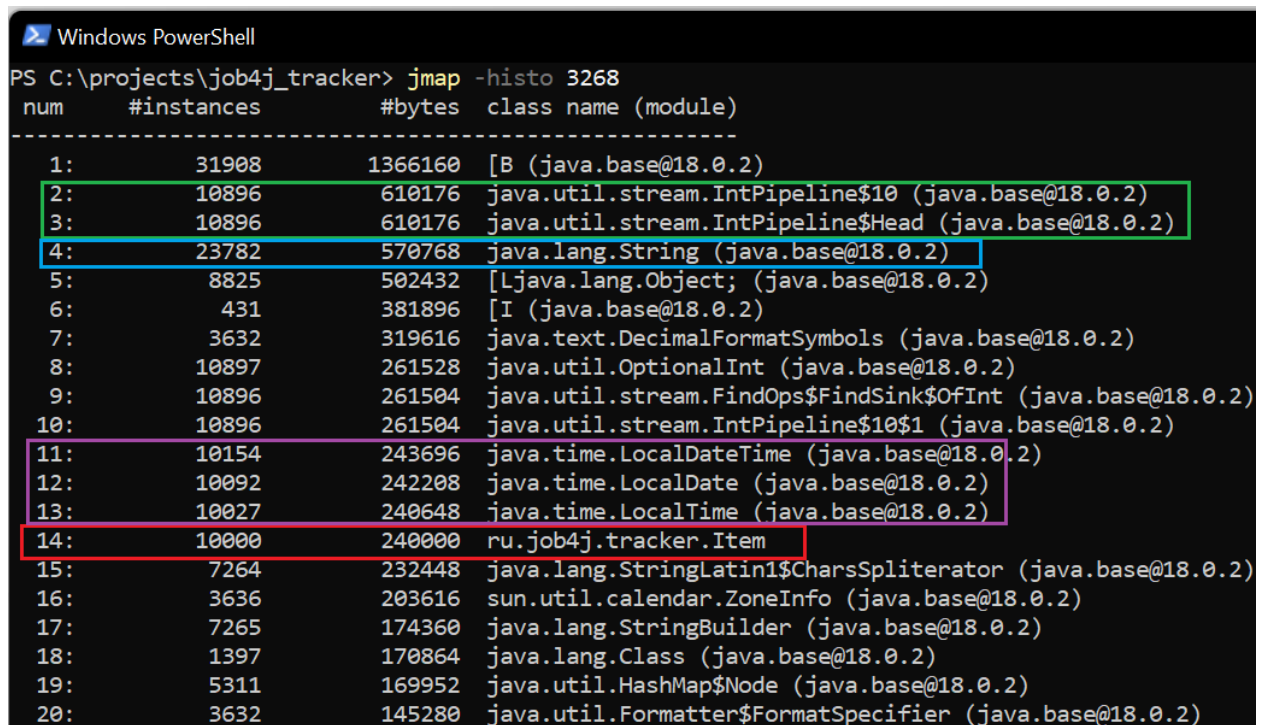


# jmap

Скриншот jmap, сделанный после добавления 10.000 заявок:



```
Windows PowerShell
PS C:\projects\job4j_tracker> jmap -histo 3268
```

num	#instances	#bytes	class name (module)
1:	31908	1366160	[B (java.base@18.0.2)
2:	10896	610176	java.util.stream.IntPipeline\$10 (java.base@18.0.2)
3:	10896	610176	java.util.stream.IntPipeline\$Head (java.base@18.0.2)
4:	23782	570768	java.lang.String (java.base@18.0.2)
5:	8825	502432	[Ljava.lang.Object; (java.base@18.0.2)
6:	431	381896	[I (java.base@18.0.2)
7:	3632	319616	java.text.DecimalFormatSymbols (java.base@18.0.2)
8:	10897	261528	java.util.OptionalInt (java.base@18.0.2)
9:	10896	261504	java.util.stream.FindOps\$FindSink\$OfInt (java.base@18.0.2)
10:	10896	261504	java.util.stream.IntPipeline\$10\$1 (java.base@18.0.2)
11:	10154	243696	java.time.LocalDateTime (java.base@18.0.2)
12:	10092	242208	java.time.LocalDate (java.base@18.0.2)
13:	10027	240648	java.time.LocalTime (java.base@18.0.2)
14:	10000	240000	ru.job4j.tracker.Item
15:	7264	232448	java.lang.StringLatin1\$CharsSplitter (java.base@18.0.2)
16:	3636	203616	sun.util.calendar.ZoneInfo (java.base@18.0.2)
17:	7265	174360	java.lang.StringBuilder (java.base@18.0.2)
18:	1397	170864	java.lang.Class (java.base@18.0.2)
19:	5311	169952	java.util.HashMap\$Node (java.base@18.0.2)
20:	3632	145280	java.util.Formatter\$FormatSpecifier (java.base@18.0.2)

