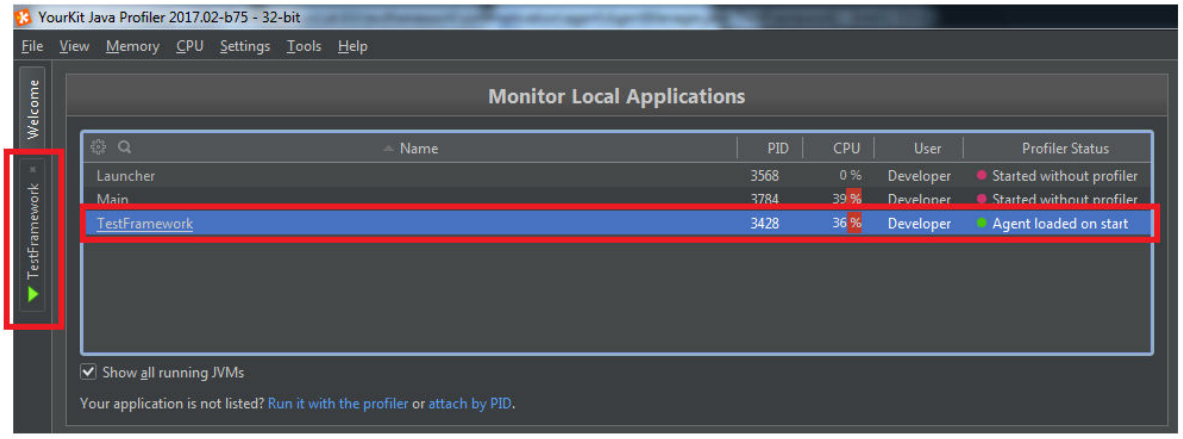
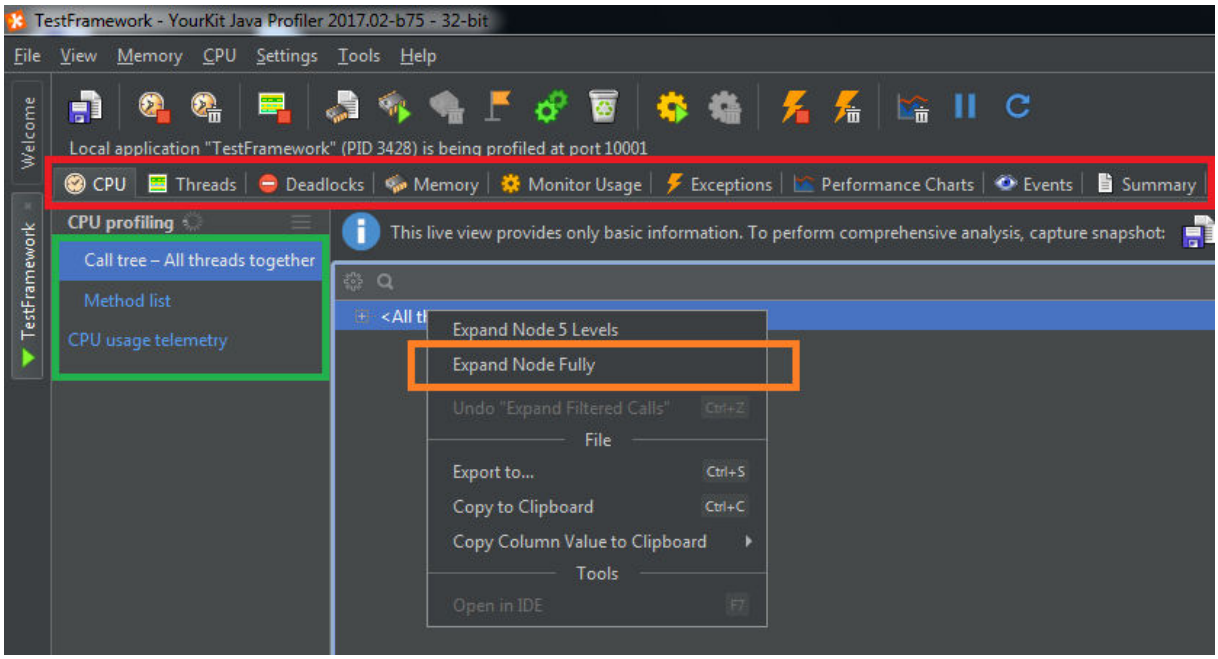
Yourkit základná príručka

1. Pri spustení projektu je vidieť viacero aplikácií, ktoré je možné pomocou Yourkitu analyzovať. Dvojklikom na náš TestFramework sa vytvorí další TAB s monitorovaním (v ľavo).



