

# Εισαγωγή στον Προγραμματισμό

## Εργαστηριακή Άσκηση

Φροντιστήριο

# Αριστείδης Ηλίας

- Διαχείριση Παραγγελιών (OrdersHandlingApp)

ΤΜΗΥΠ  
CEID

- Ένα εργοστάσιο παρασκευάζει μικρά και μεγάλα μπουκάλια εμφιάλωσης.
- Τα μικρά κοστίζουν 0.008€ και τα μεγάλα 0.02€. Η εταιρεία έχει μια πολιτική χορήγησης έκπτωσης που περιγράφεται παρακάτω:
  - α) Για παραγγελίες μεγαλύτερες από 200€ ή 3000 μπουκάλια, γίνεται έκπτωση 8%.
  - β) Για παραγγελίες μεγαλύτερες από 600€, γίνεται έκπτωση 20%.

## Η Παραγγελία (order)

---

• Θεωρήστε μια παραγγελία να αποτελείται εκτός από τον αριθμό των μικρών και τον αριθμό των μεγάλων μπουκαλιών από τα παρακάτω στοιχεία:

- 1) Ονοματεπώνυμο πελάτη,
- 2) Ημερομηνία που δόθηκε η παραγγελία,
- 3) Ημερομηνία που πρέπει να εκτελεστεί,
- 4) Αρχικό κόστος παραγγελίας (κόστος χωρίς την έκπτωση),
- 5) Έκπτωση που αντιστοιχεί στην παραγγελία (σύμφωνα με την πολιτική εκπτώσεων),
- 6) Τελικό κόστος παραγγελίας (όπου έχει υπολογιστεί και η έκπτωση), και
- 7) Ημερομηνία που τελικά εκτελέστηκε η παραγγελία.

## Ο Αλγόριθμος και κατ' επέκταση το πρόγραμμα

---

Αναπτύξτε ένα πρόγραμμα που θα δίνει τις παρακάτω επιλογές στο χρήστη

- 0 Τερματισμός Προγράμματος
- 1 Καταχώρηση παραγγελίας
- 2 Εμφάνιση παραγγελιών πελάτη
  
- **A. Πρώτη ομάδα (εκκρεμείς παραγγελίες pendingOrders):**
  - 3 Εμφάνιση εκκρεμών παραγγελιών
  - 4 Αποθήκευση εκκρεμών παραγγελιών σε αρχείο
  - 5 Φόρτωση εκκρεμών παραγγελιών από αρχείο.
  - 6 Εκτέλεση παραγγελίας (υπολογισμός αρχικού κόστους παραγγελίας)
  
- **B. Δεύτερη ομάδα (παραγγελίες προς εξόφληση readyOrders).**
  - 7 Εμφάνιση των προς εξόφληση παραγγελιών
  - 8 Εξόφληση παραγγελίας (υπολογισμός τελικού κόστους παραγγελίας)
  
- **Γ . Τρίτη ομάδα (Εκτελεσμένες παραγγελίες closedOrders**
  - 9 Εμφάνιση εκτελεσμένων παραγγελιών
  - 10 Αποθήκευση των εκτελεσμένων παραγγελιών σε ξεχωριστό αρχείο
  - 11 Φόρτωση από αρχείο εκτελεσμένων παραγγελιών

## Η εξέλιξη μιας παραγγελίας

---

- Μια παραγγελία όταν καταχωρείται είναι «εκκρεμής» (pendingOrder). Όταν εκτελεστεί είναι «προς εξόφληση» (readyOrder). Μετά την εξόφληση είναι «κλειστή» (closedOrder).
- Η επιλογή «Εκτέλεση Παραγγελίας» εκτελείται πάνω σε μια παραγγελία που είναι pending και την μετατρέπει σε ready.
- Η επιλογή «Εξόφληση Παραγγελίας» εκτελείται πάνω σε μια παραγγελία που είναι ready και την μετατρέπει σε closed.

## Η εξέλιξη μιας παραγγελίας

---

- Μια παραγγελία όταν καταχωρείται είναι «εκκρεμής» (pendingOrder). Όταν εκτελεστεί είναι «προς εξόφληση» (readyOrder). Μετά την εξόφληση είναι «κλειστή» (closedOrder).
- Η επιλογή «Εκτέλεση Παραγγελίας» εκτελείται πάνω σε μια παραγγελία που είναι pending και την μετατρέπει σε ready.
- Η επιλογή «Εξόφληση Παραγγελίας» εκτελείται πάνω σε μια παραγγελία που είναι ready και την μετατρέπει σε closed.

