ΕΠΕΞΗΓΗΣΗ ΚΩΔΙΚΑ:

Αρχικά δημιουργήσαμε τις κλάσης που θα χρησιμοποιήσουμε στον κώδικα μας την Main, HuffmanNode , TreeMaker, MyComparator, Files.

Στην HuffmanNode δηλώσαμε δύο μεταβλητές την data για τις συχνότητες και την c για τους χαρακτήρες από 0 μέχρι 127 επίσης δηλώσαμε το left και right ως δυο pointers.

Στην MyComparator συγκρίνουμε με βάση τις τιμές δεδομένων των κόμβων.

Στην TreeMaker δημιουργούμε μια function με όνομα create που παίρνει ως παραμέτρους τον πίνακα charfreq για τις συχνότητες των χαρακτήρων ένα charArray για τους χαρακτήρες και ένα n για το μέγεθος των πινάκων. Μετά δημιουργούμε μια ουρά προτεραιότητας και με την βοήθεια της for γεμίζουμε τα Nodes με τους χαρακτήρες και τις συχνότητες τους και τους βάζουμε σε ουρά(queue)

Στη συνέχεια δημιουργούμε ένα root node με τιμή null.

Μετά κάνουμε ένα while loop που παίρνει τα δύο μικρότερα Nodes από την ουρά μέχρι το size της να γίνει 1.Μέσα στο while παίρνουμε την πρώτη τιμή της ουράς με το peek και την καταχωρούμε στην μεταβλητή x και με το poll διαγράφουμε την τιμή από την ουρά μετά παίρνουμε την δεύτερη τιμή τις ουράς και κάνουμε το ίδιο για την μεταβλητή y. Στη συνέχεια δηλώνουμε ένα Node f στο οποίο αναθέτουμε την

πρόσθεση της συχνότητας των 2 Node x και y . Μετά ορίζουμε το πρώτο Node ως το αριστερό παιδί και το δεύτερο ως το δεξί παιδί. Ύστερα αναθέτουμε την τιμή του root ως το f και χρησιμοποιώντας την add προσθέτουμε την Node f στην ουρά προτεραιότητας . Τέλος επιστρέφουμε την τιμή root.

Στην κλάση Files δημιουργήσαμε μια μέθοδο Read File που παίρνει παράμετρο ReadName για το path του αρχείου, ένα πίνακα ακεραίων για τις συχνότητες και ένα πίνακα χαρακτήρων για τους χαρακτήρες ascii. Μετά χρησιμοποιήσαμε την εντολή BufferedReader σε συνδυασμό με το FileReader για να μπορέσουμε να διαβάσουμε τις συχνότητες από το αρχείο.

