

ΑΝΑΦΟΡΑ ΕΡΓΑΣΙΑΣ

Τμήμα: Πληροφορικής και Τηλεματικής του Χαροκοπείου
Πανεπιστημίου

Μάθημα: Δομές Δεδομένων

Εξάμηνο: Τρίτο

Υπεύθυνος Καθηγητής: κ. Μιχαήλ

Τίτλος: Huffman Project

Φοιτητές: Ντελής Αθανάσιος, it: 21963

Ντυμένος Ραφαήλ, it: 21965

Πράττη Σταυρούλα, it: 219151

ΠΩΣ ΞΕΚΙΝΗΣΑΜΕ

Η συνεργασία μας έγινε μέσω της πλατφόρμας του TeamViewer. Αφού μελετήσαμε καλά τα περιεχόμενα του συνδέσμου

<https://docs.oracle.com/javase/tutorial/essential/io/index.html>,

προχωρήσαμε στη συγγραφή του κώδικά μας. Το πρόγραμμα υλοποιήθηκε στο περιβάλλον του Apache NetBeans IDE 12.0, συγκεκριμένα στην κατηγορία Java with Maven και επιλογή Java Application, στο μενού, ενώ εκτελείται μέσω του JVM.

ΚΕΦΑΛΑΙΟ 1

ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΚΩΔΙΚΑ

Το πρόγραμμά μας αποτελείται από δυο κλάσεις: την FileHandling.java και την App.java.

Η πρώτη έχει ως λειτουργία να δέχεται και να διαβάζει ένα link. Στην συνέχεια, με την βοήθεια ενός counter τοποθετεί κάθε χαρακτήρα ascii

(από 0 έως 127) στις αντίστοιχες θέσεις ενός πίνακα συχνοτήτων. Έπειτα, κλείνει τα stream που υπάρχουν κι αφού ολοκληρωθεί η δημιουργία του πίνακα συχνοτήτων, τα δεδομένα αποθηκεύονται σε ένα αρχείο με όνομα frequencies.dat.

Μέσα στην App κλάση βρίσκεται η main, η οποία ,με τη σειρά της, περιέχει τον πίνακα συχνοτήτων. Εκεί δημιουργούμε τρία objects της FileHandling και χρησιμοποιούμε την μέθοδο setGutenbergUrl του κάθε object, για να περάσουμε στην κλάση FileHandling τα τρία ζητούμενα link. Στη συνέχεια, με τη βοήθεια της μεθόδου CountLettersFrequency, περνάμε τον πίνακα συχνοτήτων, ώστε να πάρουμε τα αποτελέσματα που επιθυμούμε.

~

Αποφασίσαμε να δημιουργήσουμε την κλάση FileHandling, καθώς παρατηρήσαμε πως θα χρειαστεί να χρησιμοποιήσουμε τον ίδιο κώδικα τρεις φορές για τα τρία links. Έτσι καλώντας αυτήν την μέθοδο αποφύγαμε την επανάληψη κώδικα, εκμεταλλευόμενοι την κληρονομικότητα της java.

