Υ325 - Αντικειμενοστρεφής Τεχνολογία

2021

Τμ. ΗΜ&ΤΥ - Πανεπιστήμιο Πατρών

Εβδομάδα 10 - 16/12/2021

Διαχείριση μνήμης και εργαλεία

OutOfMemoryError

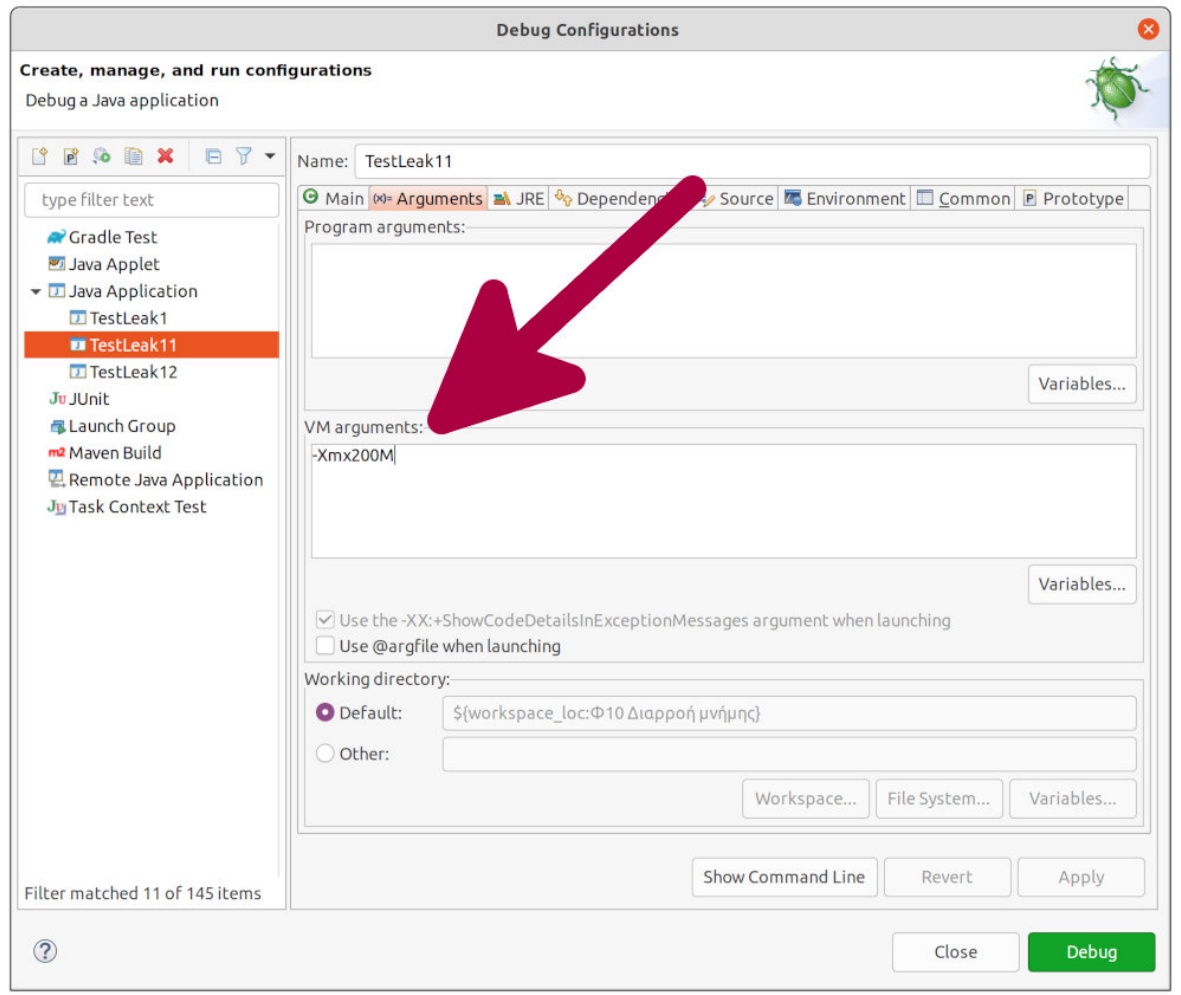
Το σφάλμα OutOfMemoryError συμβαίνει όταν εξαντλείται η διαθέσιμη μνήμη.

