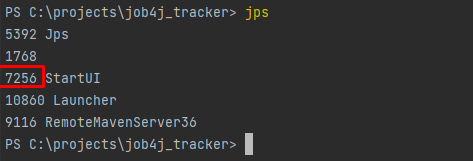
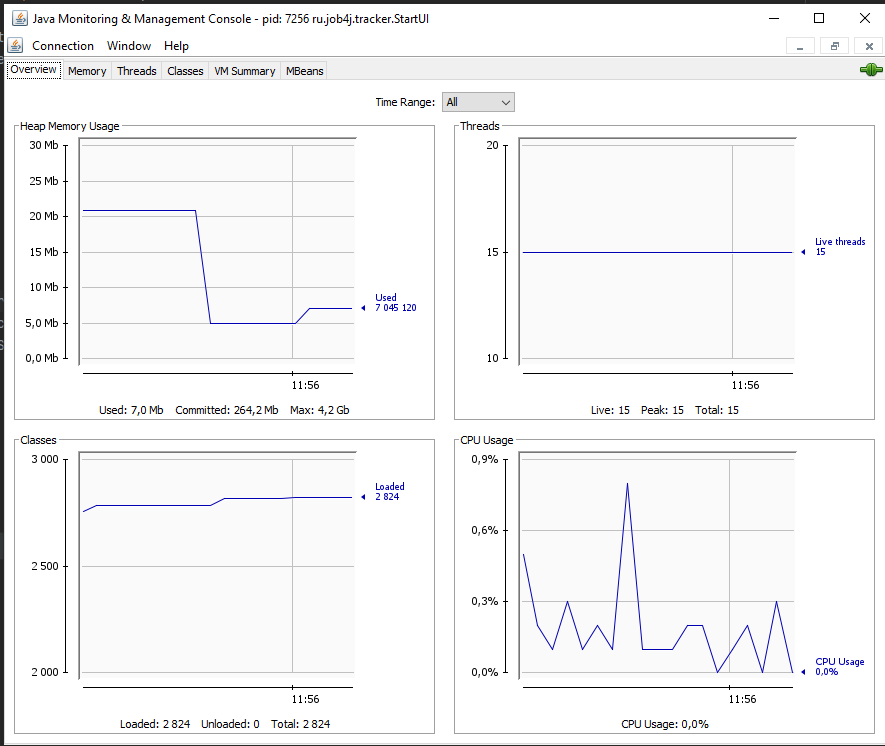
**Профилирование StartUI**

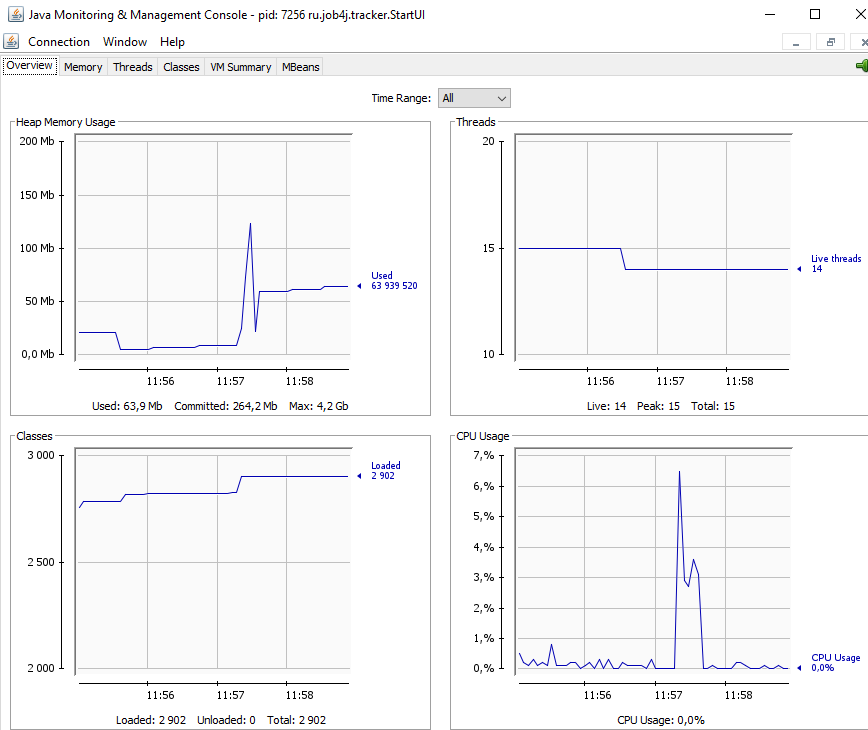
1. Запускаем приложение и используем утилиту jps, чтобы узнать PID приложения