ΠΑΡΑΔΕΙΓΜΑΤΑ ΕΚΤΕΛΕΣΗΣ ΚΩΔΙΚΑ

Δείγμα από Frequency.dat:

```
Open  ▾  [?]  frequencies.dat  Save  ≡  _  □  X
~/prog/J/data struct/project/huffman

1 Letter with ASCII 0 appered: 0 times
2 Letter with ASCII 1 appered: 0 times
3 Letter with ASCII 2 appered: 0 times
4 Letter with ASCII 3 appered: 0 times
5 Letter with ASCII 4 appered: 0 times
6 Letter with ASCII 5 appered: 0 times
7 Letter with ASCII 6 appered: 0 times
8 Letter with ASCII 7 appered: 0 times
9 Letter with ASCII 8 appered: 0 times
10 Letter with ASCII 9 appered: 0 times
11 Letter with ASCII 10 appered: 40701 times
12 Letter with ASCII 11 appered: 0 times
13 Letter with ASCII 12 appered: 0 times
14 Letter with ASCII 13 appered: 40701 times
15 Letter with ASCII 14 appered: 0 times
16 Letter with ASCII 15 appered: 0 times
17 Letter with ASCII 16 appered: 0 times
18 Letter with ASCII 17 appered: 0 times
19 Letter with ASCII 18 appered: 0 times
20 Letter with ASCII 19 appered: 0 times
21 Letter with ASCII 20 appered: 0 times
22 Letter with ASCII 21 appered: 0 times
23 Letter with ASCII 22 appered: 0 times
24 Letter with ASCII 23 appered: 0 times
25 Letter with ASCII 24 appered: 0 times
26 Letter with ASCII 25 appered: 0 times
27 Letter with ASCII 26 appered: 0 times
28 Letter with ASCII 27 appered: 0 times
29 Letter with ASCII 28 appered: 0 times
30 Letter with ASCII 29 appered: 0 times
31 Letter with ASCII 30 appered: 0 times
32 Letter with ASCII 31 appered: 0 times
33 Letter with ASCII 32 appered: 409147 times
34 Letter with ASCII 33 appered: 2719 times
35 Letter with ASCII 34 appered: 22 times
36 Letter with ASCII 35 appered: 4 times
37 Letter with ASCII 36 appered: 8 times
```

ΠΗΓΕΣ

<https://docs.oracle.com/javase/tutorial/essential/io/index.html>

<https://gist.github.com/DomPizzie/7a5ff55ffa9081f2de27c315f5018afc/>

<https://www.youtube.com/watch?v=pTCROLZLhDM/>

<https://alvinalexander.com/blog/post/java/how-open-read-url-java-url-class-example-code/>

ΚΕΦΑΛΑΙΟ 2

ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΚΩΔΙΚΑ

Σε πρώτο στάδιο θέλαμε να ανοίξουμε το αρχείο που δημιουργήσαμε στο προηγούμενο μέρος. Αυτό πραγματοποιήθηκε μέσω του tutorial I/O του site της Oracle

(<https://docs.oracle.com/javase/tutorial/essential/io/index.html>).

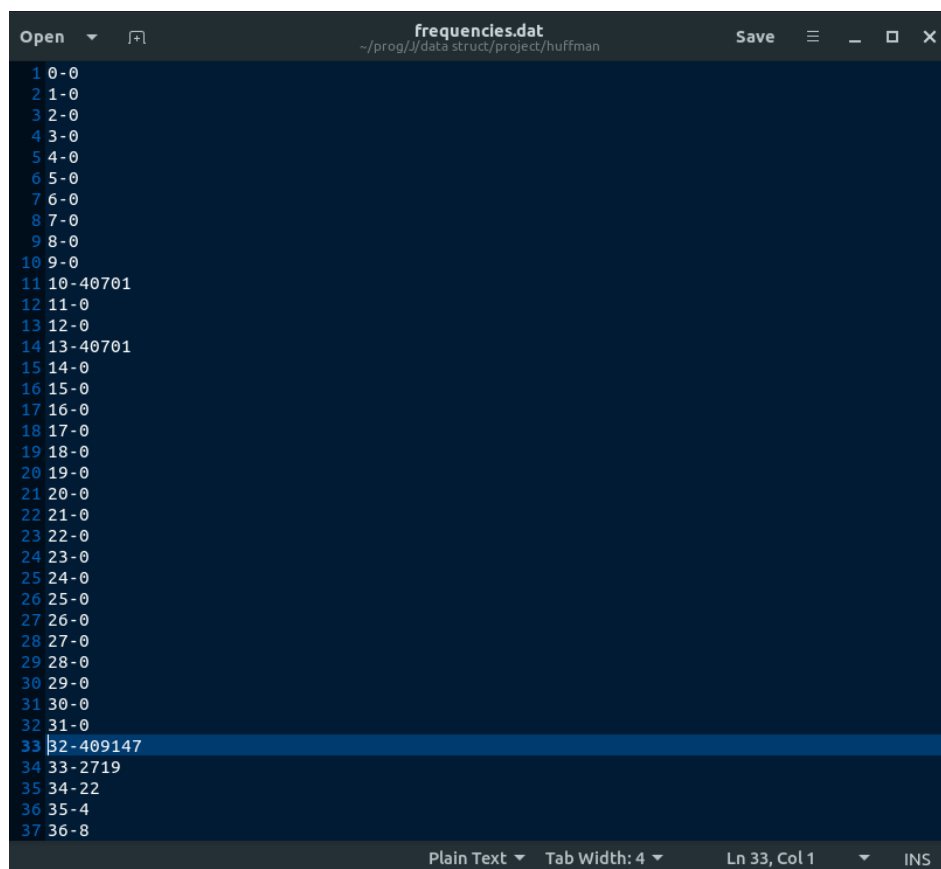
Δημιουργήσαμε τις εξής κλάσεις: HuffmanTree, TreeNodes και πήραμε απ' το εργαστήριο την Interface MinHeap και την κλάση MinHeapArray.

