

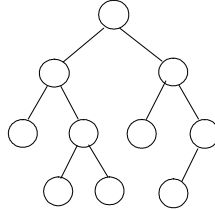
Ενδεικτικά Θέματα Παλαιότερων Εξετάσεων (Θεωρητικό Μέρος)

ΘΕΜΑ

- (α) Ποιοι θεωρούνται **τύποι αναφοράς (reference types)** στη γλώσσα Java;
- (β) Πώς ορίζουμε το **ύψος (height)** ενός κόμβου σε ένα δένδρο; Πώς ορίζεται το ύψος του δέντρου.
- (γ) Πώς ορίζεται η έννοια της **δυναμικής δομής δεδομένων (Dynamic data type)**;
- (δ) Πώς ορίζεται η έννοια της **γραμμικής δομής δεδομένων (Linear data type)**;
- (ε) Περιγράψτε τις έννοιες **αρχείο (file)** και **ρεύμα (stream)**.
- (ζ) Ποιος είναι ο ρόλος ενός **ρεύματος φίλτρου**; Θεωρήστε ότι δίνεται ένα αρχείο από **bytes**. Δώστε τις απαραίτητες δηλώσεις ώστε ένα πρόγραμμα να μπορεί να διαβάσει από το αρχείο αυτό πραγματικούς αριθμούς διπλής ακρίβειας.
- (στ) Δώστε ορισμούς για τις παρακάτω δομές δεδομένων:
 - **Πλήρες (Complete) Δυαδικό Δέντρο**
 - **Σωρός (Heap)**

ΘΕΜΑ

- (α)** Δίνεται το παρακάτω δυαδικό δέντρο το οποίο περιέχει δέκα (10) κόμβους με τη συγκεκριμένη δομή:



Με ποια σειρά πρέπει να βρίσκονται οι παρακάτω αριθμοί κατά την είσοδό τους, ώστε το δέντρο που θα προκύψει να είναι **δυαδικό δέντρο αναζήτησης (binary search tree)**:

10 20 30 40 50 60 70 80 90 99

Δικαιολογήστε την απάντησή σας και δώστε το δέντρο με τα δεδομένα.

- (β)** Υλοποιήστε μία μέθοδο η οποία να υπολογίζει τον αριθμό των τερματικών κόμβων ενός δυαδικού δέντρου.

THEMA

- (α) Δώστε ορισμούς για τις παρακάτω δομές δεδομένων:
- Δυαδικό δέντρο (*binary tree*)
 - Δυαδικό δέντρο αναζήτησης (*binary search tree*)
 - Σωρός (*heap*)

- (β) Δίνονται οι παρακάτω αριθμοί:
35 15 25 185 205 5 65

Σχηματίστε με αυτούς ένα δυαδικό δέντρο αναζήτησης και έναν σωρό. Δώστε αναλυτικά τα βήματα δημιουργίας της κάθε δομής με τη βοήθεια σχημάτων. Οι αριθμοί πρέπει να εισάγονται με τη σειρά που εμφανίζονται παραπάνω.

- (γ) Στο σωρό που έχει δημιουργηθεί εφαρμόστε τον αλγόριθμο επίσκεψης κόμβων προθεματικής διέλευσης (post order traversal) και γράψτε το αποτέλεσμα.

ΘΕΜΑ

Δίνεται το interface **LinkedList** που ορίζει τις βασικές λειτουργίες μιας συνδεδεμένης λίστας.

```
public interface LinkedList {  
    public boolean isEmpty( );  
    public int size( );  
    public void insertFirst(Object data); // εισαγωγή στην αρχή  
    public void insertLast(Object data); // εισαγωγή στο τέλος αρχή  
    public Object removeFirst( ) throws ListEmptyException;  
        // διαγράφει και επιστρέφει το πρώτο στοιχείο  
    public Object removeLast( ) throws ListEmptyException;  
        // διαγράφει και επιστρέφει το τελευταίο στοιχείο }  
}
```

Θεωρήστε ότι είναι διαθέσιμη μία κλάση υλοποίησης, έστω η **myListB** με τον αντίστοιχο δομητή (constructor) και υλοποιημένες όλες τις παραπάνω μεθόδους. Χρησιμοποιώντας τις μεθόδους αυτές προσθέστε στην κλάση **myListB** μία επιπλέον μέθοδο, την

«public Object removePreLast() throws NoSuchElementException»

η οποία να διαγράφει και επιστρέφει τον προτελευταίο κόμβο της λίστας εφόσον υπάρχει, αλλιώς να ρίχνει τη σχετική εξαίρεση. (Θεωρήστε ότι η εξαίρεση **NoElementException** είναι επίσης υλοποιημένη)

Υπόδειξη: Χρησιμοποιήστε στον κώδικά σας κατάλληλο συνδυασμό των μεθόδων **removeLast** και **insertLast**.