

# Analiza statica de cod

## -Verificare si validare software-

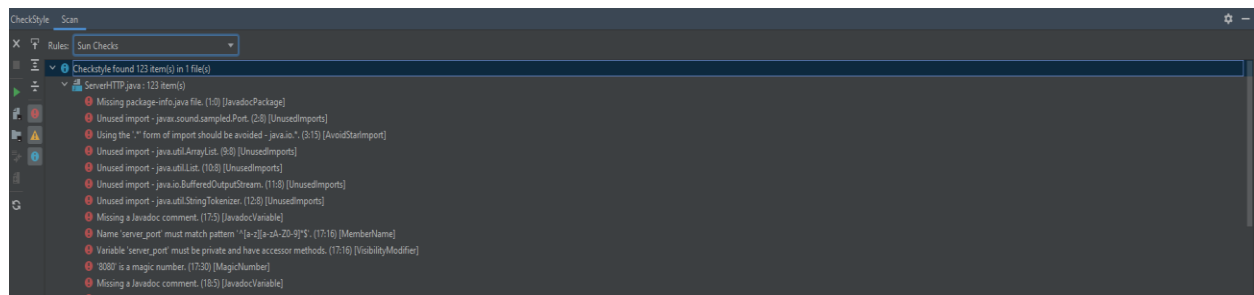
**Concept analiza statica:** reprezinta o analiza a codului sursa(fara a executa programul) cu scopul de a determina proprietati ale programului sursa.In mod principal corectitudinea programului dar si performanta.

### Exemple:

- Variabile neinitializate
- Atribuirii nefolosite
- Vulnerabilitati de cod caracterizate prin exceptii sau depasiri de indici.

Pentru partea de analiza statica am folosit plug-in-ul CheckStyle, disponibil in IntelliJ IDEA.

In urma analizei statice de cod pe clasa ServerHTTP.java am primit urmatorul feedback.



Pentru optimizarea codului am rezolvat erorile raportate de catre CheckStyle.

- Am sters tot ce era “unused”,primele erori vin de la importuri adaugate dar nefolosite. Problema care poate aparea din cauza acestor importuri este namespace conflict. Totusi aceste importuri sunt sterse de catre compilator dupa compilare,deci nu afecteaza runtime-ul.
- “Variable 'server\_port' must be private and have accessor methods. (18:16) “ si '8080' is a magic number.

Nu l-am declarat private deoarece il folosesc si in clasa de teste.Am ramas la varianta de public. Iar problema cu “8080” is a magic number am rezolvat-o,nu are nevoie de initializare la declarare deoarece el este oricum setat cand se creeaza o instanta noua de catre constructor.

- Recomandarea “**must match pattern '^[a-z][a-zA-Z0-9]\*\$**” pentru numele variabilelor. Conform conventiei de declararea variabilele trebuie sa inceapa cu litera mica si fara caractere gen “ -, \_ etc..”. Un nume corect pentru variabila ar fi serverPort in loc de server\_port. Am corectat aceasta eroare.
- **'serverPort' hides a field. (23:27) [HiddenField]** -> Exista deja o variabila definita serverPort care este disponibila intr-o metoda. Eroarea aceasta este trigger-uita de catre constructor la linia this.serverport = serverPort; (am lasat-o asa ).
- **“Class 'ServerHTTP' looks like designed for extension (can be subclassed), but the method 'modify\_serverPort' does not have javadoc that explains how to do that safely. If class is not designed for extension consider making the class 'ServerHTTP'..”** -> clasa mea, ServerHTTP nu mai este extinsa de catre nici o alta clasa,asa ca am urmat sugestia si am facut-o final.

- 'if' construct must use '{}'. (38:9) [NeedBraces] -> rezolvat, aveam un if cu o singura instructiune si nu am mai pus paranteze. La fel si pe ramura de else. Treaba asta am avut-o in multe locuri prin cod. Am adaugat peste tot.
- Multe erori primite erau din cauza codului scris urat. Cu optiunea de format code din IntelliJ (ctrl+alt+L) am formatat codul si acum arata frumos, conform standardelor de programare in Java.
- Mai am erori care spun ca liniile de cod au prea multe caractere dar nu vad a fii util in cazul meu sa modific. M-am orientat sa nu depaseasca acea linie verticala, astfel incat codul sa fie usor de citit.
- Mai am erori care se refera la javadoc comment, dar nu am stat acum sa comentez fiecare metoda pentru a scapa de ele.

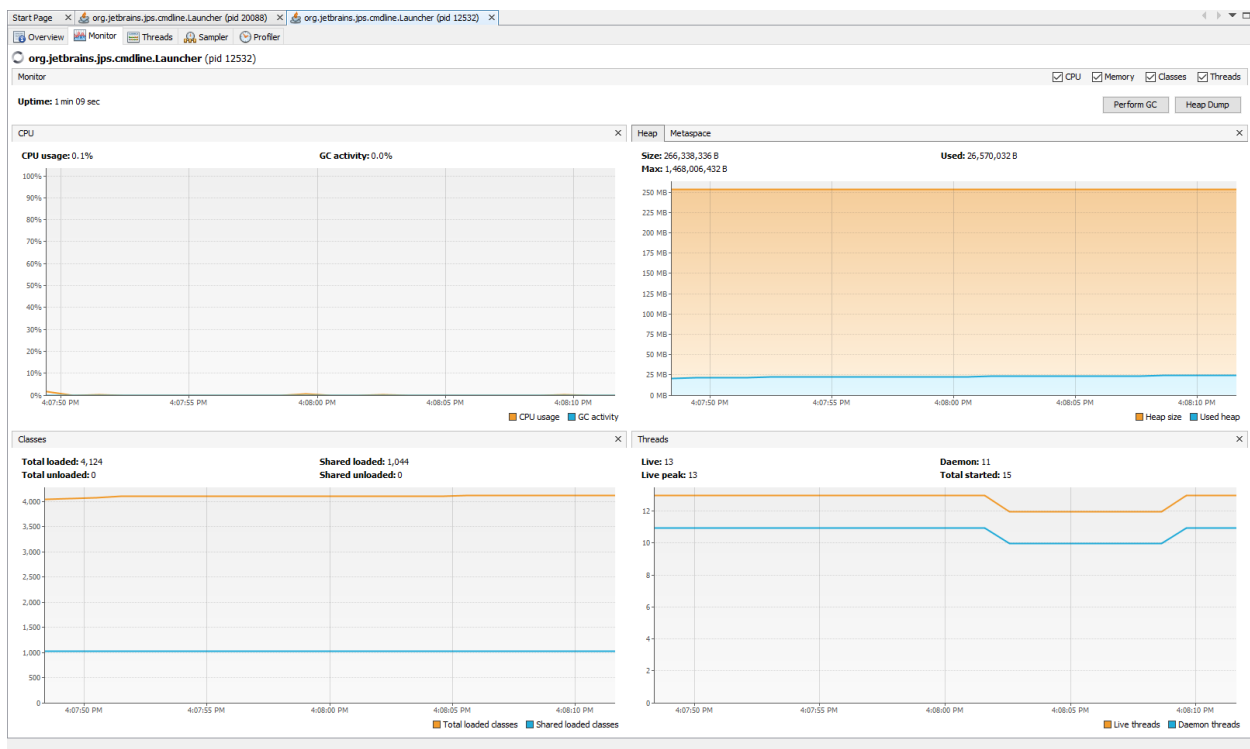
Per total, am ajuns de la un numar de 123 de erori primite in momentul cand am rulat analiza statica pentru prima data la 59 de erori. In aproximativ o ora de am rezolvat 50% din erori si pot sa zic ca analiza statica isi are rolul ei bine definit. **I would rate it 10/10.**

## Analiza dinamica

Pentru partea de analiza dinamica am ales sa folosesc **VisualVM**.