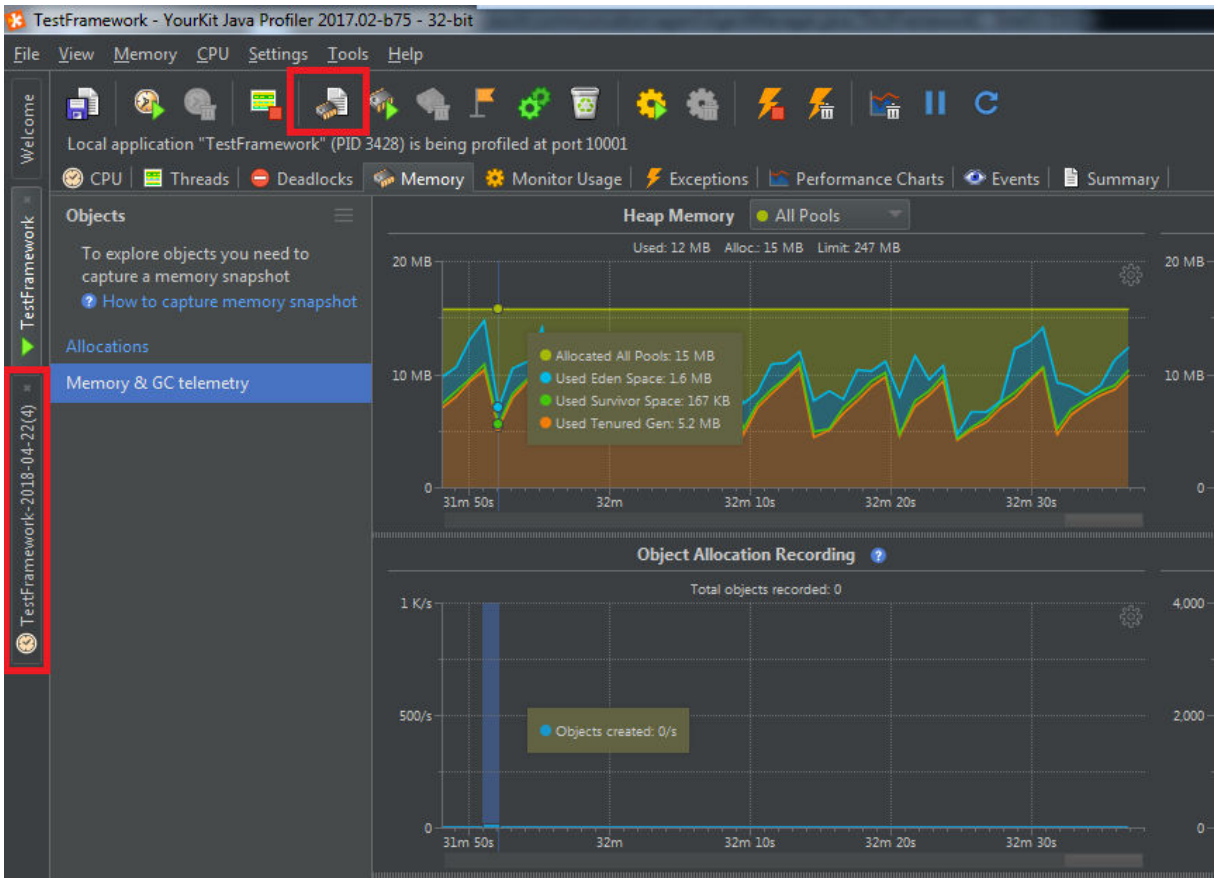
Obr.1 - všetky aplikácie

2. Zobrazí sa monitorovacie menu pre projekt. V červenom obdĺžniku je zobrazené rozdelenie monitorovania na jednotlivé typy. V zelenom obdĺžniku je zobrazené menu pre výber zobrazenia všetkých threadov a ich jednotlivých metód alebo samostatne len metód. Pomocou funkcie expand node fully sa rozbalí automaticky celý strom.



Obr.2 - monitorovanie základ

3. Existuje možnosť vytvorenia snapshotu aktuálneho stavu merania. Vytvorí sa nový súbor, ktorý sa zobrazí opäť v bočných taboch a je možné ho analyzovať v offline režime aplikácie.