2. Запускаем утилиту jconsole и подличаем ее к запущенному приложению.

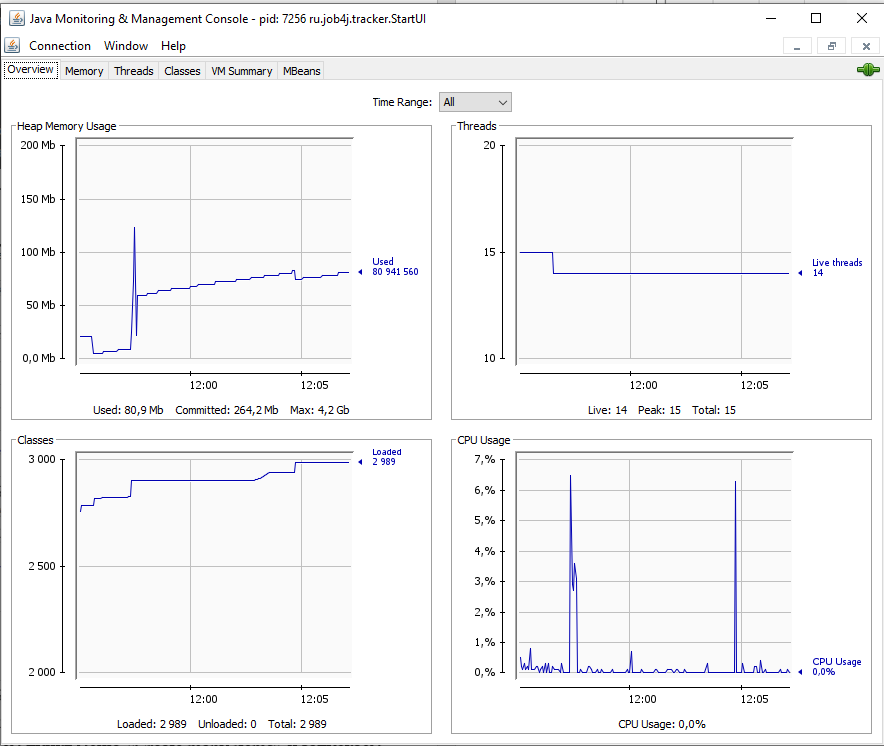


3. В приложении выбираем пункт меню «Create many items» и запускаем создание 100000 объектов. Мониторим графики в jconsole.



