

Άσκηση 2

LAB21142476 Ανουσάκης Αντώνιος

Στη δεύτερη άσκησή γράψαμε ένα πρόγραμμα που διαβάζει αρχεία με αποδείξεις πώλησης προϊόντων, να ελέγχει την ορθότητα των δεδομένων και να εκτυπώνει στατιστικά στοιχεία.

Συγκεκριμένα όταν ξεκινάει το πρόγραμμα εμφανίζει στο χρήστη ένα μενού και περιμένει την επιλογή του χρήστη. Στη συνέχεια ανάλογα με την επιλογή του χρήστη γίνεται ανάγνωση αρχείου, αφού πρώτα ρωτήσει το χρήστη για το όνομα του αρχείου, ή εκτύπωση στατιστικών για τα δεδομένα που ενδιαφέρουν τον χρήστη. Παρακάτω αναφέρονται αναλυτικά τα βήματα που ακολουθεί το πρόγραμμα ανάλογα με την επιλογή του χρήστη.

Επιλογή 1 : Ανάγνωση αρχείου

Η ανάγνωση του αρχείου ξεκινάει με τη συνάρτηση **readInputFile** στη οποία γίνονται τα εξής :

1. Ζητάει από τον χρήστη το όνομα του αρχείου
 2. Ανοίγει το αρχείο για ανάγνωση
 3. Ψάχνει να βρει την αρχή της πρώτης απόδειξης
 - Διαβάζει μια-μια τις γραμμές του αρχείου με τη συνάρτηση **read()** μέχρι να βρει την αρχή της απόδειξης “---”.
 4. Καλεί επαναλαμβανόμενα τη συνάρτηση **readReceipt**
 - Καλεί τη συνάρτηση για να διαβάσει μια απόδειξη
 - Η βρόγχος επανάληψης τερματίζει όταν η ανάγνωση φτάσει στο τέλος του αρχείου, οπού παράγεται ένα EOF exception.
 5. Αποθηκεύει τα δεδομένα
 - Μόλις η **readReceipt** διαβάσει μια σωστή απόδειξη, επιστρέφει το **αφμ**, το **σύνολο** της απόδειξης και τα **προϊόντα** που διάβασε. Τα δεδομένα αυτά, τα αποθηκεύει μόνιμα στα δύο λεξικά.
- ★ Τέλος αν προκύψει κάποιο exception κατά το άνοιγμα του αρχείου η συνάρτηση τερματίζει και εμφανίζεται το αρχικό μενού.

Στη συνέχεια αφού έχει βρει την αρχή της απόδειξης συνεχίζει στην συνάρτηση **readReceipt** όπου σύμφωνα με την εκφώνηση περιμένει να βρει το αφμ, τουλάχιστον ένα προϊόν, το σύνολο και τέλος μια γραμμή μόνο με το χαρακτήρα '-'

1. Διαβάζει το αφμ
 - Διαβάζει μια γραμμή και ελέγχει αν είναι της μορφής ΑΦΜ: AAAAAAAA
 - Αν η γραμμή που διάβασε είναι κενή, έχει φτάσει στο τέλος του αρχείου και παράγει ένα EOF exception για να σταματήσει η ανάγνωση αποδείξεων.
 - Για τον έλεγχο χρησιμοποιήθηκε η κανονική έκφραση '**^ΑΦΜ:\s*(\d{10})\s*\$**'
2. Διαβάζει τα προϊόντα.
 - Διαβάζει μια γραμμή και ελέγχει αν είναι της μορφής : ONOMA_ΠΡΟΪΟΝ:
ΠΟΣΟΤΗΤΑ ΤΙΜΗ_ΜΟΝΑΔΑΣ ΣΥΝΟΛΙΚΗ_ΤΙΜΗ
 - Διαβάζει την επόμενη γραμμή μέχρι να βρει το “Σύνολο”

- Χρησιμοποιήθηκε η κανονική έκφραση
`'(^.*):\s*(\d+)\s+(\d+|\d+\.\d+)\s+(\d+|\d+\.\d+)\s+$'`
 - Ελέγχει τα δεδομένα
 - Αν **ΠΟΣΟΤΗΤΑ** = **ΜΟΝΑΔΑΣ * ΣΥΝΟΛΙΚΗ_ΤΙΜΗ**
 - Αν υπάρχει τουλάχιστον ένα προϊόν
 - Αν όλοι οι αριθμοί είναι θετικοί
3. Αποθηκεύει προσωρινά τα προϊόντα
- Αν το προϊόν είναι σωστό το εισάγει στο λεξικό **productList**, με το όνομα του προϊόντος ως κλειδί και τη συνολική τιμή ως κλειδί.
4. Διαβάζει το σύνολο
- Αν η γραμμή που διάβασε δεν αντιστοιχεί σε προϊόν, ελέγχει αν είναι το σύνολο ώστε να σταματήσει η επαναληψη
 - Ελέγχει αν είναι της μορφής **ΣΥΝΟΛΟ: ΠΟΣΟ_ΜΕ_ΣΥΝΟΛΟ_ΑΠΟΔΕΙΞΗΣ**
 - Χρησιμοποιήθηκε η κανονική έκφραση `'^ΣΥΝΟΛΟ:\s*(\d+|\d+\.\d+)\s*$'`
 - Ελέγχει αν το **ΠΟΣΟ_ΜΕ_ΣΥΝΟΛΟ_ΑΠΟΔΕΙΞΗΣ** είναι σωστό
5. Ελέγχει αν λείπει το **ΣΥΝΟΛΟ**
- Αν η γραμμή που διάβασε δεν αντιστοιχεί σε προϊόν ή στο **ΣΥΝΟΛΟ**, ελέγχει αν αντιστοιχεί σε “---”
 - Χρησιμοποιήθηκε η κανονική έκφραση `'^--+'$'`
- 6.
7. Διαβάζει το τέλος της απόδειξης
- Αν δεν έχει προκύψει κάποιο σφάλμα, διαβάζει την επόμενη γραμμή και ελέγχει αν είναι το τέλος της απόδειξης
 - ★ Αν κάποιο από τα δεδομένα είναι λάθος ή κάποια γραμμή δεν έχει την αναμενόμενη μορφή, τότε παράγεται ένα **PARSEERROR exception**. Στη συνέχεια διαβάζονται και αγνοούνται όλες οι γραμμές μέχρι να βρεθεί η αρχή της επόμενης απόδειξης