- HuffmanTree: Στην μέθοδό της getFileData ανοίγουμε το αρχείο frequencies.dat με έναν BufferedReader και ανά γραμμή παίρνουμε το πρώτο στοιχείο ως τον χαρακτήρα ascii, το δεύτερο ως τη συχνότητά του και τα αποθηκεύουμε σε ένα νέο object του TreeNodes. Τέλος προσθέτει κάθε node που δημιουργήθηκε σε μια priority queue και κλείνει το stream. Ενώ στη μέθοδό της huffmanCoding παίρνουμε την priority queue που δημιουργήθηκε από την getFileData και φτιάχνουμε το Huffman Tree. Η ιδέα είναι ότι εξάγουμε το node με την μικρότερη συχνότητα από το priority queue, το αποθηκεύουμε προσωρινά και το διαγράφουμε από την ουρά. Στη συνέχεια, εξάγουμε πάλι το πλέον μικρότερο node, το αποθηκεύουμε προσωρινά και το διαγράφουμε από την ουρά. Εφόσον πλέον έχουμε τα δυο node με την μικρότερη συχνότητα τα βάζουμε σε μια μεταβλητή fusedNode ως το αριστερό και το δεξί παιδί (η μικρότερη συχνότητα πάει στο αριστερό και η άλλη στο δεξί), ενώ στο ASCIIletter έχουμε βάλει μια μεταβλητή εκτός ορίων ίση με -1 και στο frequency το άθροισμα των συχνοτήτων των δυο παιδιών. Όλη αυτή η διαδικασία επαναλαμβάνεται μέχρι να μείνει ένα node μέσα στην ουρά που θα είναι το ολοκληρωμένο δέντρο μας. Τέλος, αποθηκεύουμε όλο το δέντρο σε ένα αρχείο tree.dat, με τη βοήθεια της interface Serializable.
- TreeNodes: Περιέχει τις μεταβλητές ASCIIletter, frequency, leftChild και rightChild , setters και getters για αυτές αλλά και μια απλή υλοποίηση της compareTo για την σύγκριση των node.

Η ουρά που χρησιμοποιούμε είναι ο κώδικας της MinHeapArray και το Interface MinHeap, που υλοποιήσαμε στο εργαστήριο 3 του μαθήματος, τον οποίο μελετήσαμε.

