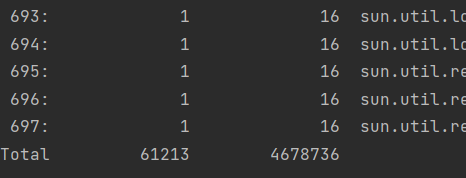
**Jmap**

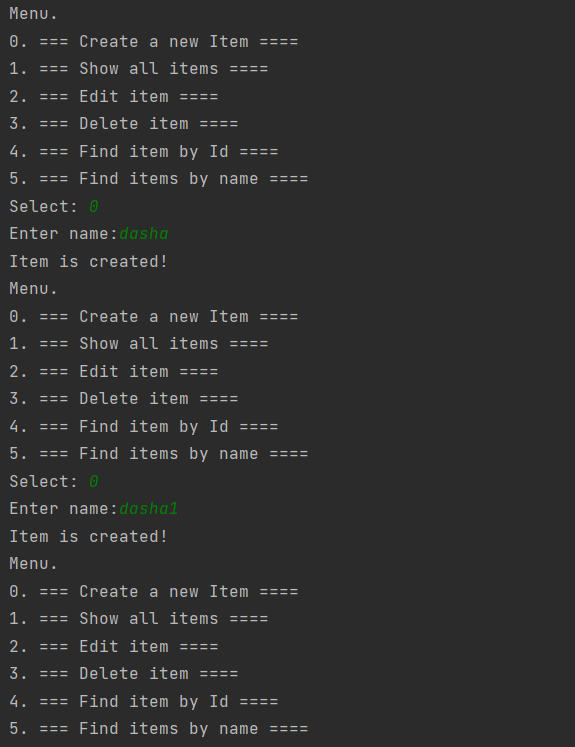
**-histo pid**

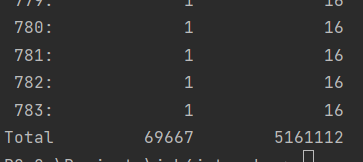
С помощью утилиты можно получить кол-во объектов, их размер и класс, к которому они принадлежат

1. До добавления объектов

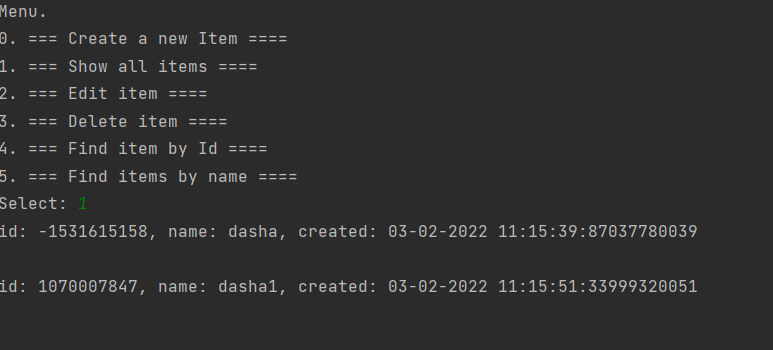


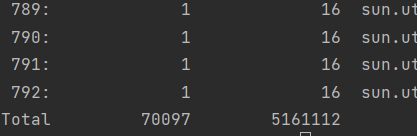
1. После добавления 2 объектов



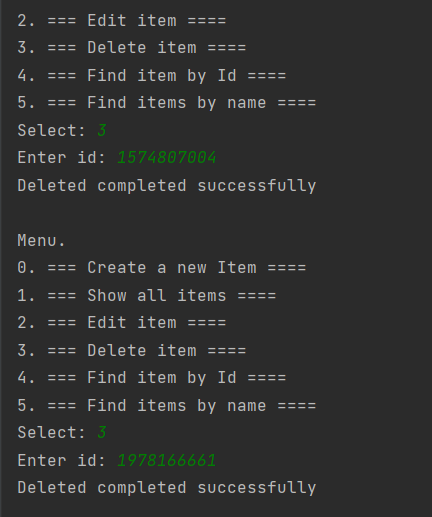


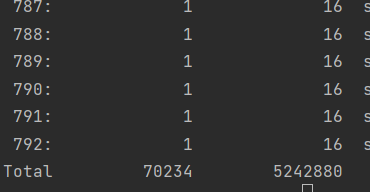
1. Выполнение команды show all items



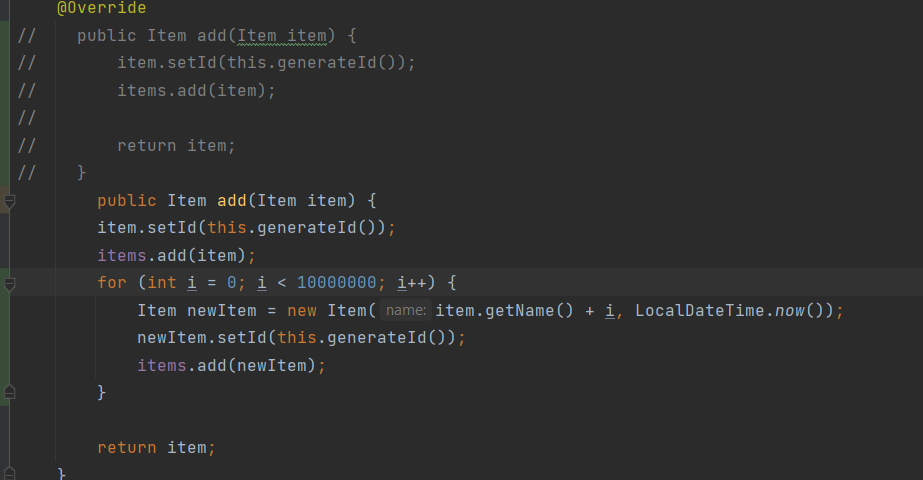


1. Удаление 2 объектов

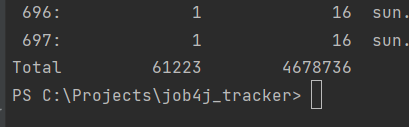




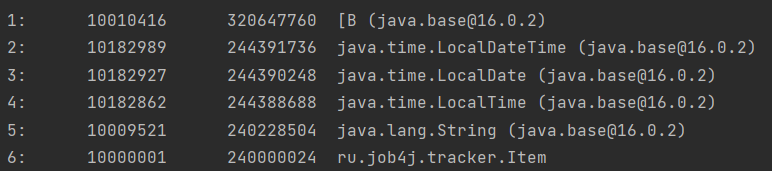
1. Выход за пределы памяти

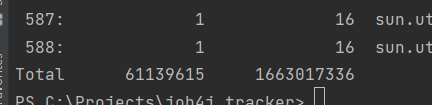


- до добавления

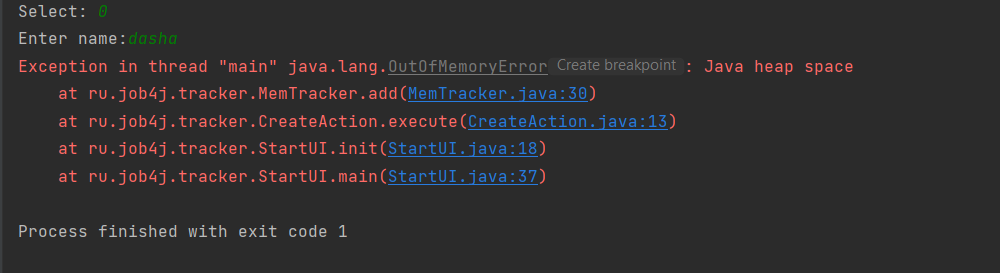


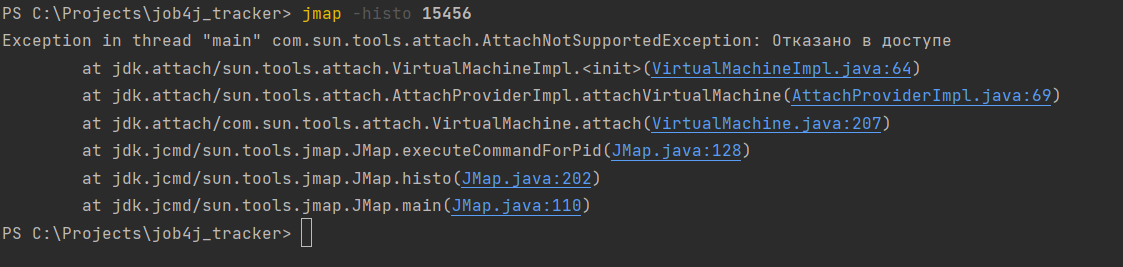
- после добавления





* Переполнение



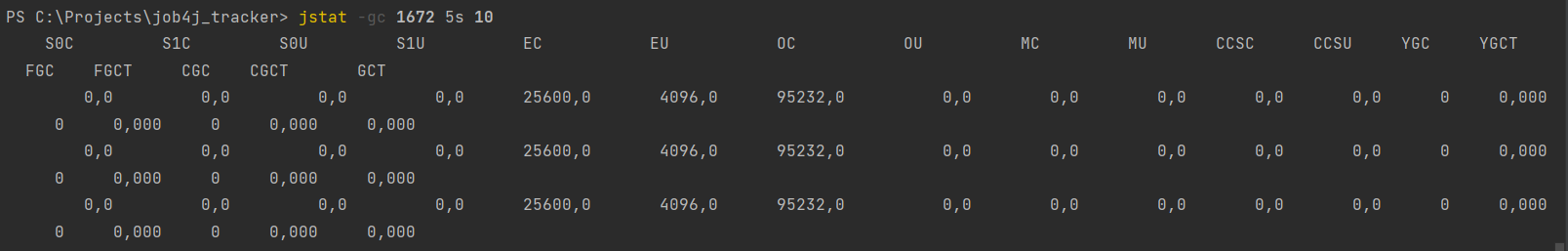


**Jstat**

Предоставляет информацию из всех отделов памяти

1. S0C: Current survivor space 0 capacity (KB).
2. S1C: Current survivor space 1 capacity (KB).
3. S0U: Survivor space 0 utilization (KB).
4. S1U: Survivor space 1 utilization (KB).
5. EC: Current eden space capacity (KB).
6. EU: Eden space utilization (KB).
7. OC: Current old space capacity (KB).
8. OU: Old space utilization (KB).
9. MC: Metaspace Committed Size (KB).
10. MU: Metaspace utilization (KB).
11. CCSC: Compressed class committed size (KB).
12. CCSU: Compressed class space used (KB).
13. YGC: Number of young generation garbage collection (GC) events.
14. YGCT: Young generation garbage collection time.
15. FGC: Number of full GC events.