Τέλος αφού έχει διαβάσει μια σωστή απόδειξη ακολουθεί η αποθήκευση των δεδομένων με τη συνάρτηση **saveData**. Χρησιμοποιήθηκαν δύο λεξικά για να γίνει πιο γρήγορη η εμφάνιση των στατιστικών. Το πρώτο λεξικό (**productDict**) έχει ως κλειδί το **ΟΝΟΜΑ_ΠΡΟΪΟΝ** και σαν τιμή ένα δεύτερο λεξικό με κλείδι το αφμ και τιμή το σύνολο πωλήσεων για το **ΟΝΟΜΑ_ΠΡΟΪΟΝ**. Στο δεύτερο λεξικό (**afmDict**) έχει ως κλειδί το **αφμ** και σαν τιμή ένα δεύτερο λεξικό με κλειδί το **ΟΝΟΜΑ_ΠΡΟΪΟΝ** και τιμή το σύνολο πωλήσεων για το αφμ.

- Για κάθε προϊόν ελέγχει αν βρίσκεται στα λεξικά
- Αν δεν βρίσκεται, εισάγει τα δεδομένα
- Αν βρίσκεται, αυξάνει την τιμή τους

Επιλογή 2: Στατιστικά προϊόντος

Αν ο χρήστης επιλέξει την εμφάνιση στατιστικών προϊόντων:

1. Ζητάει από τον χρήστη το **ΟΝΟΜΑ_ΠΡΟΪΟΝ**
2. Από το λεξικό **productDict** παίρνει την τιμή για το κλειδί **ΟΝΟΜΑ_ΠΡΟΪΟΝ**
3. Εμφάνιση στατιστικών
 - 3.1. Ταξινόμηση κλειδιών δευτερεύοντος λεξικού
 - 3.2. Για κάθε ένα από αυτά τα κλειδιά εμφανίζει το κλειδί και την τιμή του

Επιλογή 3: Στατιστικά ΑΦΜ

Αν ο χρήστης επιλέξει την εμφάνιση στατιστικών αφμ:

4. Ζητάει από τον χρήστη το **ΑΦΜ**
5. Από το λεξικό afmDict παίρνει την τιμή για το κλειδί **ΑΦΜ**
6. Εμφάνιση στατιστικών
 - 6.1. Ταξινόμηση κλειδιών δευτερεύοντος λεξικού
 - 6.2. Για κάθε ένα από αυτά τα κλειδιά εμφανίζει το κλειδί και την τιμή του

Επιλογή: ΕΞΟΔΟΣ

Αν ο χρήστης επιλέξει την έξοδο, το πρόγραμμα τερματίζει.

Ορθότητα απόδειξης

Η δομή κάθε απόδειξης ελέγχεται χρησιμοποιώντας κανονικές εκφράσεις για να ελέγξουμε αν η γραμμή έχει την αναμενόμενη μορφή σύμφωνα με την εκφώνηση. Εκτός από τη δομή ελέγχονται και τα αριθμητικά δεδομένα κάθε απόδειξης για σφάλματα. Στους ελέγχους αυτούς επιτρέπεται ένα μικρό σφάλμα στις πράξεις λόγο αριθμών κινητής υποδιαστολής. Τέλος κάθε γραμμή του αρχείου και οι είσοδοι του χρήστη, στην επιλογή 2 και 3, μετατρέπονται σε κεφάλαια για να γίνει πιο εύκολη η σύγκριση των αλφαριθμητικών δεδομένων.

Επιλογή δομών δεδομένων

Η δομή δεδομένων που χρησιμοποιήθηκε είναι εμφωλευμένο λεξικό ώστε η εμφάνιση των δεδομένων να είναι όσο το δυνατών πιο γρήγορη. Επιπλέον η δομή αυτή έχει το πλεονέκτημα ότι δεν επαναλαμβάνουμε τα δεδομένα αν υπάρχουν ήδη στο λεξικό, καθώς επίσης και η αναζήτηση είναι πιο εύκολη.

Ανάγνωση μεγάλων αρχείων

Η ανάγνωση πολύ μεγάλων αρχείων γίνεται διαβάζοντας τα γραμμή-γραμμή με τη συνάρτηση `readline()`. Με τη συνάρτηση αυτή, η ανάγνωση μεγάλων αρχείων δεν χρειάζεται πολύ μνήμη καθώς δεσμεύεται μνήμη μόνο για την γραμμή διαβάστηκε. Στις χαμηλές απαιτήσεις μνήμης συμβάλει και η παραπάνω δομή δεδομένων με την οποία δεν έχουμε επανάληψη πληροφορίας.