- `java.lang.OutOfMemoryError: Java Heap Space`

Μπορεί να σημαίνει πως πραγματικά χρειαζόμαστε περισσότερη μνήμη για να λειτουργήσει η εφαρμογή μας.

- Αν χρειαζόμαστε περισσότερη μνήμη, μπορούμε αλλάξουμε το μέγεθος της heap που διαθέτει η Java VM στην εφαρμογής μας, π.χ. αν χρησιμοποιούμε το HotSpot ή OpenJ9 JVM.

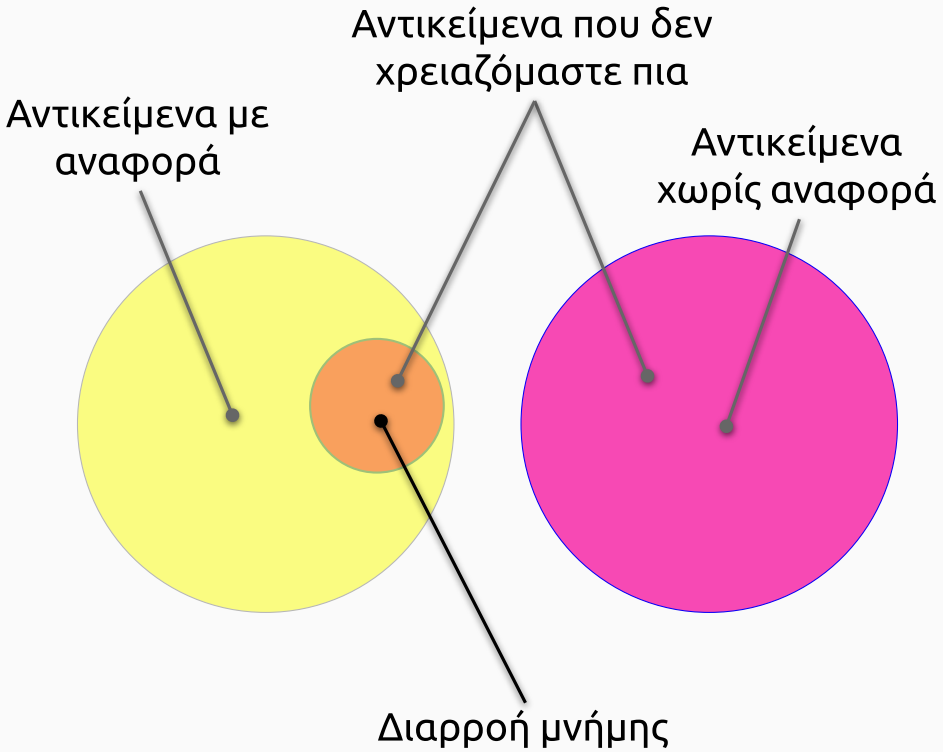
Αλλαγή του Heap space



Εικόνα 1: Run -> Run ή Debug Configurations ... -> Επιλέγουμε προφίλ στο Java applications -> Arguments -> VM arguments

OutOfMemoryError: Java Heap Space

Μπορεί όμως να κρατάμε αναφορές σε αντικείμενα που δεν χρειαζόμαστε.



Εικόνα 2: Ο garbage collector θα αφαιρέσει από τη μνήμη τα αντικείμενα χωρίς αναφορά (ροζ). Αντικείμενα όμως τα οποία δεν χρειαζόμαστε, αλλά για το οποία διατηρούμε αναφορά (πορτοκαλί), θα μείνουν στη μνήμη.

Διαρροή μνήμης στη Java #2

Συνήθεις τρόποι δημιουργία αυτού του προβλήματος:

Στατική αναφορά σε αντικείμενο, δηλ. χρησιμοποιούμε μια στατική μεταβλητή για να αναφερθούμε σε κάποιο μεγάλο αντικείμενο (π.χ. ένα πίνακα με πολλά περιεχόμενα).

Η στατικές αναφορές έχουν ισχύ για όλη τη διάρκεια ζωής της εφαρμογής. Αν δεν είναι απολύτως απαραίτητες οι στατικές αναφορές, μπορούμε να χρησιμοποιήσουμε αναφορές με πιο περιορισμένη ζωή (τοπικές μεταβλητές ή μεταβλητές στιγμιοτύπου).

Επιπλέον, αν αποθηκεύουμε αναφορές σε συλλογές, είναι πιθανό να τις διατηρούμε παραπάνω από όσο χρειαζόμαστε.

Αναλογιστείτε τί θα γίνει με τα `Laser` που έχουν ξεπεράσει τα όρια της οθόνης στην άσκηση 8.

Διαρροή μνήμης στη Java #3

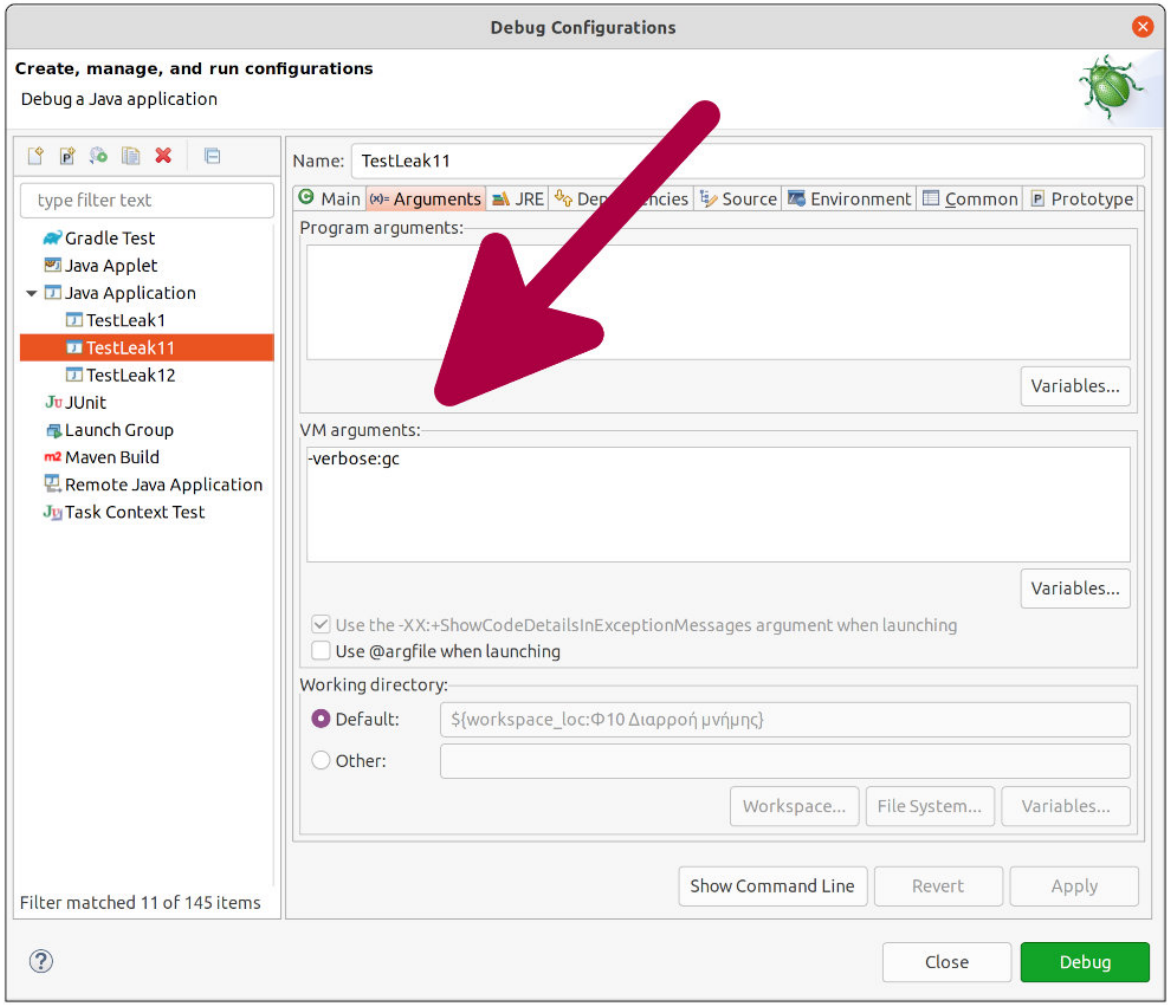
Άλλες πηγές διαρροών είναι

- Ανοιχτά stream, π.χ. αρχεία που ανοίγουμε και έχουμε ξεχάσει να τα κλείσουμε,
- Ανοιχτές συνδέσεις, π.χ. σε μια βάση δεδομένων, έναν εξυπηρετητή
- ...

Για να εντοπίσουμε διαρροές μπορούμε

- να ενεργοποιήσουμε *φλύαρη* garbage collector (Run -> Run ή Debug Configurations ... -> Επιλέγουμε προφίλ στο Java applications -> Arguments -> VM arguments και γράφουμε `-verbose:gc`)

Διαρροή μνήμης στη Java #3



Εικόνα 3: Run -> Run ή Debug Configurations ... -> Επιλέγουμε προφίλ στο Java applications -> Arguments -> VM arguments

Heap Dump

Η ανάλυση γίνεται εξετάζοντας τα περιεχόμενα του heap, ή συγκρίνοντας δύο στιγμιότυπα του heap. Ένα **heap dump** περιέχει πληροφορίες για όλα τα αντικείμενα Java που υπήρχαν στη μνήμη μια χρονική στιγμή.

Για να δημιουργήσουμε heap dump μπορούμε να χρησιμοποιήσουμε το jmap

```
jmap -dump:format=b,file=<heap-dump-file-path> <process-id>
```

ή με κάποιον άλλον τρόπο.

Heap Dump - Πληροφορίες

Τυπικά, σε ένα heap dump υπάρχουν

- **πληροφορίες για όλα τα αντικείμενα**

  κλάση, πεδία, primitives και αναφορές,

- **πληροφορίες για όλες τις κλάσεις**

  classloader, όνομα, υπερκλάση, στατικά πεδία,

- πληροφορίες για τα αντικείμενα που είναι **προσβάσιμα από τον garbage collector**,

- οι **στοίβες** και οι **τοπικές τους μεταβλητές** τη στιγμή που δημιουργήθηκε το αντίγραφο.

Heap Dump - Αναφορές σε αντικείμενα

Τα αντικείμενα στο heap dump έχουν αναφορές σε άλλα αντικείμενα:

- αναφορά στιγμιοτύπου (πεδίο) που δείχνει σε άλλο αντικείμενο,
- πίνακας αντικειμένων, που περιέχει αναφορές σε αντικείμενα,
- με άλλους τρόπους π.χ. ένα αντικείμενο έχει αναφορά στην κλάση του κ.ο.κ.

Τα αντικείμενα και οι αναφορές τους συνιστούν έναν γράφο, όπου τα αντικείμενα είναι οι κόμβοι. Οι ρίζες του γράφου είναι τα **garbage collection roots**.

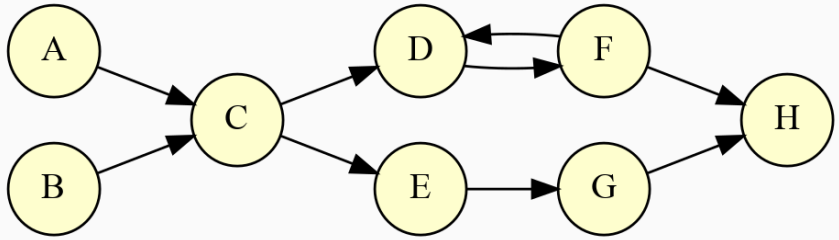
Ένα αντικείμενο στο οποίο μπορούμε να φτάσουμε ακολουθώντας τις ακμές του γράφου, είναι προσβάσιμο (**reachable**). Αν δεν είναι προσβάσιμο (**unreachable**), αυτό σημαίνει πως δεν υπάρχει αναφορά σε αυτό και άρα μπορεί να αφαιρεθεί από τη heap.

Heap Dump - Shallow Vs. Retained heap

Η μνήμη που καταλαμβάνεται από ένα αντικείμενο ονομάζεται **shallow heap size** (χώρος που χρειάζεται το αντικείμενο + την αναφορά σε αυτό).

Η μνήμη που καταλαμβάνεται από όλα τα αντικείμενα στα οποία αναφέρεται ένα αντικείμενο (και από τα αντικείμενα που αναφέρονται αυτά κοκ) ονομάζεται **retained heap size**. Με άλλα λόγια, η μνήμη που θα ελευθερωθεί όταν αφαιρέσουμε το αντικείμενο από τη μνήμη.

Παράδειγμα



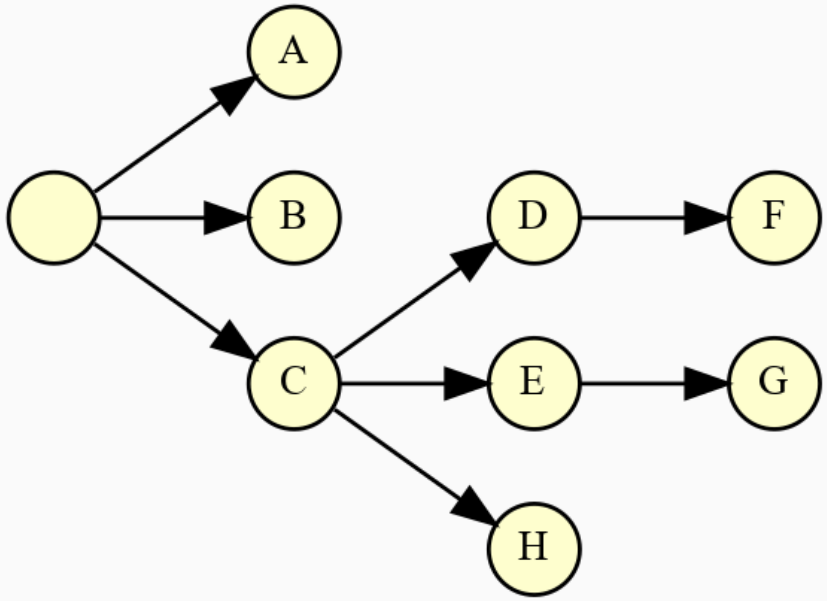
Εικόνα 4: Έστω ο γράφος αναφορών που αποτυπώνει τις αναφορές μεταξύ των αντικειμένων `A` ως `H` στη μνήμη.

Στο παραπάνω στιγμιότυπο:

- Τα αντικείμενα `A`, `B` είναι **garbage collection roots** (τοπικά αντικείμενα, παράμετροι μεθόδου κλπ),
- το retained set του `E` είναι τα `E`, `G`,

  δηλ. αν σβήσουμε την αναφορά από το `C` στο `E` τότε και το `G` θα είναι μη προσβάσιμο,
- το retained set του `C` είναι τα `C`, `D`, `E`, `F`, `G`, `H`.

Dominator tree - Δέντρο επικράτησης



Εικόνα 5: Από το γράφο αναφορών μπορεί να παραχθεί ο γράφος του δέντρου επικράτησης. Οι ακμές εδώ δεν αντιστοιχούν σε αναφορές. Ένα αντικείμενο Χ επικρατεί του Υ όταν όλα τα μονοπάτια προς το Υ περνάνε από το Χ. Το δέντρο επικράτησης δίνει μια γρήγορη εικόνα για τις εξαρτήσεις μεταξύ των αντικειμένων. Για παράδειγμα, φαίνεται ξεκάθαρα το retained set του `C`.

