
Αναφορά 2ης άσκησης

ΠΛΗ211-Εργαλεία ανάπτυξης λογισμικού και προγραμματισμός συστημάτων
Διδάσκων:

Ονοματεπώνυμο φοιτητή: Εμμανουήλ Λιουδάκης

Αρ. Μητρώου φοιτητή:

Αριθμός ομάδας εργασίας:

Το πρόγραμμα αποτελείται από 4 συναρτήσεις, τις `main()`, `choice2()`, `choice3()` και `checkFile`(περιγραφητής αρχείου).

Όταν εκκινείται το πρόγραμμα, εμφανίζεται το μενού χρήστη, όπως ζητείται από την εκφώνηση. Εν συνεχεία, ανάλογα με την επιλογή του χρήστη, προβαίνουμε σε μια από τις παρακάτω ενέργειες:

- Αν η επιλογή είναι το 1, ζητείται από το χρήστη το όνομα του αρχείου, ανοίγεται με σκοπό την ανάγνωση από αυτό, και δίδεται ως όρισμα στη μέθοδο `checkFile`. Η λειτουργικότητα της `checkFile` θα περιγραφεί παρακάτω. Έπειτα, αφού η κληθείσα συνάρτηση επιστρέψει τη ροή εκτέλεσης του προγράμματος στη `main`, κλείνεται το αρχείο.
- Αν η επιλογή είναι 2, τότε η `main` καλεί τη συνάρτηση `choice2`.
- Αν η επιλογή είναι 3, τότε η `main` καλεί τη συνάρτηση `choice3`.
- Αν η επιλογή είναι 4, τότε μέσα στη `main` διαγράφεται το προσωρινό αρχείο `"tmp.txt"`, και τερματίζεται το πρόγραμμα.

Όταν δίνεται από το χρήστη ένα αρχείο (επιλογή 1 του μενού), μέσω της συνάρτησης `checkFile` ελέγχεται η ορθότητα των αποδείξεων, και μόνον οι σωστές γράφονται στο συνολικό αρχείο `"tmp.txt"`. Αν ο χρήστης στη συνέχεια δώσει και άλλο αρχείο, τότε οι ορθές αποδείξεις του θα εγγραφούν στο τέλος του `"tmp.txt"` (`append`).

Ως απόδειξη νοείται το τμήμα του αρχείου που βρίσκεται ανάμεσα σε δυο γραμμές αποτελούμενες μόνο από τον χαρακτήρα `"-"`. Ελέγχεται αν υπάρχει οποιοσδήποτε χαρακτήρας εκτός του `"-"` στις διαχωριστικές γραμμές.

Έπειτα, σε μια λίστα αποθηκεύονται οι αριθμοί τως γραμμών με τις παύλες (παραδείγματος χάριν αν οι γραμμές 1 και 15 αποτελούν διαχωριστικές

γραμμές, η λίστα αυτή θα περιέχει δυο στοιχεία, το 1 και το 15). Με μια επαναληπτική δομή, διατρέχεται η λίστα αυτή και ανάμεσα σε δυο διαδοχικές τιμές διαβάζεται μία απόδειξη.

Οι έλεγχοι που γίνονται σε κάθε απόδειξη είναι οι εξής:

- Αν υπάρχει μόνο μια φορά η γραμμή του ΑΦΜ και αν είναι η πρώτη γραμμή της απόδειξης.
- Αν το ΑΦΜ είναι ορθό (10 ψηφία).
- Αν υπάρχει μόνο μία φορά η γραμμή του συνόλου και αν είναι η τελευταία γραμμή της απόδειξης.
- Αν για κάθε προϊόν υπάρχουν όλα τα πεδία(όνομα, ποσότητα, τιμή μονάδας και συνολική τιμή), και αν ο υπολογισμός της συνολικής τιμής είναι σωστός (με σφάλμα το πολύ 0.0001, ώστε να μην επηρεαστεί η λειτουργία του προγράμματος από σφάλματα που προκύπτουν από τη χρήση αριθμών κινητής υποδιαστολής (float)).
- Αν είναι σωστά υπολογισμένο το συνολικό ποσό της απόδειξης (με σφάλμα το πολύ 0.0001, για τον ίδιο λόγο με προηγούμενως).
- Αφού η κάθε απόδειξη "σπάσει" σε μια λίστα, ελέγχεται αν το μήκος της λίστας αυτής διαιρείται ακριβώς με το 4. Αν δε διαιρείται, η απόδειξη απορρίπτεται (προφανώς υπάρχουν κάποια περιττά πεδία). Στην περίπτωση σωστής απόδειξης, το πλήθος των στοιχείων της λίστας δίνεται από την εξής σχέση:
$$2(\text{ΑΦΜ}, 10\text{ψηφιος αριθμός}) + 4 * \text{πλήθος προϊόντων} + 2(\text{ΣΥΝΟΛΟ}, \text{αριθμός})$$
, το οποίο είναι πολλαπλάσιο του 4.