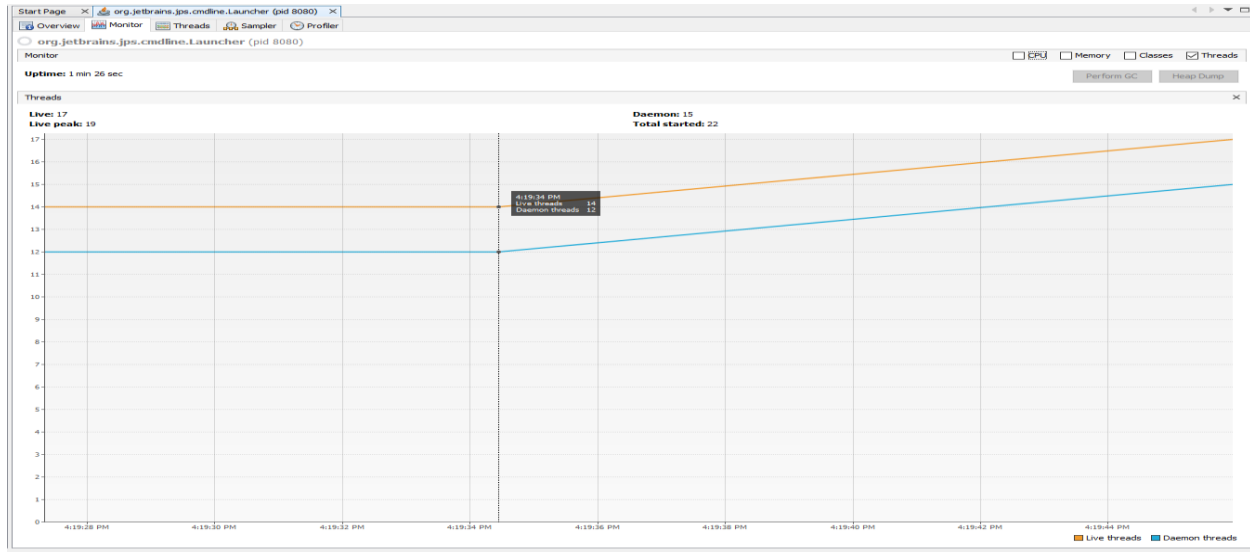
VisualVM este un tool folosit pentru vizualizarea informatiilor detaliate despre aplicatii scrise in Java in timp ce ele ruleaza.

Panoul de vizualizare inainte de rularea codului:

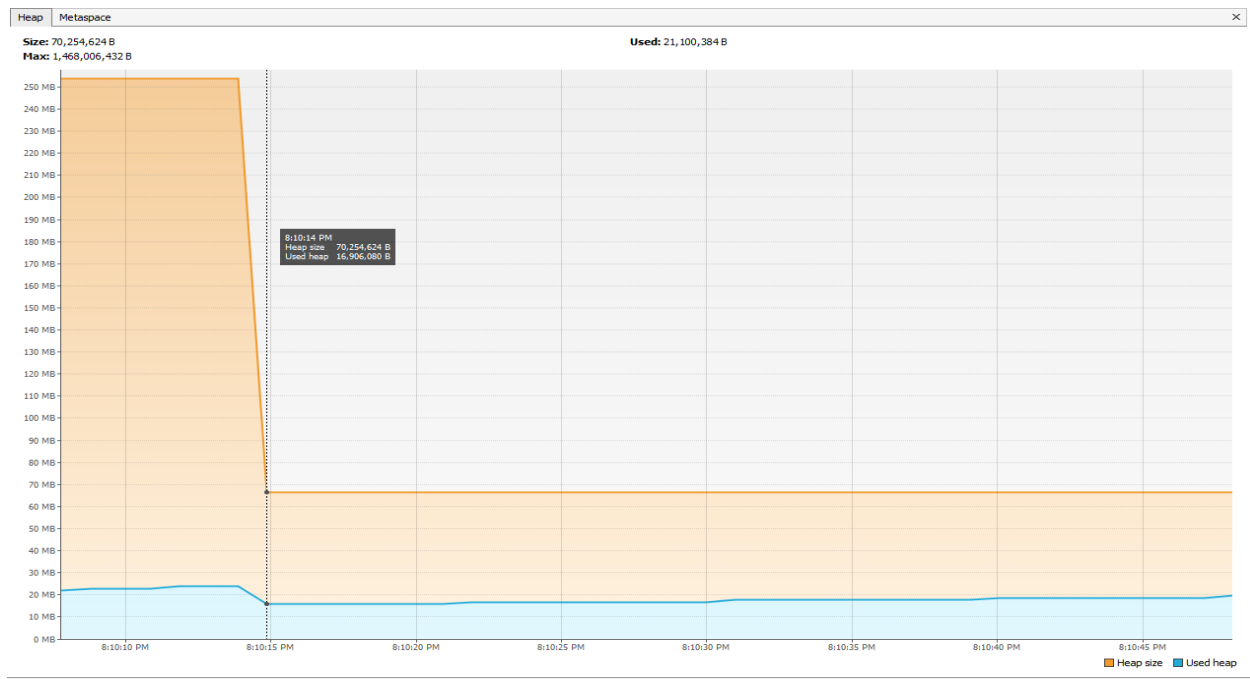


Odata cu rularea codului,putem vedea in urma analizei urmatoarele:

## 1) cresterea numarului de thread-uri



## 2) Dupa activarea Garbage Collector-ului se curata memoria alocata nefolosita





## 3) Sampler

Sampler





































☐ SettingsSample:  CPU  Memory  Stop

Status: memory sampling in progress

Heap histogram

Results:    Collected data:  Snapshot Perform GC

Heap Dump

Name	Live Bytes	Live Objects
 boolean[]	1,664 B (0%)	13 (0%)
 byte[]	19,635,024 B (35.5%)	104,061 (12.2%)
 byte[][]	1,856 B (0%)	10 (0%)
 char[]	2,847,080 B (5.1%)	9,902 (1.2%)
 com.google.protobuf.CodedOutputStream\$ArrayEncoder	32 B (0%)	1 (0%)
 com.google.protobuf.GeneratedMessageLite\$MethodToInvoke	240 B (0%)	10 (0%)
 com.google.protobuf.GeneratedMessageLite\$MethodToInvoke[]	56 B (0%)	1 (0%)
 com.google.protobuf.UnknownFieldSetLite	32 B (0%)	1 (0%)
 com.google.protobuf.UnsafeUtils1	16 B (0%)	1 (0%)
 com.google.protobuf.UnsafeUtilsJvmMemoryAccessor	16 B (0%)	1 (0%)
 com.intellij.diagnostic.LoadingState	192 B (0%)	8 (0%)
 com.intellij.diagnostic.LoadingState[]	48 B (0%)	1 (0%)
 com.intellij.openapi.diagnostic.Log4jBasedLogger	1,024 B (0%)	64 (0%)
 com.intellij.openapi.diagnostic.Logger\$Lambda\$19/0x0000000800bc0b78	16 B (0%)	1 (0%)
 com.intellij.openapi.fileTypes.impl.FileTypeAssocTable	32 B (0%)	1 (0%)
 com.intellij.openapi.fileTypes.impl.IgnoredPatternSet	24 B (0%)	1 (0%)
 com.intellij.openapi.fileTypes.WildcardFileNameMatcher	24 B (0%)	1 (0%)
 com.intellij.openapi.fileTypes.WildcardFileNameMatcher\$SuffixMatcher	16 B (0%)	1 (0%)
 com.intellij.openapi.util.EmptyRunnable	16 B (0%)	1 (0%)
 com.intellij.openapi.util.io.FileAttributes	40 B (0%)	1 (0%)
 com.intellij.openapi.util.io.FileAttributes\$Type	72 B (0%)	3 (0%)
 com.intellij.openapi.util.io.FileAttributes\$Type[]	32 B (0%)	1 (0%)
 com.intellij.openapi.util.io.FileFilters\$1	64 B (0%)	4 (0%)
 com.intellij.openapi.util.io.FileSystemUtils\$IdeaWin32MediatorImpl	16 B (0%)	1 (0%)
 com.intellij.openapi.util.io.FileUtils\$1	16 B (0%)	1 (0%)
 com.intellij.openapi.util.io.FileUtils\$2	16 B (0%)	1 (0%)
 com.intellij.openapi.util.io.FileUtils\$3	16 B (0%)	1 (0%)
 com.intellij.openapi.util.io.NullAppendable	16 B (0%)	1 (0%)
 com.intellij.openapi.util.PathEx\$LazyValue	48 B (0%)	2 (0%)
 com.intellij.openapi.util.win32.IdeaWin32	16 B (0%)	1 (0%)
 com.intellij.openapi.util.JDOMUtils\$EmptyTextFilter	16 B (0%)	1 (0%)
 com.intellij.openapi.util.Key	720 B (0%)	30 (0%)
 com.intellij.openapi.util.Key[]	16 B (0%)	1 (0%)
 com.intellij.openapi.util.LowMemoryWatcher	192 B (0%)	8 (0%)
 com.intellij.openapi.util.LowMemoryWatcher\$LowMemoryWatcherType	48 B (0%)	2 (0%)
 com.intellij.openapi.util.LowMemoryWatcher\$LowMemoryWatcherType[]	24 B (0%)	1 (0%)