16. FGCT: Full garbage collection time.
17. GCT: Total garbage collection time.

- До добавления объектов



1. **EC** Размер области Eden New Generation 25600 кбайт.

В этой области выделяется память под все создаваемые программой объекты. Жизненный цикл большей части объектов, к которым относятся итераторы, объекты внутри методов и т.п., недолгий.

1. **EU** В Eden используются 4096 кбайт
2. **OC** Размер Old Generation 95232 кбайт

Tenured Space хранит долгоживущие объекты. Когда данная область памяти заполняется, выполняется полная сборка мусора (full, major collection).

- После добавления объектов



1. **S1C** survivor space 1 = 33792 кбайт

Здесь хранятся перемещенные из Eden Space объекты после первой сборки мусора. Объекты, пережившие несколько сборок мусора, перемещаются в следующую сборку Tenured Generation.

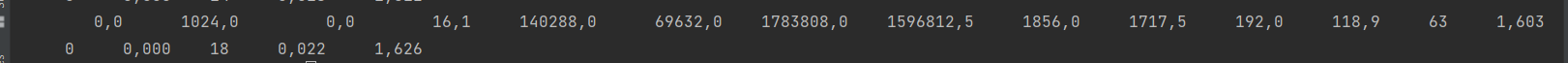
1. **EC** Размер области Eden New Generation = 201728
2. **EU** В Eden используются 169984 кбайт
3. **OC** Размер Old Generation 1049600
4. **OU** Использование Old Generation 737792
5. **MC** Гарантированный размер метапространства

Область памяти Permanent Generation используется виртуальной машиной JVM для хранения необходимых для управления программой данных, в том числе метаданные о созданных объектах.

The permanent generation holds all the reflective data of the virtual machine itself, such as class and method objects.

1. **MU** использование метапространства 1488
2. YGC: Количество сборки мусора молодого поколения = 17
3. YGCT Время сборки мусора молодого поколения = 0.845 с
4. FGC: Число полных сборок 4
5. FGCT: Время полной сборки 0.002
6. GCT: Total garbage collection time.0.848

- Состояние после отображения всех объектов



Исходя из статистики видим, что большая часть объектов перемистилась в Old generation

Количество сборки мусора молодого поколения = 63

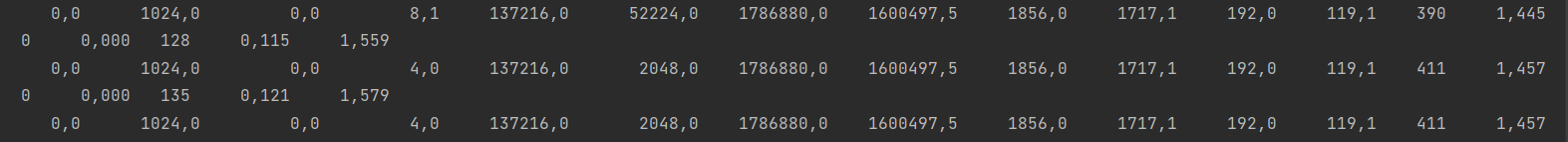
Время сборки мусора молодого поколения = 1.603 с

