Университет ИТМО

Факультет программной инженерии и компьютерной техники

Направление подготовки 09.03.04 Программная инженерия

Дисциплина «Методы и средства программной инженерии»

**Отчет**

По лабораторной работе №4

Вариант 1780

Выполнили:

Янь Пэйсинь

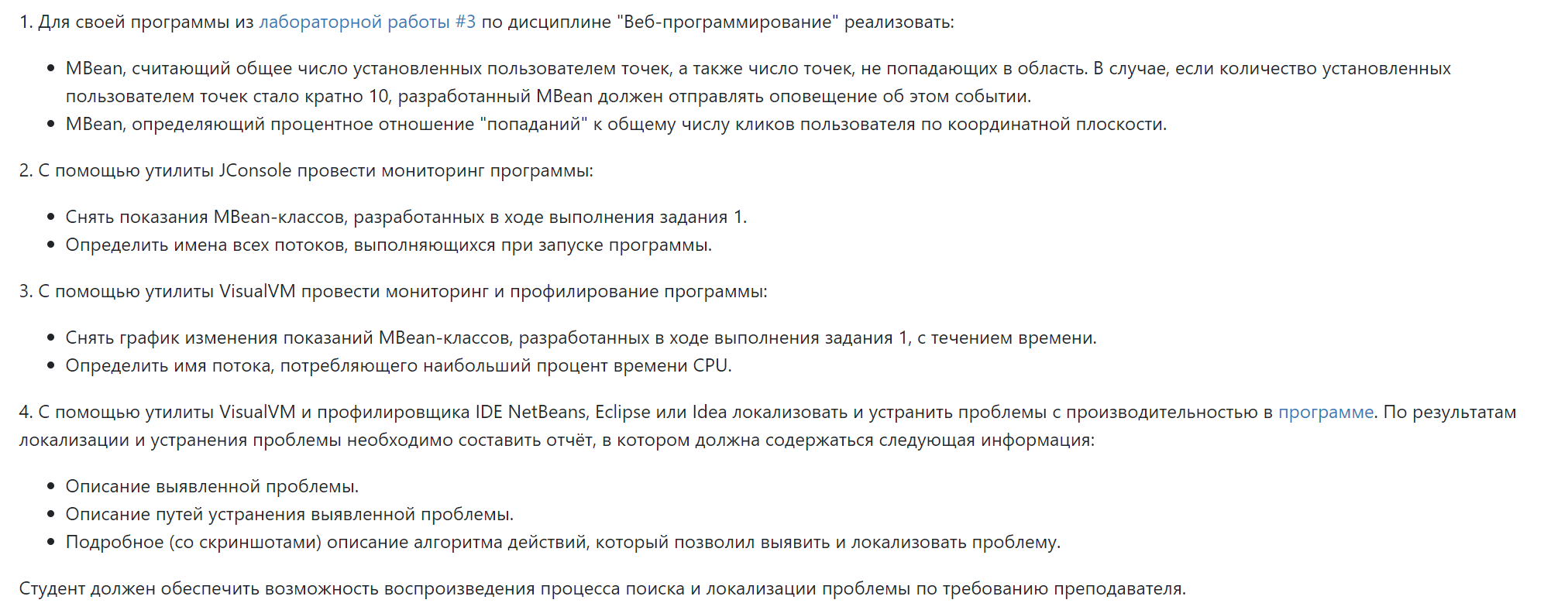
Е Хэн

Преподаватель:

Цопа Евгений Алексеевич

Санкт-Петербург, 2023 г.

Задание:



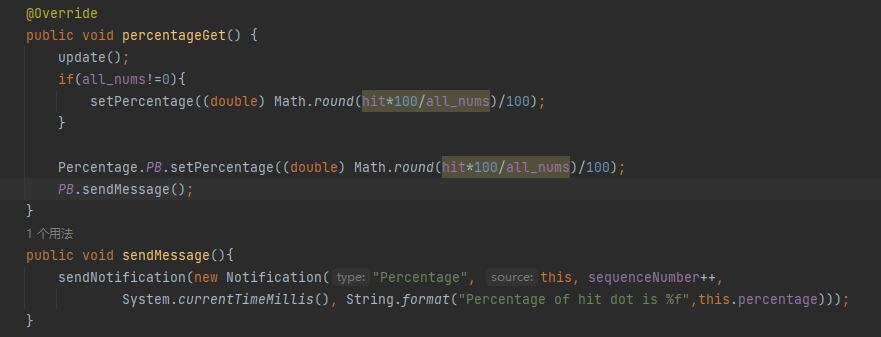
**Исходный код:**

**(<https://github.com/LiaoYihong-1/Method-4>)**

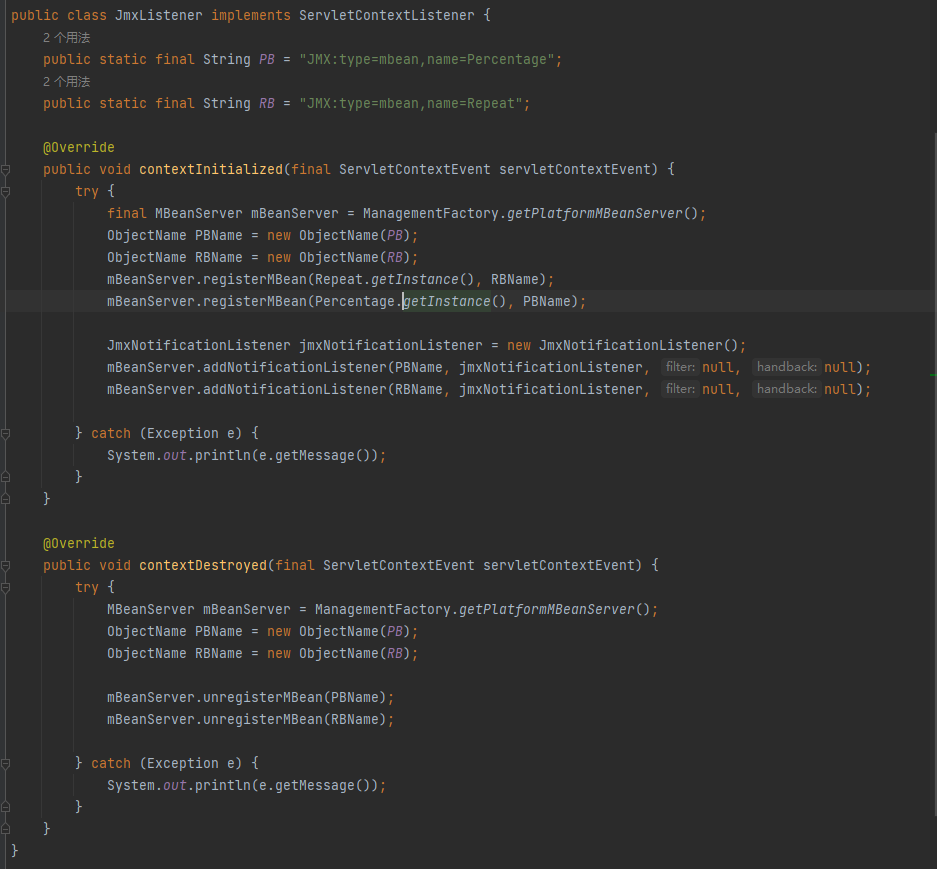
1. Mbean, который проверяет если количество установленных пользователем точек стало кратно 10



1. Mbean, определяющий процентное отношение “попаданий” к общему числу кликтов



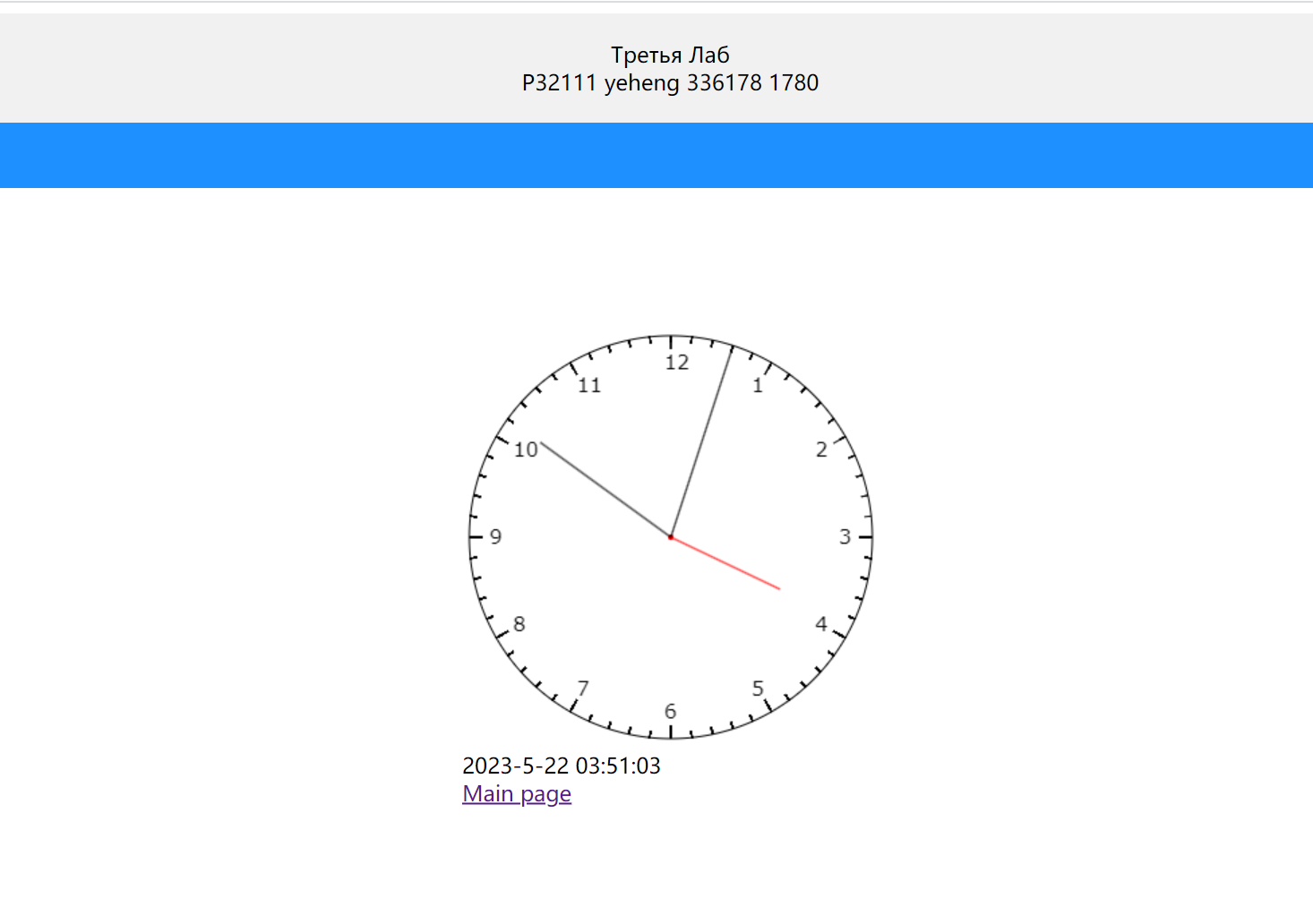
1. Читает написанного Mbean

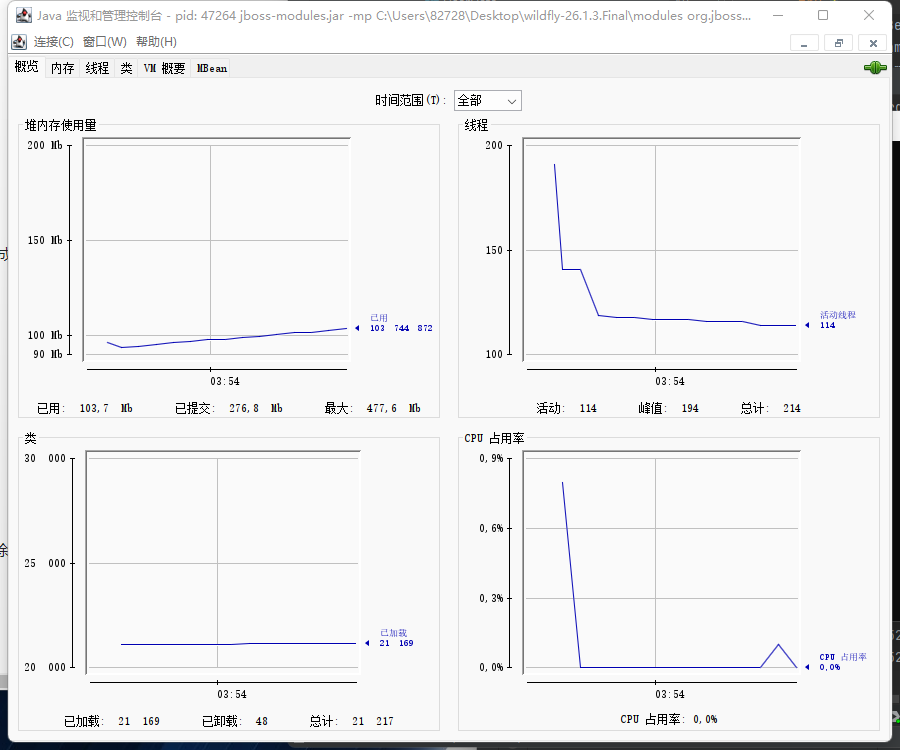