На скриншоте видно, что:

- Было создано в точности 10.000 объектов типа `ru.job4j.tracker.Item` (выделено красным).
- Было создано 23.156 объектов типа `String`. Здесь, видимо дважды учитываются строки, использованные при автогенерации имени заявки через `String.format()` (выделено синим).
- Создается около 10.000 объектов `LocalDateTime`, `LocalDate` и `LocalTime` (выделено фиолетовым).
- Создается около 10.000 объектов `stream.IntPipeline` (выделено зеленым).

Скриншот jmap, сделанный после удаления 10.000 заявок:

Windows PowerShell

```
PS C:\projects\job4j_tracker> jmap -histo 3268
```

num	#instances	#bytes	class name (module)
1:	31933	1367120	[B (java.base@18.0.2)
2:	10899	610344	java.util.stream.IntPipeline\$10 (java.base@18.0.2)
3:	10899	610344	java.util.stream.IntPipeline\$Head (java.base@18.0.2)
4:	8829	582544	[Ljava.lang.Object; (java.base@18.0.2)
5:	23802	571248	java.lang.String (java.base@18.0.2)
6:	436	338352	[I (java.base@18.0.2)
7:	3633	319704	java.text.DecimalFormatSymbols (java.base@18.0.2)
8:	10900	261600	java.util.OptionalInt (java.base@18.0.2)
9:	10899	261576	java.util.stream.FindOps\$FindSink\$OfInt (java.base@18.0.2)
10:	10899	261576	java.util.stream.IntPipeline\$10\$1 (java.base@18.0.2)
11:	10154	243696	java.time.LocalDateTime (java.base@18.0.2)
12:	10092	242208	java.time.LocalDate (java.base@18.0.2)
13:	10027	240648	java.time.LocalTime (java.base@18.0.2)
14:	10000	240000	ru.job4j.tracker.Item
15:	7266	232512	java.lang.StringLatin1\$CharsSpliterator (java.base@18.0.2)
16:	3636	203616	sun.util.calendar.ZoneInfo (java.base@18.0.2)
17:	7267	174408	java.lang.StringBuilder (java.base@18.0.2)
18:	1397	170864	java.lang.Class (java.base@18.0.2)
19:	5311	169952	java.util.HashMap\$Node (java.base@18.0.2)
20:	3633	145320	java.util.Formatter\$FormatSpecifier (java.base@18.0.2)

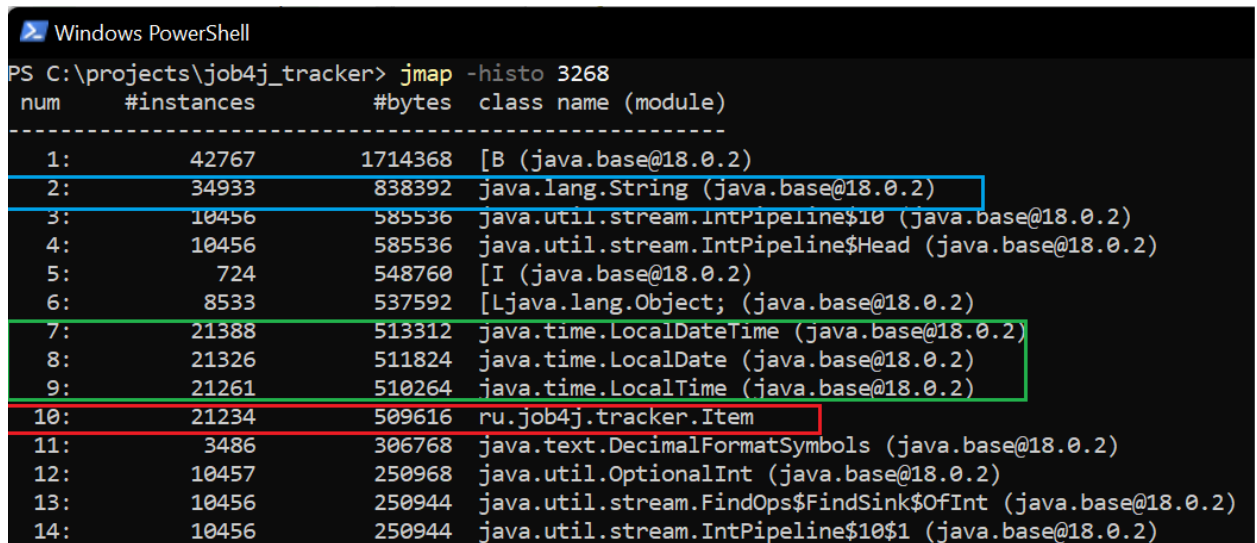
Из скриншота видно, что количество объектов Item в памяти не изменилось даже после удаления. Также не изменилось или незначительно изменилось количество объектов, предположительно хранящих значения полей объектов Item.