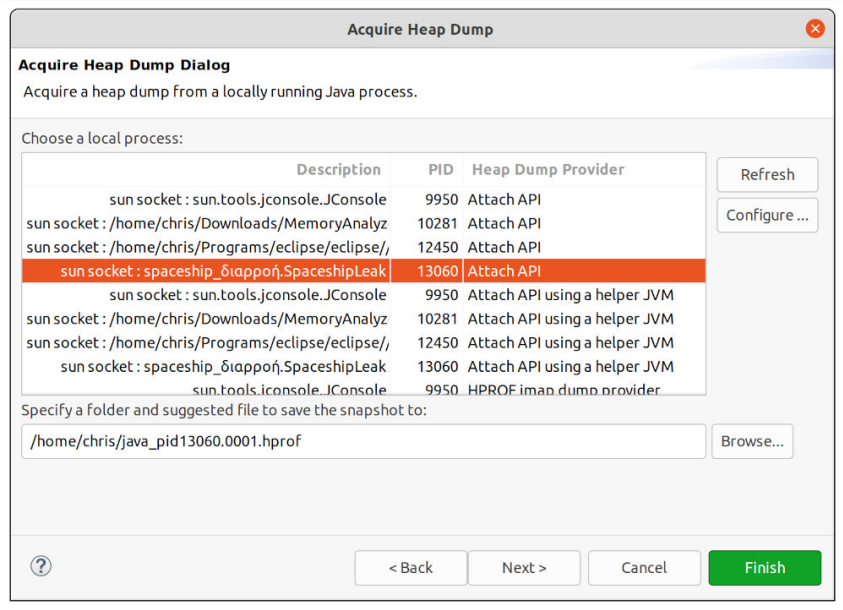
Διάγνωση σφαλμάτων

Για την ανάλυση των περιεχομένων της μνήμης υπάρχει μια μεγάλη επιλογή από εργαλεία, μεταξύ άλλων:

- Java Visual VM
- Eclipse MAT, γρήγορος και ικανός αναλυτής σωρού που βοηθά να εντοπιστούν διαρροές μνήμης και να μειωθεί η κατανάλωση μνήμης.
    - Υπάρχει σε δύο εκδοχές,
        - Plugin για το Eclipse,
        - Αυτόνομο εργαλείο.
- Java Mission Control - από την Oracle (eclipse plugin)
- Heap Hero (online tool),

Eclipse Memory Analyzer (MAT)

Το εργαλείο Eclipse MAT δουλεύει αναλύοντας heap dumps που μπορούμε να αποκτήσουμε διάφορους τρόπους, αλλά πιο εύκολο είναι μέσα από το MAT επιλέγοντας `File ->Acquire Heap Dump ...` και επιλέγουμε στη συνέχεια τη διεργασία (process) για την οποία θέλουμε να αποκτήσουμε αντίγραφο της μνήμης heap:



Εικόνα 6: Το MAT έχει διάφορους τρόπους για να πάρει το αντίγραφο.

Eclipse MAT #2

Στη συνέχεια, αφού αγνοήσουμε τον Getting Started Wizard, παρουσιάζεται μια επισκόπηση των αποτελεσμάτων. Τα αντικείμενα που ήταν σημειωμένα για να αφαιρεθούν από την heap με το επόμενο πέρασμα του garbage collector δεν εμφανίζονται, αλλά μπορούμε να τα δούμε επιλέγοντας “Unreachable Objects Histogram”.

Παραδείγματα

Στο φροντιστήριο είδαμε εποπτικά το Java Visual VM και το Eclipse MAT.

Χρησιμοποιήστε τον κώδικα που συνοδεύει αυτές τις διαφάνειες για να πειραματιστείτε με τα εργαλεία αυτά.