Число полных сборок 18

Время полной сборки 0.022

Total garbage collection time.0.848

**- Состояние при удалении**

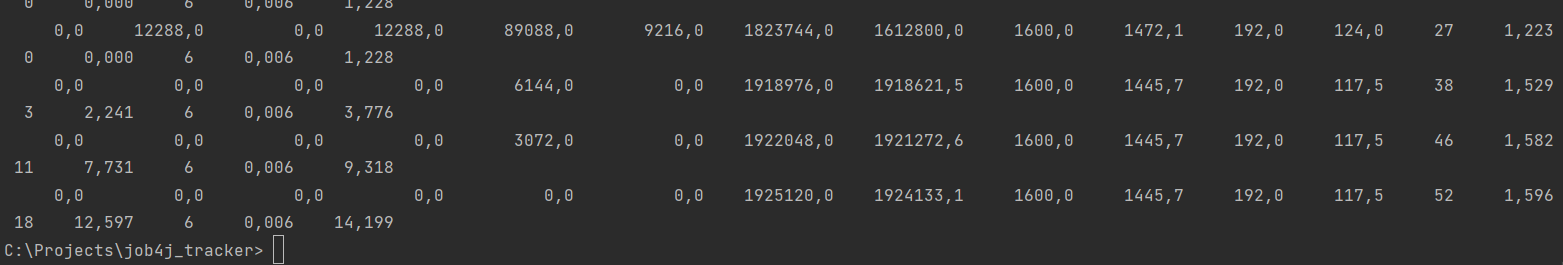
****

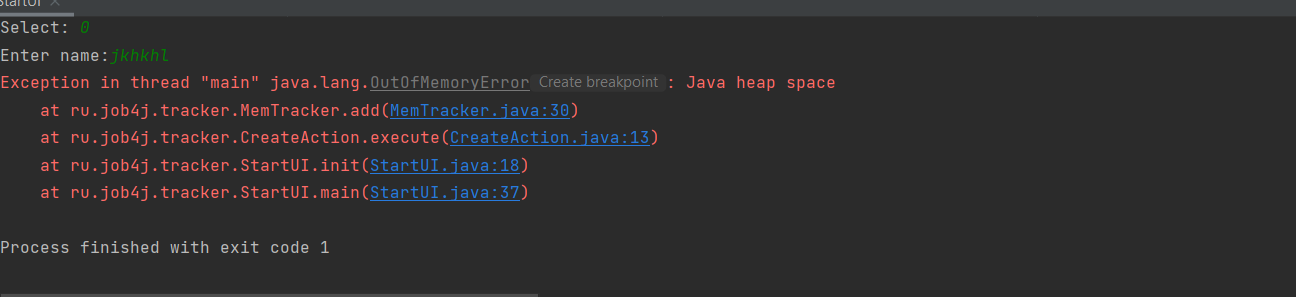
Уменьшились показатели EU MU

Увеличились – время сборки мусора, кол-во сборок

- Выход за пределы памяти

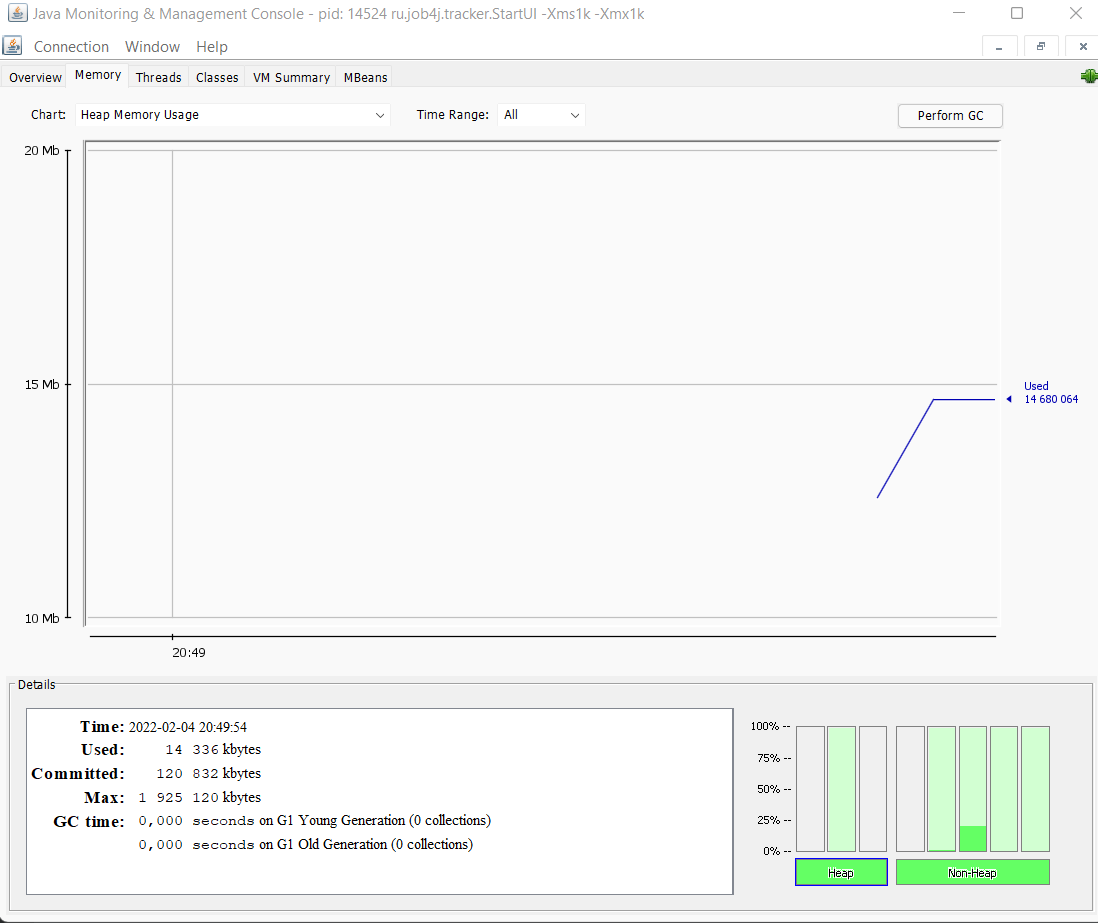
При появлении ошибки закончился сбор статистики





Jconsole

* Подключаемся с помощью команды jconsole
* Состояние памяти до добавления объектов



