

A)

searchByAFM : Η μέθοδος αυτή υλοποιείται με την χρήση ενός while οπου ανατρέχει όλο το δέντρο σύμφωνα με μια int μεταβλητή η οποία δόθηκε σαν όρισμα , έως ότου βρει το κλειδί η πέσει πάνω σε Null κόμβο οπου η μέθοδος τερματίζεται.

Insert: Η μέθοδος αυτή υλοποιείται επίσης με ένα while το οποίο ε κάθε μια επανάληψη κοιτάει αν το όρισμα που δέχτηκε είναι μεγαλύτερο η μικρότερο από τον κόμβο τον οποίο ελέγχει (στην τάδε επανάληψη) αν είναι μικρότερο προχωράει στο αριστερό παιδί του κόμβου αν είναι μεγαλύτερο προχωράει στο δεξί παιδί του κόμβου μέχρι να βρεθεί η κατάλληλη θέση για το όρισμα.

Insert_at_root: Αυτή η μέθοδος εισάγει τον κόμβο που δέχτηκε σαν όρισμα στη ριζά του δέντρου και στη συνέχεια κάνει τις κατάλληλες τροποποιήσεις ώστε το δέντρο να παραμείνει ΔΔΑ.

remove: Αυτή η μέθοδος είναι λίγο πιο περιπλοκή , αρχικά χρησιμοποιεί έναν βρόγχο while ώστε να βρει τον κόμβο που θέλουμε να διαγράψουμε , στην συνέχεια ελέγχει τις ακόλουθες περιπτώσεις 1^{ov} ο κόμβος που θέλουμε να διαγράψουμε δεν έχει κανένα παιδί οπότε η διαγραφή γίνεται απλά κάνοντας τον βρόγχο Null. Στην δεύτερη περίπτωση ελέγχουμε αν έχει 1 παιδί και αν έχει τότε κάνουμε μερικές τροποποιήσεις και στην συνέχεια ενώνουμε τον πατέρα του κόμβου που διαγράψαμε με το παιδί του. Και τελευταία και πιο περιπλοκή είναι η τρίτη περίπτωση οπου ο κόμβος έχει 2 παιδιά σε αυτήν την περίπτωση ψάχνουμε να βρούμε το κόμβο που έχει την μικρότερη τιμή από όλα τα παιδιά - εγγόνια που βρίσκονται δεξιά του κόμβου που θέλουμε να διαγράψουμε , αφού τον βρούμε κάνουμε μια αντιμετάθεση και ύστερα αφού ο κόμβος που θέλουμε να διαγράψουμε έχει πάρει την θέση του εγγονιού -παιδιού του τότε τον κάνουμε Null;

printTreeByAFM: Αυτή η μέθοδος καλεί την returnedOrderAFM η οποία καλεί τον εαυτό της αναδρομικά μέχρι να συμπληρώσει το αντικείμενο τύπου List σε σειρά αύξουσα(inorder traverse).

Ύστερα αφού γίνει αυτό η **printTreeByAFM** εμφανίζει τα στοιχεία στην κονσόλα

load: Σε αυτή τη μέθοδο, το πρόγραμμα φορτώνει τα Αντικείμενα τύπου Suspect στο δυαδικό δέντρο. Μέσα σε έναν βρόγχο , διαβάζει κάθε γραμμή του αρχείου του οποίου το μονοπάτι θα πάρει σαν όρισμα και για κάθε γραμμή δημιουργεί έναν StringReader. Έπειτα διαβάζει κάθε χαρακτήρα και ανάλογα που βρίσκει κενά δημιουργεί τις μεταβλητές που είναι απαραίτητες για την δημιουργία και εισαγωγή του Suspect. (Χρησιμοποιείται ένας μετρητής i τύπου int για να μετρήσει τους χαρακτήρες που έχει διαβάσει και να αποτρέψει τον StringReader απο το να συνεχίσει να διαβάζει μόλις φτάσει στο τέλος της γραμμής.)

searchByLastName: Αυτή η μέθοδος επιστρέφει μια λίστα με τα αντικείμενα τύπου Suspect με το ζητούμενο επίθετο. Για να δημιουργήσει την αρχικά κενή λίστα, καλεί μια άλλη αναδρομική μέθοδο που κάνει προδιατεταγμένη διάσχιση στο δένδρο και όπου βρεί αντικείμενο με το ζητούμενο Last name το προσθέτει στη λίστα.

getMeanSavings: Σε αυτή τη μέθοδο, για να υπολογιστεί ο μέσος όρος, χρησιμοποιείται μια μέθοδος που κάνει προδιατεταγμένη διάσχιση του δέντρου για να βρεί τα συνολικά savings.

Έπειτα αυτά διαιρούνται με τον συνολικό αριθμό των suspects (που δίνεται από το N του root αυξημένο κατά 1).

printTopSuspects: Αυτή η μέθοδος εμφανίζει ταξινομημένα K πιο ύποπτους. Αρχικά δημιουργείται ένας πίνακας με μέγεθος ίσο με το πλήθος όλων των υπόπτων. Έπειτα καλείται μία αναδρομική μέθοδος που κάνει προδιατεταγμένη διάσχιση του δένδρου και προσθέτει όλους τους ύποπτους στον πίνακα. Έπειτα ταξινομεί τον πίνακα με την μέθοδο **quicksort** σύμφωνα με την διαφορά savings – income που δίνεται από την μέθοδο getDifference στην κλάση Suspect. Μόλις ο πίνακας ταξινομηθεί, εμφανίζεται το ζητούμενο πλήθος από τον πιο ύποπτο στον λιγότερο.

B)Μενού Διαχείρισης:

Στην main, το μενού διαχείρισης υλοποιείται με έναν βρόγχο while. Έχει τις εξής επιλογές:

- 0: τερματίζει το πρόγραμμα βγαίνοντας από τον βρόγχο
- 1: Φορτώνει το ζητούμενο αρχείο με τα στοιχεία των υπόπτων, ζητώντας το μονοπάτι του.
- 2: Διαγράφει τον ύποπτο με το ζητούμενο ΑΦΜ. Αν δεν υπάρχει εμφανίζει καταλλήλο μήνυμα.
- 3: Εμφανίζει τον ύποπτο με το ζητούμενο ΑΦΜ, αν υπάρχει.
- 4: Εμφανίζει την λίστα με τους ύποπτους που έχουν το συγκεκριμένο επίθετο.
- 5: Επιστρέφει τον μέσο όρο των savings.
- 6: Εμφανίζει τους περισσότερους ύποπτους ανάλογα το πλήθος που ζητά ο χρήστης.
- 7: Εμφανίζει όλους τους ύποπτους της λίστας ταξινομημένους ανάλογα με το ΑΦΜ.
- 8: Επιτρέπει στον χρήστη να εισάγει χειροκίνητα νέο ύποπτο. Δίνει ένα ένα τα στοιχεία του υπόπτου και αν δεν υπάρξει πρόβλημα (δηλαδή exception) και το δωθέν ΑΦΜ δεν υπάρχει, τον ρωτά με ποιόν τρόπο θα ήθελε να γίνει η εισαγωγή. Με insert_at_root ή insert.

Μόλις ο χρήστης επιλέξει μία από τις επιλογές 0-8 εισέρχεται σε μία δομή switch για να κάνει τις απαιτούμενες διεργασίες.