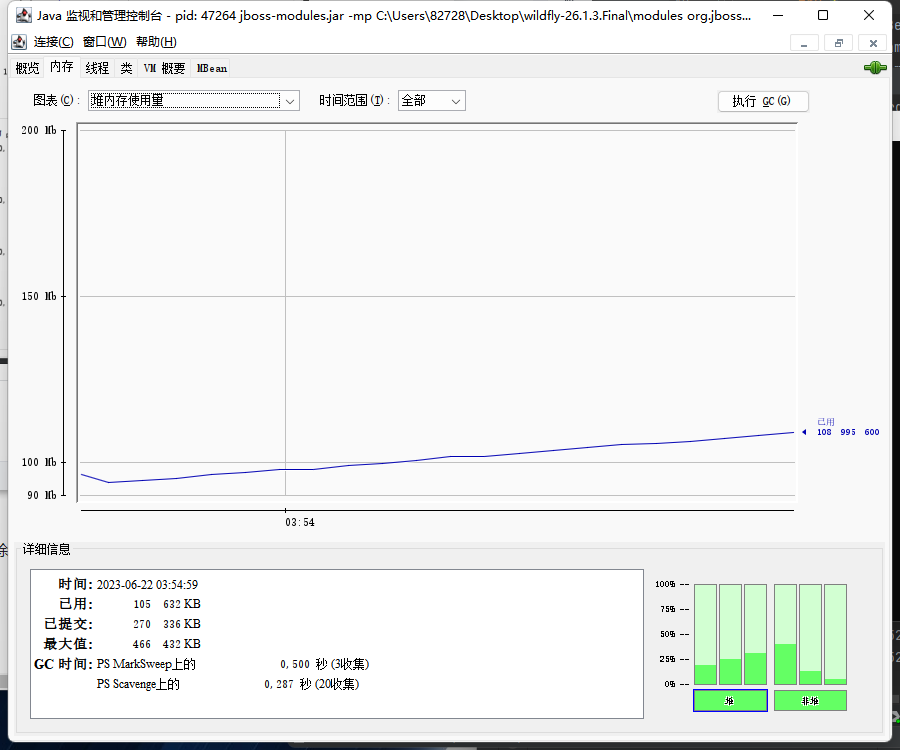
**Выполнение задач:**

Чтение количества классов(Jconsole)

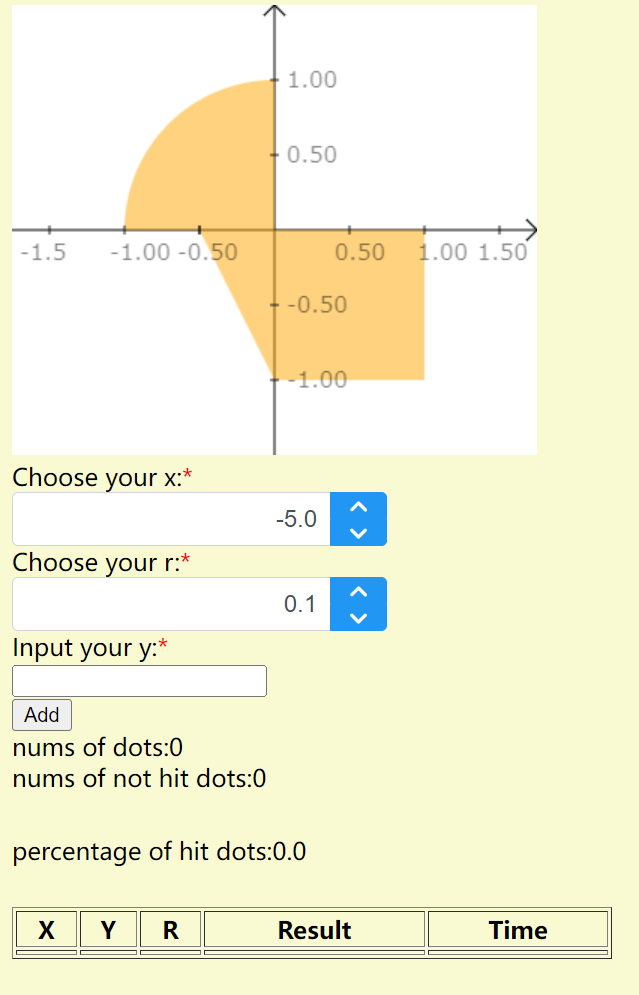
Первая страница:

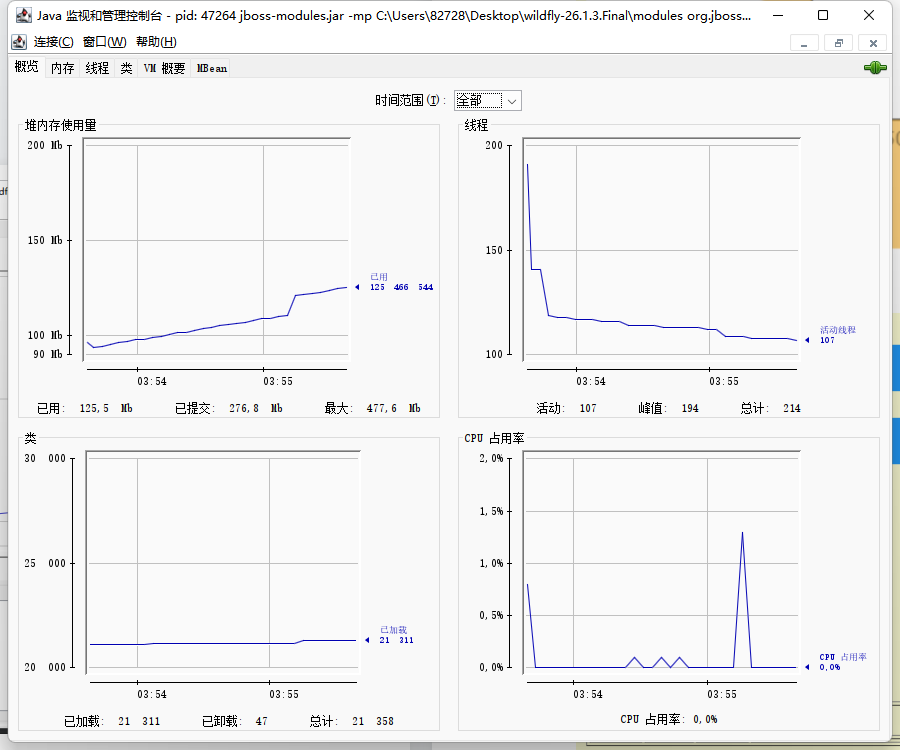


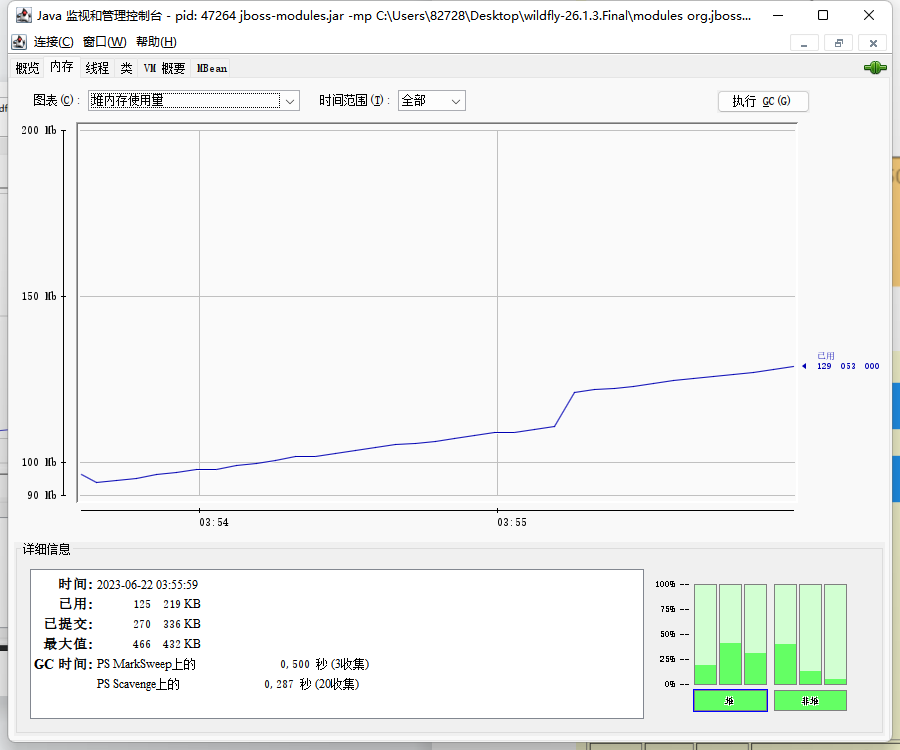




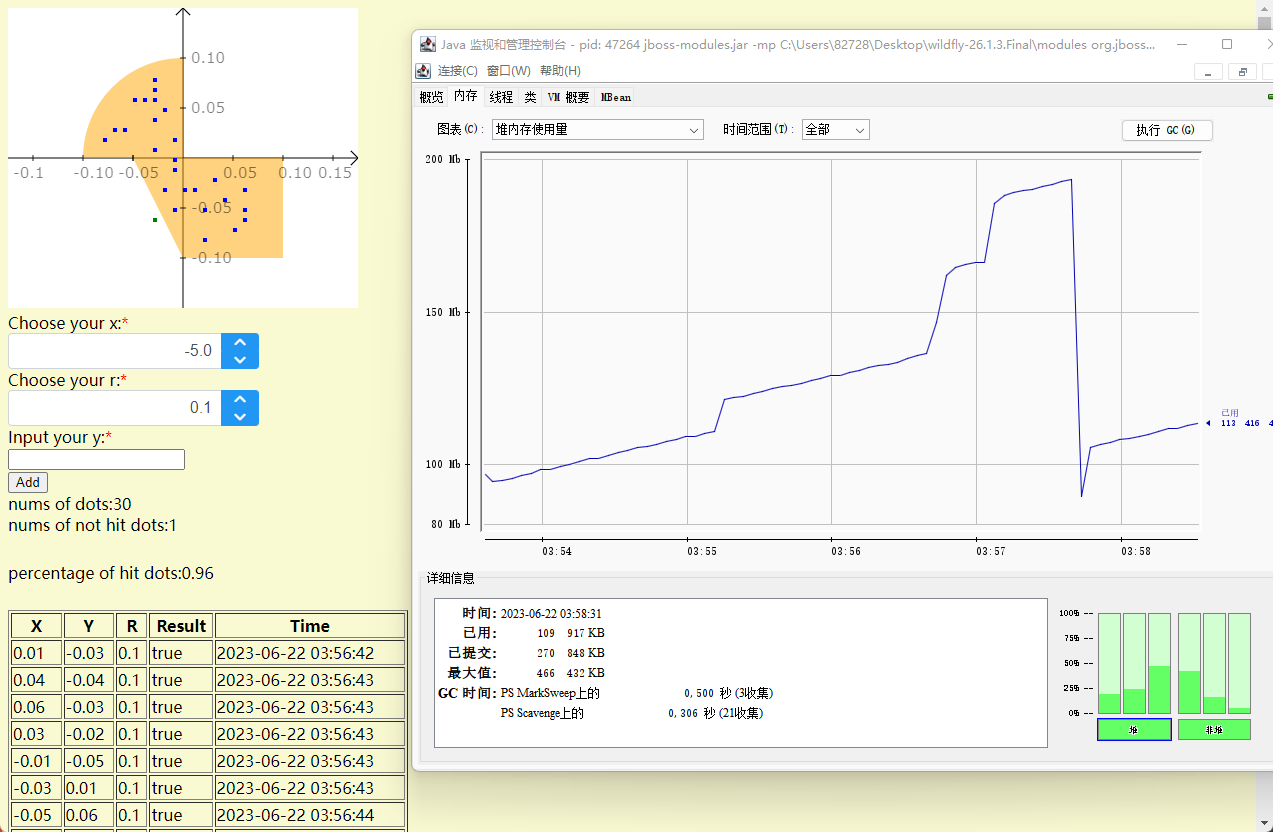
Переход на вторую:

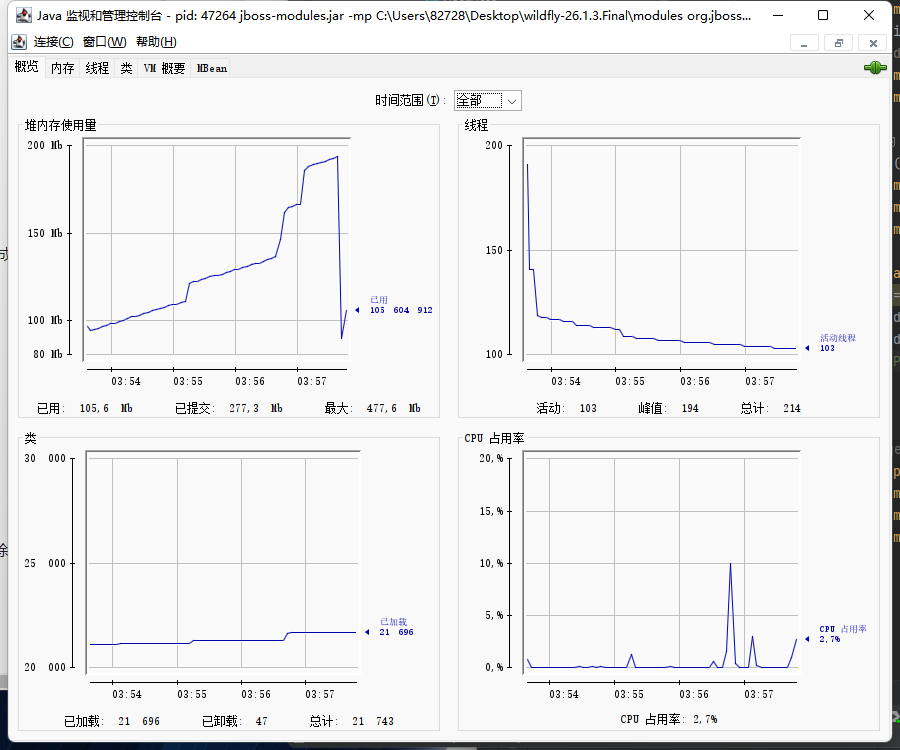


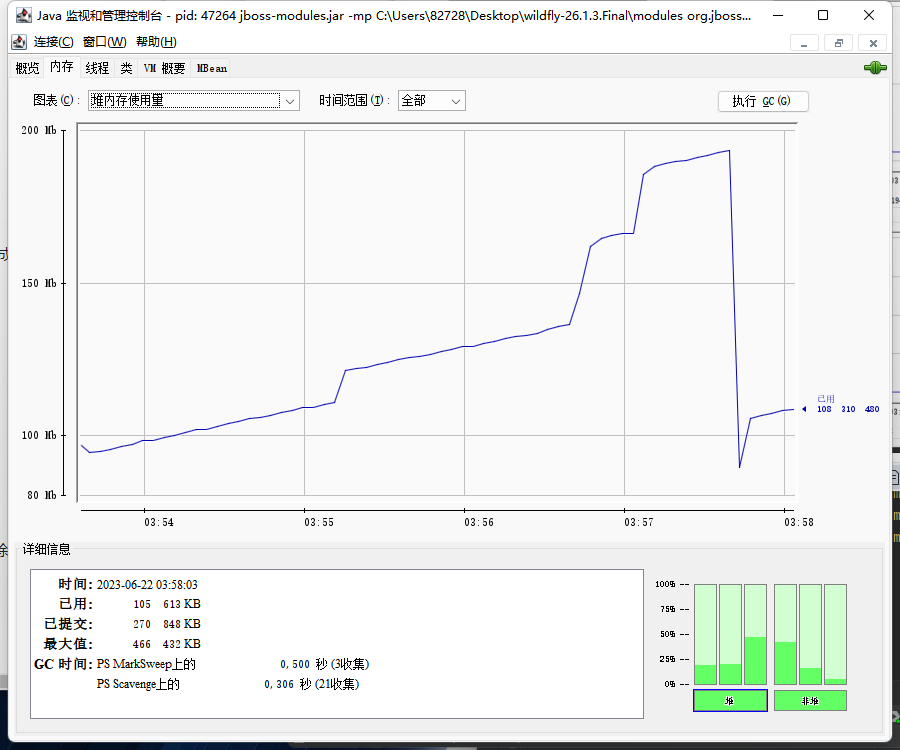




Главная страница работает:

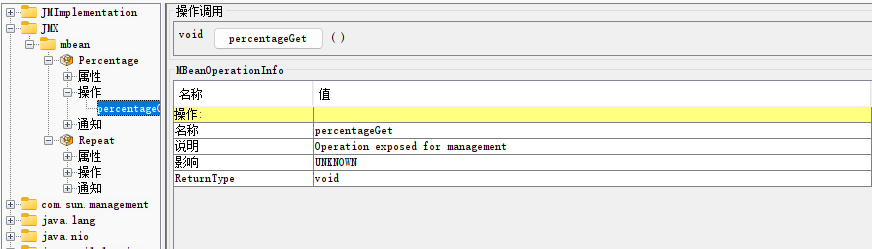


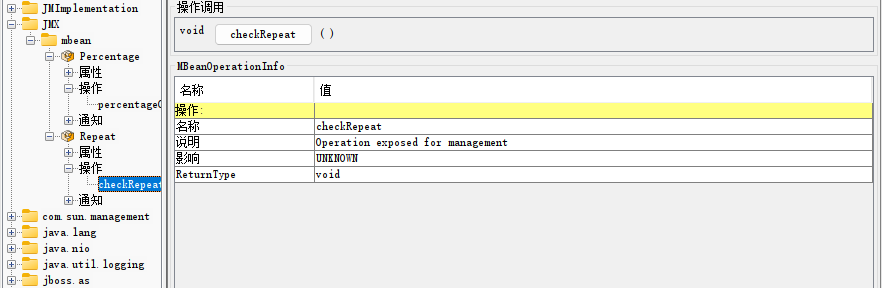




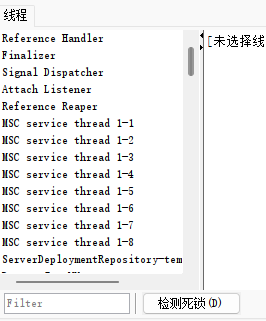
Видимо что в первых, в любом случае хотя мы нечего не делаем тут нагрузка памяти повышает с примерно 102 MB . А когда нагрузка слишком высока, он сам сразу уменьшается. Приводится что обычно нагрузка не больше 200 MB. Это потому что там происходит GC. Поскольку в Jconsole не можем проверять что-нибудь о GC, пока доказывть не можем. И иследовательнно, что когда пользователь использует программу, нагрузка значительно повышает. В том случае главная восходящая часть в heap. Одновременно, количество загруженных классов почти не изменяется, когда мы нечего не делать. И в состоянии работы главной страницы, оно увеличается(около 900 больше, измение которого не большое сравняя с изменением нагрузкой памяти). Количество живых процессов мало спустя некоторое время(около 100).

**Чтение Mbean(Jconsole):**

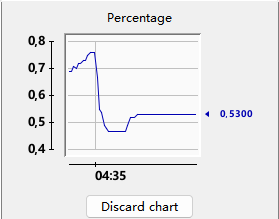




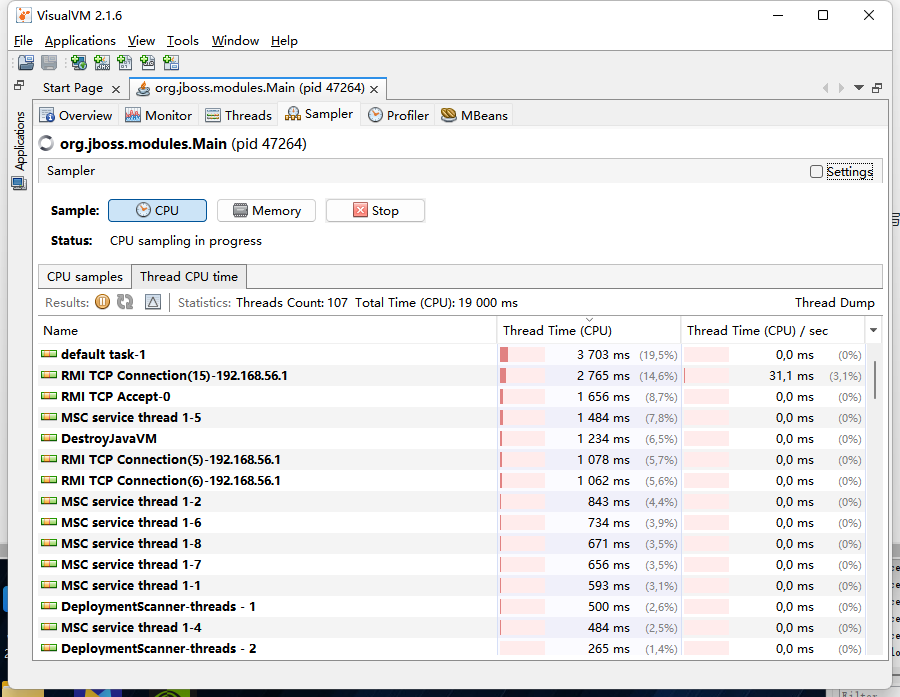
Определить имена всех потоков, выполняющихся при запуске программы.