Used – Количество памяти, которое на данный момент используется

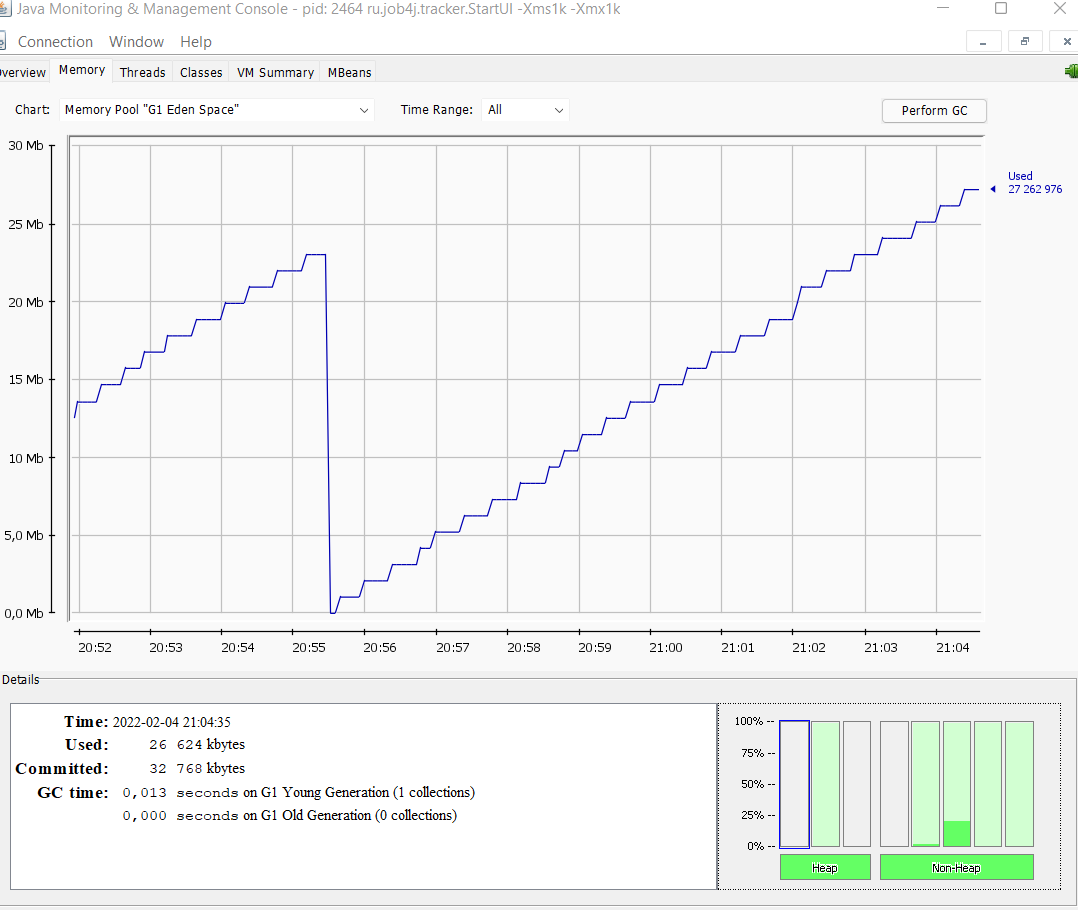
Commited – кол-во памяти гарантированно доступное для использования. Может изменяться с течением времени.

Должно быть больше чем Used mem

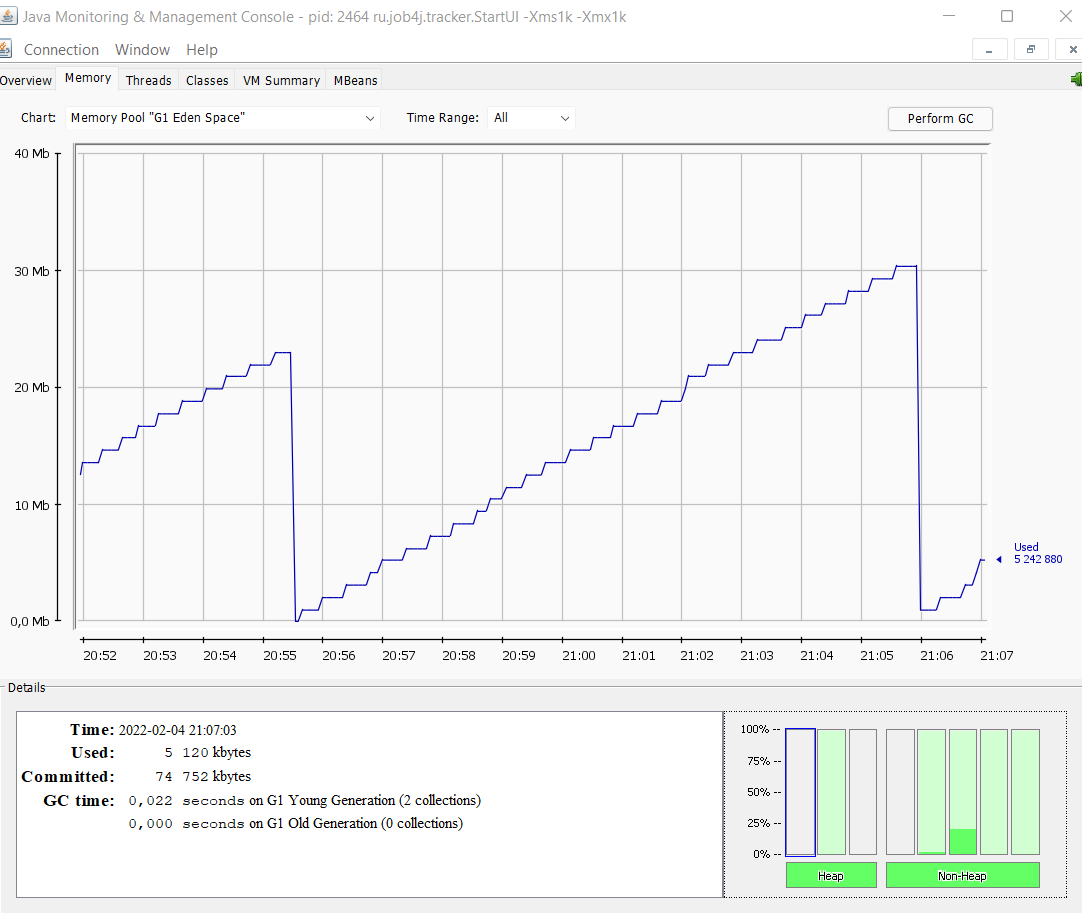
Max – максимальное кол-во памяти, которое может быть использовано