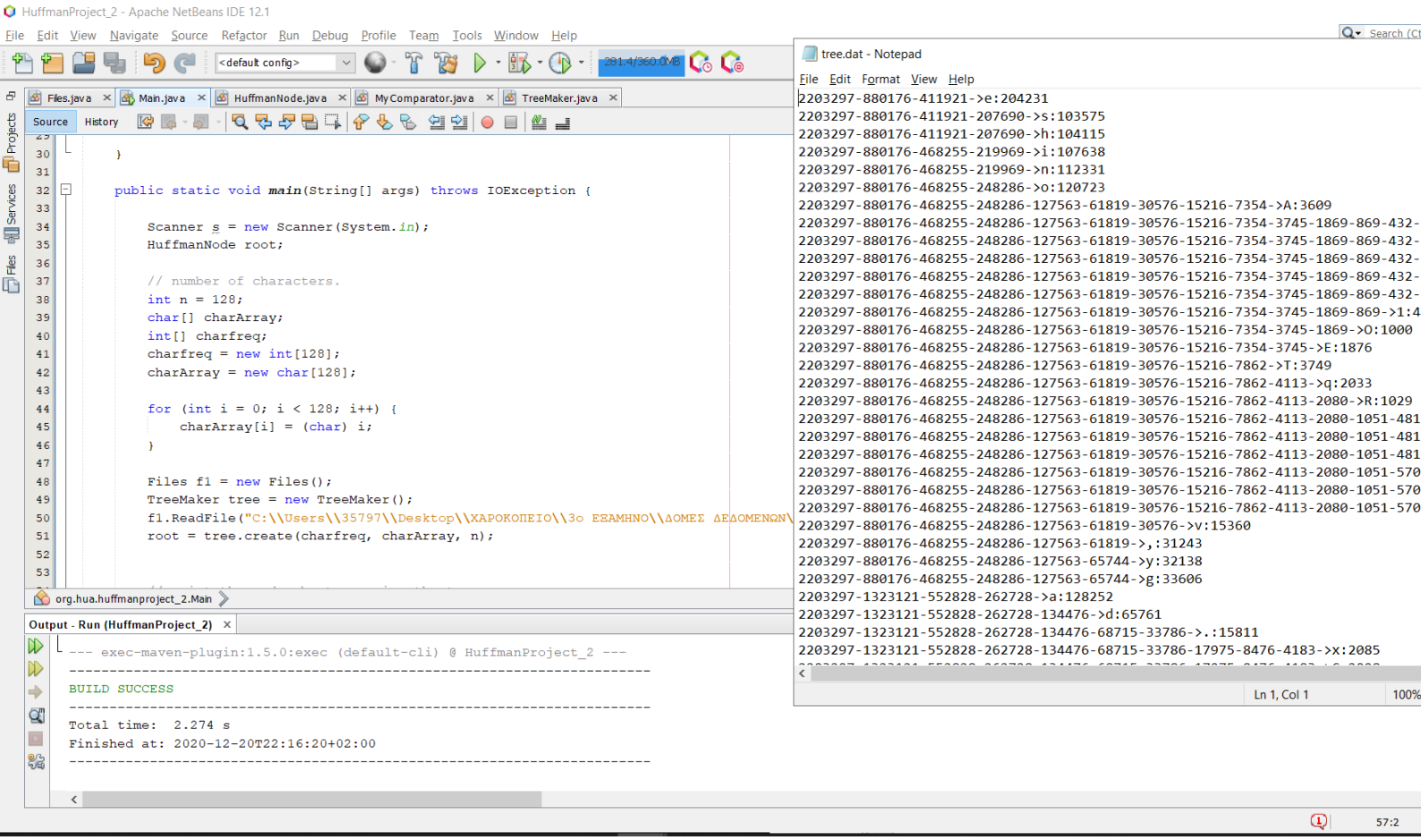
Συνεχίζοντας χρησιμοποιήσαμε ένα while loop .Μέσα στο loop διαβάζουμε τις συχνότητες από το αρχείο με την βοήθεια του indexOf και substring(indexOf: βρίσκει την θέση μετά από ένα συγκεκριμένο string που έχουμε δήλωση, substring: παίρνει ότι υπάρχει μετα από την θέση του indexOf) και τις αποθηκεύουμε στον πίνακα asciitable χρησιμοποιώντας το Integer.valueOf(μετατρέπει το string σε integer).Χρησιμοποιήσαμε μια for για την σωστή τοποθέτηση των frequencies.

Μετά δημιουργήσαμε 2 μεθόδους writeToFile η μια παίρνει ως παραμέτρους ένα χαρακτήρα ένα ακέραιο και μια συμβολοσειρά και η άλλη παίρνει ως παραμέτρους ένα ακέραιο και μια συμβολοσειρά για να για να γράψουμε το δέντρο Huffman.

Στην κλάση Main δημιουργήσαμε μια function printcode που παίρνει τους παραμέτρους ένα node και μια συμβολοσειρά δηλώσαμε ένα constructor για να μπορούμε να χρησιμοποιήσουμε την κλάση files και κάναμε ένα if για να ελέγξουμε εάν στο αριστερό και δεξιό παιδί η τιμή είναι null, εάν είναι και οι δυο null τυπώνουμε στο αρχείο tree.dat το μονοπάτι του κάθε φύλλου και τερματίζει η printcode, εάν δεν είναι null τότε αυτοκαλειται με άλλες παραμετρους,και συνεχίζει μέχρι να φτάσει στην ρίζα του δέντρου.

Στην Main δηλώσαμε ένα πίνακα charArray του οποίου αρχικοποιήσαμε τις θέσεις του με χαρακτήρες ascii από 0 έως και 127.Μετα δηλώσαμε ένα πίνακα charfreq με 128 θέσεις στον οποίο αποθηκεύσαμε τις συχνότητες των χαρακτήρων με την βοήθεια της κλάσης files. Δηλώσαμε το root στο οποίο αποθηκεύσουμε την τιμή που επιστρέφει η create και τέλος καλούμε την printcode για να τυπώσουμε το δέντρο huffman.

ΠΑΡΑΔΕΙΓΜΑΤΑ ΕΚΤΕΛΕΣΗΣ:



Created from: Μυριάνθη Αποστολίδη , Χάρη Παπαμιχαήλ, Φώτη Γαλανό