بسم الله الرحمن الرحیم

محمدعلی سیف کاشانی 97106024

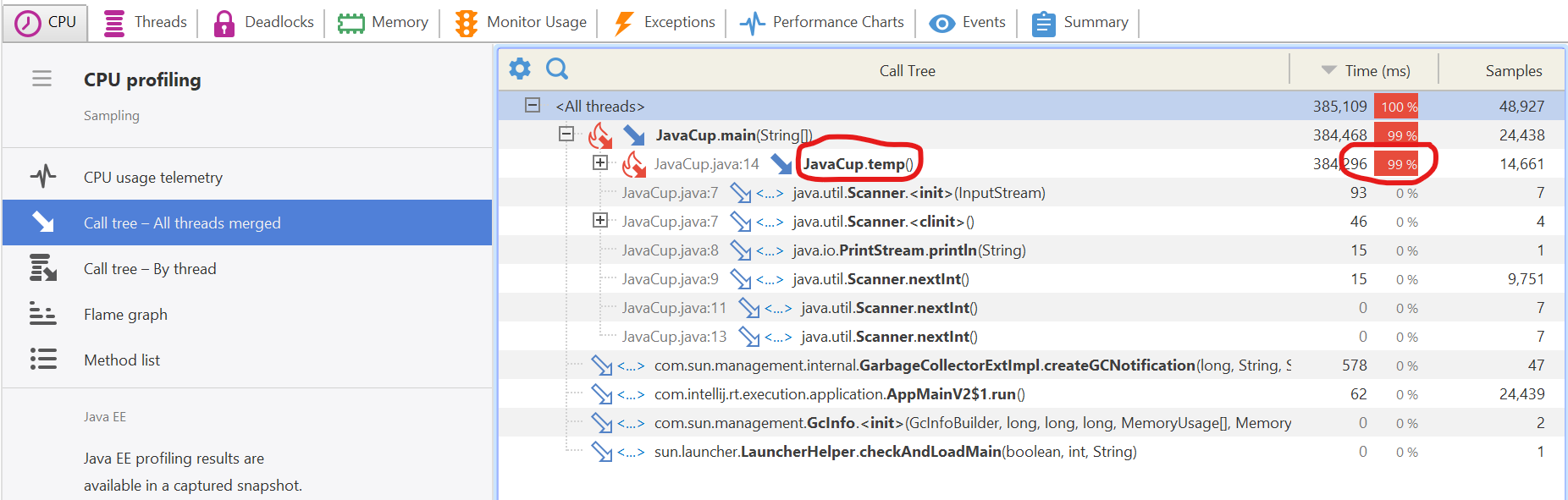
ابوالفضل اسد 97105714

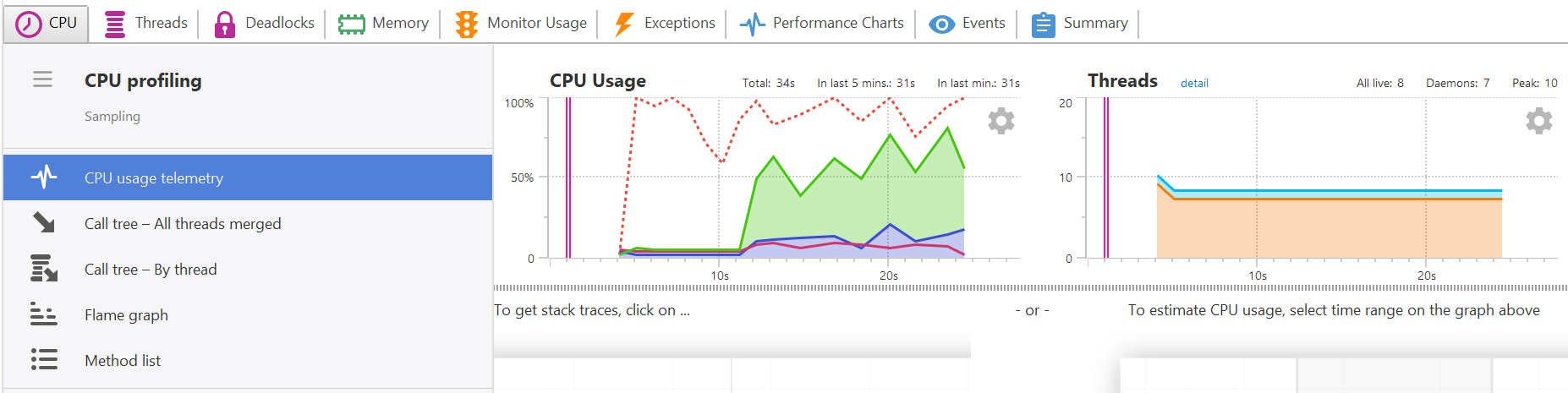
گزارش آزمایش بخش Profiling:

تمرین1:

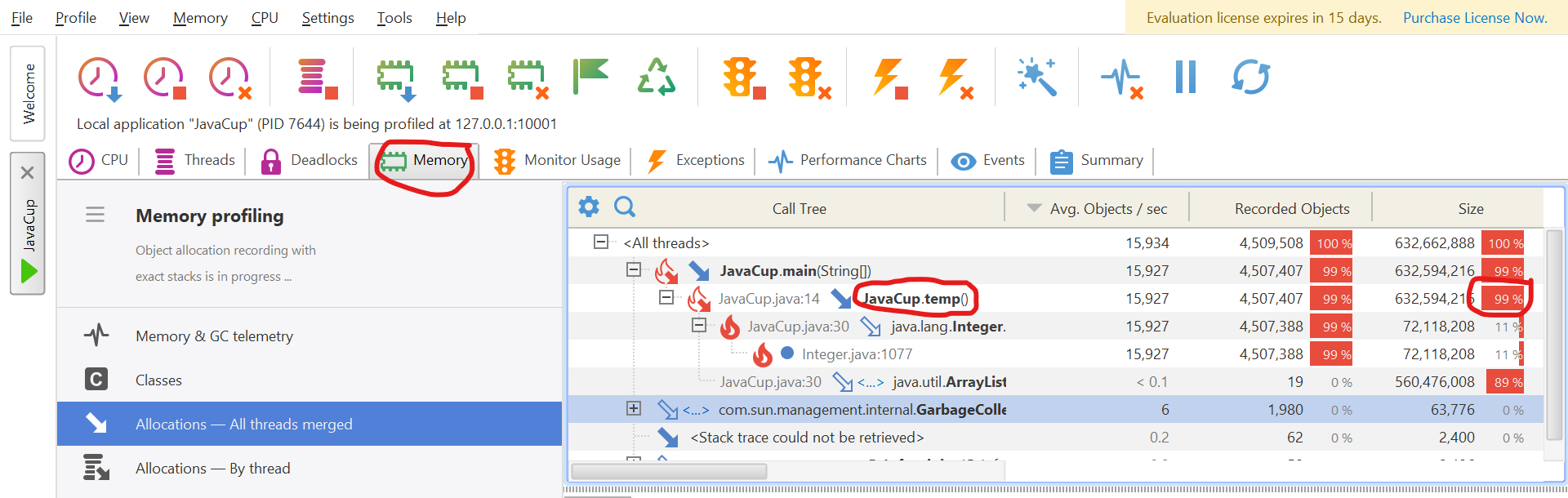
مرحله اول: اجرای کد و شناسایی تابعی که بیشترین منابع را صرف می کند ⭠ تابع temp()

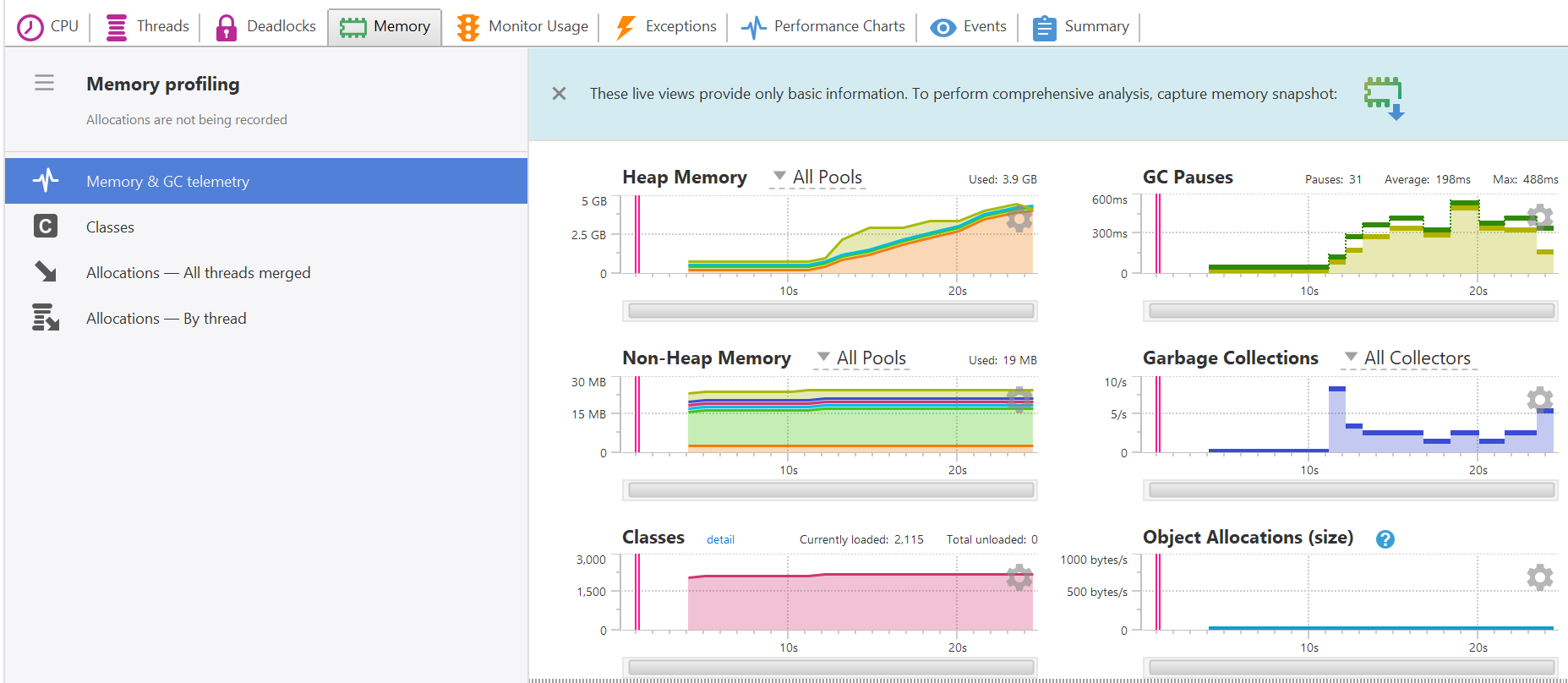
**CPU**



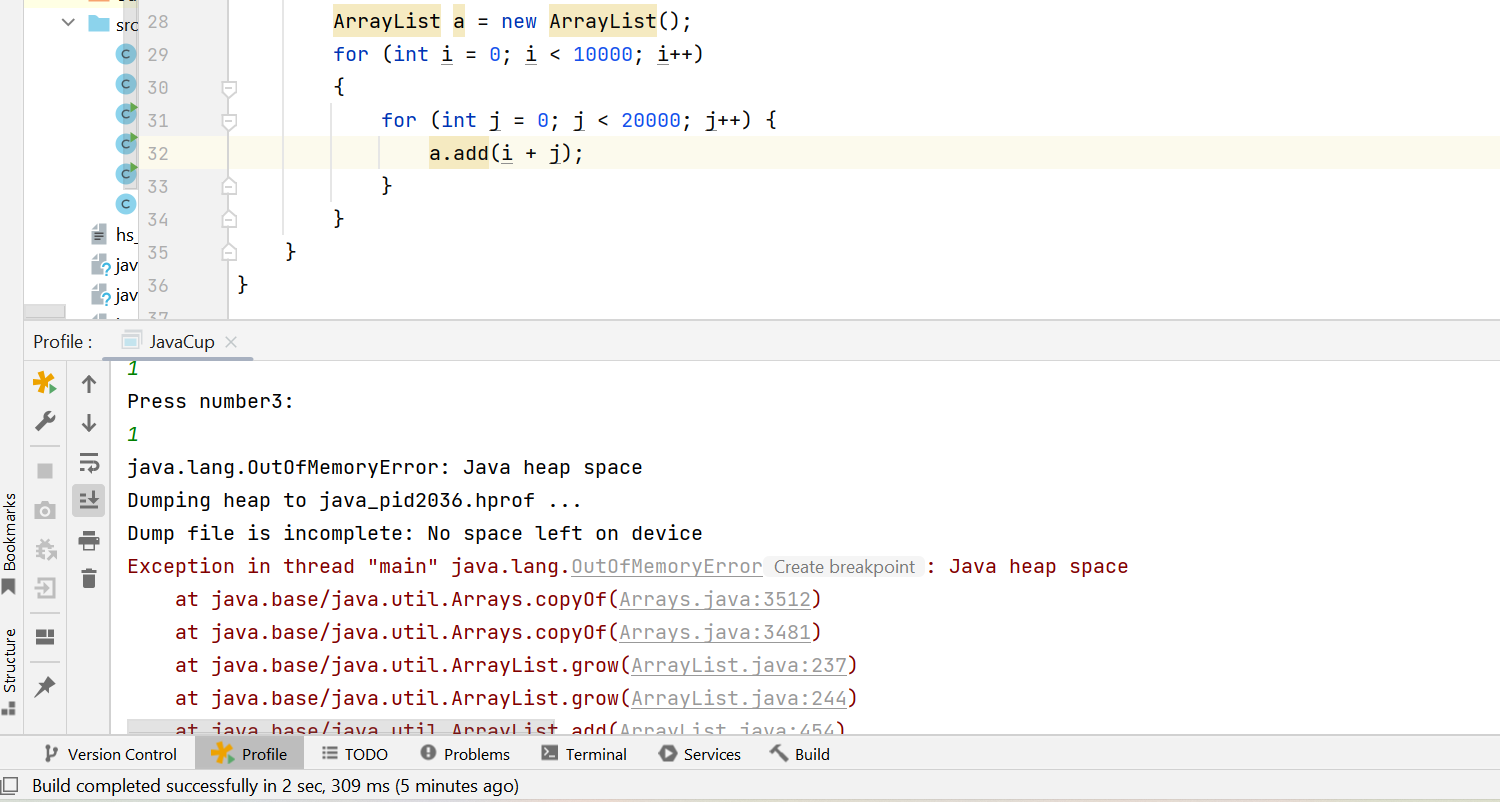


**MEMORY**



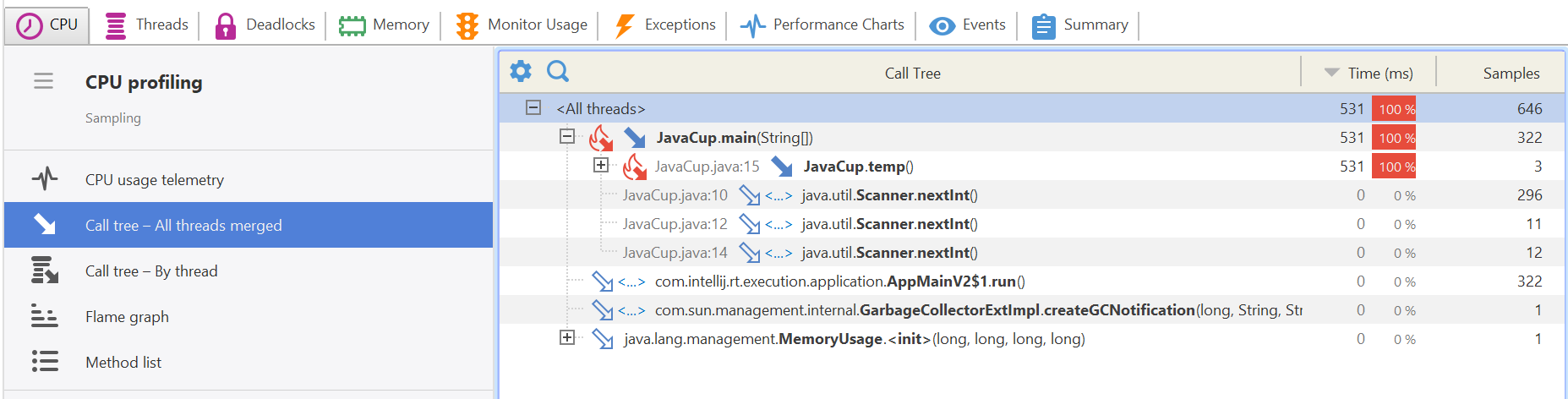


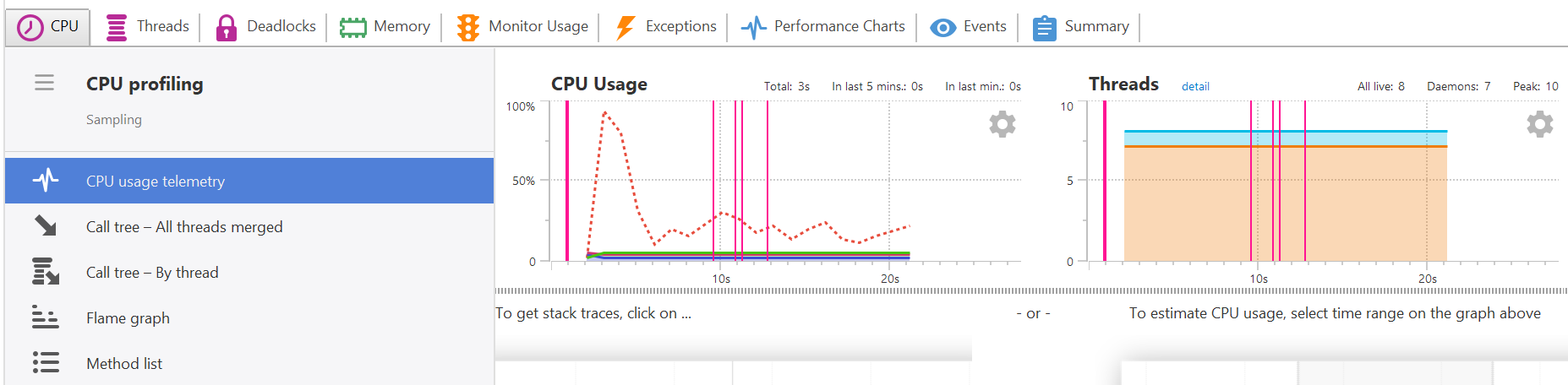
خوب است این تصویر را هم ببینیم که اجرای این برنامه به ارور مموری می خورد.



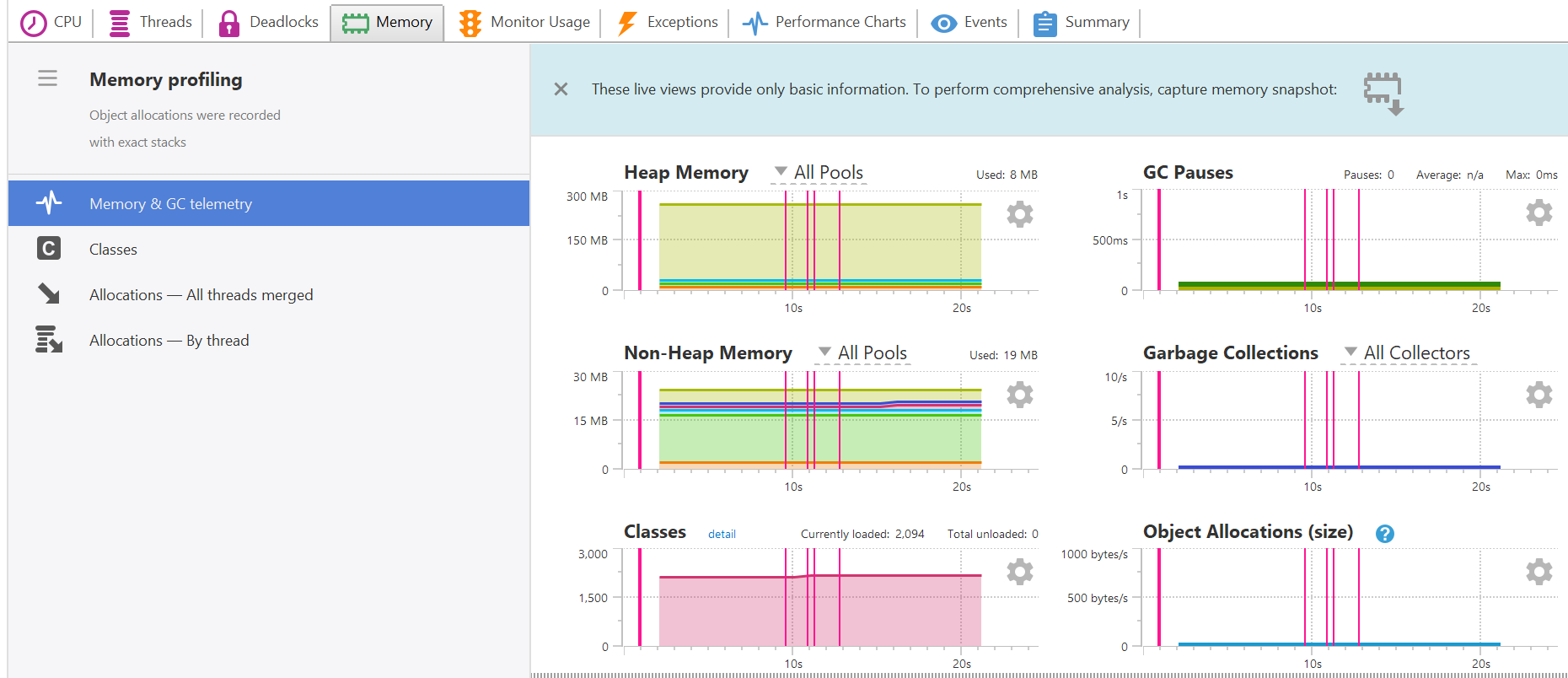
در مرحله بعدی کد را به گونه ای اصلاح کردیم که نه تنها به ارور نمی خورد، بلکه در اثر اجرای سریع حتی نمی توان منابع آن را track کرد.

**CPU**





**MEMORY**

****

Heap Memory که در اینجا مد نظر ما است، خم به ابرو نمی آورد (بر خلاف حالت قبل).

همچنین در تصویر زیر مشاهده می شود که کد بدون هیچ خطایی اجرا و تمام شده است.



تمرین2:

کدهای مربوط به این قسمت را در خود فایل Main.java نوشته ایم.

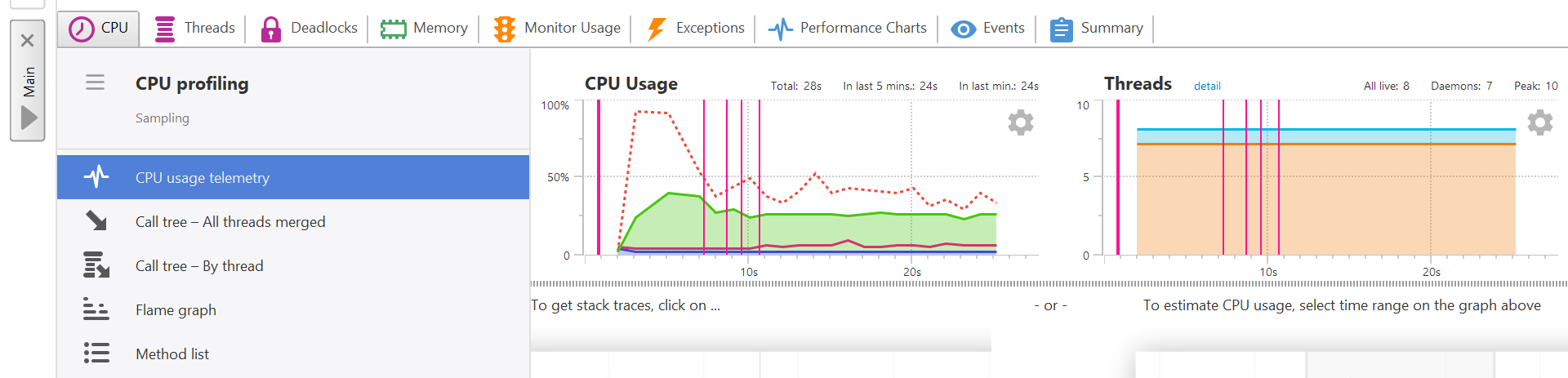
پیاده سازی فرمول جمع انتخاب های پاسکال از دو روش:

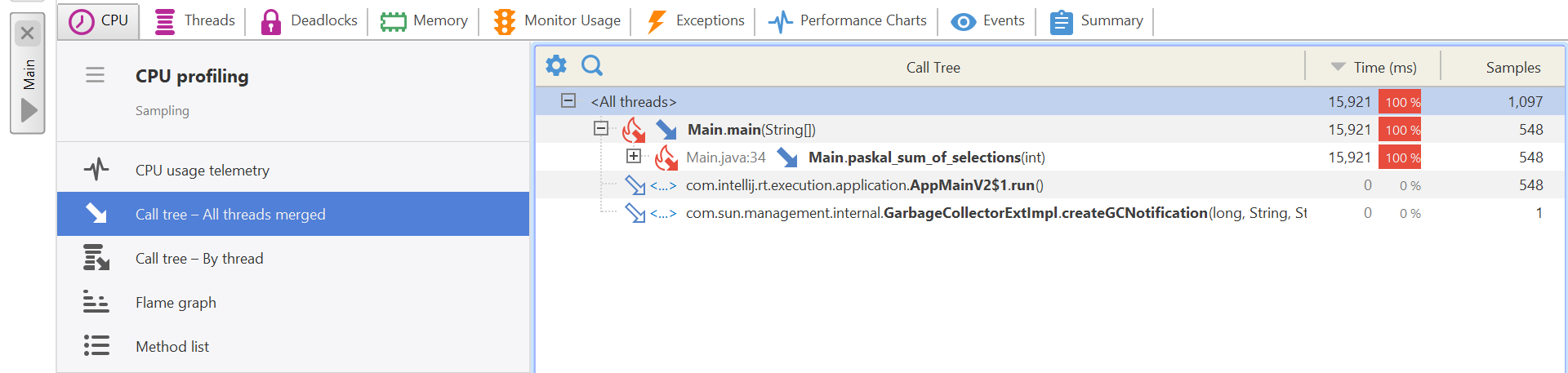
1. پیاده سازی جمع انتخاب ها
2. پیاده سازی توان به عنوان سمت راست تساوی



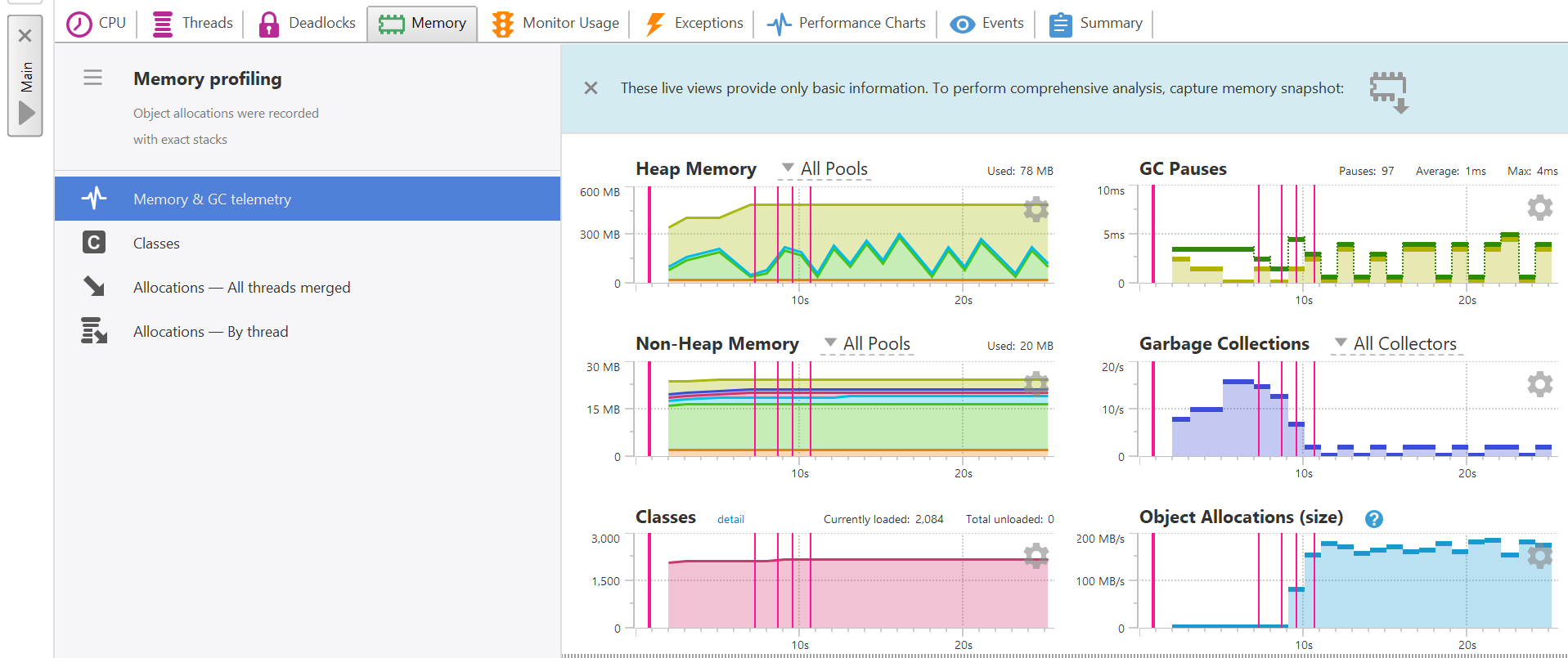
گام اول: پیاده سازی حالت اول:

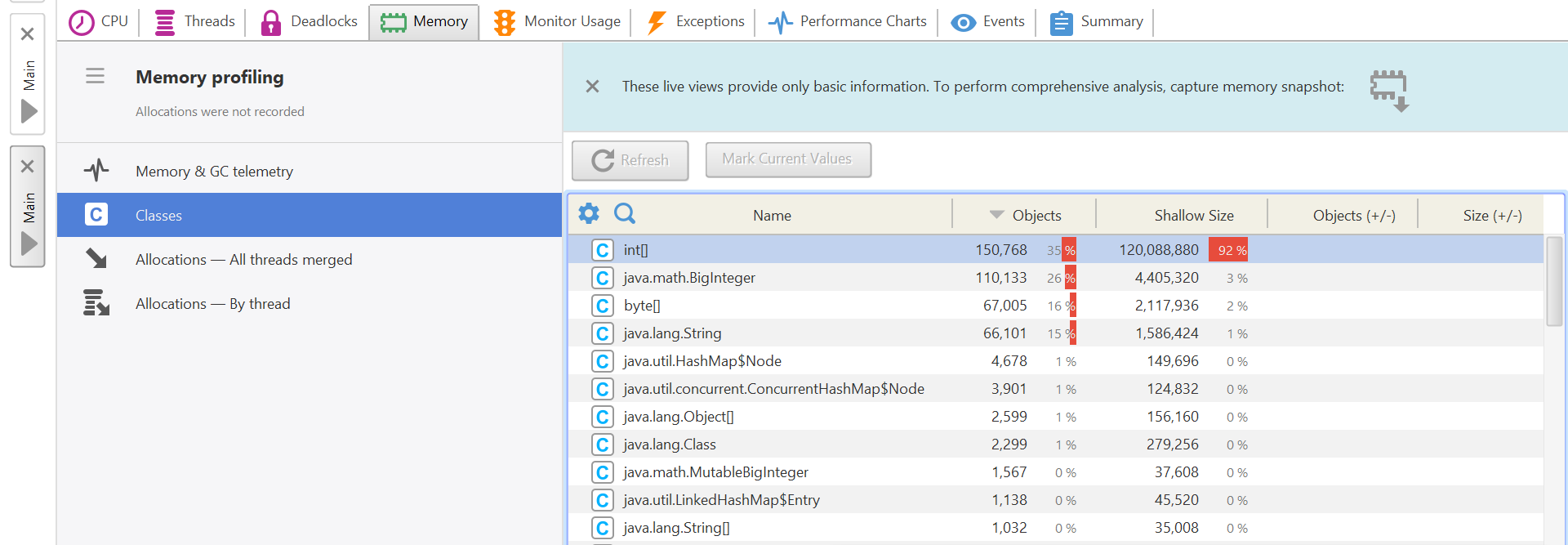
**CPU**

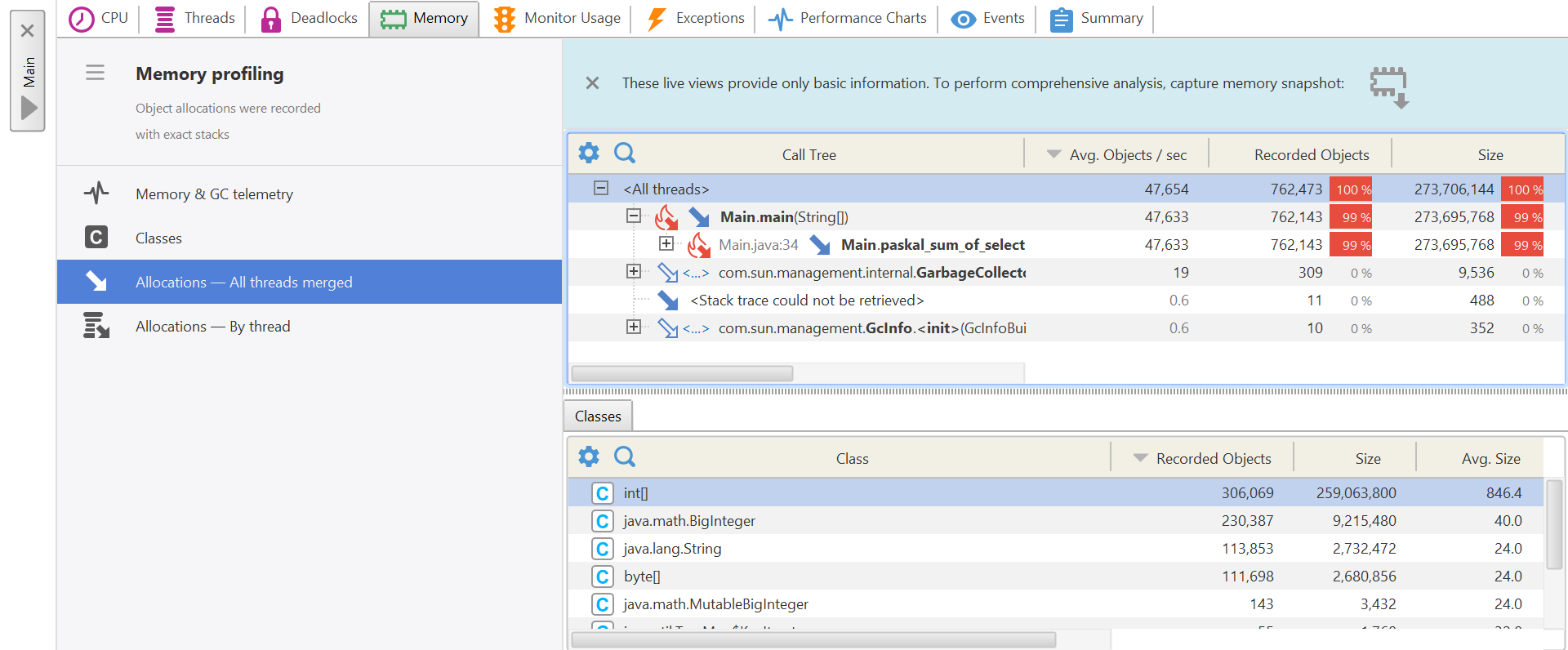




**MEMORY**

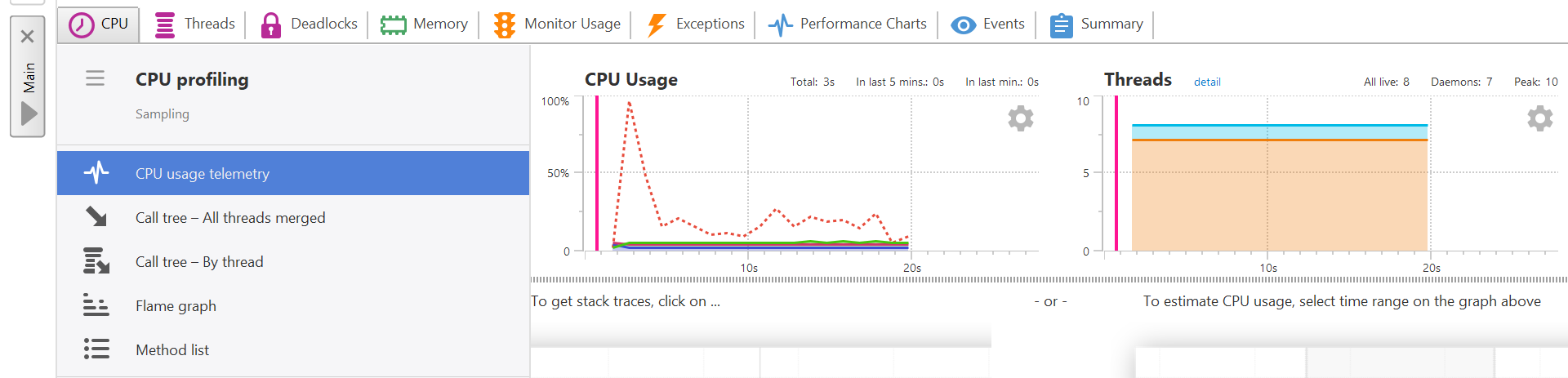
****

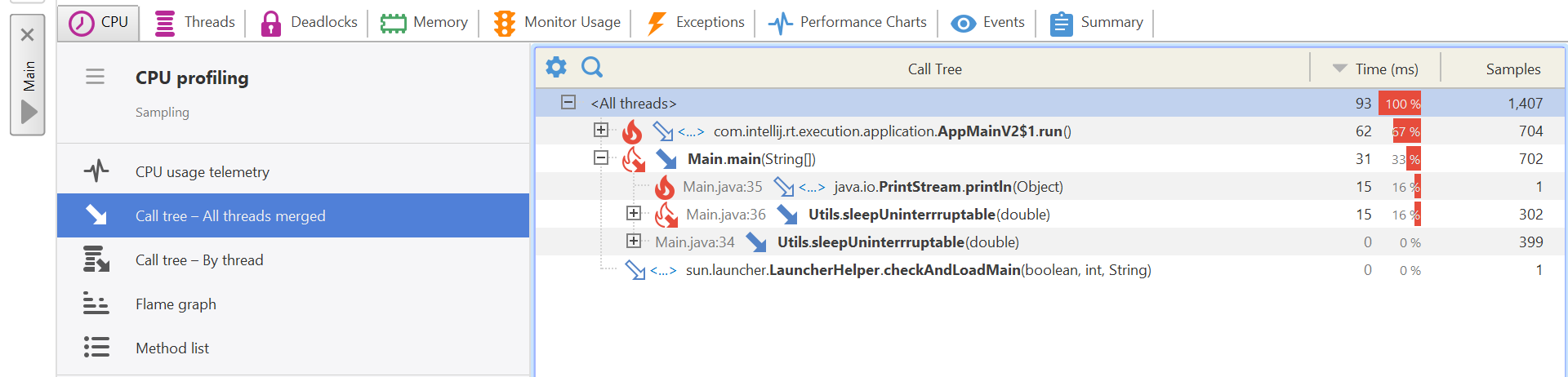
****

****

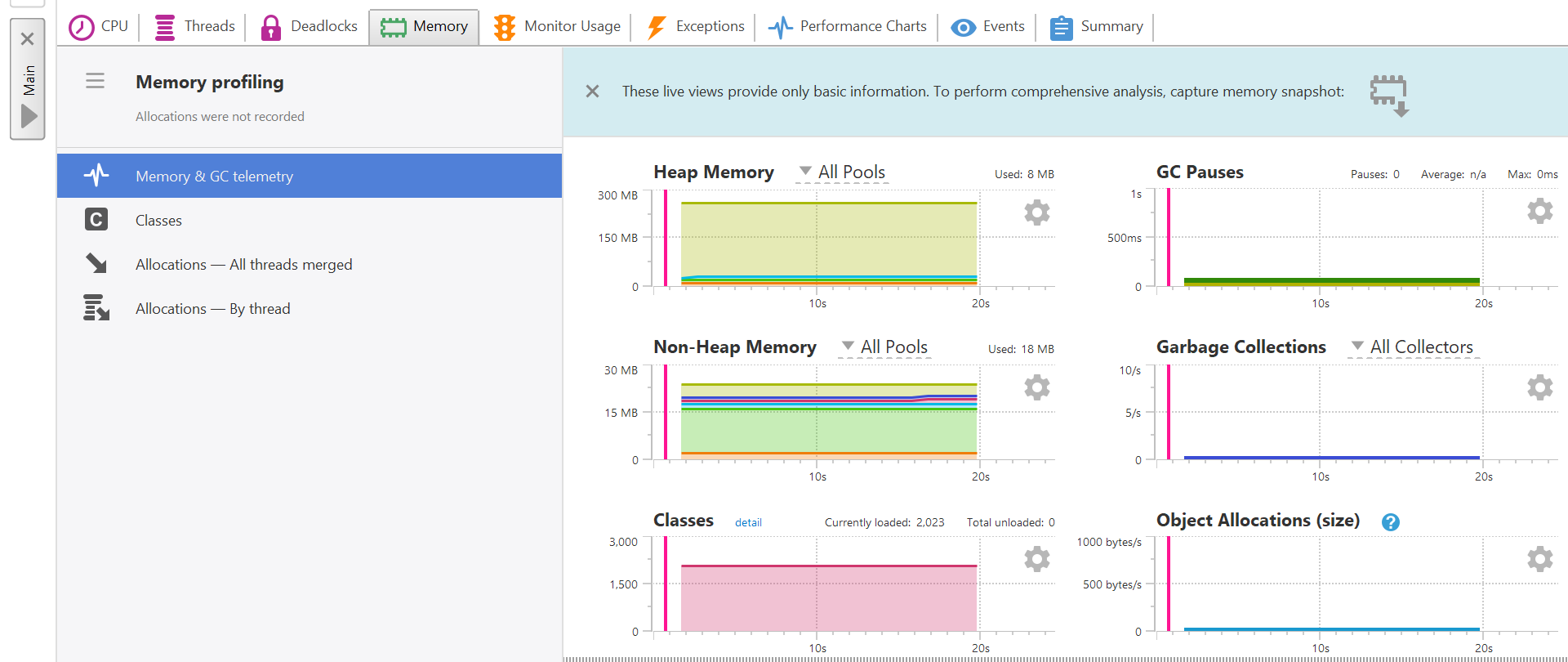
گام دوم: پس از تغییر برنامه به حالت استفاده مستقیم از فرمول پاسکال:

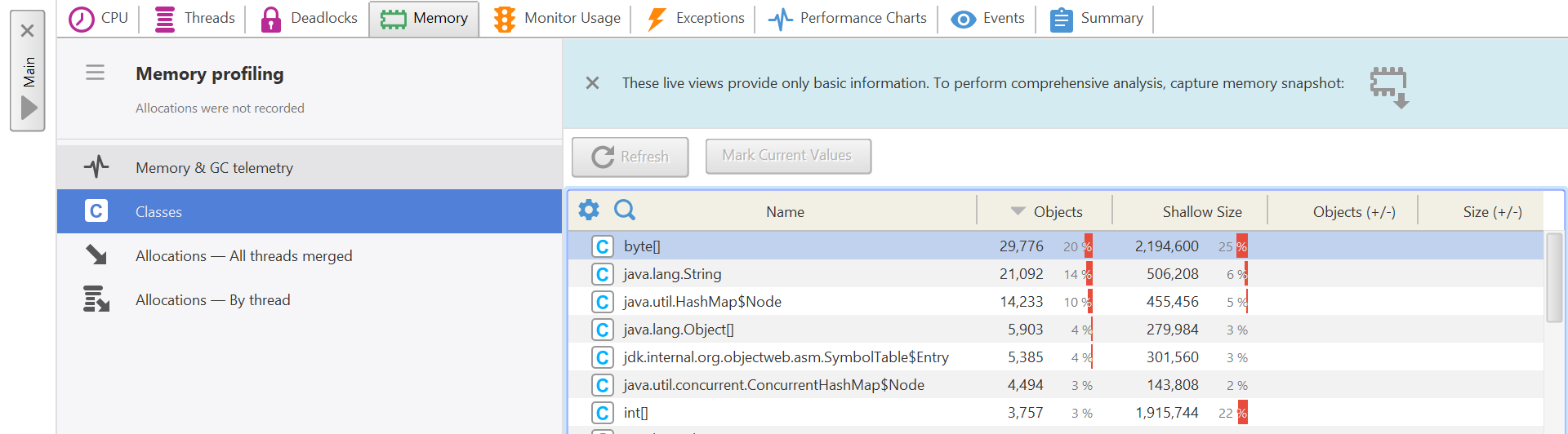
**CPU**

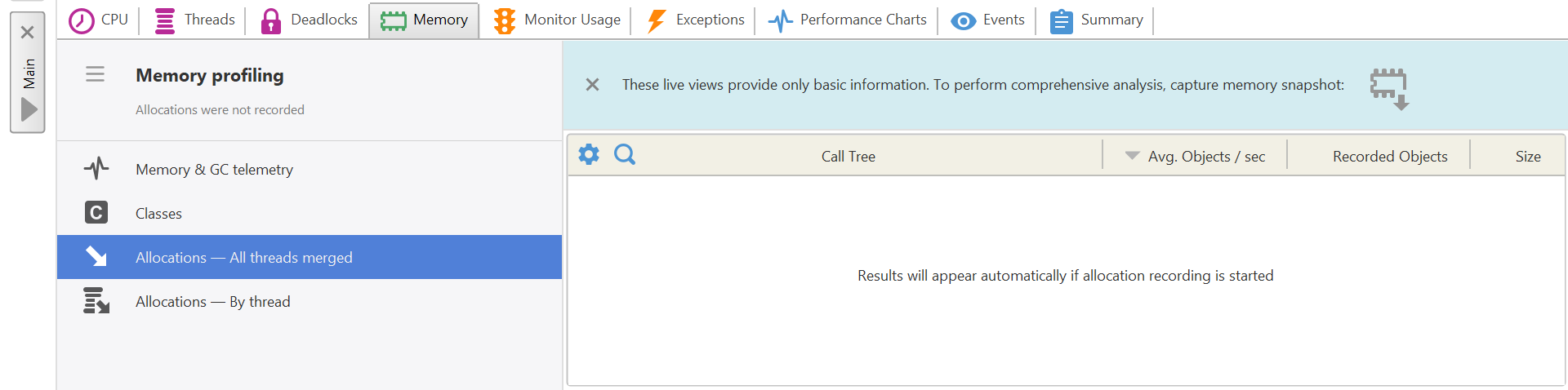
****

****

**MEMORY**

****

****

****

همانطور که مشهودا دیده می شود هم در CPU و هم در Memory بهبود بسیار داریم. (نمودار ها را متناظرا مقایسه کنید.)