Για την επιλογή 2 του μενού, η διαδικασία που ακολουθείται είναι η εξής: Ανοίγεται (με σκοπό την ανάγνωση) το αρχείο με τις ορθές αποδείξεις ("tmp.txt"), και δημιουργείται ένα λεξικό (dictionary) της μορφής:

{όνομα_προϊόντος: {ΑΦΜ:συνολικό ποσό πωλήσεων για το ΑΦΜ } }

Ζητείται από το χρήστη το όνομα του προϊόντος. Κατόπιν ταξινομείται ως προς το ΑΦΜ το υπολεξικό που υπάρχει με κλειδί το όνομα του προϊόντος που έδωσε ο χρήστης, και εκτυπώνεται.

Για την επιλογή 3 του μενού, ακολουθείται παρόμοια διαδικασία: Ανοίγεται (με σκοπό την ανάγνωση) το αρχείο με τις ορθές αποδείξεις ("tmp.txt"), και δημιουργείται ένα λεξικό (dictionary) της μορφής:

{ΑΦΜ: {όνομα_προϊόντος:συνολικό ποσό πωλήσεων για το προϊόν } }

Ζητείται από το χρήστη το ΑΦΜ. Κατόπιν ταξινομείται ως προς το όνομα προϊόντος το υπολεξικό που υπάρχει με κλειδί το ΑΦΜ που έδωσε ο χρήστης, και εκτυπώνεται.

Για να μπορούν να απαντηθούν γρήγορα τα ερωτήματα του χρήστη επιλέχθηκε η δομή δεδομένων λεξικού που παρέχει η Python. Η συγκεκριμένη επιλογή έγινε καθώς το λεξικό χρησιμοποιεί συνάρτηση κατακερματισμού (όπως το HashMap στη Java), με αποτέλεσμα να είναι η ιδανική λύση για γρήγορη αναζήτηση δεδομένων.

Ακολουθεί δείγμα ενδεικτικής εκτέλεσης του αρχείου:
Το αρχείο "input_file" έχει το εξής περιεχόμενο:

```
-----  
ΑΦΜ: 5123456789  
ΣΑΛΑΤΑ: 3 7.00 21.00  
ΤΖΑΤΖΙΚΙ: 1 3.50 3.50  
ΤΥΡΙ: 2 1.00 2.00  
ΠΟΙΚΙΛΙΑ: 2 14.225 28.45  
ΚΡΑΣΙ: 1 50.00 50.00  
ΣΥΝΟΛΟ: 104.95  
-----  
ΑΦΜ: 5123456789  
ΝΤΟΜΑΤΕΣ: 1 1.00 1.00  
ΠΑΤΑΤΕΣ: 2 2.20 4.40  
ΠΑΤΑΤΕΣ: 2 2.20 4.40  
ΣΑΛΑΤΑ: 1 7.00 7.00  
ΣΥΝΟΛΟ: 16.80  
-----  
ΑΦΜ: 1722525333  
ΚΡΑΣΙ: 1 50.00 50.00  
ΚΡΑΣΙ: 2 50.00 100.00  
ΣΑΛΑΤΑ: 1 7.00 7.00  
ΣΥΝΟΛΟ: 157.00  
-----
```

Και ένα στιγμιότυπο της εκτέλεσης του προγράμματος:

```
elioudakis@elioudakis-VirtualBox:~$ cd Desktop
elioudakis@elioudakis-VirtualBox:~/Desktop$ python3 computeSales.py
Give your preference: (1: read new input file, 2: print statistics for a specific product, 3: print
statistics for a specific AFM, 4: exit the program)1
Please enter the file's name...test_input
Give your preference: (1: read new input file, 2: print statistics for a specific product, 3: print
statistics for a specific AFM, 4: exit the program)2
Give the product's name... σαλατα
1722525333 7.00
5123456789 28.00
Give your preference: (1: read new input file, 2: print statistics for a specific product, 3: print
statistics for a specific AFM, 4: exit the program)3
Give the afm... 5123456789
ΚΡΑΣΙ 50.00
ΝΤΟΜΑΤΕΣ 1.00
ΠΑΤΑΤΕΣ 8.80
ΠΟΙΚΙΛΙΑ 28.45
ΣΑΛΑΤΑ 28.00
ΤΖΑΤΖΙΚΙ 3.50
ΤΥΠΙ 2.00
Give your preference: (1: read new input file, 2: print statistics for a specific product, 3: print
statistics for a specific AFM, 4: exit the program)4
```