

## **ΠΑΛΙΑ ΘΕΜΑΤΑ ΘΕΩΡΙΑΣ**

### **1. Ποιοι θεωρούνται τύποι αναφοράς (reference types) στη γλώσσα Java;**

Οι τύποι αναφοράς που ορίζονται στην Java ανήκουν σε τρεις κατηγορίες:

- τους τύπους κλάσης (class types),
  - τους τύπους πίνακα (array types), και
  - τους τύπους διασύνδεσης (interface types).
- Η κλάση αποτελεί τον βασικό μηχανισμό παραγωγής νέων τύπων δεδομένων από τον χρήστη.
  - Ο τύπος πίνακα ορίζεται με τη βοήθεια ενός ειδικού μηχανισμού που ορίζεται από τη γλώσσα.
  - Ο τύπος διασύνδεσης αποτελεί έναν μηχανισμό ορισμού αφηρημένων τύπων δεδομένων.

### **2. Πώς ορίζεται η έννοια της γραμμικής δομής δεδομένων (Linear Data Types);**

Πρόκειται για μία δομή δεδομένων, το σύνολο των στοιχείων της οποίας είναι διατεταγμένο με τέτοιο τρόπο ώστε να ισχύουν τα εξής:

- Υπάρχει ένα μόνο στοιχείο, το οποίο ονομάζεται αρχή, και έχει ένα και μόνο ένα επόμενο στοιχείο.
- Υπάρχει ένα μόνο στοιχείο, το οποίο ονομάζεται τέλος, και έχει ένα και μόνο ένα προηγούμενο στοιχείο.
- Κάθε άλλο στοιχείο έχει ένα και μόνο ένα προηγούμενο στοιχείο και ένα και μόνο ένα επόμενο στοιχείο.

### **3. Πώς ορίζεται η έννοια της δυναμικής δομής δεδομένων (Dynamic Data Types);**

Πρόκειται για μία δομή δεδομένων, η οποία μπορεί να δημιουργηθεί και να μετατρέπεται στη διάρκεια της εκτέλεσης του προγράμματος, αυξομειώνοντας το μέγεθός της ή τον χώρο που καταλαμβάνει στη μνήμη.

### **4. Τι είναι γραμμικές και μη γραμμικές δομές δεδομένων; Αναφέρετε τουλάχιστον τέσσερις γραμμικές και δύο μη γραμμικές.**

- Γραμμική δομή δεδομένων: Πρόκειται για μία δομή δεδομένων, το σύνολο των στοιχείων της οποίας είναι διατεταγμένο με τέτοιο τρόπο ώστε να ισχύουν τα εξής:
  - Υπάρχει ένα μόνο στοιχείο, το οποίο ονομάζεται αρχή, και έχει ένα και μόνο ένα επόμενο στοιχείο.
  - Υπάρχει ένα μόνο στοιχείο, το οποίο ονομάζεται τέλος, και έχει ένα και μόνο ένα προηγούμενο στοιχείο.
  - Κάθε άλλο στοιχείο έχει ένα και μόνο ένα προηγούμενο στοιχείο και ένα και μόνο ένα επόμενο στοιχείο.

- Μη γραμμική δομή δεδομένων: Πρόκειται για μία δομή δεδομένων, κάθε στοιχείο της οποίας μπορεί να έχει πολλά επόμενα στοιχεία. Σε μία μη γραμμική δομή δεδομένων υπάρχει μία ιεραρχική σχέση μεταξύ των δεδομένων, καθώς υπάρχουν ρίζες, παιδιά και κόμβοι.