GC time – время на сборку мусора и кол-во сборок

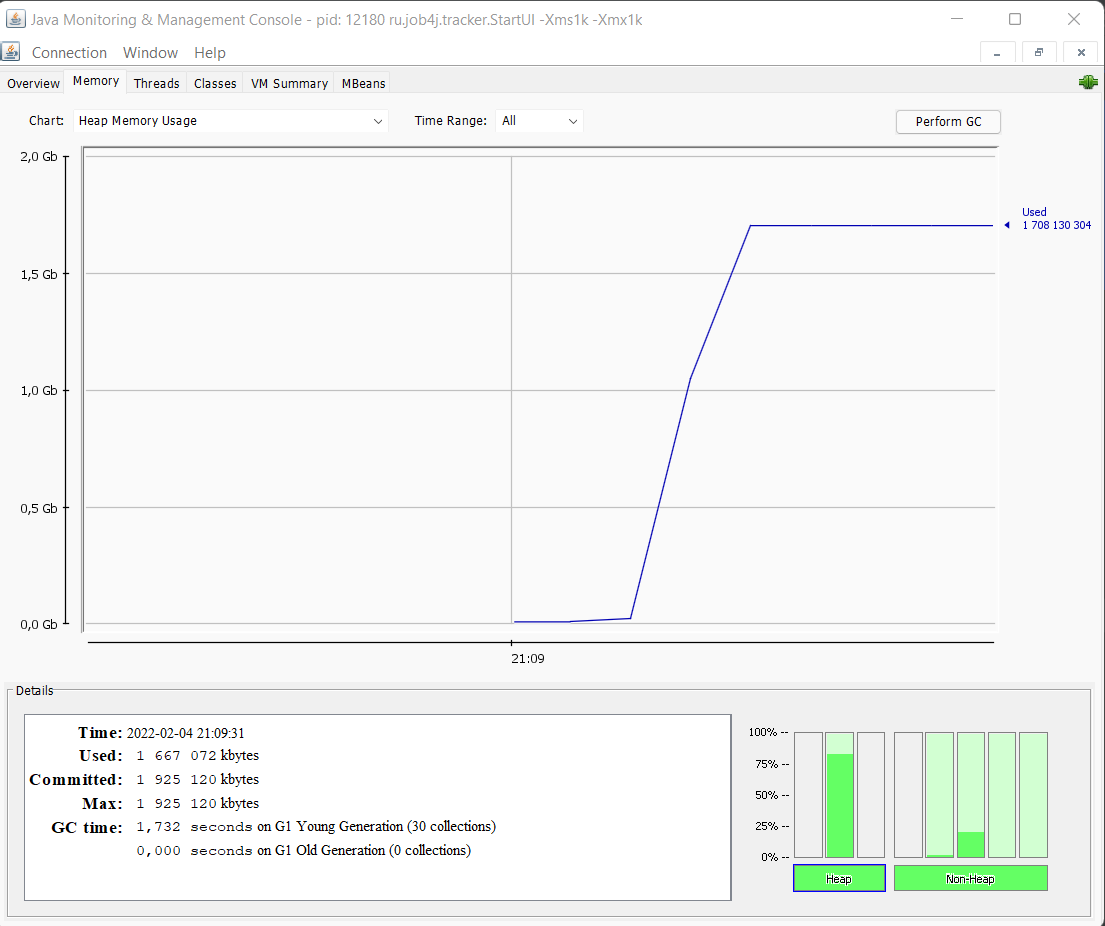
- После добавления объектов проведена сборка Young Gen



