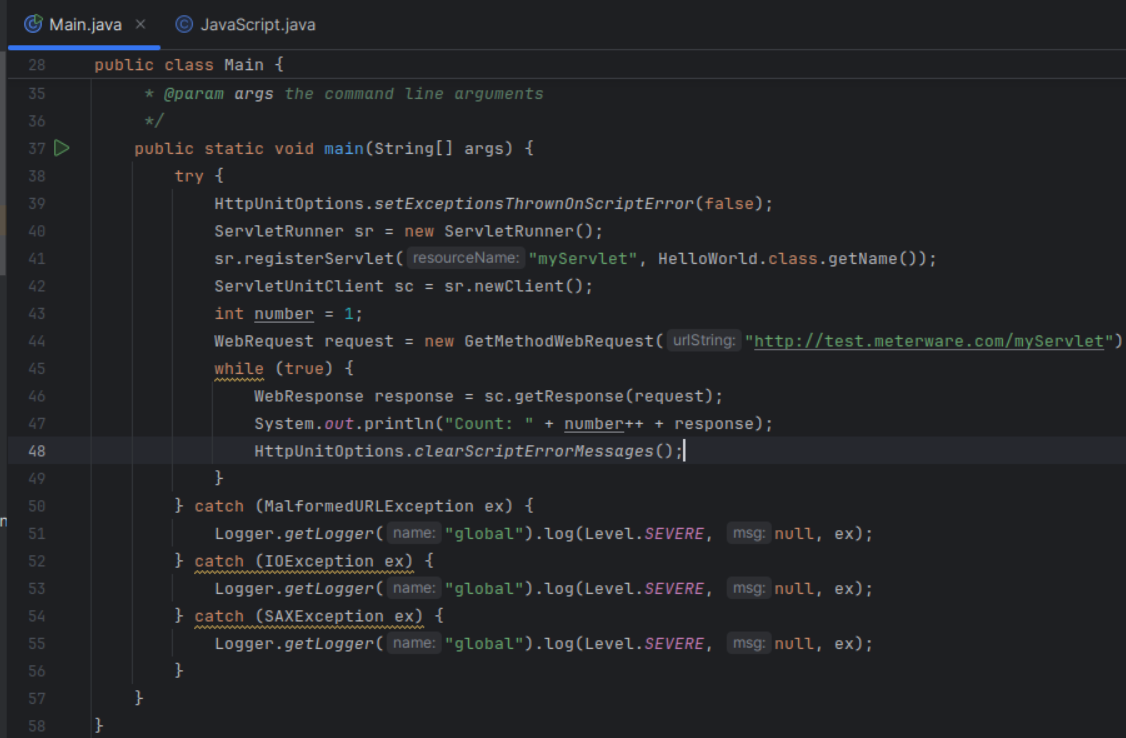
Большое количество строк одинакового содержания. Найдем место их создания



Строки сохраняются здесь…

а затем удаляются в других частях программы.

Будем очищать список строк в цикле:

Проверим результат

# 

Утечка устранена

# **Вывод**

В ходе выполнения лабораторной работы мы ознакомились с MBeans, утилитами JConsole, VisualVM и получили ценный и незабываемый опыт по оптимизации плохо оптимизированного приложения

.∧\_∧

(・ω・)つ━☆・\*。

⊂ ノ ・ ゚+.

しーJ °。+ \* ́ ̈)

.· ́ ̧.·\* ́ ̈)

( ̧.· ́ ( ̧.·'\* ☆ JSF всё ещё пора на свалку истории