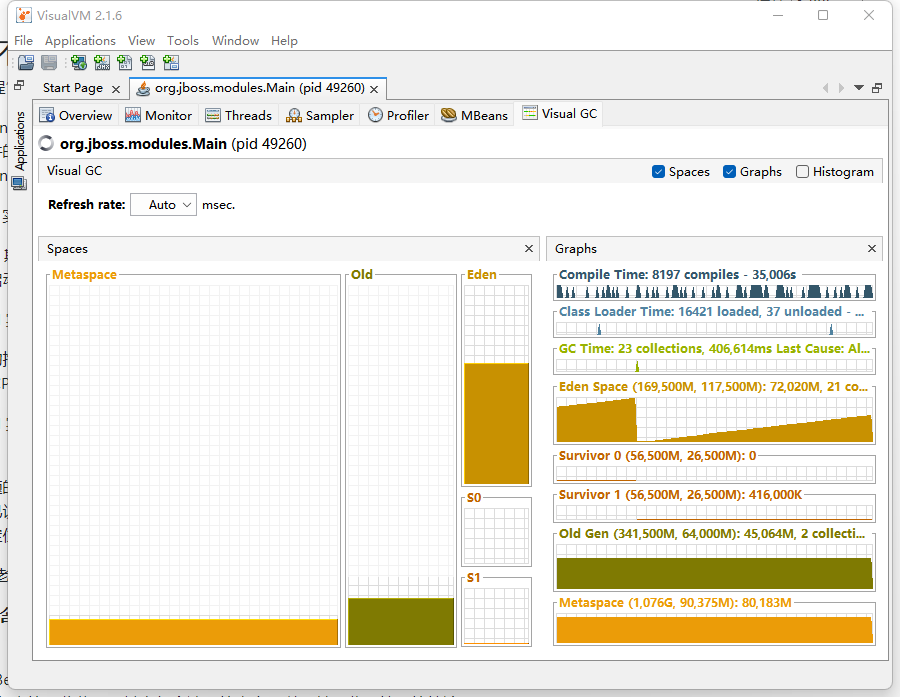
Снять график изменения показаний MBean-классов, разработанных в ходе выполнения задания 1, с течением времени.

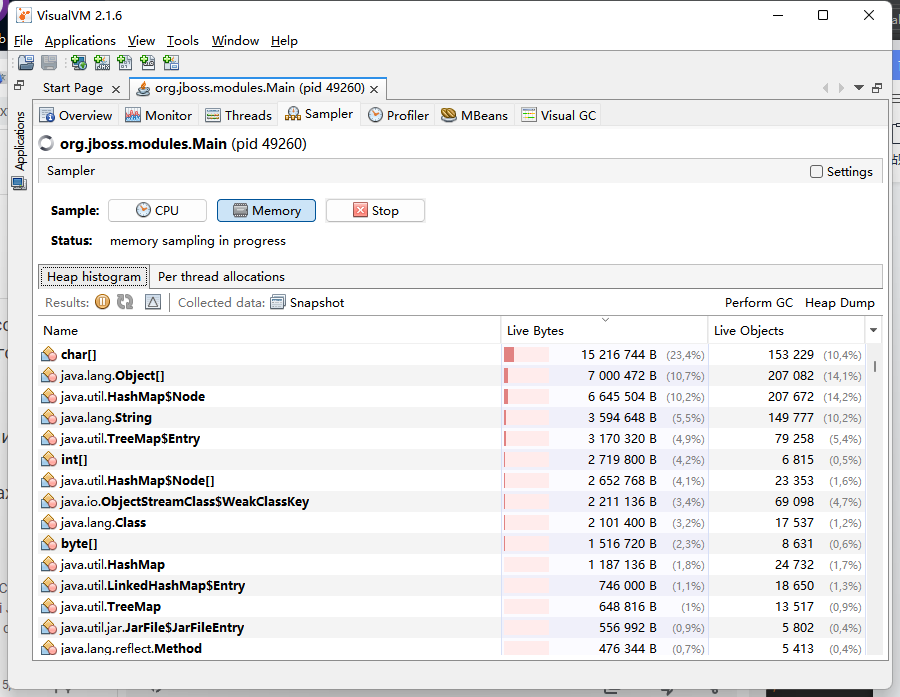


Cpu time



4.





Здесь по-первых видим что если нагрузка слишком высока, то происходит GC. Во-вторых, класс char[], объекты которого занимают наибольший объём памяти JVM

Учетка памяти:

Запускал программу, чтобы она работала минут 9. Там

5 old collections и их нагрузки нормально изменяется. Кажется

что учетки нет.