При этом приложение показывает, что в хранилище заявок нет:

```
=== Delete all items ===
Удалено заявок: 10000
Menu:
0. Add new Item
1. Show all items
2. Edit item
3. Delete item
4. Find item by id
5. Find items by name
6. Create multiple random items
7. Delete all items
8. Quit
Select: 1
=== Show all items ===
Хранилище еще не содержит заявок
```

Теперь попробуем добавить 20.000 новых заявок в хранилище сразу же после описанных выше операций.

Результат jmap в этом случае будет следующим:



```
PS C:\projects\job4j_tracker> jmap -histo 3268
```

num	#instances	#bytes	class name (module)
1:	42767	1714368	[B (java.base@18.0.2)
2:	34933	838392	java.lang.String (java.base@18.0.2)
3:	10456	585536	java.util.stream.IntPipeline\$10 (java.base@18.0.2)
4:	10456	585536	java.util.stream.IntPipeline\$Head (java.base@18.0.2)
5:	724	548760	[I (java.base@18.0.2)
6:	8533	537592	[Ljava.lang.Object; (java.base@18.0.2)
7:	21388	513312	java.time.LocalDateTime (java.base@18.0.2)
8:	21326	511824	java.time.LocalDate (java.base@18.0.2)
9:	21261	510264	java.time.LocalTime (java.base@18.0.2)
10:	21234	509616	ru.job4j.tracker.Item
11:	3486	306768	java.text.DecimalFormatSymbols (java.base@18.0.2)
12:	10457	250968	java.util.OptionalInt (java.base@18.0.2)
13:	10456	250944	java.util.stream.FindOps\$FindSink\$OfInt (java.base@18.0.2)
14:	10456	250944	java.util.stream.IntPipeline\$10\$1 (java.base@18.0.2)

Как видно из скриншота, количество объектов Item оказалось даже больше, чем количество вновь добавленных заявок, но значительно меньше 30.000, т.е. меньше суммы предыдущего количества с новым.

Таким образом, при удалении заявок сборщик мусора не удалил из памяти объекты.

Однако, поскольку при тестировании используется сравнительно малый размер памяти – всего 16Мб, часть объектов, добавленных ранее и более недоступных, была удалена, чтобы освободить память под новые 20.000 объектов (плюс новые объекты даты-времени и какое-то количество новых строк, добавленных в пул).

При добавлении 100.000 заявок, приложение ожидаемо падает с ошибкой java.lang.OutOfMemoryError, что невозможно отследить через jmap.

**jstat:**

## Добавление 10.000 заявок

[illegible]

Как видно из скриншота, после добавления 10.000 заявок значения занятого места в некоторых областях памяти изменились в большую сторону.

