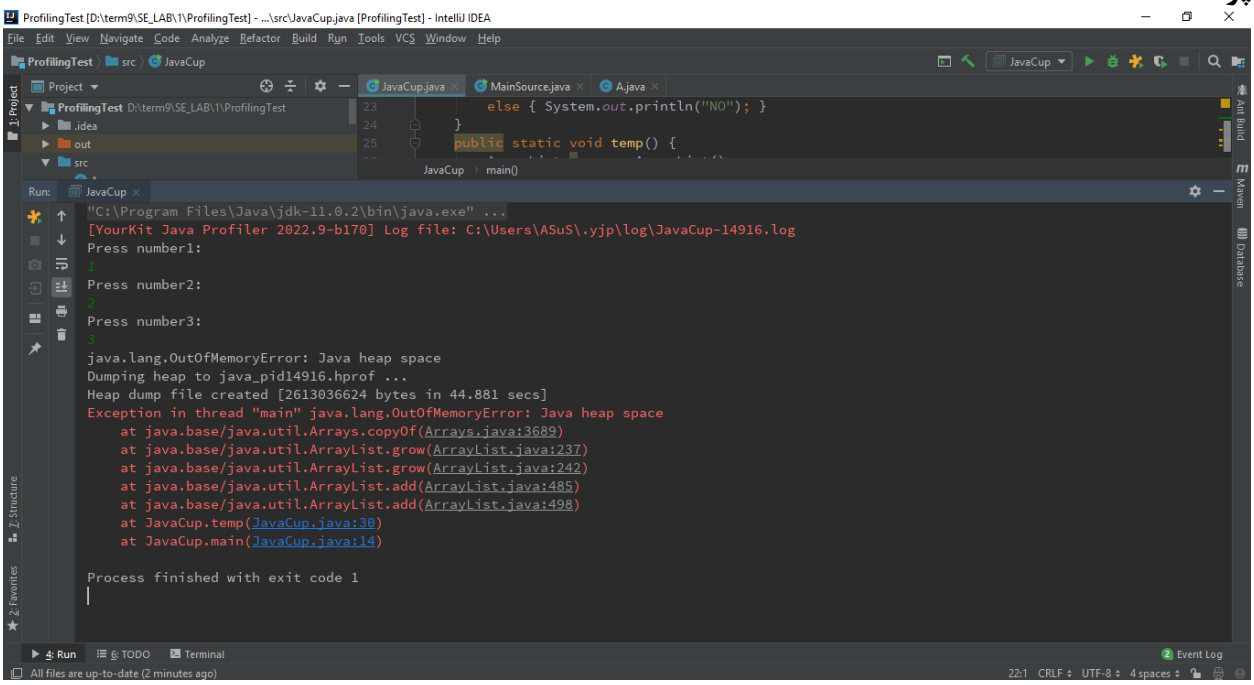
به نام خدا

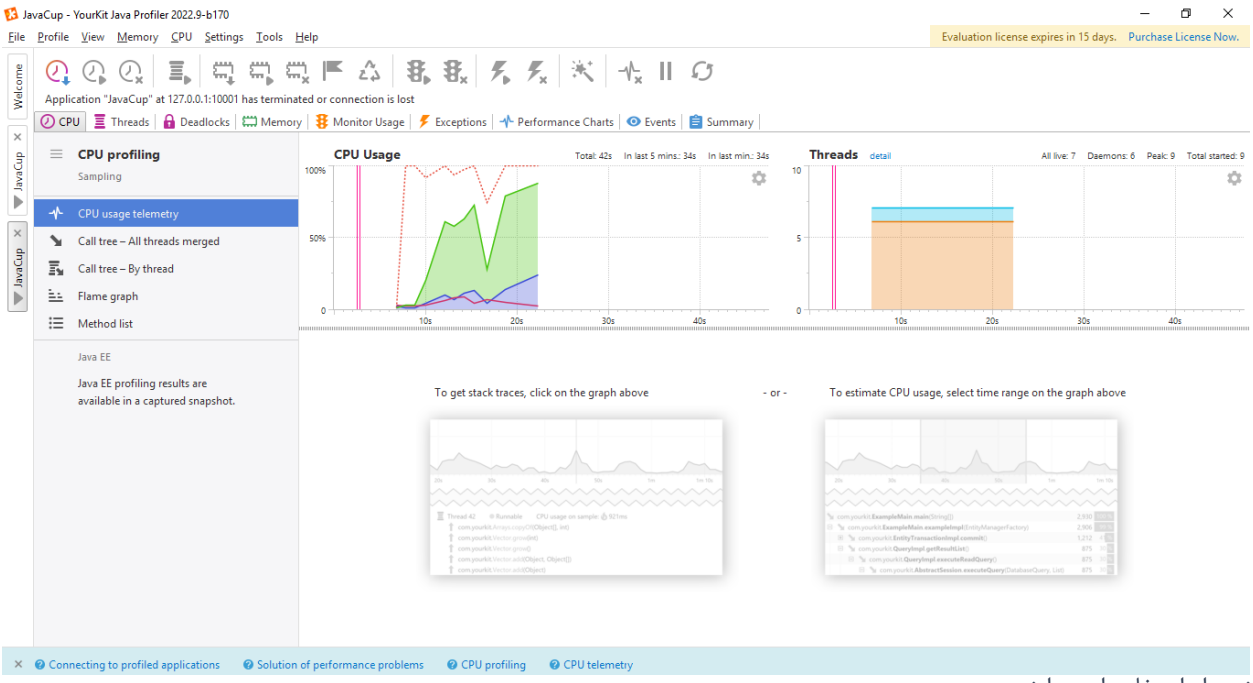
تمرین ۴
آز نرم افزار

احمد نصرت بخش
97110217

ابتدا به بررسی تصاویری از مراحل نصب برنامه YourKit-JavaProfiler میپردازیم. با مراجعه به وبسایت https://www.yourkit.com/download ، آخرین نسخه مطابق با سیستم عامل(در اینجا ویندوز) را دانلود کرده و سپس آن را اجرا میکنیم. برای دریافت مجوز ۱۵ روزه ایمیل خود را وارد و lisence key را دریافت میکنیم.
سپس کلید ارسال شده را در برنامه وارد کرده و مجوز استفاده از آن فعال میشود.
در پروژه ProfilingTest ، عملیات Profiling را بر روی کلاس JavaCup اجرا کرده و با استفاده از YourKit به بررسی مصرف منابع، تحلیل حافظه و CPU و Thread ها خواهیم پرداخت.