Γραμμικές δομές δεδομένων:

- Πίνακας (array)
- Διάνυσμα (vector)
- Συμβολοσειρά (string)
- Στοίβα (stack)
- Ουρά (queue)
- Συνδεδεμένη λίστα (linked list)

Μη γραμμικές δομές δεδομένων:

- Δένδρα
- Γραφήματα

#### **5. Ποιες είναι, κατά τη γνώμη σας, οι διαφορές ανάμεσα στους τύπους δεδομένων Πίνακας (Array) και Διάνυσμα (Vector).**

- Ένας πίνακας είναι στατικός ως προς το μέγεθός του. Δηλαδή το μέγεθός του δεν μπορεί να αλλάζει κατά τη διάρκεια της εκτέλεσης του προγράμματος.
- Ένα διάνυσμα μπορεί να μεταβάλλει το μήκος του κατά τη διάρκεια της εκτέλεσης του προγράμματος.

#### **6. Ποιος είναι ο ορισμός της στοίβας (stack);**

Στοίβα είναι μία λίστα, στην οποία μπορούν να εισάγονται και να διαγράφονται στοιχεία μόνο από τη μία άκρη της (κορυφή της στοίβας), βάσει της αρχής «Όποιο στοιχείο εισάγεται τελευταίο, διαγράφεται πρώτο.». Δηλαδή, το στοιχείο που θα προστεθεί τελευταίο στη στοίβα θα αφαιρεθεί πρώτο, ή ισοδύναμα, το πρώτο στοιχείο που θα προστεθεί στη στοίβα αναγκαστικά πρέπει να αφαιρεθεί πρώτο. Γι' αυτό και αναφέρεται και σαν λίστα τύπου LIFO (Last – In – First – Out).

#### **7. Ποιος είναι ο ορισμός της ουράς (queue);**

Ουρά είναι μία λίστα, μία συλλογή από στοιχεία που εισάγονται και διαγράφονται βάσει της αρχής «Όποιο στοιχείο εισάγεται πρώτο, διαγράφεται και πρώτο.». Δηλαδή, τα στοιχεία εισάγονται οποιαδήποτε στιγμή, αλλά το στοιχείο που διαγράφεται οποιαδήποτε στιγμή είναι αυτό που έχει μείνει τον περισσότερο χρόνο. Γι' αυτό και αναφέρεται και σαν λίστα τύπου FIFO (First – In – First – Out).

**8. Ποιος είναι ο ορισμός της λίστας (list);**

Λίστα είναι ένα διατεταγμένο σύνολο από μηδέν (0) ή περισσότερα στοιχεία, τα οποία, κατά κανόνα, είναι όλα του ίδιου τύπου. Το μέγεθός της είναι μεταβλητό.

**9. Σε τι διαφέρει μία απλή συνδεδεμένη λίστα από μία κυκλική λίστα;**

Μία κυκλική λίστα έχει το ίδιο είδος κόμβων με μία απλά συνδεδεμένη λίστα. Δηλαδή, κάθε κόμβος έχει έναν δείκτη next και μία αναφορά σε ένα στοιχείο. Αλλά δεν υπάρχει κεφαλή ή ουρά σε μία κυκλική λίστα. Αντί ο τελευταίος κόμβος να έχει δείκτη next null, δείχνει πίσω στον πρώτο κόμβο. Επομένως, σε μία κυκλική λίστα δεν υπάρχει πρώτος ή τελευταίος κόμβος. Έτσι, αν σαρώσουμε τους κόμβους μίας κυκλικά συνδεδεμένης λίστας από οποιονδήποτε κόμβο ακολουθώντας τους δείκτες next, θα περάσουμε κυκλικά τους κόμβους.

**10. Τι εννοούμε όταν λέμε ότι η Java είναι μία ισχυρά τυποποιημένη (strongly typed) γλώσσα προγραμματισμού;**

Σημαίνει ότι κάθε μεταβλητή και κάθε έκφραση έχει κάποιον τύπο, ο οποίος πρέπει να είναι γνωστός κατά το χρόνο της μεταγλώττισης. Ουσιαστικά, η Java απαιτεί από τον προγραμματιστή να κάνει σωστή χρήση των τύπων και δεν επιτρέπει αυθαίρετες μετατροπές.

**11. Τι ονομάζουμε μετάπτωση τύπου (type casting); Ποιο είναι το βασικό πρόβλημα που συνεπάγεται η μετάπτωση τύπου σε κάποιες περιπτώσεις; Δώστε ένα παράδειγμα.**

- **Μετάπτωση** ονομάζεται η μετατροπή του τύπου μίας μεταβλητής σε έναν άλλον.
- Η διαδικασία της μετάπτωσης δεν είναι πάντοτε ασφαλής, καθώς είναι πιθανόν να χαθεί μέρος της πληροφορίας ή να αλλοιωθεί η ακρίβεια μίας τιμής.
- **Παραδείγματος χάριν**, όταν έχουμε μετάπτωση μίας μεταβλητής τύπου double σε μία τιμή τύπου float, ο μεταγλωττιστής στρογγυλοποιεί αναγκαστικά την τιμή.

**12. Στην περίπτωση υλοποίησης της ουράς με τη βοήθεια πίνακα, σε τι διαφέρει η έννοια της υπερχείλισης (overflow) από την εικονική υπερχείλιση;**

- **Υπερχείλιση** συμβαίνει όταν προσπαθούμε να εισάγουμε ένα στοιχείο στον ήδη γεμάτο πίνακα που υλοποιεί την ουρά.
- **Εικονική υπερχείλιση** δημιουργείται όταν ο αριθμός των εισαγωγών είναι μεγαλύτερος από το μέγεθος του πίνακα που υλοποιεί την ουρά και έχει πραγματοποιηθεί ένας αριθμός εξαγωγών.

13. Η έκφραση “a+b” θα αναπαρασταθεί ως “ab+” σε μεταθεματική σημειολογία (postfix notation). Για την αναπαράσταση μίας μεταθεματικής έκφρασης (postfix expression) θα χρησιμοποιήσουμε δυαδικό δέντρο αναζήτησης, στοίβα ή ουρά; Εξηγείστε.

Θα χρησιμοποιήσουμε στοίβα, καθώς αποτελεί μία λίστα στην οποία μπορούν να προστίθενται και να αφαιρούνται στοιχεία μόνο από την μία της άκρη (από την κορυφή της), γεγονός που εξυπηρετεί τον αλγόριθμο που χρειαζόμαστε.

14. Πώς μπορούμε να ελέγχουμε αν ένα δυαδικό δένδρο είναι δυαδικό δένδρο αναζήτησης;

Ελέγχοντας εάν όλες οι τιμές που βρίσκονται στο αριστερό υποδένδρο είναι μικρότερες από την τιμή της ρίζας του δένδρου, και εάν όλες οι τιμές που βρίσκονται στο δεξιό υποδένδρο είναι μεγαλύτερες από την τιμή της ρίζας του δένδρου.

15. Πώς ορίζεται το ύψος (height) ενός κόμβου σε ένα δένδρο; Πώς ορίζεται το ύψος του δένδρου;

- Ορίζουμε σαν **ύψος** (height), ή βάθος (depth), ενός **κόμβου** τον αριθμό του επιπέδου στο οποίο βρίσκεται ο κόμβος αυτός.
- Το **ύψος** (ή βάθος) ενός **δένδρου** ταυτίζεται με το μέγιστο ύψος (ή βάθος) των κόμβων του δένδρου.

16. Δώστε έναν αναδρομικό ορισμό για τη δομή δεδομένων δένδρο (tree).

**Δένδρο** (tree) είναι ένα σύνολο  $T$  από κόμβους (nodes), τέτοιο ώστε είτε:

- Το  $T$  να είναι κενό, ή
- Το  $T$  περιλαμβάνει έναν ξεχωριστό κόμβο,  $R$ , που ονομάζεται ρίζα (root) του  $T$  και οι υπόλοιποι κόμβοι  $T-\{R\}$  χωρίζονται σε μηδέν ή περισσότερα σύνολα κόμβων,  $T_1, T_2, \dots, T_n$ , που είναι ξένα μεταξύ τους και τα οποία είναι με τη σειρά τους δένδρα. Τα  $T_1, T_2, \dots, T_n$  ονομάζονται υποδένδρα του  $T$ .