Удаление 10.000 заявок:

PS C:\projects\job_tracker> jstat -gc 3216 is 10																		
S0C	S1C	S0U	S1U	EC	EU	OC	OU	MC	MU	CCSC	CCSU	YGC	YGCT	FGC	FGCT	CGC	CGCT	GCT
0.0	1024.0	0.0	1024.0	6144.0	0.0	9216.0	2904.0	1664.0	1522.1	192.0	119.4	4	0.008	0	0.000	0	0.000	0.008
0.0	1024.0	0.0	1024.0	6144.0	0.0	9216.0	2904.0	1664.0	1522.1	192.0	119.4	4	0.008	0	0.000	0	0.000	0.008
0.0	1024.0	0.0	1024.0	6144.0	0.0	9216.0	2904.0	1664.0	1522.1	192.0	119.4	4	0.008	0	0.000	0	0.000	0.008
0.0	1024.0	0.0	1024.0	6144.0	0.0	9216.0	2904.0	1664.0	1522.1	192.0	119.4	4	0.008	0	0.000	0	0.000	0.008
0.0	1024.0	0.0	1024.0	6144.0	0.0	9216.0	2904.0	1664.0	1522.1	192.0	119.4	4	0.008	0	0.000	0	0.000	0.008
0.0	1024.0	0.0	1024.0	6144.0	0.0	9216.0	2904.0	1664.0	1522.1	192.0	119.4	4	0.008	0	0.000	0	0.000	0.008
0.0	1024.0	0.0	1024.0	6144.0	0.0	9216.0	2904.0	1664.0	1522.1	192.0	119.4	4	0.008	0	0.000	0	0.000	0.008
0.0	1024.0	0.0	1024.0	6144.0	0.0	9216.0	2904.0	1664.0	1522.1	192.0	119.4	4	0.008	0	0.000	0	0.000	0.008
0.0	1024.0	0.0	1024.0	6144.0	0.0	9216.0	2904.0	1664.0	1522.1	192.0	119.4	4	0.008	0	0.000	0	0.000	0.008
0.0	1024.0	0.0	1024.0	6144.0	0.0	9216.0	2904.0	1664.0	1522.1	192.0	119.4	4	0.008	0	0.000	0	0.000	0.008
0.0	1024.0	0.0	1024.0	6144.0	0.0	9216.0	2904.0	1664.0	1522.1	192.0	119.4	4	0.008	0	0.000	0	0.000	0.008
0.0	1024.0	0.0	1024.0	6144.0	0.0	9216.0	2904.0	1664.0	1522.1	192.0	119.4	4	0.008	0	0.000	0	0.000	0.008
0.0	1024.0	0.0	1024.0	6144.0	0.0	9216.0	2904.0	1664.0	1522.1	192.0	119.4	4	0.008	0	0.000	0	0.000	0.008
0.0	1024.0	0.0	1024.0	6144.0	0.0	9216.0	2904.0	1664.0	1522.1	192.0	119.4	4	0.008	0	0.000	0	0.000	0.008
0.0	1024.0	0.0	1024.0	6144.0	0.0	9216.0	2904.0	1664.0	1522.1	192.0	119.4	4	0.008	0	0.000	0	0.000	0.008
0.0	1024.0	0.0	1024.0	6144.0	0.0	9216.0	2904.0	1664.0	1522.1	192.0	119.4	4	0.008	0	0.000	0	0.000	0.008
0.0	1024.0	0.0	1024.0	6144.0	0.0	9216.0	2904.0	1664.0	1522.1	192.0	119.4	4	0.008	0	0.000	0	0.000	0.008
0.0	1024.0	0.0	1024.0	6144.0	0.0	9216.0	2904.0	1664.0	1522.1	192.0	119.4	4	0.008	0	0.000	0	0.000	0.008
0.0	1024.0	0.0	1024.0	6144.0	0.0	9216.0	2904.0	1664.0	1522.1	192.0	119.4	4	0.008	0	0.000	0	0.000	0.008
0.0	1024.0	0.0	1024.0	6144.0	0.0	9216.0	2904.0	1664.0	1522.1	192.0	119.4	4	0.008	0	0.000	0	0.000	0.008
0.0	1024.0	0.0	1024.0	6144.0	0.0	9216.0	2904.0	1664.0	1522.1	192.0	119.4	4	0.008	0	0.000	0	0.000	0.008