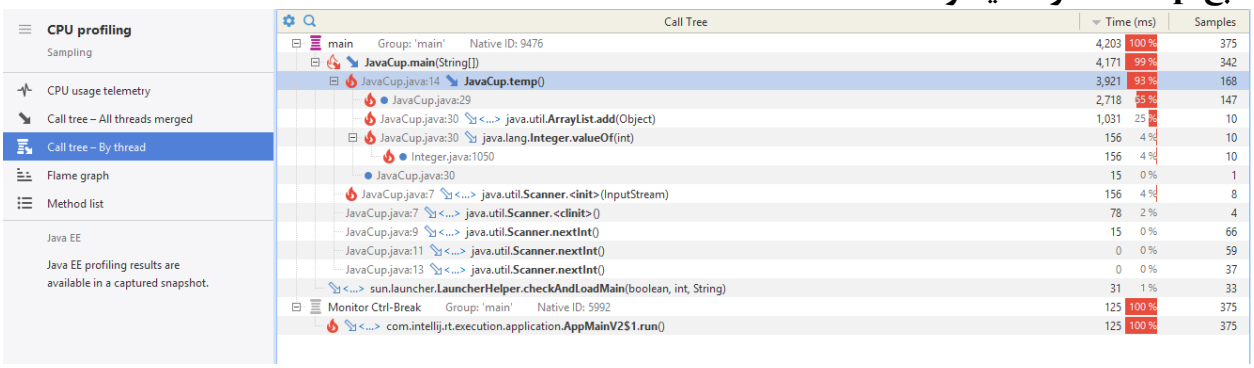


خروجی برنامه بعد از اجرای عملیات profiling

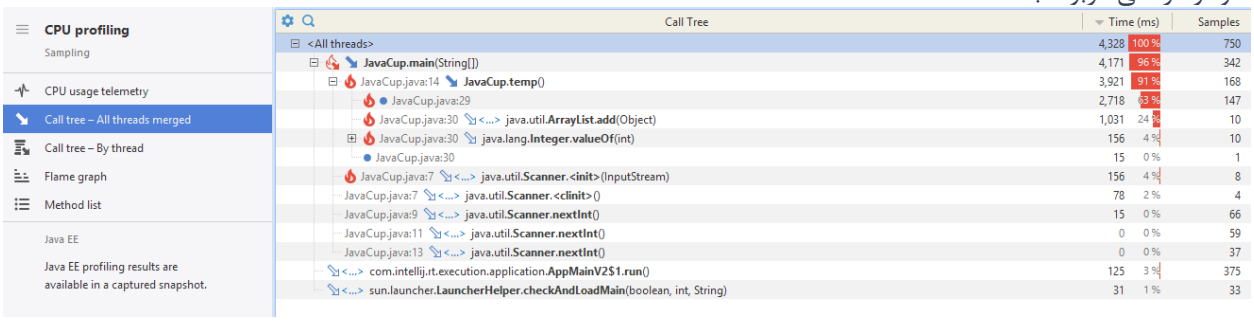


نمودار استفاده از پردازنده

طبق نمودار مربوط به thread ها و زمان اجرای توابع مشاهده میشود ۹۱درصد زمان اجرایی توسط تابع temp مصرف میشود.