17. Δώστε ορισμούς για τις παρακάτω δομές δεδομένων: Πλήρες (Complete) δυαδικό δένδρο, Σωρός (Heap).

- Ένα δυαδικό δένδρο βάθους  $N$  ονομάζεται **πλήρες** (complete) όταν έχει όλους τους κόμβους του επιπέδου  $N$  συμπληρωμένους.
- Ένα δυαδικό δένδρο βάθους  $N$  ονομάζεται **σχεδόν πλήρες** (almost complete) όταν έχει όλους τους κόμβους του επιπέδου  $N-1$  συμπληρωμένους και οι κόμβοι που υπάρχουν στο  $N$ -οστό επίπεδο είναι τοποθετημένοι όσο το δυνατόν πιο αριστερά.
- Ένα δυαδικό δένδρο ονομάζεται **σωρός** (heap) όταν ισχύουν οι παρακάτω δύο προϋποθέσεις:
  - Το δυαδικό δένδρο είναι σχεδόν πλήρες, και

- Οι τιμές των κόμβων είναι τοποθετημένες με τέτοιο τρόπο, ώστε ο κάθε κόμβος πατέρας να έχει μεγαλύτερη τιμή από τα παιδιά του.

**18. Δώστε ορισμούς για τις παρακάτω δομές δεδομένων: Δυαδικό δένδρο (binary tree), Δυαδικό δένδρο αναζήτησης (binary search tree), Σωρός (heap).**

- **Δυαδικό δένδρο** (binary tree) είναι ένα δένδρο, κάθε κόμβος του οποίου έχει το πολύ δύο υποδένδρα. Τα υποδένδρα του δυαδικού δένδρου ονομάζονται αριστερό και δεξιό υποδένδρο αντίστοιχα.
- Το **δυαδικό δένδρο αναζήτησης** αποτελεί μία ειδική κατηγορία δυαδικού δένδρου, το οποίο κατασκευάζεται βάσει του παρακάτω αλγορίθμου:
  - Βήμα 1<sup>ο</sup>: Το πρώτο δεδομένο χρησιμοποιείται για τη δημιουργία της ρίζας του δένδρου.
  - Βήμα 2<sup>ο</sup>: Τα επόμενα δεδομένα τοποθετούνται στο δένδρο, έτσι ώστε σε σχέση με οποιονδήποτε κόμβο να ισχύει το εξής: Όλες οι τιμές που βρίσκονται στο αριστερό υποδένδρο είναι μικρότερες από την τιμή του κόμβου και όλες οι τιμές που βρίσκονται στο δεξιό υποδένδρο είναι μεγαλύτερες.
- Ένα δυαδικό δένδρο ονομάζεται **σωρός** (heap) όταν ισχύουν οι παρακάτω δύο προϋποθέσεις:
  - Το δυαδικό δένδρο είναι σχεδόν πλήρες, και
  - Οι τιμές των κόμβων είναι τοποθετημένες με τέτοιο τρόπο, ώστε ο κάθε κόμβος πατέρας να έχει μεγαλύτερη τιμή από τα παιδιά του.

**19. Σε ποιες περιπτώσεις και γιατί είναι βολικό να αναπαραστήσουμε τον τύπο δεδομένων δυαδικό δένδρο (binary tree) με τη βοήθεια πίνακα;**

Στην περίπτωση ενός πλήρους δυαδικού δένδρου μπορούμε να χρησιμοποιήσουμε έναν πίνακα, καθώς δεν χρειάζονται δείκτες για τους εκάστοτε κόμβους, καθώς και δε θα υπάρχουν κενές θέσεις σε αυτόν. Έτσι, και εξοικονομούμε μνήμη και η διαδικασία είναι πιο απλή.

