Санкт-Петербургский национальный исследовательский университет  
 информационных технологий, механики и оптики  
Кафедра информатики и прикладной математики

Основы программной инженерии  
Лабораторная работа №4  
“Фреймворк MBeans, Мониторинг и Профилирование программ”

Выполнил: Шкаруба Н.Е.  
Проверила: Харитонова   
группа: P3218  
год: 2015

Требования

1. Для своей программы из [лабораторной работы #4](http://helios.cs.ifmo.ru/disciplines/iaps#lab4) по дисциплине "Программирование интернет-приложений" реализовать:

* MBean, считающий общее число установленных пользователем точек, а также число точек, попадающих в область. В случае, если количество установленных пользователем точек стало кратно 15, разработанный MBean должен отправлять оповещение об этом событии.
* MBean, определяющий средний интервал между кликами пользователя по координатной плоскости.

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

* Снять показания MBean-классов, разработанных в ходе выполнения задания 1.
* Определить значение переменной classpath для данной 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 потока, потребляющего наибольший процент времени CPU.

4. Получить HeapDump, и с помощью утилиты **VisualVM** локализовать и устранить "утечку памяти" в указанной программе ( Её смотреть в соответственном пункте выполнения ниже)

Результат (Найти более удобное слово)

1. Немного полезной теории:

**Профилирование** – Сбор характеристик работы рограммы или системы с целью их дальнейшей оптимизации.   
**Характеристики**:

* Время выолнения строк кода
* Количество вызовов функций
* Дерево вызовов
* Hotspots
* Доля параллелизма
* Загрузка CPU \ сути \ шины доступа
* Количество промахов в кэше
* И прочее.

**MBean** это:

* Последовательность аттрибутов, которые можно изменять
* Последовательность вызывемых операций
* Metadata

**JMX specification** разделяет Mbean’ы на

* Standart MBean
* Dynamic MBean
* Open MBean
* Model MBean
* MXBean

1. Исходный код разработанных MBean-классов и сопутствующих классов
2. Мониторинг написанных Bean’ов с помощью Jconsole
3. Профилирование написанных Bean’ов с помощью VisualVM
4. Поиск утечки памяти с помощью VisualVM
5. Общий вывод по работе