0.0	1024.0	0.0	1024.0	6144.0	0.0	9216.0	2904.0	1664.0	1522.1	192.0	119.4	4	0.008	0	0.000	0	0.000	0.008
0.0	1024.0	0.0	1024.0	6144.0	0.0	9216.0	2904.0	1664.0	1522.1	192.0	119.4	4	0.008	0	0.000	0	0.000	0.008
0.0	1024.0	0.0	1024.0	6144.0	0.0	9216.0	2904.0	1664.0	1522.1	192.0	119.4	4	0.008	0	0.000	0	0.000	0.008
0.0	1024.0	0.0	1024.0	6144.0	0.0	9216.0	2904.0	1664.0	1522.1	192.0	119.4	4	0.008	0	0.000	0	0.000	0.008
0.0	1024.0	0.0	1024.0	6144.0	0.0	9216.0	2904.0	1664.0	1522.1	192.0	119.4	4	0.008	0	0.000	0	0.000	0.008
0.0	1024.0	0.0	1024.0	6144.0	0.0	9216.0	2904.0	1664.0	1522.1	192.0	119.4	4	0.008	0	0.000	0	0.000	0.008
0.0	1024.0	0.0	1024.0	6144.0	0.0	9216.0	2904.0	1664.0	1522.1	192.0	119.4	4	0.008	0	0.000	0	0.000	0.008
0.0	1024.0	0.0	1024.0	6144.0	0.0	9216.0	2904.0	1664.0	1522.1	192.0	119.4	4	0.008	0	0.000	0	0.000	0.008
0.0	1024.0	0.0	1024.0	6144.0	0.0	9216.0	2904.0	1664.0	1522.1	192.0	119.4	4	0.008	0	0.000	0	0.000	0.008
0.0	1024.0	0.0	1024.0	6144.0	0.0	9216.0	2904.0	1664.0	1522.1	192.0	119.4	4	0.008	0	0.000	0	0.000	0.008
0.0	1024.0	0.0	1024.0	6144.0	0.0	9216.0	2904.0	1664.0	1522.1	192.0	119.4	4	0.008	0	0.000	0	0.000	0.008
0.0	1024.0	0.0	1024.0	6144.0	0.0	9216.0	2904.0	1664.0	1522.1	192.0	119.4	4	0.008	0	0.000	0	0.000	0.008
0.0	1024.0	0.0	1024.0	6144.0	0.0	9216.0	2904.0	1664.0	1522.1	192.0	119.4	4	0.008	0	0.000	0	0.000	0.008
0.0	1024.0	0.0	1024.0	6144.0	0.0	9216.0	2904.0	1664.0	1522.1	192.0	119.4	4	0.008	0	0.000	0	0.000	0.008
0.0	1024.0	0.0	1024.0	6144.0	0.0	9216.0	2904.0	1664.0	1522.1	192.0	119.4	4	0.008	0	0.000	0	0.000	0.008
0.0	1024.0	0.0	1024.0	6144.0	0.0	9216.0	2904.0	1664.0	1522.1	192.0	119.4	4	0.008	0	0.000	0	0.000	0.008
0.0	1024.0	0.0	1024.0	6144.0	0.0	9216.0	2904.0	1664.0	1522.1	192.0	119.4	4	0.008	0	0.000	0	0.000	0.008
0.0	1024.0	0.0	1024.0	6144.0	0.0	9216.0	2904.0	1664.0	1522.1	192.0	119.4	4	0.008	0	0.000	0	0.000	0.008
0.0	1024.0	0.0	1024.0	6144.0	0.0	9216.0	2904.0	1664.0	1522.1	192.0	119.4	4	0.008	0	0.000	0	0.000	0.008
0.0	1024.0	0.0	1024.0	6144.0	0.0	9216.0	2904.0	1664.0	1522.1	192.0	119.4	4	0.008	0	0.000	0	0.000	0.008
0.0	1024.0	0.0	1024.0	6144.0	0.0	9216.0	2904.0	1664.0	1522.1	192.0	119.4	4	0.008	0	0.000	0	0.000	0.008
0.0	1024.0	0.0	1024.0	6144.0	0.0	9216.0	2904.0	1664.0	1522.1	192.0	119.4	4	0.008	0	0.000	0	0.000	0.008
0.0	1024.0	0.0	1024.0	6144.0	0.0	9216.0	2904.0	1664.0	1522.1	192.0	119.4	4	0.008	0	0.000	0	0.000	0.008
