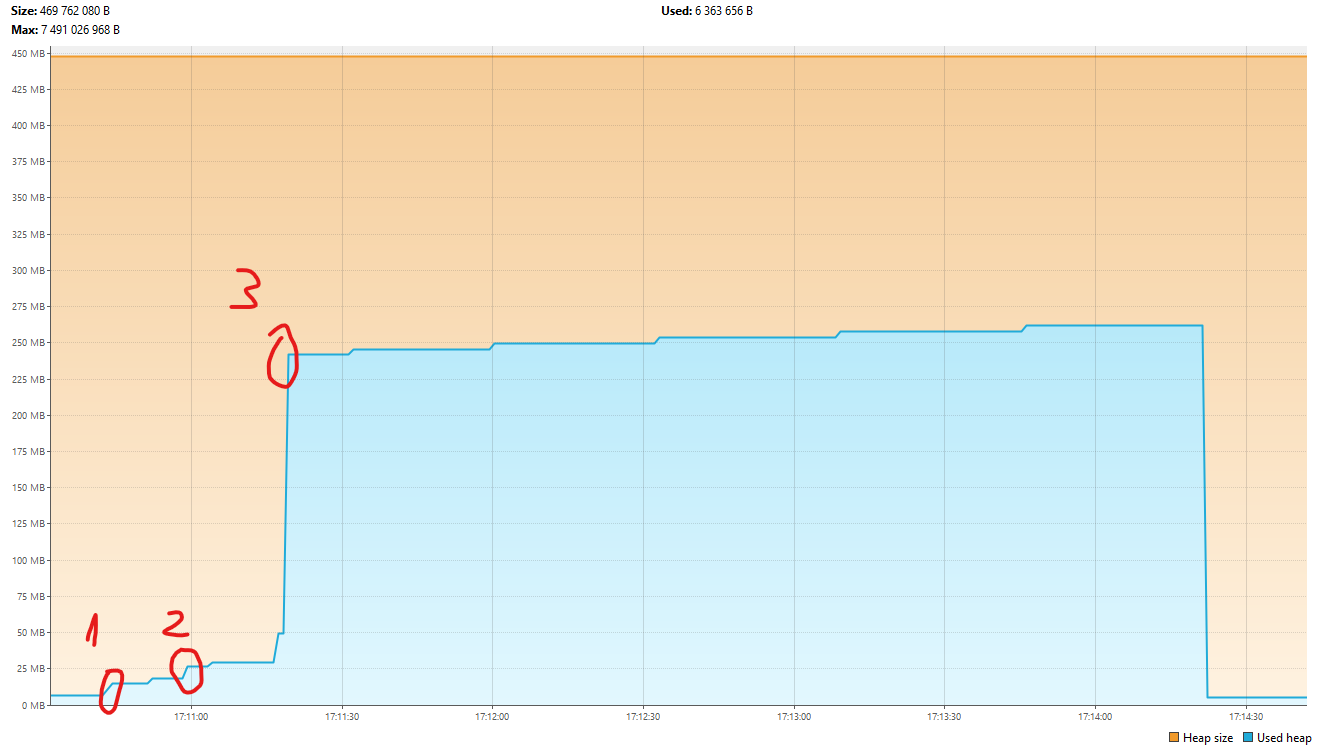
Анализ работы job4j\_tracker:

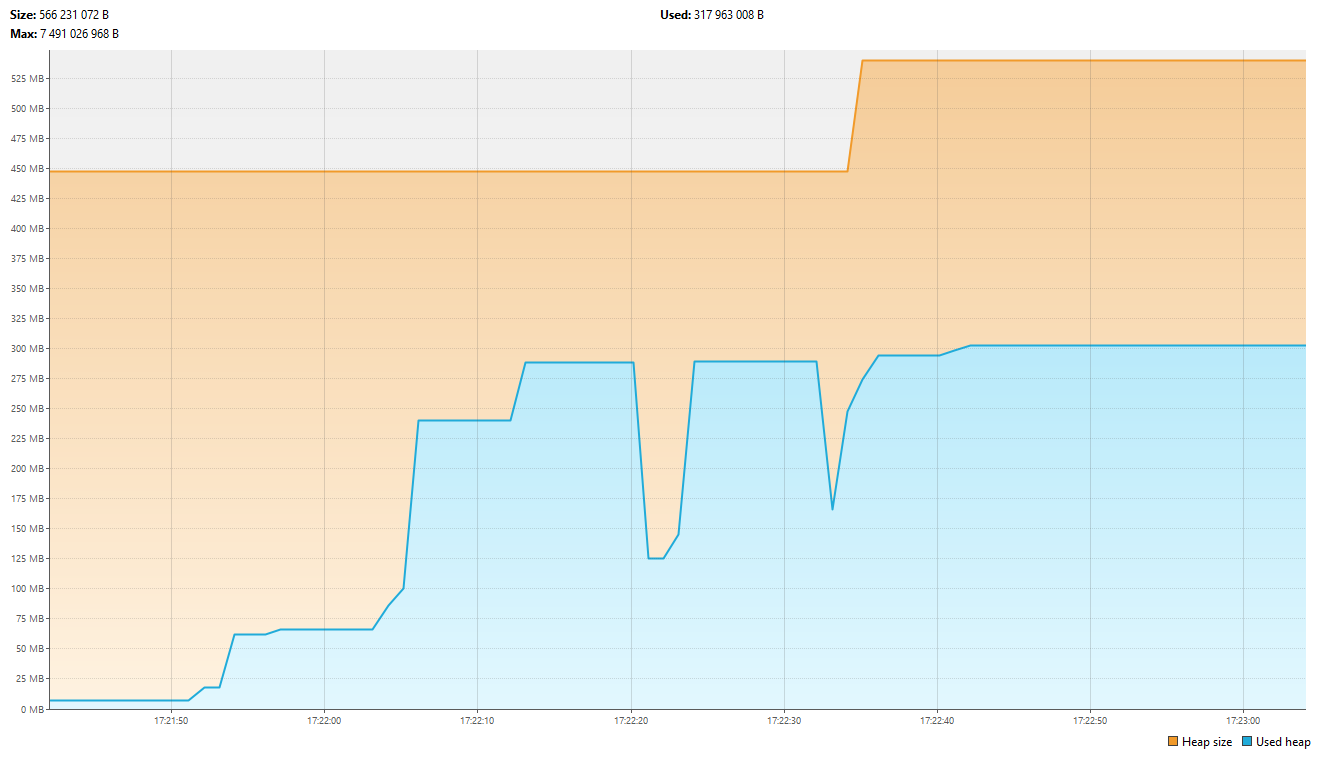
Были добавлены два класса DeleteAllAction, CreateManyAction, которые очищали весь трекер и заполняли по заданному числу итемов.



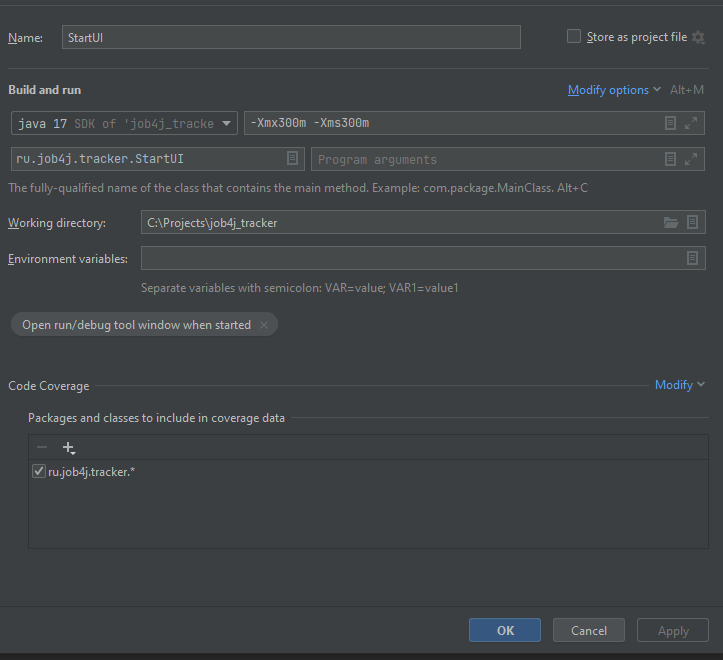
1. Добавили 70000 элементов и вывели на экран, график изменился незначительно, т.е. потребление памяти незначительно увеличилось
2. Добавили еще 20000 элементов, произошло тоже самое, что и в пункте 1
3. Добавили еще 100000 элементов и вывели на экран, тем самым произошел скачек графика, т.е. потребление памяти многократно возрасло

После выполнения п.3 были удалены все элементы, но на график это никак не повлияло, тем самым сбор мусора не производился долгое время (3 мин.), но после все-таки произошел FullGC.

При попытке добиться выхода за пределы памяти было добавлено 200000 элементов несколько раз, при этом происходили скачки графика, т.е. при частом добавлении элементов сборка мусора происходит быстрее:



Ограничим выделенную под мусор память:



Постепенно добавляя по 200000 элементов, затем добавляя 1000000 удалось выйти за пределы памяти и получить OutOfMemoryError: Java heap space:

