



دانشگاه صنعتی شریف
دانشکده مهندسی کامپیوتر

آزمایش چهارم

مهندسی نرم افزار

محمد رضا دولتی

۹۷۱۱۰۴۱۱

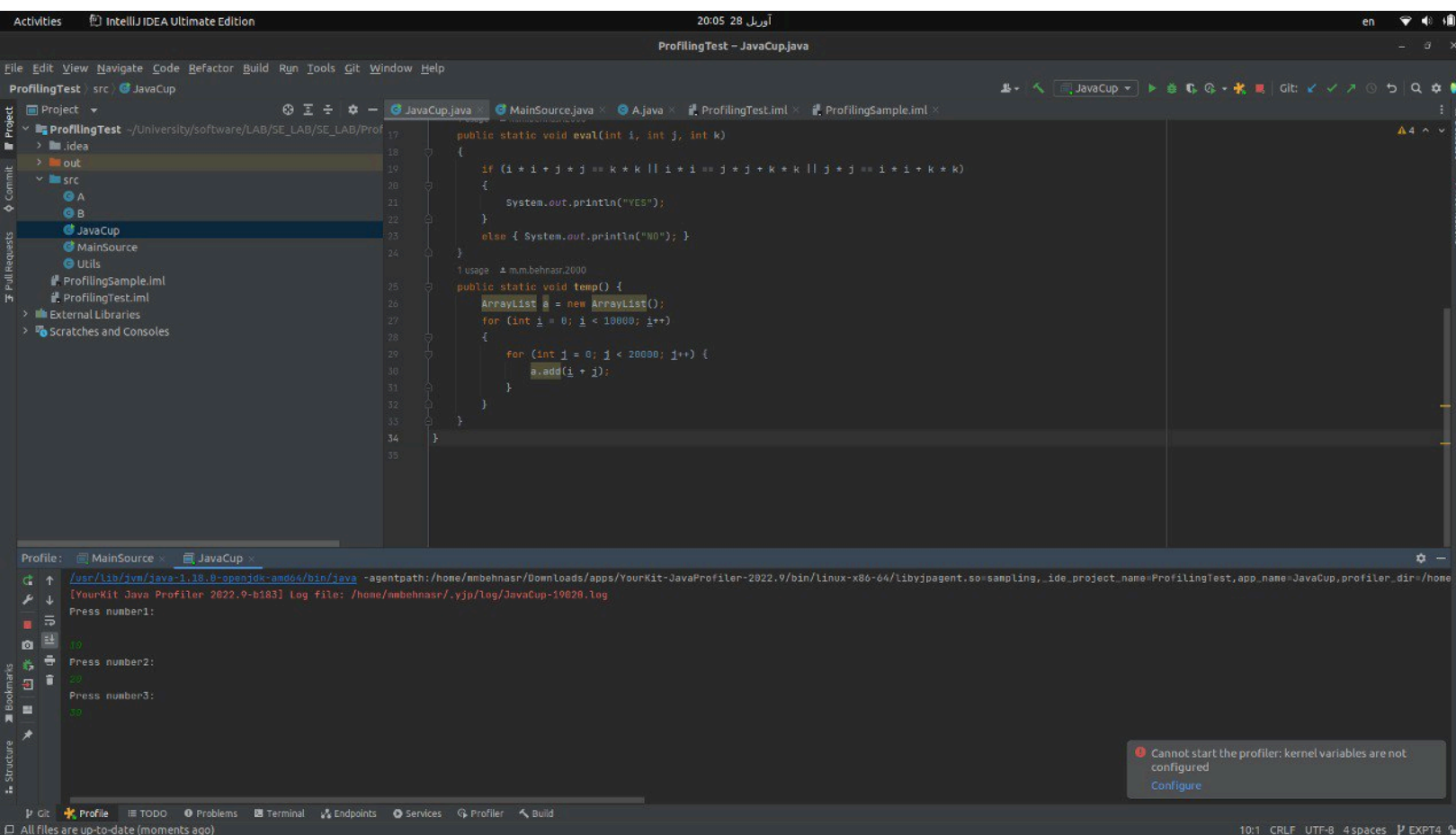
محمد مهدی به نصر

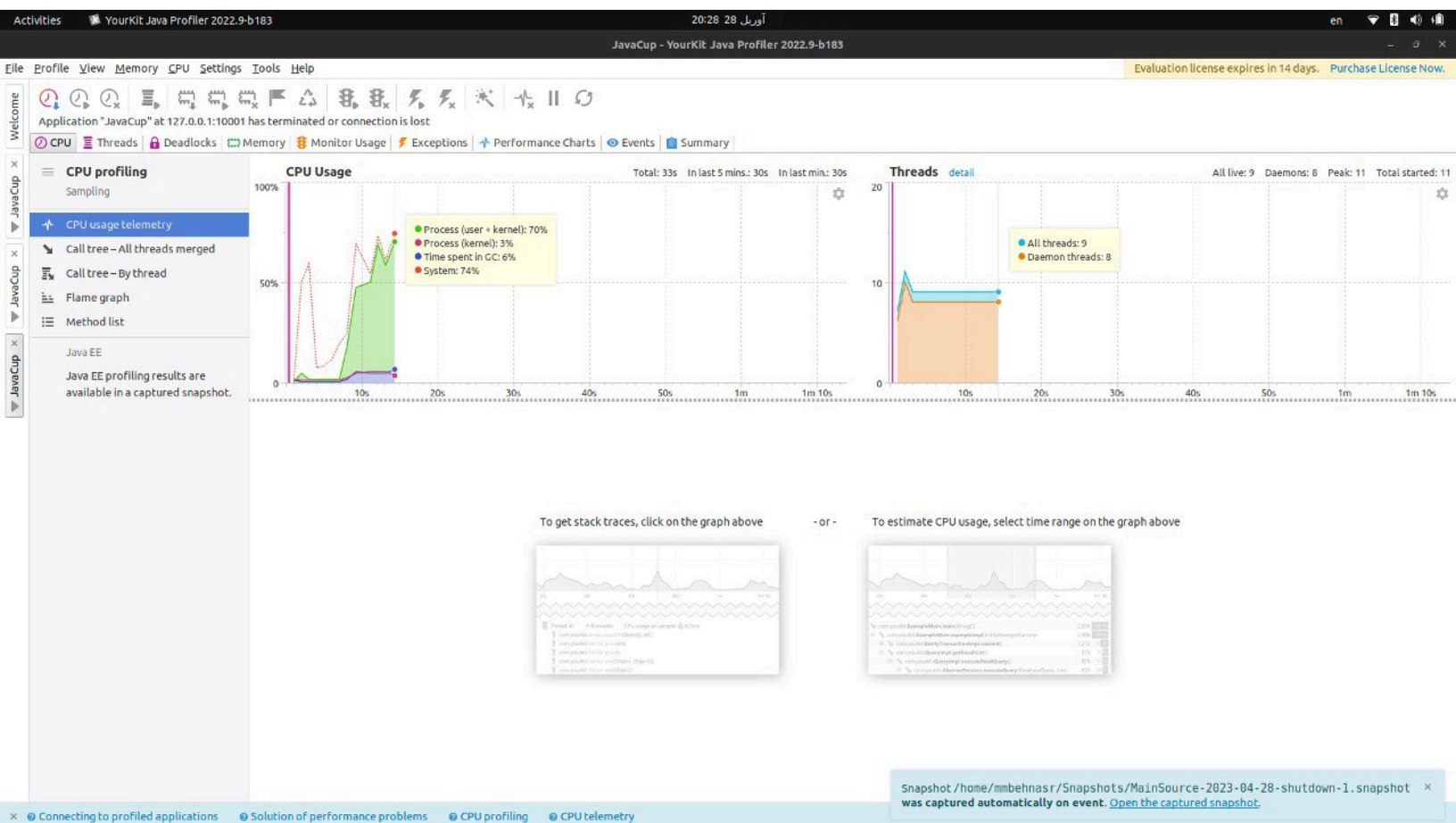
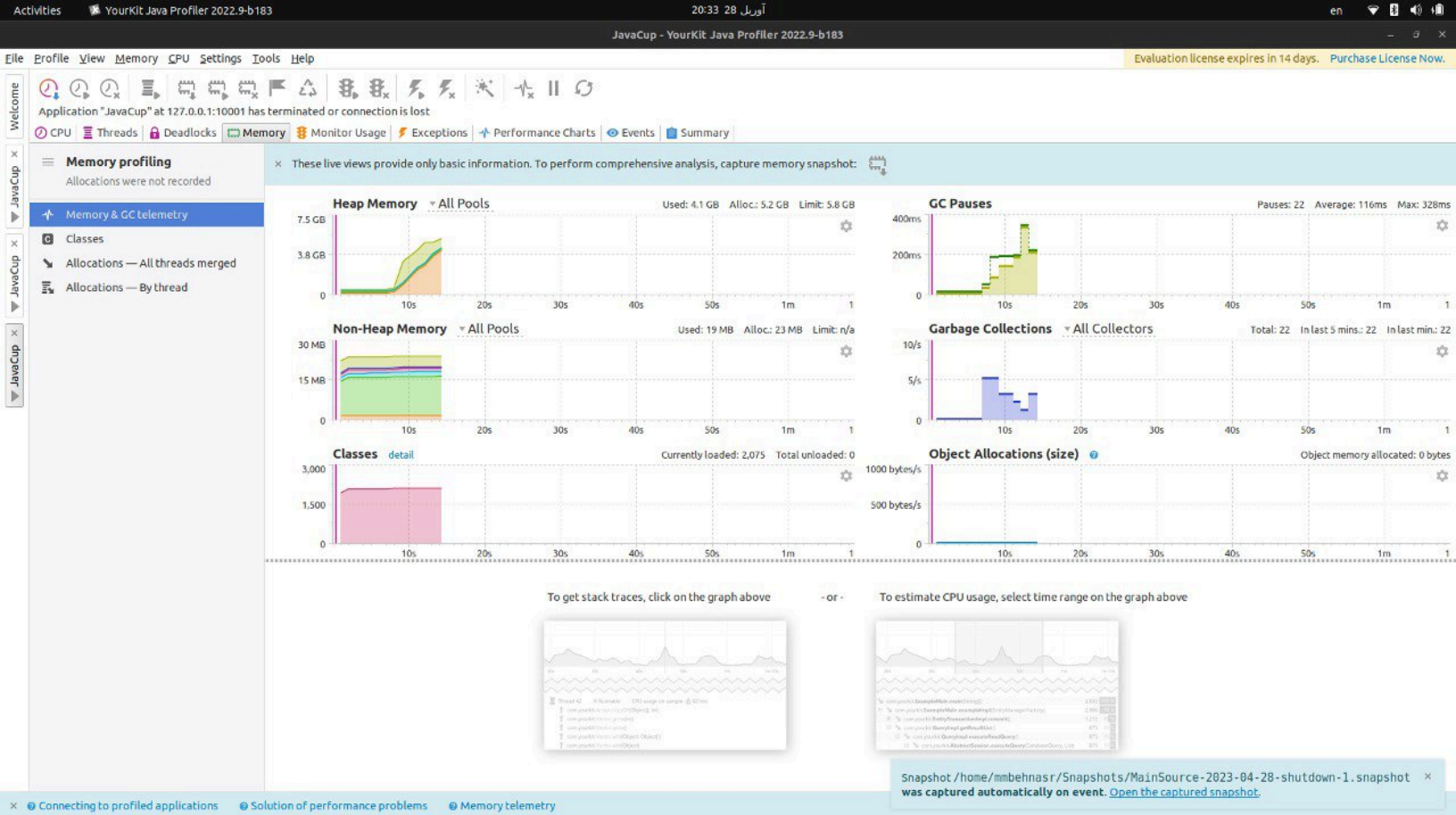
۹۷۱۰۵۷۹۳

اردیبهشت ۱۴۰۲

سوال ۱) برای این سوال در نرم افزار YourKit مشاهده می کنیم که بیشترین منبعی که مصرف می شود توسط تابع temp می باشد که در یک for با 20000 تکرار درون یک for دیگر که 10000 بار تکرار دارد در حال اجراست و در هر تکرار یک مقدار را به ArrayList اضافه می کند. این موضوع سربار زیادی را برای CPU به وجود می آورد.

در تصویر زیر کد نوشته شده در JavaCup و تصاویر مربوط به اجرای برنامه در YourKit و IntelliJ قابل مشاهده است.

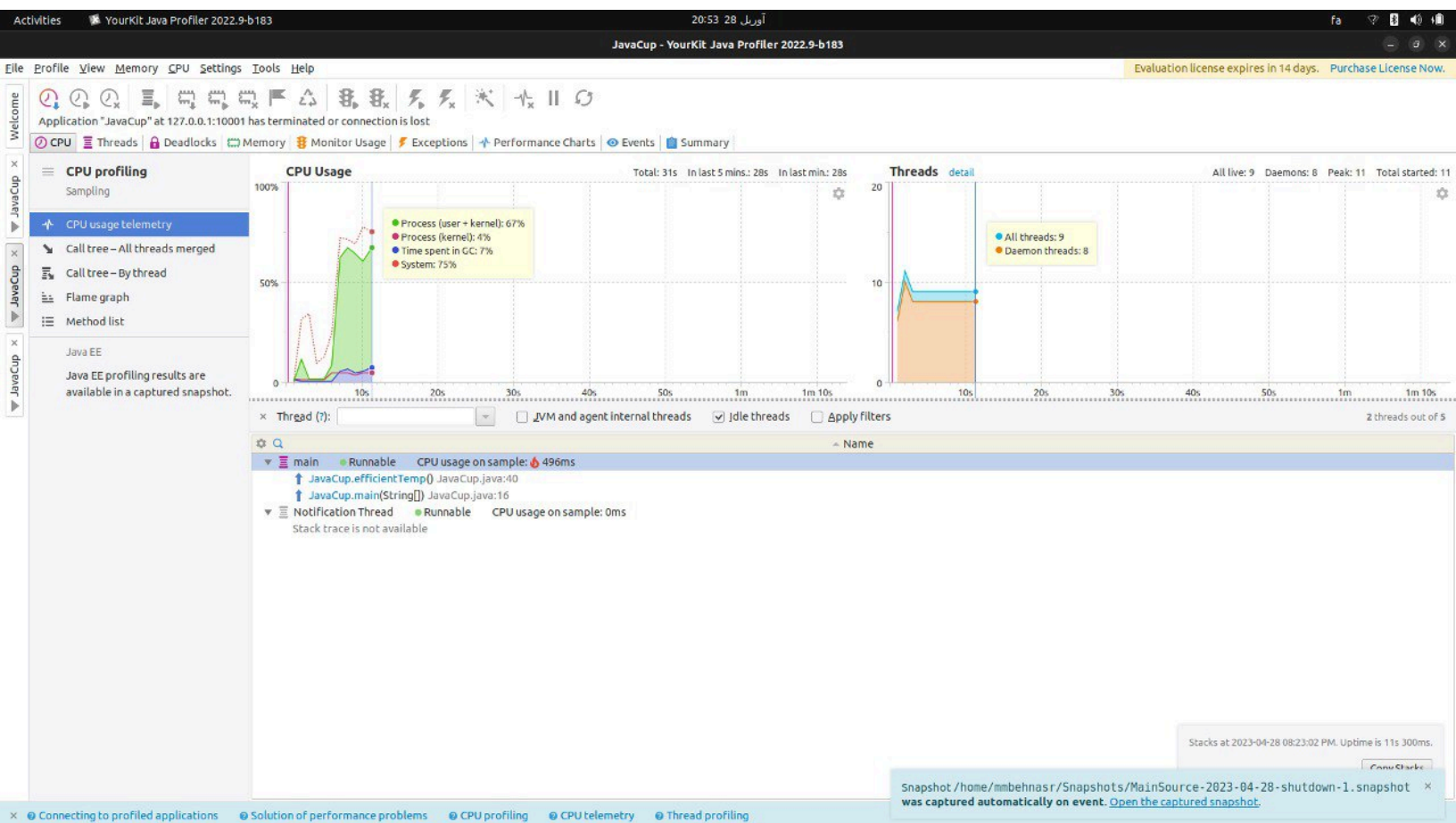




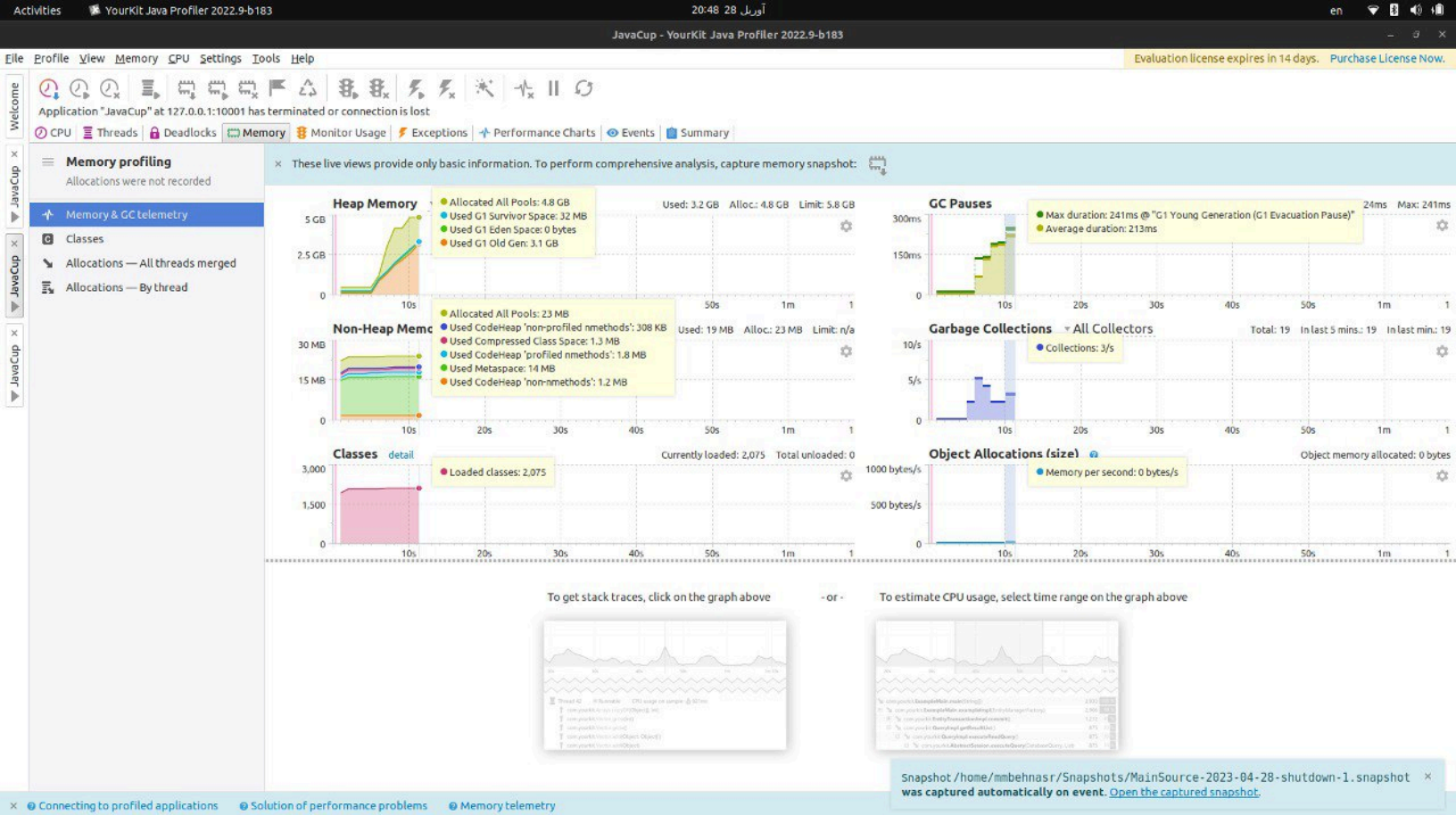
همانطور که از تصاویر صفحه‌ی قبل مشخص است، مصرف حافظه‌ی Heap در طول زمان افزایش می‌یابد (به علت بزرگی حجم آرایه نهایی نیز می‌باشد) و در نهایت برنامه با پر شدن کامل این حافظه خاتمه پیدا می‌کند.

هم‌چنین با مشاهده‌ی نمودارهای دیگر می‌توان برداشت کرد که مصرف حافظه‌ی غیر Heap ای در طول زمان تغییری نخواهد کرد.

سوال ۲) (بهبود بخشیدن temp) برای راه‌حل اول بهبود این نمودارها و تابع temp از تابع efficientTemp استفاده می‌کنیم، این تابع در واقع همان وظایف تابع temp را انجام می‌دهد، با این تفاوت که باید سائز آرایه‌ی a را از همان ابتدا مشخص کنیم، در نتیجه از کپی کردن‌ها بیهوده‌ی internal برای تعیین سائز آرایه جلوگیری می‌کند و همانطور که در تصاویری که در ادامه آمده‌اند، قابل مشاهده است، میزان CPU Usage از 70% به 67% کاهش پیدا کرده است.

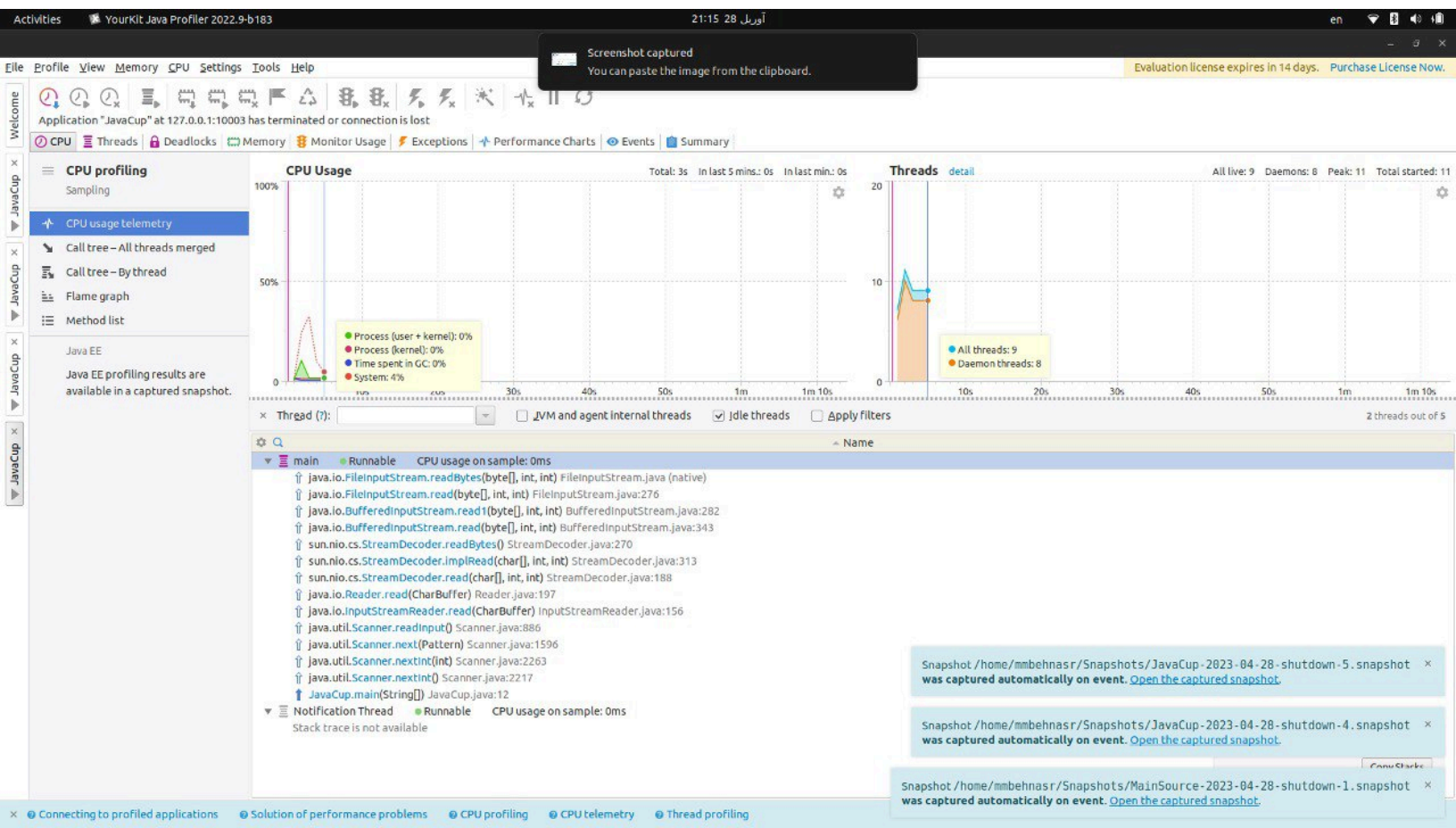


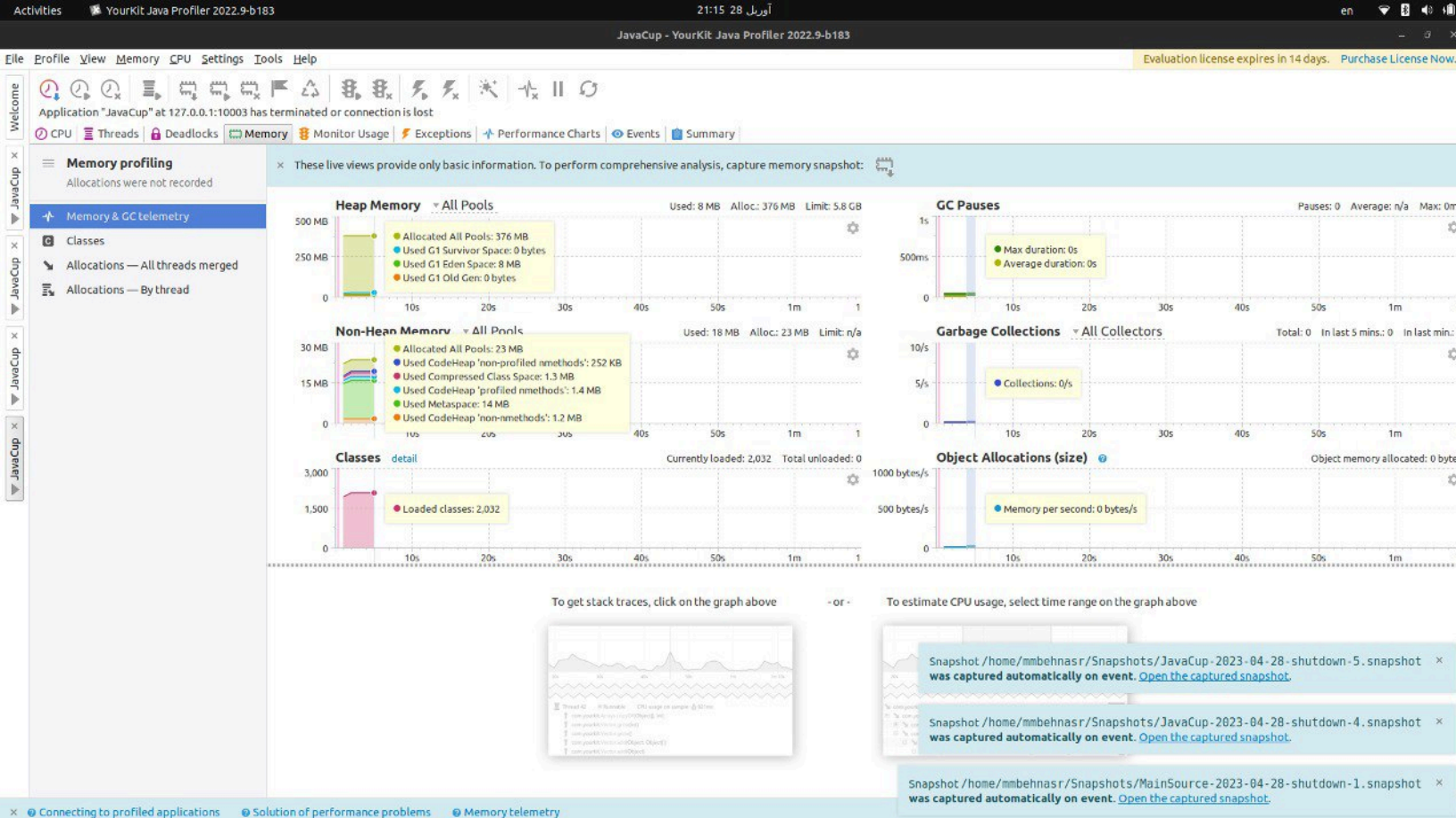
البته مصرف مموری همچنان مشابه قبل است، چون در نهایت هدف تابع پر کردن چنین لیست بزرگی است.



راه ديگر بهبود عملکرد خاص تابع temp را بهينه تر مي کند و مصرف CPU و heap را بسيار بهبود مي بخشد.

در ادامه از تابع efficientTempPlus استفاده مي کنيم، اين تابع مقادير مورد نظر تابع temp را در arraylist مي ريزد. حال با اجراي برنامه با YourKit داريم:





همانطور که مشخص است، مشکل heap نیز برطرف شده است.