

- **Τυχαιοποιημένο Δέντρο Δυναμικής Αναζήτησης**

Δημιουργούμε την κλάση *TreeNode*, η οποία αναπαριστά το τυχαιοποιημένο δέντρο και περιέχει τις μεθόδους:

- 1) *private TreeNode (int id, String city)* : constructor του δέντρου,
- 2) *protected void unlink()* : χρησιμοποιείται όταν αφαιρούμε έναν κόμβο από το δέντρο,
- 3) *private int getN()* : επιστρέφει τον αριθμό των παιδιών που υπάρχουν κάτω από το συγκεκριμένο κόμβο,
- 4) *private void setBooklist (int isbn , int copies)* : Προσθέτει ένα βιβλίο μέσα στη λίστα *booklist* (εξηγείται παρακάτω). Αν το isbn του βιβλίου υπάρχει ήδη στη *booklist*, τότε απλά ο αριθμός των αντιτύπων του αυξάνεται κατά *copies*,
- 5) *private int searchBooklist (int isbn)* : δέχεται σαν όρισμα το isbn ενός βιβλίου και το αναζητά στη *booklist*. Επιστρέφει τη θέση που υπάρχει το βιβλίο μέσα στη λίστα. Αν το βιβλίο δεν υπάρχει επιστρέφει την τιμή -1,
- 6) *private void printList (int key , String name)* : δέχεται σαν ορίσματα το id μια αποθήκης (*key*) και το όνομα της πόλης που βρίσκεται και τυπώνει όλα τα βιβλία που υπάρχουν στη συγκεκριμένη αποθήκη,
- 7) *public int key()* : επιστρέφει το id κάποιας αποθήκης,
- 8) *public String name()* : επιστρέφει το όνομα της πόλης κάποιας αποθήκης.

Επιπλέον μέθοδοι μέσα στην κλάση ST :

- 1) *public void insert (int id, String city)* : δέχεται σαν ορίσματα τον κωδικό *id* και το όνομα *name* της πόλης μιας αποθήκης και το προσθέτει στο δέντρο με τη βοήθεια των παρακάτω μεθόδων:
  - a. *public TreeNode insert (TreeNode h, int id, String city)* : η μέθοδος αυτή καλείται αναδρομικά και προσθέτει μια

αποθήκη στον κατάλληλο κόμβο. Με τη χρήση αυτής της μεθόδου δημιουργείται το τυχαιοποιημένο δέντρο,

- b. *public TreeNode insertR (TreeNode h, int id, String city)* : η μέθοδος αυτή καλείται αναδρομικά και προσθέτει μια αποθήκη στο δέντρο,
  - c. *public TreeNode insertT (TreeNode h, int id, String city)* : η μέθοδος αυτή καλείται αναδρομικά και προσθέτει μια αποθήκη στη ρίζα του δέντρου,
  - d. *public TreeNode rotateLeft (TreeNode pivot)* : η μέθοδος αυτή περιστρέφει το δέντρο αριστερά με βάση τον κόμβο *pivot*,
  - e. *public TreeNode rotateRight (TreeNode pivot)* : η μέθοδος αυτή περιστρέφει το δέντρο δεξιά με βάση τον κόμβο *pivot*,
- 2) *public boolean remove (int id)* : δέχεται σαν όρισμα τον κωδικό *id* μιας αποθήκης και την αφαιρεί από το δέντρο με τη βοήθεια των παρακάτω μεθόδων:
- a. *private void remove (TreeNode p)* : η μέθοδος αυτή καλείται αναδρομικά και αφαιρεί μια αποθήκη από το δέντρο,
  - b. *private TreeNode succ (TreeNode q)* : η μέθοδος αυτή βοηθάει στην ενοποίηση των υποδέντρων με το υπόλοιπο δέντρο μετά την αφαίρεση του κόμβου *q*.
- 3) *public boolean isEmpty()* : επιστρέφει την τιμή *true*, όταν το δέντρο είναι άδειο, αλλιώς επιστρέφει την τιμή *false*,
- 4) *public int compare (int a, int b)* : συγκρίνει τους αριθμούς *a* και *b*. Αν  $a > b$  επιστρέφει 1, αν  $a == b$  επιστρέφει 0, αν  $a < b$  επιστρέφει -1,
- 5) *private TreeNode find (int id)* : δέχεται σαν όρισμα το *id* μιας αποθήκης και ψάχνει στο δέντρο αν υπάρχει η αποθήκη αυτή. Αν υπάρχει επιστρέφει τον κόμβο στον οποίο υπάρχει, αλλιώς επιστρέφει *null*,
- 6) *private void inOrder (int flag)* : το όρισμα που δέχεται αυτή η μέθοδος υποδεικνύει αν θα διασχίσουμε όλο το δέντρο για να τυπώσουμε τα περιεχόμενα όλων των κόμβων ή για να κάνουμε αναζήτηση,

- 7) *private void inOrder (TreeNode n, int flag)* : αν το *flag* είναι -1 τότε τυπωθούν τα περιεχόμενα όλων των κόμβων του δέντρου, αλλιώς ψάχνουμε το βιβλίο με κωδικό *flag*.

Μέθοδοι εισαγωγής/εξαγωγής:

- 1) *public void insertWarehouse (int nodeid, String name)* : εισάγει την αποθήκη με κωδικό *nodeid* και όνομα πόλης *name* στο δέντρο. Αν η αποθήκη με το ίδιο *nodeid* υπάρχει ήδη, εμφανίζει κατάλληλο μήνυμα,
- 2) *public void insertBookAtWarehouse (int nodeid, int isbn, int copies)* : εισάγει στη λίστα των βιβλίων της αποθήκης με κωδικό *nodeid* ένα βιβλίο με κωδικό *isbn* μέσω της *setBooklist*. Αν το βιβλίο υπάρχει ήδη μέσα στη λίστα, τότε αυξάνεται ο αριθμός των αντιτύπων του,
- 3) *public void removeWarehouse (int nodeid)* : αφαιρεί την αποθήκη με κωδικό *nodeid* από το δέντρο,
- 4) *public void removeBook (int nodeid, int isbn)* : αφαιρεί από την αποθήκη *nodeid* το βιβλίο με κωδικό *isbn*,

Μέθοδοι αναζήτησης:

- 1) *public void searchByWarehouse (int nodeid)* : ψάχνει την αποθήκη με κωδικό *nodeid*. Αν υπάρχει, τυπώνει όλα τα βιβλία της και των αριθμό των αντιτύπων κάθε βιβλίου,
- 2) *public void searchBookInWarehouse (int nodeid, int isbn)* : ψάχνει αν το βιβλίο με κωδικό *isbn* υπάρχει στην αποθήκη με κωδικό *nodeid*. Αν υπάρχει, τυπώνεται ο κωδικός του βιβλίου με τον αριθμό των αντιτύπων του,
- 3) *public void searchBook (int isbn)* : ψάχνει το βιβλίο με κωδικό *isbn* σε όλες τις αποθήκες . Αν υπάρχει, τυπώνει τις περιοχές στις οποίες υπάρχει το συγκεκριμένο βιβλίο,
- 4) *public void printTree (PrintStream stream)* : τυπώνει όλες τις αποθήκες με όλα τα διαθέσιμα βιβλία και τον αριθμό αντιτύπων του κάθε βιβλίου.