**20. Πώς ορίζεται η έννοια του ρεύματος (stream) στην Java; Τι είναι ένα ρεύμα εισόδου (input stream) και τι ένα ρεύμα εξόδου (output stream);**

- **Ρεύμα** (Stream) είναι ένα αντικείμενο, το οποίο αναπαριστά μία σειριακή ροή δεδομένων από μία πηγή προς έναν προορισμό.
- Ένα **ρεύμα εισόδου** (InPut Stream) μπορεί να οριστεί ως μία ακολουθία δεδομένων και χρησιμοποιείται για την ανάγνωση δεδομένων από μία πηγή.
- Ένα **ρεύμα εξόδου** (OutPut Stream) μπορεί να οριστεί ως μία ακολουθία δεδομένων και χρησιμοποιείται για την εγγραφή δεδομένων σε κάποιον προορισμό.

**21. Τι είναι κοινό και τι διαφορετικό στα παρακάτω ρεύματα (streams): InputStream, OutputStream, Reader, Writer.**

- Η **InputStream** είναι μία αφηρημένη κλάση, με τις υποκλάσεις της οποίας υλοποιούνται όλες οι διαδικασίες εισόδου για ακολουθιακά ρεύματα και ακολουθιακά αρχεία, τα οποία αποτελούνται από bytes, όπως θεωρεί η Java.
- Η **OutputStream** είναι μία αφηρημένη κλάση, με τις υποκλάσεις της οποίας υλοποιούνται όλες οι διαδικασίες εξόδου για ακολουθιακά ρεύματα και ακολουθιακά αρχεία, τα οποία αποτελούνται από bytes, όπως θεωρεί η Java.
- Η κλάση **Reader** υλοποιεί τις διαδικασίες εισόδου από ρεύματα χαρακτήρων.
- Η κλάση **Writer** υλοποιεί τις διαδικασίες εξόδου από ρεύματα χαρακτήρων.

**22. Ποιες είναι οι προϋποθέσεις για επιτυχή σειριοποίηση (serialization) ενός αντικειμένου;**

- Θα πρέπει το αντικείμενο αυτό να υλοποιεί την διασύνδεση `java.io.Serializable`.
- Η κλάση του «σειριοποιημένου» αντικειμένου να κωδικοποιείται μαζί με το όνομα και την υπογραφή της κλάσης, τις τιμές των πεδίων και των πινάκων του αντικειμένου, καθώς και την συμπερίληψη άλλων αντικειμένων που αναφέρονται από το αρχικό αντικείμενο.
- Η ανάγνωση των αντικειμένων να γίνει με την ίδια σειρά με την οποία έγινε η εγγραφή τους σε αρχείο.

**23. Ένα αρχείο έχει δημιουργηθεί ως αρχείο κατ' ευθείαν πρόσβασης (random access file). Ποιες είναι οι προϋποθέσεις (τι πρέπει να γνωρίζουμε) για να χρησιμοποιήσουμε το αρχείο αυτό ως ακολουθιακό αρχείο (sequential file);**

Πρέπει να ισχύουν κατ' ελάχιστο τα εξής:

- Να είναι γνωστός ο τρόπος αρίθμησης των στοιχείων του αρχείου (Για αρχείο με N στοιχεία: σχετική θέση πρώτου στοιχείου 0 (μηδέν) και τελευταίου N-1).
- Να υπάρχει η δυνατότητα τοποθέτησης του δείκτη του αρχείου κατευθείαν σε οποιοδήποτε στοιχείο.
- Να υπάρχει η δυνατότητα να μάθουμε σε ποιο στοιχείο του αρχείου είναι τοποθετημένος ο δείκτης του αρχείου.

**24. Σε τι διαφέρει ένα αρχείο τυχαίας πρόσβασης (random access file) από ένα αρχείο ακολουθιακής πρόσβασης (sequential file);**

- Όταν ανοίξουμε ένα **αρχείο τυχαίας πρόσβασης** (random access file), ή αλλιώς αρχείο κατευθείαν πρόσβασης (direct access file), μπορούμε να το χρησιμοποιήσουμε ταυτόχρονα σαν αρχείου εισόδου και εξόδου.
- Ένα **ακολουθιακό αρχείο** (sequential file) κατά τη διάρκεια της επεξεργασία του ανοίγει κατ' αποκλειστικότητα είτε σαν αρχείο εισόδου είτε σαν αρχείο εξόδου.

**25. Ποιος είναι ο ρόλος ενός ρεύματος φίλτρου;**

Ένα **ρεύμα φίλτρο** φιλτράρει δεδομένα όπως αυτά διαβάζονται ή γράφονται στο ρεύμα.

**26. Ποιες είναι οι διαφορές ανάμεσα στον τύπο String και στον StringBuffer;**

Ο τύπος StringBuffer είναι μία δυναμικά μεταβαλλόμενη ακολουθία χαρακτήρων. Είναι όπως ένα String, αλλά μπορεί να μεταβληθεί, τόσο το μήκος της όσο και το περιεχόμενό της, μέσω κλήσης συγκεκριμένων μεθόδων.

**27. Ορίστε τις έννοιες αρχείο (file) και ρεύμα (stream). Τι είναι ένα ρεύμα από bytes (byte stream) στη Java και σε τι διαφέρει από ένα ρεύμα χαρακτήρων (character stream);**

- Ένα **αρχείο** (file) είναι μία σύνθετη δομή δεδομένων, που αποτελείται από μία σειρά στοιχείων (elements), τα οποία είναι (συνήθως) του ίδιου τύπου και είναι αποθηκευμένα σε έναν χώρο αποθήκευσης (π.χ. σκληρό δίσκο, usb κ.λπ.).
- **Ρεύμα** (stream) είναι ένα αντικείμενο, το οποίο αναπαριστά μία σειριακή ροή δεδομένων από μία πηγή προς έναν προορισμό.
- Ρεύμα από bytes (byte stream) στην Java είναι μία ακολουθία από bytes (1 byte = 8 bits), χρησιμοποιούμενη για είσοδο και έξοδο δεδομένων ανά byte, δηλαδή ανά 8 bits.
- Ενώ, ένα ρεύμα χαρακτήρων (character stream) χρησιμοποιείται για είσοδο και έξοδο δεδομένων ανά 16 bit Unicode.

**28. Περιγράψτε με συντομία τον ρόλο της κλάσης DataInputStream.**

Η κλάση DataInputStream υλοποιεί τις λειτουργίες που σχετίζονται με την ανάγνωση δεδομένων που αντιστοιχούν σε πρωταρχικούς τύπους της Java (primitive data types). Φροντίζει για την μετατροπή των δεδομένων που βρίσκονται με την μορφή bytes στα δεδομένα που αντιστοιχούν σε βασικούς τύπους.

**29. Τι περιλαμβάνει ένα Πλαίσιο Συλλογών (Collection Framework); Τι είναι ο επαναλήπτης (iterator) στο Πλαίσιο Συλλογών της Java (Java Collection Framework); Βασικές μέθοδοι επαναλήπτη.**

- Ένα **Πλαίσιο Συλλογών** (Collection Framework) περιλαμβάνει 1) *διασυνδέσεις*, που επιτρέπουν τον χειρισμό συλλογών ανεξάρτητα από τις λεπτομέρειές τους, 2) *υλοποιήσεις*, δηλαδή επαναχρησιμοποιήσιμες δομές δεδομένων, και 3) *αλγορίθμους*, δηλαδή διάφορες χρήσιμες λειτουργίες.
- Ο **επαναλήπτης** (iterator) είναι αντικείμενο, το οποίο μάς επιτρέπει να «κινούμαστε» μέσα σε μία συλλογή αντικειμένων και να επιλέγουμε το κάθε αντικείμενο με τη σειρά του, χωρίς να γνωρίζουμε ή να μάς ενδιαφέρει η δομή της συγκεκριμένης συλλογής.
- **Βασικές μέθοδοι επαναλήπτη:**  
public interface Iterator<E> {

**boolean hasNext();** //Επιστρέφει true εάν υπάρχει άλλο στοιχείο.

**E next();** //Επιστρέφει το επόμενο στοιχείο.

**void remove();** //Αφαιρεί το στοιχείο που επιστρέφει η next.

}