0.0	1024.0	0.0	1024.0	6144.0	0.0	9216.0	2904.0	1664.0	1522.1	192.0	119.4	4	0.008	0	0.000	0	0.000	0.008
0.0	1024.0	0.0	1024.0	6144.0	0.0	9216.0	2904.0	1664.0	1522.1	192.0	119.4	4	0.008	0	0.000	0	0.000	0.008
0.0	1024.0	0.0	1024.0	6144.0	0.0	9216.0	2904.0	1664.0	1522.1	192.0	119.4	4	0.008	0	0.000	0	0.000	0.008
0.0	1024.0	0.0	1024.0	6144.0	0.0	9216.0	2904.0	1664.0	1522.1	192.0	119.4	4	0.008	0	0.000	0	0.000	0.008
0.0	1024.0	0.0	1024.0	6144.0	0.0	9216.0	2904.0	1664.0	1522.1	192.0	119.4	4	0.008	0	0.000	0	0.000	0.008
0.0	1024.0	0.0	1024.0	6144.0	0.0	9216.0	2904.0	1664.0	1522.1	192.0	119.4	4	0.008	0	0.000	0	0.000	0.008
0.0	1024.0	0.0	1024.0	6144.0	0.0	9216.0	2904.0	1664.0	1522.1	192.0	119.4	4	0.008	0	0.000	0	0.000	0.008
0.0	1024.0	0.0	1024.0	6144.0	0.0	9216.0	2904.0	1664.0	1522.1	192.0	119.4	4	0.008	0	0.000	0	0.000	0.008
0.0	1024.0	0.0	1024.0	6144.0	0.0	9216.0	2904.0	1664.0	1522.1	192.0	119.4	4	0.008	0	0.000	0	0.000	0.008
0.0	1024.0	0.0	1024.0	6144.0	0.0	9216.0	2904.0	1664.0	1522.1	192.0	119.4	4	0.008	0	0.000	0	0.000	0.008
0.0	1024.0	0.0	1024.0	6144.0	0.0	9216.0	2904.0	1664.0	1522.1	192.0	119.4	4	0.008	0	0.000	0	0.000	0.008
0.0	1024.0	0.0	1024.0	6144.0	0.0	9216.0	2904.0	1664.0	1522.1	192.0	119.4	4	0.008	0	0.000	0	0.000	0.008
0.0	1024.0	0.0	1024.0	6144.0	0.0	9216.0	2904.0	1664.0	1522.1	192.0	119.4	4	0.008	0	0.000	0	0.000	0.008
0.0	1024.0	0.0	1024.0	6144.0	0.0	9216.0	2904.0	1664.0	1522.1	192.0	119.4	4	0.008	0	0.000	0	0.000	0.008
0.0	1024.0	0.0	1024.0	6144.0	0.0	9216.0	2904.0	1664.0	1522.1	192.0	119.4	4	0.008	0	0.000	0	0.000	0.008
0.0	1024.0	0.0	1024.0	6144.0	0.0	9216.0	2904.0	1664.0	1522.1	192.0	119.4	4	0.008	0	0.000	0	0.000	0.008
0.0	1024.0	0.0	1024.0	6144.0	0.0	9216.0	2904.0	1664.0	1522.1	192.0	119.4	4	0.008	0	0.000	0	0.000	0.008
0.0	1024.0	0.0	1024.0	6144.0	0.0	9216.0	2904.0	1664.0	1522.1	192.0	119.4	4	0.008	0	0.000	0	0.000	0.008
0.0	1024.0	0.0	1024.0	6144.0	0.0	9216.0	2904.0	1664.0	1522.1	192.0	119.4	4	0.008	0	0.000	0	0.000	0.008
0.0	1024.0	0.0	1024.0	6144.0	0.0	9216.0	2904.0	1664.0	1522.1	192.0	119.4	4	0.008	0	0.000	0	0.000	0.008
0.0	1024.0	0.0	1024.0	6144.0	0.0	9216.0	2904.0	1664.0	1522.1	192.0	119.4	4	0.008	0	0.000	0	0.000	0.008
0.0	1024.0	0.0	1024.0	6144.0	0.0	9216.0	2904.0	1664.0	1522.1	192.0	119.4	4	0.008	0	0.000	0	0.000	0.008
0.0	1024.0	0.0	1024.0	6144.0	0.0	9216.0	2904.0	1664.0	1522.1	192.0	119.4	4	0.008					

