

A.

- `insert()` : Θέτει ως ρίζα το αποτέλεσμα της `rootInsert()`. Η `rootInsert()` δέχεται ένα κόμβο, το αντικείμενο που θέλουμε να προσθέσουμε στο δέντρο μας και τον πατέρα του κόμβου. Αν η ρίζα είναι κενή τότε φτιάχνει νέο κόμβο και την προσθέτει στο δέντρο. Αν το αντικείμενο που θέλουμε να προσθέσουμε υπάρχει ήδη μέσα στο δέντρο τυπώνουμε αντίστοιχο μήνυμα. Τέλος αν το ΑΦΜ του αντικειμένου είναι μικρότερο από το ΑΦΜ του κόμβου τότε καλείται αναδρομικά η `rootInsert()` με ορίσματα το αριστερό παιδί του κόμβου το αντικείμενο και το κόμβο για πατέρα, το αποτέλεσμα το θέτουμε στο αριστερό παιδί του κόμβου μας και για το κόμβο μας θέτουμε το αποτέλεσμα της `rotateRight()` δηλαδή κάνουμε περιστροφή στο δέντρο μας προς την αντίθετη πλευρά που προσθέσαμε το αντικείμενο. Ομοίως κάνουμε την αντίστοιχη διαδικασία αν το ΑΦΜ του αντικειμένου είναι μεγαλύτερο από το ΑΦΜ του κόμβου. Η `rotateRight()` δέχεται ένα κόμβο, κρατάμε τον πατέρα και το αριστερό παιδί του κόμβου σε τοπικές μεταβλητές. Αν ο κόμβος είναι άδειος επιστρέφουμε το παιδί. Αν το αριστερό παιδί του πατέρα είναι ίδιο με τον κόμβο τότε θέτουμε το αριστερό παιδί του ίσο με το παιδί του κόμβου (δηλαδή το αριστερό παιδί του κόμβου) αλλιώς αν ισχύει για το δεξί παιδί η αντίστοιχη διαδικασία. Στη τοπική μεταβλητή 'παιδί' θέτουμε για πατέρα το πατέρα του κόμβου και για πατέρα του κόμβου το 'παιδί' και στο αριστερό παιδί του κόμβου θέτουμε το δεξί παιδί του 'παιδιού', ουσιαστικά τους αλλάζουμε θέση. Αν υπάρχει το δεξί παιδί του 'παιδιού' του θέτουμε για πατέρα τον κόμβο. Θέτουμε για δεξί παιδί του πατέρα τον κόμβο και τέλος επιστρέφουμε το παιδί.
- `updateSavings()`: Καλούμε την μέθοδο `searchByAFM()` και αν είναι null το αποτέλεσμά της τυπώνουμε αντίστοιχο μήνυμα αλλιώς με setter αλλάζουμε τα `savings` του επιθυμητού στοιχείου.
- `searchByAFM()`:Κρατάμε τοπική μεταβλητή(`temp`) για τον τρέχων κόμβο. Όσο το ΑΦΜ δεν είναι ίσο με το ΑΦΜ μας , αν το ΑΦΜ μας είναι μικρότερο του ΑΦΜ του `temp` θέτουμε το `temp` ίσο με το αριστερό παιδί, αλλιώς το θέτουμε ίσο με το δεξί παιδί . Αν το `temp` είναι null (δηλαδή έχουμε φτάσει στο τέλος του δέντρου ) επιστρέφουμε null. Τέλος , επιστρέφουμε το περιεχόμενο του `temp`.
- `remove()`:Κάνουμε `search` με βάση το ΑΦΜ και αν δεν υπάρχει κόμβος με το αντίστοιχο πεδίο τυπώνουμε μήνυμα λάθους και τερματίζει. Στη συνέχεια ελέγχουμε αν ο κόμβος της κορυφής είναι ο ζητούμενος.Αν ναι τότε τον αφαιρούμε.
- `getMeanSavings()`: Επιστρέφει το αποτέλεσμα της διαίρεσης του πλήθους των κόμβων με το άθροισμα του πεδίου `savings` των αντίστοιχων κόμβων. Το πλήθος και το άθροισμα τα υπολογίζουμε με αναδρομική διάσχιση του δέντρου.
- `load()`: Διαβάζουμε σειριακά όλο το αρχείο το οποίο δέχεται σαν όρισμα χωρίζουμε τα δεδομένα της κάθε γραμμής και κάνουμε `insert()` τα δεδομένα μας.