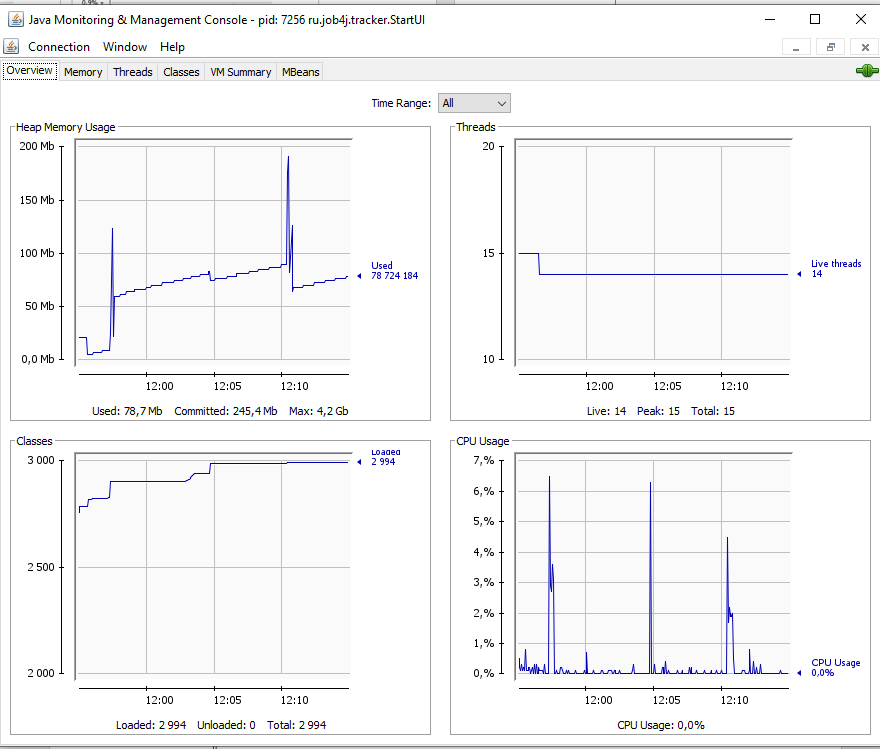
Видно, что резко вырос, чуть просел, но продолжает дальше расти объем кучи, на короткое время выросла нагрузка на процессор, также немного увеличилось число используемых классов.

4. В приложении выбираем пункт меню «Show all Items». Мониторим графики в jconsole.



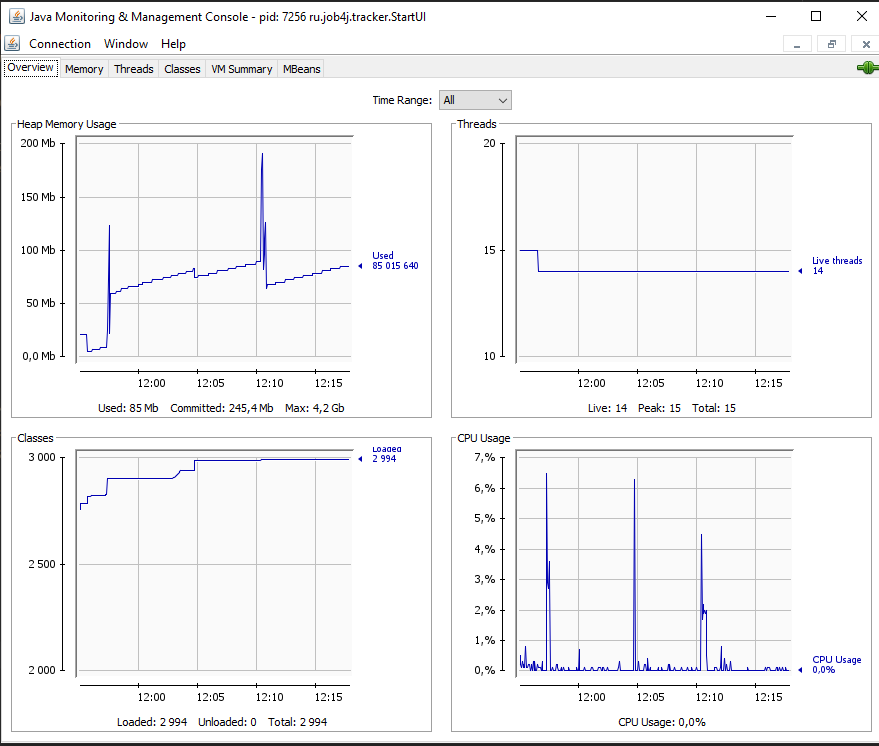
Видно, что на короткое время незначительно уменьшился размер кучи, но потом снова начал расти, также на короткое время увеличился нагрузка на процессор, слегка выросло количество используемых классов.

5. В приложении выбираем пункт меню Delete all items. Мониторим графики в jconsole.