После удаления всех заявок видимых изменений в памяти не происходит.

Попробуем теперь по аналогии с предыдущим экспериментом добавить 20.000 заявок:

[illegible]

Как видно из скриншота, объем занимаемой памяти возрастает в областях EU и OU.

Добавление 100.000 заявок приводит к переполнению памяти и падению приложения с ошибкой `java.lang.OutOfMemoryError`:

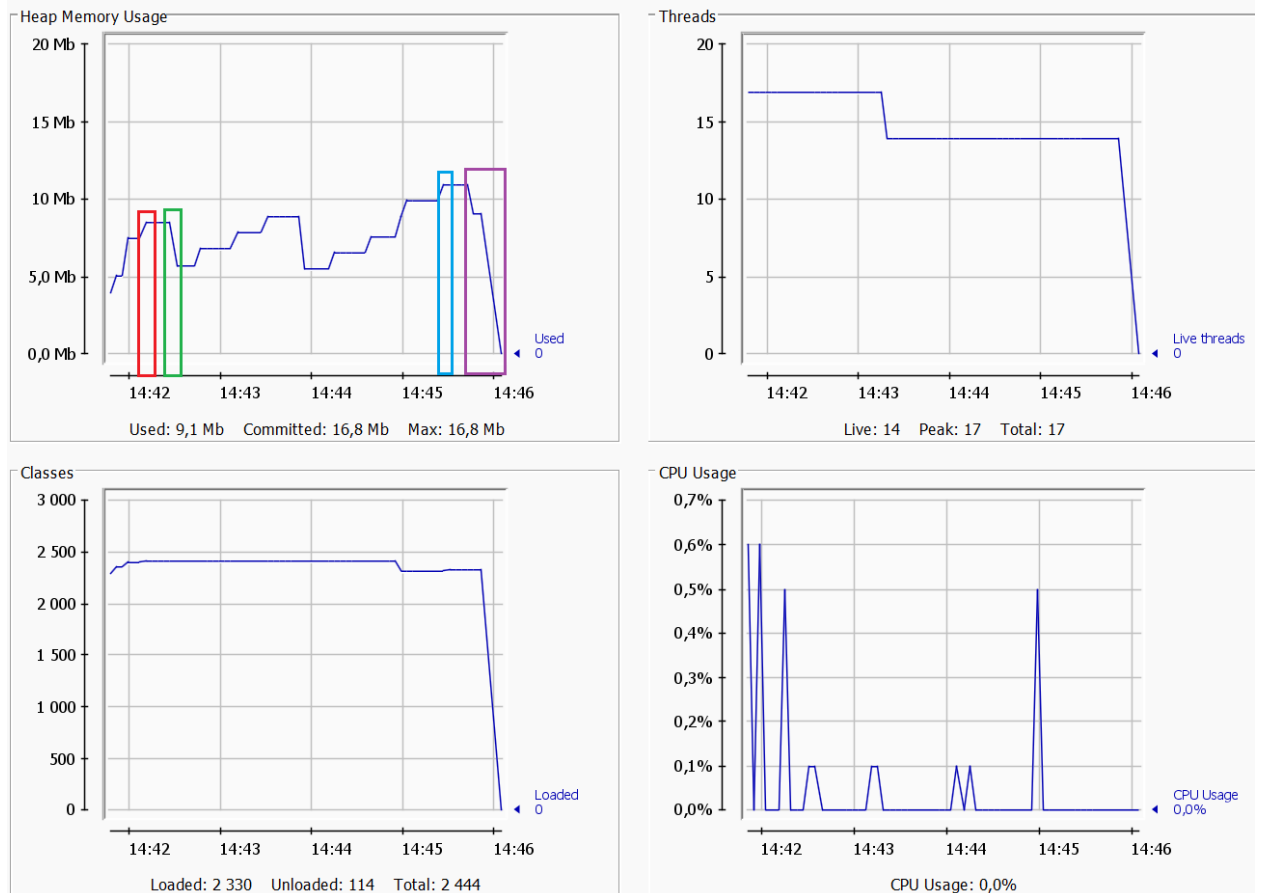
Windows PowerShell

```
PS C:\projects\job4j_tracker> jstat -gc 3216 1s 20
```