نمودار درختی مربوط به thread ها

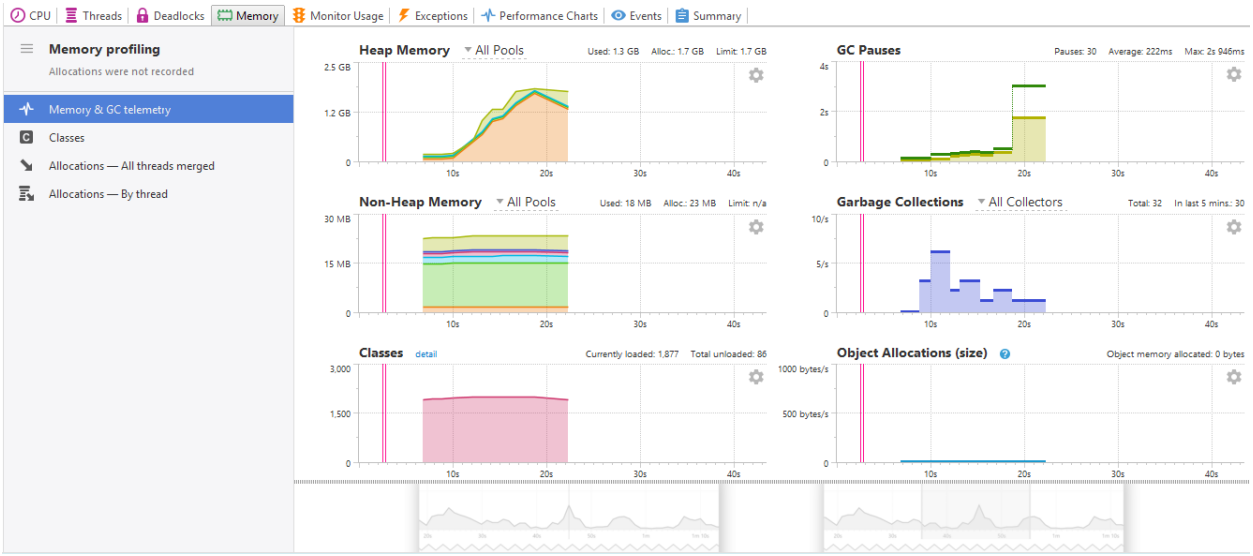


نمودار درختی thread ها و زمان اجرای توابع

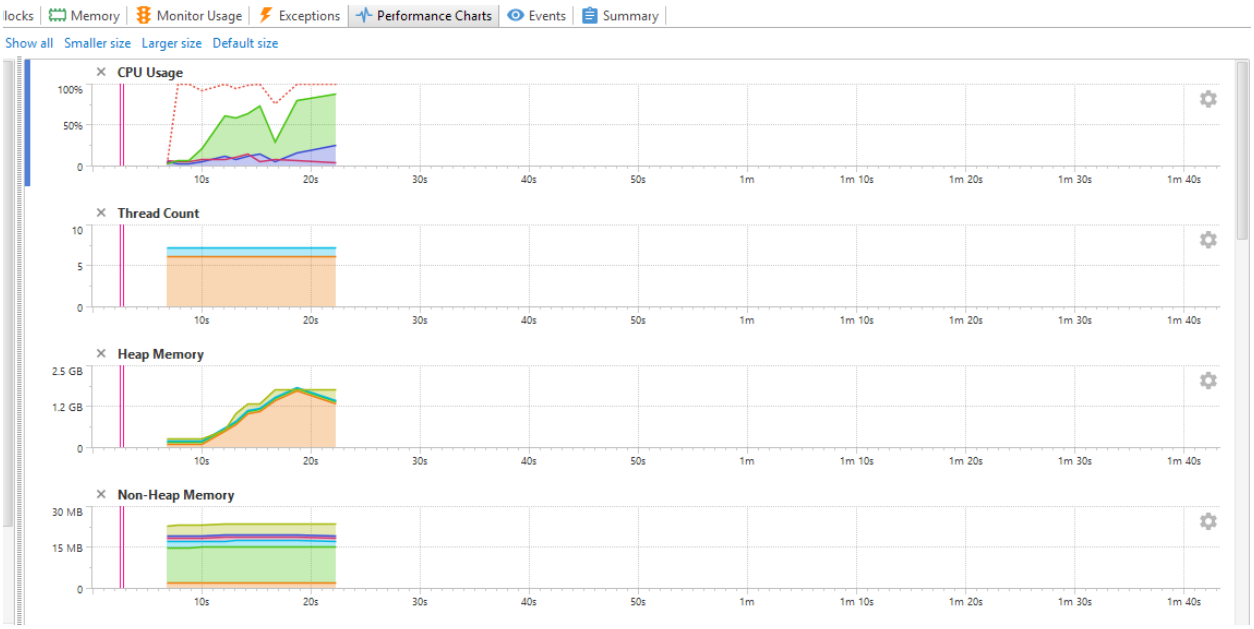
Method	Time (ms)	Own Time (ms)	Samples
JavaCup.main(String[]) JavaCup.java	4,171 96 %	0	342
JavaCup.temp() JavaCup.java	3,921 91 %	2,734	168
java.util.**ArrayList.add**(Object) ArrayList.java	1,031 24 %	1,031	10
java.lang.**Integer.valueOf**(int) Integer.java	156 4 %	156	10
java.util.**Scanner.<init>**(InputStream) Scanner.java	156 4 %	156	8
com.intellij.rt.execution.application.**AppMainV2$1.run**() AppMainV2.java	125 3 %	125	375
java.util.**Scanner.<clinit>**() Scanner.java	78 2 %	78	4
sun.launcher.**LauncherHelper.checkAndLoadMain**(boolean, int, String) LauncherHelper.java	31 1 %	31	33
java.util.**Scanner.nextInt**() Scanner.java	15 0 %	15	162

نمودار زمان مصرفی و درصد زمانی مربوط به توابع

همچنین طبق نمودارهای مربوط به حافظه مصرفی داریم:



نمودار حافظه مصرفی



نمودار کارایی پردازنده و حافظه وthread ها

طبق بررسی نمودارهای بالا و حجم بالای مصرف منابع و زمان اجرایی توسط تابع temp ، به بررسی دقیقتر آن میپردازیم.

تابعtemp

در این تابع، ازArrayList جاوا استفاده شده و ۱۰۰۰۰ * ۲۰۰۰۰ مرتبه به آن عضوی ر اضافه میکند. این داده ساختار به صورت فضای پویا پیادهسازی شده و هربار که فضای آن تکمیل شود، سایز خود را دوبرابر میکند. به همین علت منابع و زمان اجرایی بالایی را مصرف کرده و heap پر میشود. (خطای پس از اجرای برنامه نیز به همین علت بود).