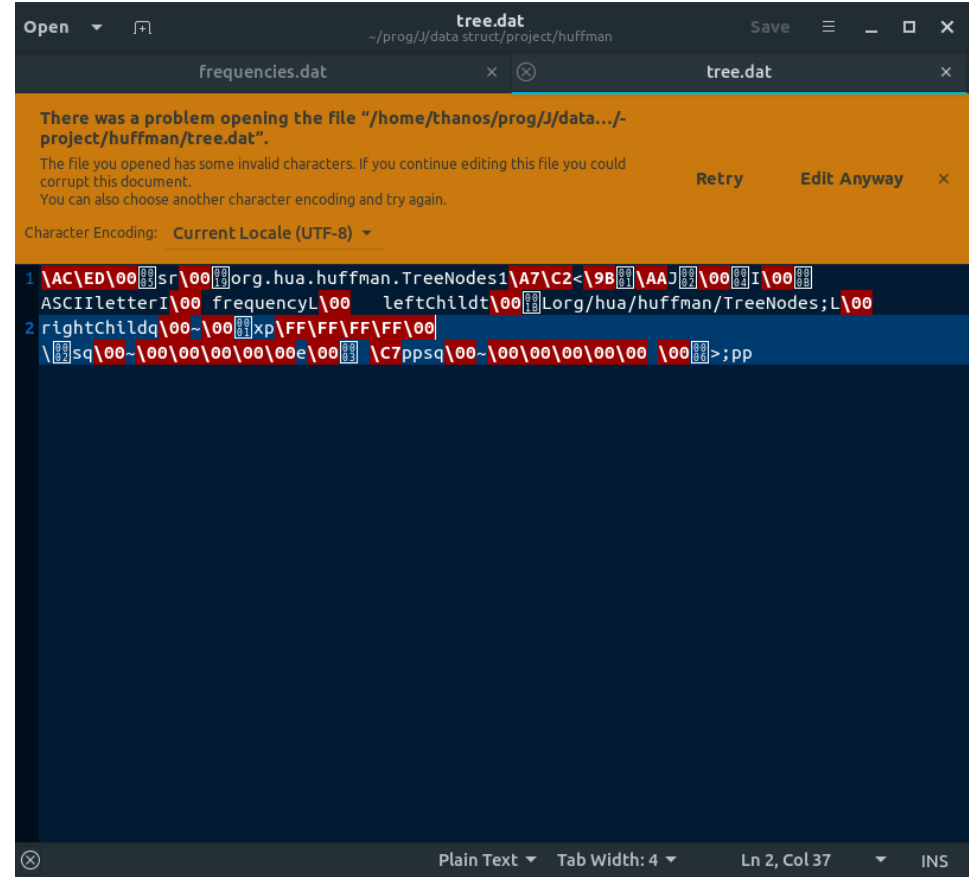
ΠΑΡΑΔΕΙΓΜΑΤΑ ΕΚΤΕΛΕΣΗΣ ΚΩΔΙΚΑ

Δείγμα από Frequency.dat:



```
1 0-0
2 1-0
3 2-0
4 3-0
5 4-0
6 5-0
7 6-0
8 7-0
9 8-0
10 9-0
11 10-40701
12 11-0
13 12-0
14 13-40701
15 14-0
16 15-0
17 16-0
18 17-0
19 18-0
20 19-0
21 20-0
22 21-0
23 22-0
24 23-0
25 24-0
26 25-0
27 26-0
28 27-0
29 28-0
30 29-0
31 30-0
32 31-0
33 32-409147
34 33-2719
35 34-22
36 35-4
37 36-8
```

Δείγμα από tree.dat:



ΠΗΓΕΣ

<https://docs.oracle.com/javase/tutorial/essential/io/index.html>

https://www.tutorialspoint.com/java/io/bufferedReader_readline.htm

<https://www.guru99.com/how-to-split-a-string-in-java.html>

[https://docs.oracle.com/javase/tutorial/essential/
io/objectstreams.html](https://docs.oracle.com/javase/tutorial/essential/io/objectstreams.html)

<https://docs.oracle.com/javase/8/docs/api/java/lang/Comparable.html>