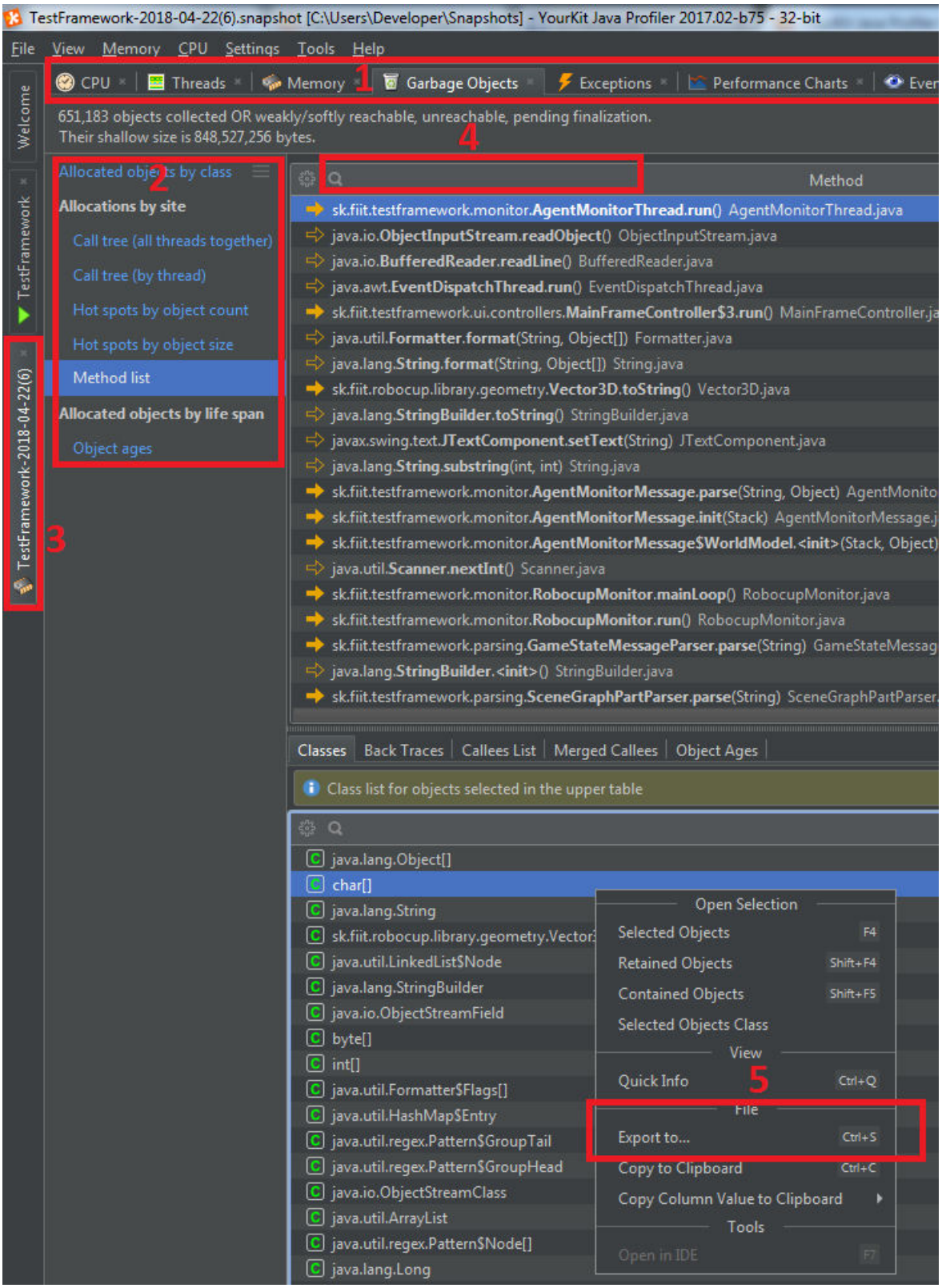


Obr.3 - snapshot

4. Pre zaznamenanie Garbage collectora pre aplikáciu je potrebné pred snapshotom spustiť Object allocation recording.

Obr.4 - allocation recording

5. Pri analýze snapshotu môžeme pracovať s následovným:
    - 1 - panel pre prepínanie rôznych monitorovacích módov
    - 2 - v rámci niektorého monitorovacieho módu je možné zobrazovať rôzne štruktúry (thready, metódy, triedy...) a sledovať ich využitie
    - 3 - tab snapshotu
    - 4 - pole pre vyhľadávanie
    - 5 - je možnosť exportu monitorovacieho módu napríklad do .csv súboru



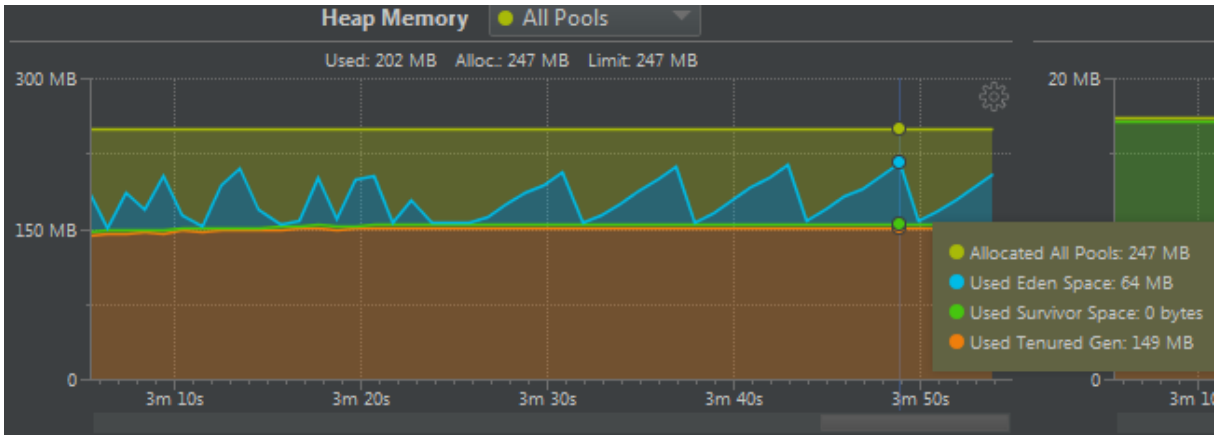
Obr.5 - práca so snapshotom

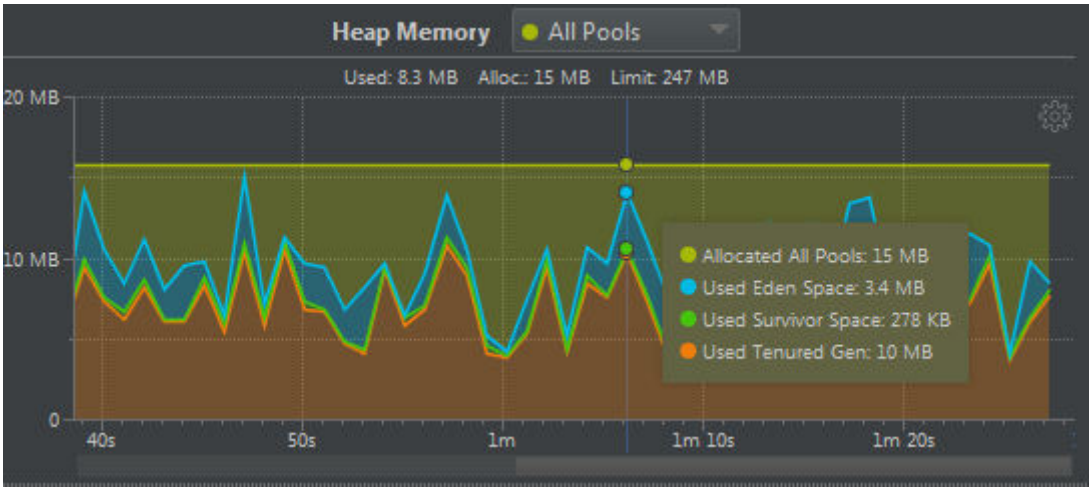
6. Prehľad performance a jednotlivých tried je možné zobraziť v grafoch.

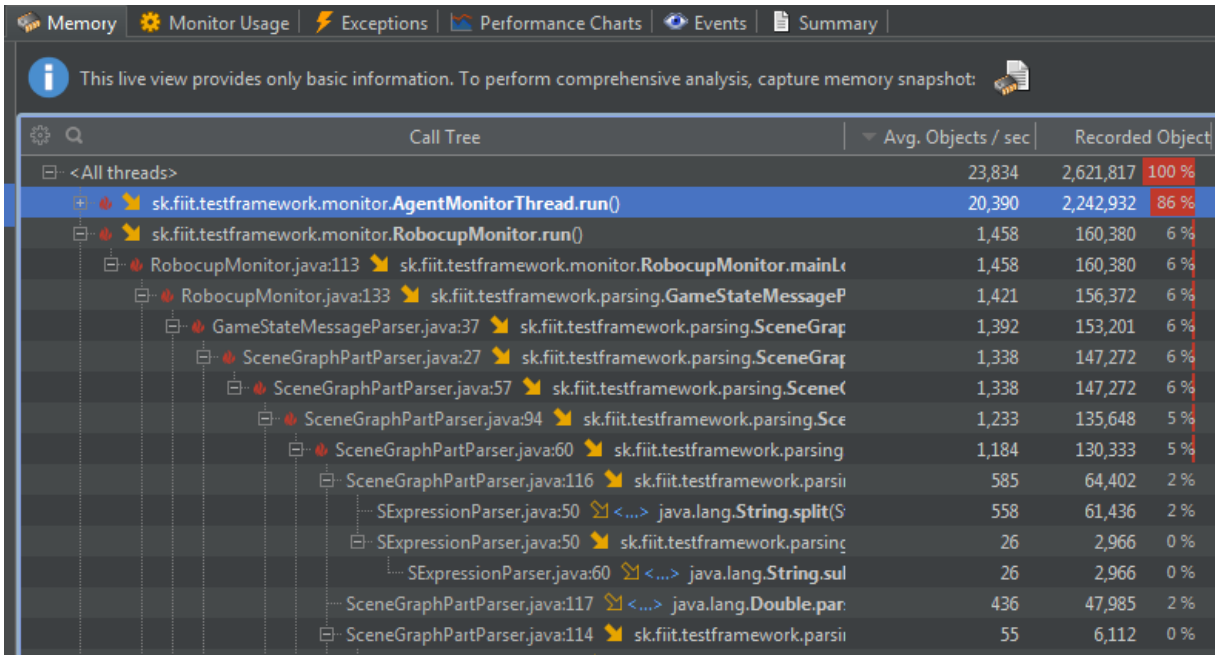
Obr.6 - performance

Zaujímavosti:

- sledovanie exceptions
- detekcia deadlockov
- grafy performance
- základný sumár
- sledovanie memor
- sledovanie CPU
- rozdelenia úrovní
    - na základe threadov
    - na základe metód
    - na základe tried
    - na základe packagov
    - celková stromová štruktúra
- export do .csv
- grabage collection hodnoty


Heap Memory
All Pools
Used: 202 MB Alloc.: 247 MB Limit: 247 MB
300 MB
150 MB
0
3m 10s
3m 20s
3m 30s
3m 40s
3m 50s
20 MB
0
Allocated All Pools: 247 MB
Used Eden Space: 64 MB
Used Survivor Space: 0 bytes
Used Tenured Gen: 149 MB


Heap Memory
All Pools
Used: 8.3 MB Alloc.: 15 MB Limit: 247 MB
20 MB
10 MB
0
40s
50s
1m
1m 10s
1m 20s
Allocated All Pools: 15 MB
Used Eden Space: 3.4 MB
Used Survivor Space: 278 KB
Used Tenured Gen: 10 MB


Memory
Monitor Usage
Exceptions
Performance Charts
Events
Summary
This live view provides only basic information. To perform comprehensive analysis, capture memory snapshot:
Call Tree
Avg. Objects / sec
Recorded Object
<All threads> 23,834 2,621,817 100 %
sk.fiit.testframework.monitor.AgentMonitorThread.run() 20,390 2,242,932 86 %
sk.fiit.testframework.monitor.RobocupMonitor.run() 1,458 160,380 6 %
RobocupMonitor.java:113 sk.fiit.testframework.monitor.RobocupMonitor.mainL 1,458 160,380 6 %
RobocupMonitor.java:133 sk.fiit.testframework.parsing.GameStateMessageP 1,421 156,372 6 %
GameStateMessageParser.java:37 sk.fiit.testframework.parsing.SceneGrap 1,392 153,201 6 %
SceneGraphPartParser.java:27 sk.fiit.testframework.parsing.SceneGrap 1,338 147,272 6 %
SceneGraphPartParser.java:57 sk.fiit.testframework.parsing.Scene(1,338 147,272 6 %
SceneGraphPartParser.java:94 sk.fiit.testframework.parsing.Sce 1,233 135,648 5 %
SceneGraphPartParser.java:60 sk.fiit.testframework.parsing 1,184 130,333 5 %
SceneGraphPartParser.java:116 sk.fiit.testframework.parsi 585 64,402 2 %
SExpressionParser.java:50 <...> java.lang.String.split(S 558 61,436 2 %
SExpressionParser.java:50 sk.fiit.testframework.parsing 26 2,966 0 %
SExpressionParser.java:60 <...> java.lang.String.sul 26 2,966 0 %
SceneGraphPartParser.java:117 <...> java.lang.Double.par 436 47,985 2 %
SceneGraphPartParser.java:114 sk.fiit.testframework.parsi 55 6,112 0 %