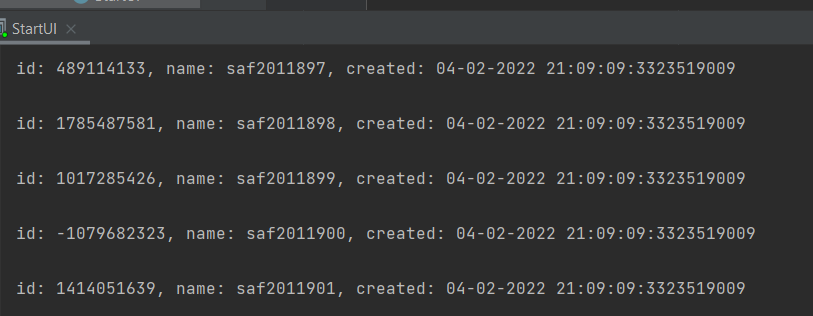
* Удаление объектов. Произошла еще 1 сборка

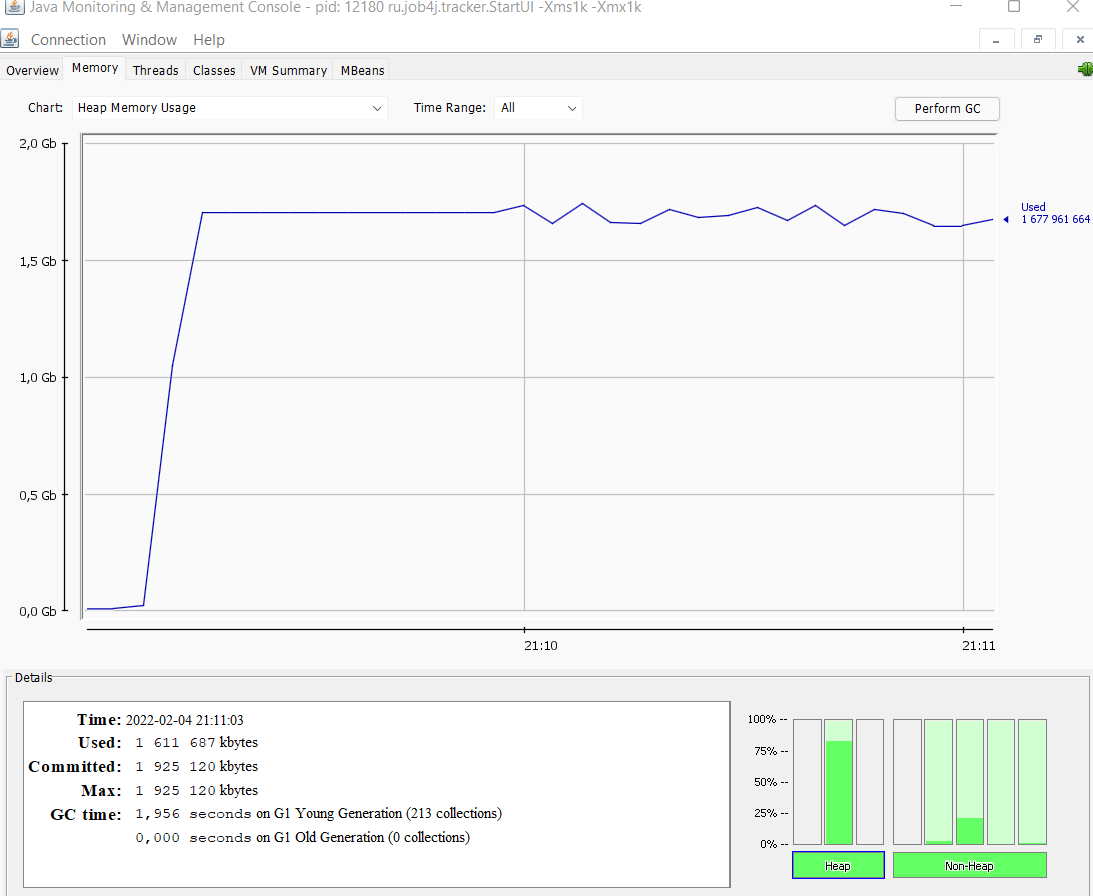


- Добавление большого количества объектов



* Запрос вывода всех объектов





- Переполнение памяти