Σχεδιάζουμε και αναπτύσσουμε **αυξητικά**, αξιοποιώντας την **αφαιρετικότητα** στις διεργασίες, δηλαδή «βηματικά» με βάση την «εκφώνηση» (μελλοντικά σαν μηχανικοί τις απαιτήσεις του προβλήματος)

ΤΜΗΜΑ  
CEID



Αποτελεί την 1η έκδοση του προγράμματος OrdersHandling με το μενού επιλογής ενεργειών.

Ορίστε την δομή της main () με την παράλληλη ανάπτυξη του μενού επιλογής λειτουργίας.

Η έκδοση αυτή υλοποιεί μόνο την επιλογή Τερματισμού προγράμματος.

Υλοποιεί τις λειτουργίες 1 , 2 και 3 του μενού επιλογής. Θα πρέπει να επιλέξετε να δουλέψετε μεταξύ των δυο επιλογών:

1. πίνακα παραγγελιών, ή,
2. πίνακα δεικτών σε παραγγελίες.

Αν δεν είστε εξοικειωμένοι με τους δείκτες και τη δυναμική διαχείριση μνήμης (malloc και free) επιλέξτε τον πίνακα παραγγελιών. Και στις 2 περιπτώσεις υποθέστε ένα σταθερό μέγεθος για τον πίνακα σας.

Για να διευκολύνετε την διαδικασία ανάπτυξης του προγράμματος αρχικοποιήστε τον πίνακα των παραγγελιών σας με ορισμένες παραγγελίες. Θα αποφύγετε με τον τρόπο αυτό να εισάγετε παραγγελίες κάθε φορά που θα ελέγχετε το πρόγραμμα σας.

Η υλοποίηση της λειτουργίας Εμφάνιση ς εκκρεμών παραγγελιών είναι απλή στην φάση αυτή καθώς όλες οι παραγγελίες μας είναι ακόμη εκκρεμείς (pending). Προχωρήστε στην υλοποίηση της. Θα την τροποποιήσετε αργότερα όταν θα έχετε και παραγγελίες άλλης κατάστασης, π.χ. readyOrders

Οδηγίες για την υλοποίηση της έκδοσης ς αυτή ς θα βρείτε στο τέλος της άσκησης. Μελετήστε τις μόνο μετά την δική σας προσπάθεια για τη δημιουργία της έκδοσης αυτής.

Ο έλεγχος της καλής λειτουργίας των λειτουργιών 1 , 2 και 3 σηματοδοτεί και την ολοκλήρωση της 2 ης έκδοσης του προγράμματος σας.

Επιλέγουμε για την έκδοση αυτή τις λειτουργίες 6 , 7 και 8 . Αφήνουμε τις λειτουργίες που έχουν σχέση με τα αρχεία για επόμενη έκδοση.

- **Εκτέλεση παραγγελίας**

Η λειτουργία αυτή περιλαμβάνει τον υπολογισμό του αρχικού κόστους της παραγγελίας, δηλαδή του κόστους χωρίς την έκπτωση. Η λειτουργία εκτελείται πάνω σε μία εκκρεμή ( pending) παραγγελία και την μετατρέπει σε ready. Στην φάση αυτή θα πρέπει να επιλέξετε αν θα διατηρείτε τις ready παραγγελίες στον ίδιο πίνακα με τις pending orders) ή θα τις αποθηκεύετε σε διαφορετικό πίνακα . Στις οδηγίες περιγράφονται και οι δύο επιλογές υλοποίησης.

Η πρώτη επιλογή είναι πιο απλή από την δεύτερη. Και για τις δύο επιλογές υιοθετούμε το μοντέλο εξυπηρέτησης first come first served. Αυτό σημαίνει πως η διαχείριση των παραγγελιών ( Εκτέλεση Παραγγελίας και Εξόφληση Παραγγελίας ) γίνεται σειριακά με την σειρά που αυτές καταχωρήθηκαν.

Αυτό σημαίνει πως η λειτουργία « Εκτέλεση παραγγελίας » επιδρά πάνω στην πιο παλιά εκκρεμή παραγγελία

- **Εξόφληση παραγγελίας**

Η επιλογή «Εξόφληση Παραγγελίας» εκτελείται πάνω σε μια παραγγελία που είναι ready και την μετατρέπει σε closed. Περιλαμβάνει τον υπολογισμό του τελικού κόστους της παραγγελίας με βάση την πολιτική έκπτωσης. Αν έχετε επιλέξει για την υλοποίηση της επιλογής Εκτέλεσης παραγγελίας, ένα ξεχωριστό πίνακα για τις ready παραγγελίες θα ορίσετε τώρα έναν νέο πίνακα για τις closed παραγγελίες. Διαφορετικά και οι closed παραγγελίες θα είναι στον αρχικό πίνακα σας.

# Εισαγωγή στον Προγραμματισμό

Αρχεία

Φροντιστήριο

Αριστείδης Ηλίας

- Μορφές αρχείων
  - αρχεία κειμένου (text files) και
  - δυαδικά αρχεία (binary files)
- Γραμμογράφηση αρχείου κειμένου (π.χ. myfile.txt)

1	1	0000-00-00 00:00:00	Το δικαίωμα στην εργασία	The right to work
3	2	0000-00-00 00:00:00	Το δικαίωμα σε δίκαιες συνθήκες εργασίας	The right to fair working conditions
2	3	0000-00-00 00:00:00	Το δικαίωμα σε συνθήκες ασφάλειας	The right to safety and hygiene at work
3	4	0000-00-00 00:00:00	Το δικαίωμα σε δίκαιη αμοιβή	The right to a fair remuneration
3	5	0000-00-00 00:00:00	Το δικαίωμα στο συνδικαλισμό	The right to trade unionism
3	6	0000-00-00 00:00:00	Το δικαίωμα στις διαπραγματεύσεις	The right to collective bargaining
4	7	0000-00-00 00:00:00	Το δικαίωμα των παιδιών στην προστασία	The right of children and young people to protection
4	8	0000-00-00 00:00:00	Το δικαίωμα των εργαζομένων γυναικών	The right of working women to maternity protection
1	9	0000-00-00 00:00:00	Το δικαίωμα στον προσανατολισμό	The right to vocational guidance
1	10	0000-00-00 00:00:00	Το δικαίωμα στην επαγγελματική κατάρτιση	The right to vocational training
2	11	0000-00-00 00:00:00	Το δικαίωμα στην προστασία της υγείας	The right to health protection
2	12	0000-00-00 00:00:00	Το δικαίωμα στην κοινωνική ασφάλιση	The right to social security
2	13	0000-00-00 00:00:00	Το δικαίωμα στην κοινωνική πρόνοια	The right to social welfare
...				

- Τα αρχεία κειμένου αποτελούνται από μία ή περισσότερες γραμμές, οι οποίες περιέχουν χαρακτήρες σε αναγνώσιμη μορφή, σύμφωνα με κάποια κωδικοποίηση, όπως ο ASCII κώδικας
- Κάθε γραμμή τελειώνει με τον ειδικό χαρακτήρα που χρησιμοποιεί το λειτουργικό σύστημα για να καθορίσει το τέλος της γραμμής
  - Στα Windows για παράδειγμα, το ζευγάρι των χαρακτήρων '\r' (Carriage Return) και '\n' (Line Feed), CR/LF, με ASCII κωδικούς 13 και 10 αντίστοιχα, χαρακτηρίζει το τέλος της γραμμής
  - Άρα, ο χαρακτήρας νέας γραμμής '\n' αντικαθίσταται από αυτό το ζευγάρι, όταν γράφεται στο αρχείο ενώ η αντίστροφη αντικατάσταση συμβαίνει όταν διαβάζουμε αυτό το ζευγάρι χαρακτήρων από το αρχείο
  - Αντίθετα, σε Unix συστήματα, δεν συμβαίνει αυτή η μετατροπή, γιατί ο χαρακτήρας '\n' συμβολίζει το τέλος της γραμμής
  - Στα αρχεία κειμένου το λειτουργικό σύστημα μπορεί να προσθέσει έναν ειδικό χαρακτήρα που να δηλώνει το τέλος του αρχείου κειμένου, Π.χ. στα Windows αυτός ο χαρακτήρας είναι ο Control-Z (CTRL-Z)

## Άνοιγμα Αρχείου...

---

- Για το **άνοιγμα ενός αρχείου** χρησιμοποιείται η συνάρτηση **fopen()**
- Το πρωτότυπο της συνάρτησης δηλώνεται στο αρχείο **stdio.h** και είναι το ακόλουθο:

```
FILE *fopen(const char *filename, const char *mode);
```
- Η παράμετρος **filename** δηλώνει το όνομα του αρχείου
- Η παράμετρος **mode** καθορίζει τις ενέργειες που μπορούν να εκτελεστούν στο αρχείο, σύμφωνα με τον πίνακα της επόμενης διαφάνειας
- Αν η συνάρτηση **fopen()** εκτελεστεί **επιτυχημένα**, τότε **επιστρέφει έναν δείκτη** σε μία δομή τύπου **FILE**
- Το πρότυπο της δομής **FILE** δηλώνεται στο αρχείο **stdio.h** και στα πεδία αυτής της δομής αποθηκεύεται πληροφορία σχετικά με το αρχείο
- Όλες οι επόμενες **λειτουργίες** που θα εκτελεστούν στο αρχείο, π.χ. ανάγνωση ή εγγραφή σε αυτό, θα γίνονται **με χρήση αυτού του δείκτη**

## ...Άνοιγμα Αρχείου...

---

Επιλογή	Ενέργεια
r	Ανοίγει το αρχείο για ανάγνωση.
w	Ανοίγει το αρχείο για εγγραφή. Αν το αρχείο δεν υπάρχει, δημιουργείται. Αν υπάρχει, το περιεχόμενό του διαγράφεται.
a	Ανοίγει το αρχείο για προσάρτηση. Αν το αρχείο δεν υπάρχει, δημιουργείται. Αν υπάρχει, τα υπάρχοντα δεδομένα διατηρούνται και τα νέα δεδομένα προστίθενται στο τέλος.
r+	Ανοίγει το αρχείο για ανάγνωση και εγγραφή.
w+	Ανοίγει το αρχείο για ανάγνωση και εγγραφή. Αν το αρχείο δεν υπάρχει, δημιουργείται. Αν υπάρχει, το περιεχόμενό του διαγράφεται.
a+	Ανοίγει το αρχείο για ανάγνωση και προσάρτηση. Αν το αρχείο δεν υπάρχει, δημιουργείται. Αν υπάρχει, τα υπάρχοντα δεδομένα διατηρούνται και τα νέα δεδομένα προστίθενται στο τέλος.



- Για το άνοιγμα ενός αρχείου κειμένου επιλέγουμε κάποιον από τους παραπάνω τρόπους
- Αν **αποτύχει** η εκτέλεση της συνάρτησης **fopen()**, τότε αυτή επιστρέφει την τιμή **NULL**  
Παράδειγμα αποτυχίας:
  - ◆ Περίπτωση προσπάθειας ανάγνωσης ενός αρχείου (δηλ. η παράμετρος **mode** είναι **"r"**), ενώ η παράμετρος **filename** περιέχει το όνομα ενός αρχείου που δεν υπάρχει
- Όταν **τελειώσουμε** με την επεξεργασία του αρχείου, το **κλείνουμε** με τη συνάρτηση **fclose()**

## ...Άνοιγμα Αρχείου...

---

- Αν το αρχείο βρίσκεται στον ίδιο φάκελο με το εκτελέσιμο πρόγραμμα, αρκεί να γράψουμε μόνο το όνομα του αρχείου σε διπλά εισαγωγικά

Π.χ.

- ◆ `fopen("test.dat", "r");` ανοίγει για ανάγνωση το αρχείο κειμένου `test.dat`, το οποίο βρίσκεται στον ίδιο φάκελο με το εκτελέσιμο πρόγραμμα
  - ◆ `fopen("test.dat", "wb");` ανοίγει για εγγραφή το δυαδικό αρχείο `test.dat` (αν δεν υπάρχει, δημιουργείται στον ίδιο φάκελο με το εκτελέσιμο πρόγραμμα, ενώ, αν υπάρχει, τα περιεχόμενά του διαγράφονται)
- Αν το αρχείο βρίσκεται σε διαφορετικό φάκελο από το εκτελέσιμο πρόγραμμα, τότε πρέπει να καθοριστεί η πλήρης διαδρομή

## ...Άνοιγμα Αρχείου...

Αν το λειτουργικό σύστημα χρησιμοποιεί τον χαρακτήρα της ανάστροφης κεκλιμένης \ για τον διαχωρισμό των φακέλων (π.χ. Windows), τότε, στη διαδρομή του αρχείου, πρέπει να προστεθεί μια ακόμα \, γιατί, θυμηθείτε, η C χειρίζεται τον χαρακτήρα \ σαν την αρχή μίας ακολουθίας διαφυγής

Π.χ.

αν το πρόγραμμα εκτελείται σε Windows και θέλουμε να ανοίξουμε για διάβασμα το αρχείο **test.txt** που υπάρχει στη διαδρομή **d:\dir1\dir2**, πρέπει να γράψουμε:

```
fopen("d:\\dir1\\dir2\\test.txt", "r");
```

Ωστόσο, σε περίπτωση που το όνομα του αρχείου εισάγεται μέσω της γραμμής εντολών, δεν χρειάζεται να προσθέσουμε τη δεύτερη \, δηλαδή θα πρέπει να πληκτρολογήσουμε:

```
d:\dir1\dir2\test.txt (με μία \)
```

## ...Άνοιγμα Αρχείου...

```
fopen("test.txt", "r");
```

Ανοίγει για ανάγνωση το αρχείο κειμένου **test.txt**, το οποίο βρίσκεται στον ίδιο φάκελο με το εκτελέσιμο πρόγραμμα

```
fopen("c:\\src\\test.bin", "a+b");
```

Ανοίγει (σε λειτουργικό σύστημα Windows) για ανάγνωση και προσάρτηση το δυαδικό αρχείο **test.bin**, το οποίο βρίσκεται στον φάκελο **c:\\src**

Αν το αρχείο δεν υπάρχει, δημιουργείται στον φάκελο **c:\\src**, ενώ, αν ήδη υπάρχει, τα υπάρχοντα δεδομένα διατηρούνται και τα νέα δεδομένα προστίθενται στο τέλος του

```
fopen("test.dat", "wb");
```

Ανοίγει για εγγραφή το δυαδικό αρχείο **test.dat**, που βρίσκεται στον ίδιο φάκελο με το εκτελέσιμο πρόγραμμα

Αν το **test.txt** δεν υπάρχει, τότε δημιουργείται στον ίδιο φάκελο με το εκτελέσιμο πρόγραμμα, ενώ αν υπάρχει το περιεχόμενό του διαγράφεται

## ...Άνοιγμα Αρχείου...

---

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    FILE* fp;
    char fname[100];

    printf("Enter file name: ");
    gets(fname);

    fp = fopen(fname, "r"); /* Άνοιγμα αρχείου για ανάγνωση. */
    if(fp == NULL)
    {
        printf("Error: File can not be opened\n");
        exit(1); /* Το αρχείο δεν άνοιξε, οπότε το πρόγραμμα
τερματίζεται. */
    }
    else
    {
        printf("File is opened successfully\n");
        fclose(fp); /* Κλείσιμο αρχείου. */
    }
    return 0;
}
```

Επαναλαμβάνεται, ότι η τιμή επιστροφής της συνάρτησης **fopen()** πρέπει πάντα να ελέγχεται και μόνο αν αυτή είναι διαφορετική από **NULL**, θα πρέπει να επιτρέπεται να γίνουν διάφορες ενέργειες στο αρχείο

- Ο έλεγχος της επιστροφής της **fopen()** θα μπορούσε να γίνει σε μία γραμμή κώδικα ως εξής:

```
if((fp = fopen(fname, "r")) == NULL)
```

Οι εσωτερικές παρενθέσεις μπαίνουν για λόγους προτεραιότητας

## Κλείσιμο Αρχείου...

- Η συνάρτηση **fclose()** χρησιμοποιείται **για το κλείσιμο ενός ανοικτού αρχείου**
- Το πρωτότυπο της συνάρτησης δηλώνεται στο αρχείο **stdio.h** και είναι το ακόλουθο:

```
int fclose(FILE *fp);
```

- Η παράμετρος **fp** είναι ο δείκτης σε μία δομή τύπου **FILE**, που έχει επιστραφεί από μία προηγούμενη κλήση της **fopen()**
- Η συνάρτηση **fclose()** επιστρέφει **0** αν το αρχείο έκλεισε επιτυχώς, αλλιώς επιστρέφει την ειδική τιμή **EOF**

- Η ειδική τιμή **EOF** δηλώνεται στο αρχείο **stdio.h** , έχει τιμή **-1** και χρησιμοποιείται για να μας ενημερώσει ότι:
  - ◆ είτε φτάσαμε στο **τέλος του αρχείου**
  - ◆ είτε **συνέβη κάποιο λάθος** στην εκτέλεση μίας λειτουργίας στο αρχείο

Αν και ένα **ανοικτό αρχείο κλείνει αυτόματα όταν το πρόγραμμα τερματίζει**, προτείνεται **να το κλείνετε** όταν τελειώσετε με την επεξεργασία του

Ένας καλός λόγος για να πειστείτε είναι ότι, ακόμα και αν το πρόγραμμά σας τερματιστεί ανώμαλα (π.χ. crash), το αρχείο θα παραμείνει ανέπαφο και τα δεδομένα του δεν θα χαθούν



**Ευχαριστώ**

**Ερωτήσεις?**