S0C	S1C	S0U	S1U	EC	EU	OC	OU	MC	MU	CCSC	CCSU	YGC	YGCT	FGC	FGCT	CSC	CGCT	GCT
0,0	1024,0	0,0	744,9	6144,0	4896,0	9216,0	4949,5	1664,0	1528,2	192,0	119,4	8	0,013	0	0,000	0	0,000	0,013
0,0	1024,0	0,0	744,9	6144,0	4896,0	9216,0	4949,5	1664,0	1528,2	192,0	119,4	8	0,013	0	0,000	0	0,000	0,013
0,0	0,0	0,0	0,0	10740,0	0,0	6144,0	1944,6	1664,0	1514,1	192,0	113,7	64	0,066	24	0,407	6	0,006	0,474

PS C:\projects\job4j\_tracker>

## jconsole



События на данном скриншоте по цвету блока:

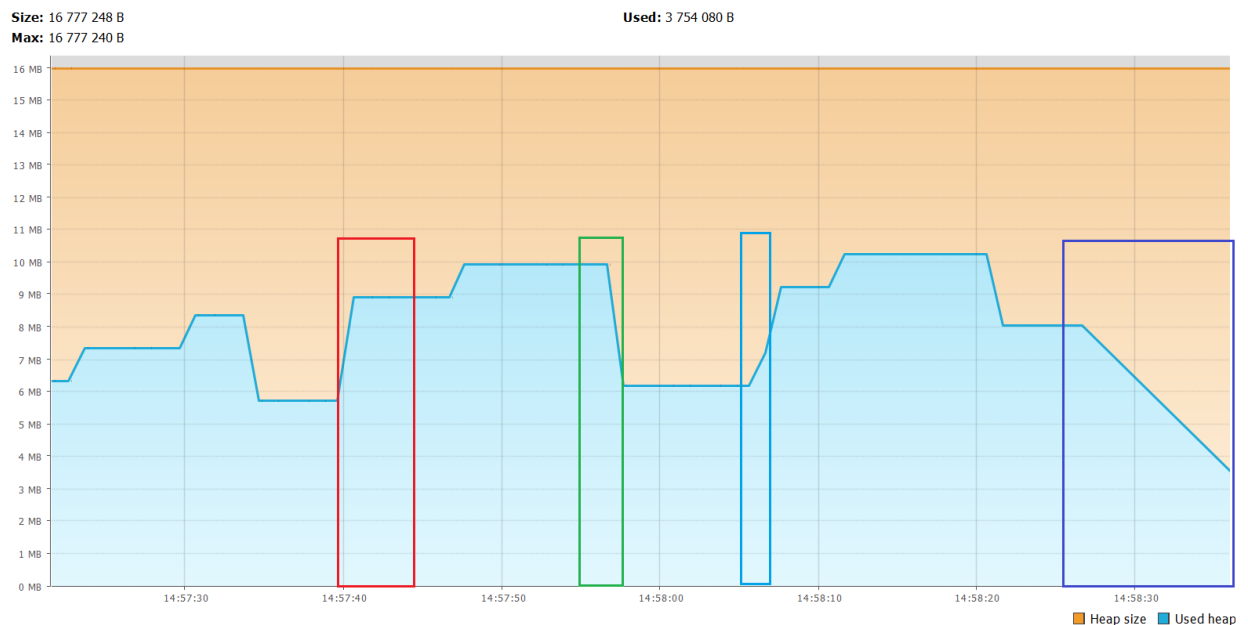
- Красный – создание 10.000 заявок
- Зеленый – удаление 10.000 заявок
- Синий – создание 20.000 заявок
- Фиолетовый – создание 100.000 заявок.

Судя по графику, в промежутке между удалением заявок и следующим добавлением, память сначала освобождалась, а потом занималась какими-то данными практически до исходных значений.

При попытке добавить 100.000 заявок приложение упало с ошибкой:

```
Exception in thread "main" java.lang.OutOfMemoryError: Java heap space: failed reallocation of scalar replaced objects
```

## Visual VM



События на данном скриншоте по цвету блока:

- Красный – создание 10.000 заявок
- Зеленый – удаление 10.000 заявок
- Синий – создание 20.000 заявок
- Фиолетовый – создание 100.000 заявок.

Как видно из данного скриншота, разница между пиковыми значениями памяти для 10.000 заявок и 20.000 заявок незначительна. Скорее всего это вызвано тем, что строки, сгенерированные для первого массива заявок

остались в пуле даже после удаления первого массива, и были повторно использованы при создании второго массива.

Как и в прошлых экспериментах, добавление 100.000 заявок привело к падению приложения.