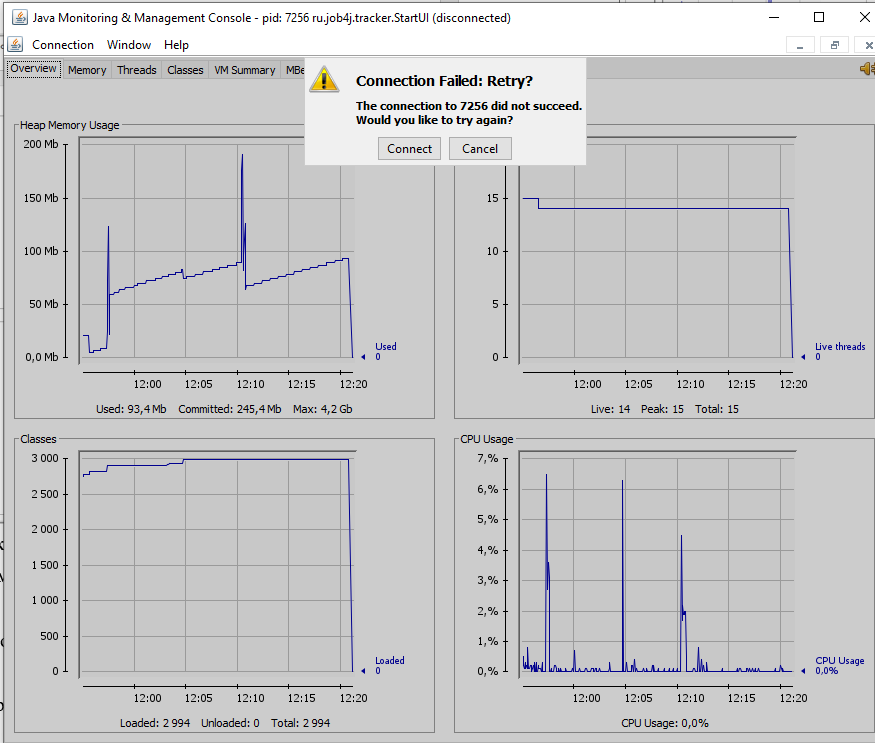
Видно, что на короткое время резко увеличился, а затем резко уменьшился размер кучи. Размер кучи стал заметно меньше, чем до удаления объектов, но тем не менее продолжает расти. Снова на короткое время увеличилась нагрузка на процессор, и снова слегка выросло количество используемых классов.

6. В приложении выбираем пункт меню «Show all Items». Мониторим графики в jconsole.

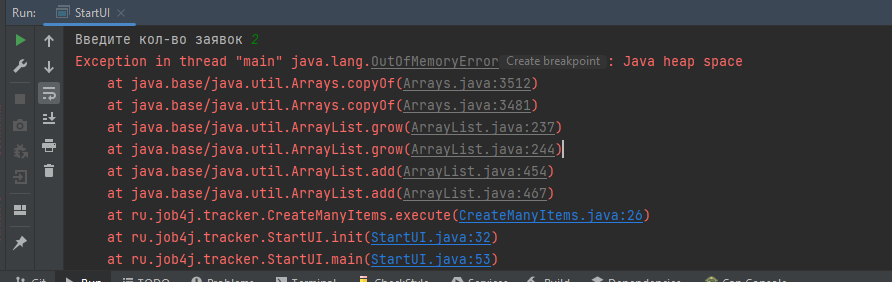


Видим, что данное действие никак не повлияло на потребление ресурсов: процессор не отреагировал, количество используемых классов осталось прежним, а тенденция роста размера кучи осталась на прежнем уровне.

7. Останавливаем приложении с помощью пункт меню «Exiting the program». Утилита jconsole потеряла связь с приложением.



Как только программа выходит за пределы памяти, IDEA прекращает работу программы и кидает OutOfMemoryError: Java heap space.



В jconsole отображено это на графике, где отслеживается размер кучи:

