**Εισαγωγή στον Προγραμματισμό**

**Εργασία OrdersHandling**

**Ονοματεπώνυμο: Κωνσταντίνος Λάιος**

**Α.Μ. : 1110124**

Η εργασία αφορά την Δράση Δ1 του [Εργαστηρίου της 1HYPERLINK "https://eclass.upatras.gr/modules/blog/index.php?course=CEID1416&action=showPost&pId=1525"1HYPERLINK "https://eclass.upatras.gr/modules/blog/index.php?course=CEID1416&action=showPost&pId=1525"ης HYPERLINK "https://eclass.upatras.gr/modules/blog/index.php?course=CEID1416&action=showPost&pId=1525"Εβδομάδας](https://eclass.upatras.gr/modules/blog/index.php?course=CEID1416&action=showPost&pId=1525).

* **Εκδόσεις που αναπτύξατε**

Η αριστερή στήλη του πίνακα περιέχει τις λειτουργικότητες της εφαρμογής OrdersHandling που σας ζητήθηκαν να υλοποιήσετε. Η πρώτη γραμμή καταγράφει τέσσερεις εκδόσεις. Για κάθε μία από τις εκδόσεις που υλοποιήσατε κατά τη διαδικασία ανάπτυξης του προγράμματος σημειώστε με 1 τις λειτουργικότητες που η έκδοση υλοποιεί.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Λειτουργικότητα** | **Έκδοση 1η** | **Έκδοση 2η** | **Έκδοση 3η** | **Έκδοση 4η** |
| 0 – Τερματισμός Προγράμματος | 1 | 1 | 1 | 1 |
| 1 - Καταχώρηση παραγγελίας |  | 1 | 1 | 1 |
| 2- Εμφάνιση παραγγελιών πελάτη |  | 1 | 1 | 1 |
| 3 - Εμφάνιση εκκρεμών παραγγελιών |  | 1 | 1 | 1 |
| 4 - Αποθήκευση εκκρεμών παραγγελιών σε αρχείο |  |  |  | 1 |
| 5 - Φόρτωση εκκρεμών παραγγελιών από αρχείο. |  |  |  | 1 |
| 6 - Εκτέλεση παραγγελίας |  |  | 1 | 1 |
| 7 - Εμφάνιση των προς εξόφληση παραγγελιών |  |  | 1 | 1 |
| 8 - Εξόφληση παραγγελίας |  |  | 1 | 1 |
| 9 - Εμφάνιση εκτελεσμένων παραγγελιών |  |  | 1 | 1 |
| 10 - Αποθήκευση των εκτελεσμένων παραγγελιών σε ξεχωριστό αρχείο |  |  |  | 1 |
| 11 - Φόρτωση από αρχείο εκτελεσμένων παραγγελιών |  |  |  | 1 |

* **Λειτουργικότητα που υλοποιεί το πρόγραμμα του παραδοτέου**

Για κάθε λειτουργικότητα της τελευταίας έκδοσης του προγράμματος που αναπτύξατε, βάλτε 1 στην στήλη που περιγράφει την κατάσταση της λειτουργικότητας.

|  |  |  |  |
| --- | --- | --- | --- |
| **Λειτουργικότητα** | **Έχει υλοποιηθεί πλήρως** | **Έχει υλοποιηθεί μερικώς** | **Δεν έχει υλοποιηθεί** |
| 0 – Τερματισμός Προγράμματος | 1 |  |  |
| 1 - Καταχώρηση παραγγελίας | 1 |  |  |
| 2- Εμφάνιση παραγγελιών πελάτη | 1 |  |  |
| 3 - Εμφάνιση εκκρεμών παραγγελιών | 1 |  |  |
| 4 - Αποθήκευση εκκρεμών παραγγελιών σε αρχείο | 1 |  |  |
| 5 - Φόρτωση εκκρεμών παραγγελιών από αρχείο. | 1 |  |  |
| 6 - Εκτέλεση παραγγελίας | 1 |  |  |
| 7 - Εμφάνιση των προς εξόφληση παραγγελιών | 1 |  |  |
| 8 - Εξόφληση παραγγελίας | 1 |  |  |
| 9 - Εμφάνιση εκτελεσμένων παραγγελιών | 1 |  |  |
| 10 - Αποθήκευση των εκτελεσμένων παραγγελιών σε ξεχωριστό αρχείο | 1 |  |  |
| 11 - Φόρτωση από αρχείο εκτελεσμένων παραγγελιών | 1 |  |  |

* **Συναρτήσεις που υλοποιούν την ζητούμενη Λειτουργικότητα**

Για κάθε λειτουργικότητα του προγράμματος την οποία υλοποιήσατε, δώστε στον παρακάτω πίνακα το function prototype της συνάρτησης που την υλοποιεί.

|  |  |
| --- | --- |
| **Λειτουργικότητα** | **Function prototype συνάρτησης** |
| 1 - Καταχώρηση παραγγελίας | void addOrder(OrderDetails orderArray[], int \*lastOrder, OrderDetails newOrder); |
| 2- Εμφάνιση παραγγελιών πελάτη | void printClientOrders(OrderDetails orderArray[], int currentOrder, int lastOrder); |
| 3 - Εμφάνιση εκκρεμών παραγγελιών | void printArray(OrderDetails orderArray[], int currentOrder, int lastOrder); |
| 4 - Αποθήκευση εκκρεμών παραγγελιών σε αρχείο | void saveOrdersToFile(OrderDetails \*orderArray[MAX\_ORDERS], int currentOrder, int lastOrder, const char \*filename); |
| 5 - Φόρτωση εκκρεμών παραγγελιών από αρχείο. | void loadOrdersFromFile(OrderDetails \*orderArray[MAX\_ORDERS], int \*currentOrder, int \*lastOrder, const char \*filename); |
| 6 - Εκτέλεση παραγγελίας | void readyOrder(OrderDetails pendingOrderArray[], OrderDetails readyOrderArray[], int \*lastReadyOrder, int \*currentPendingOrder, int lastPendingOrder); |
| 7 - Εμφάνιση των προς εξόφληση παραγγελιών | void printArray(OrderDetails orderArray[], int currentOrder, int lastOrder); |
| 8 - Εξόφληση παραγγελίας | void closeOrder(OrderDetails readyOrderArray[], OrderDetails closedOrderArray[], int \*lastClosedOrder, int \*currentReadyOrder, int lastReadyOrder); |
| 9 - Εμφάνιση εκτελεσμένων παραγγελιών | void printArray(OrderDetails orderArray[], int currentOrder, int lastOrder); |
| 10 - Αποθήκευση των εκτελεσμένων παραγγελιών σε ξεχωριστό αρχείο | void saveOrdersToFile(OrderDetails \*orderArray[MAX\_ORDERS], int currentOrder, int lastOrder, const char \*filename); |
| 11 - Φόρτωση από αρχείο εκτελεσμένων παραγγελιών | void loadOrdersFromFile(OrderDetails \*orderArray[MAX\_ORDERS], int \*currentOrder, int \*lastOrder, const char \*filename); |

* **Βασικές αρχές υλοποίησης**

Στον παρακάτω πίνακα βάλτε 1 στις γραμμές που περιγράφουν τους τρόπους υλοποίησης που ακολουθήσατε για την ανάπτυξη του προγράμματος σας.

|  |  |
| --- | --- |
| **Τρόπος Υλοποίησης** | **Αξιοποιήθηκε** |
| Πίνακας παραγγελιών |  |
| Πίνακας δεικτών σε παραγγελίες | 1 |
| Χρήση ενός πίνακα για όλες τις παραγγελίες |  |
| Χρήση 3 διαφορετικών πινάκων για τις παραγγελίες | 1 |

* **1η έκδοση του προγράμματος**

Για την πρώτη έκδοση του προγράμματος που αναπτύξατε δώστε: την λεκτική περιγραφή, τον πηγαίο κώδικα και ένα screenshot από την εκτέλεση του προγράμματος.

* **Λεκτική περιγραφή**

Δώστε την λεκτική περιγραφή της λειτουργικότητας που υλοποιεί η 1η έκδοση του προγράμματος σας.

-Δείξε στον χρήστη τις επιλογές

-Πάρε την επιλογή του χρήστη

-Αν επιλογή 0 τερμάτισε

* **Πηγαίος Κώδικας**

Δώστε τον πηγαίο κώδικα της 1ης έκδοσης του προγράμματος σας (σε μορφή κειμένου). Κάντε Export σε μορφή RTF τον πηγαίο κώδικα και στη συνέχεια μεταφέρεται τον από το αρχείο που τον αποθηκεύσατε στην παρακάτω θέση με copy-paste χωρίς να χαλάσει η στοίχιση και η εμφάνιση του. Ο κώδικας σας θα πρέπει να είναι στην μορφή του παρακάτω κώδικα που σας δίνεται ενδεικτικά. Τον διαγράφεται και τοποθετείται τον δικό σας.

#include <stdio.h>

**enum** menu **{**

TERMINATE**,**

GET\_ORDER**,**

SHOW\_CLIENT\_ORDER**,**

SHOW\_PENDING\_ORDERS

**};**

**int** menu**();**

**void** getOrder**();**

**void** showOrders**();**

**void** showClientOrder**();**

**int** main**(){**

**int** choice **=** **-**1**;**

**while((**choice **=** menu**())!=**TERMINATE**){**

**switch(**choice**){**

**case** TERMINATE**:**

**return** 0**;**

**case** GET\_ORDER**:**

getOrder**();**

**break;**

**case** SHOW\_CLIENT\_ORDER**:**

showOrders**();**

**break;**

**case** SHOW\_PENDING\_ORDERS**:**

showClientOrder**();**

**break;**

**default:**

printf**("INVALID CHOICE\n");**

**break;**

**}**

**}**

**}**

**int** menu**(){**

**int** choice**;**

*// Get choice*

printf**("Exit: 0\n");**

printf**("Get order: 1\n");**

printf**("Show client orders: 2\n");**

printf**("Show Pending Orders: 3\n");**

printf**("Save pending orders: 4\n");**

printf**("Load pending order: 5\n");**

printf**("Ready order: 6\n");**

printf**("Show ready orders: 7\n");**

printf**("Close order: 8\n");**

printf**("Show closed order: 9\n");**

printf**("Save closed orders: 10\n");**

printf**("Load closed orders: 11\n");**

printf**("Enter choice:\n");**

scanf**("%d",** **&**choice**);**

**return** choice**;**

**}**

**void** getOrder**(){**

**};**

**void** showOrders**(){**

**};**

**void** showClientOrder**(){**

**};**

* **Screenshot εκτέλεσης του προγράμματος**

Δώστε Screenshot της εκτέλεσης της 1ης έκδοσης του προγράμματος σας.

* **Εκτέλεση παραγγελίας**

Για την λειτουργικότητα «Εκτέλεση παραγγελίας» δώστε: την λεκτική περιγραφή και τον πηγαίο κώδικα της συνάρτησης που την υλοποιεί.

* **Λεκτική περιγραφή**

Δώστε τη λεκτική περιγραφή της συνάρτησης που υλοποιεί τη λειτουργικότητα «Εκτέλεση παραγγελίας».

-Πάρε την παρραγελία

-Υπολόγισε αρχικό κόστος

-Βάλε την παρραγελία στον πίνακα ready

* **Πηγαίος Κώδικας**

Δώστε τον πηγαίο κώδικα της συνάρτησης που υλοποιεί τη λειτουργικότητα «Εκτέλεση παραγγελίας».

OrderDetails**\*** calculateSum**(**OrderDetails**\*** pendingOrderArray**[],** **int** currentPendingOrder**)** **{**

**float** sum**;**

sum **=** pendingOrderArray**[**currentPendingOrder**]->**bBottles **\*** 0.02 **+** pendingOrderArray**[**currentPendingOrder**]->**sBottles **\*** 0.002**;**

pendingOrderArray**[**currentPendingOrder**]->**startingCost **=** sum**;**

**return** pendingOrderArray**[**currentPendingOrder**];**

**}**

**void** readyOrder**(**OrderDetails**\*** pendingOrderArray**[],** OrderDetails**\*** readyOrderArray**[],** **int\*** lastReadyOrder**,** **int\*** currentPendingOrder**,** **int** lastPendingOrder**)** **{**

**if** **(\***currentPendingOrder **==** lastPendingOrder**)** **{**

printf**("No orders to ready.\n");**

**return;**

**}**

OrderDetails**\*** order **=** calculateSum**(**pendingOrderArray**,** **\***currentPendingOrder**);**

addOrder**(**readyOrderArray**,** lastReadyOrder**,** order**);**

**(\***currentPendingOrder**)++;**

**}**

* **Εξόφληση παραγγελίας**

Για την λειτουργικότητα «Εξόφληση παραγγελίας» δώστε: την λεκτική περιγραφή και τον πηγαίο κώδικα της συνάρτησης που την υλοποιεί.

* **Λεκτική περιγραφή**

Δώστε τη λεκτική περιγραφή της συνάρτησης που υλοποιεί τη λειτουργικότητα «Εκτέλεση παραγγελίας».

-Πάρε την παρραγελία

-Υπολόγισε έκπτωση και τελικό κόστος

-Βάλε τελική ημερομηνία την σημερινή

-Βάλε παραγγελία στον πίνακα closed

* **Πηγαίος Κώδικας**

Δώστε τον πηγαίο κώδικα της συνάρτησης που υλοποιεί τη λειτουργικότητα «Εκτέλεση παραγγελίας».

OrderDetails**\*** calculateFinalSum**(**OrderDetails**\*** readyOrderArray**[],** **int** currentReadyOrder**)** **{**

**int** totalBottles**;**

**float** sum**;**

totalBottles **=** readyOrderArray**[**currentReadyOrder**]->**bBottles **+** readyOrderArray**[**currentReadyOrder**]->**sBottles**;**

sum **=** readyOrderArray**[**currentReadyOrder**]->**startingCost**;**

**if** **(**sum **>** 600**)** **{**

readyOrderArray**[**currentReadyOrder**]->**discount **=** 20**;**

readyOrderArray**[**currentReadyOrder**]->**finalCost **=** sum **\*** 0.8**;**

**}** **else** **if** **(**sum **>** 200 **||** totalBottles **>** 3000**)** **{**

readyOrderArray**[**currentReadyOrder**]->**discount **=** 8**;**

readyOrderArray**[**currentReadyOrder**]->**finalCost **=** sum **\*** 0.92**;**

**}** **else** **{**

readyOrderArray**[**currentReadyOrder**]->**discount **=** 0**;**

readyOrderArray**[**currentReadyOrder**]->**finalCost **=** sum**;**

**}**

**return** readyOrderArray**[**currentReadyOrder**];**

**}**

**void** closeOrder**(**OrderDetails**\*** readyOrderArray**[],** OrderDetails**\*** closedOrderArray**[],** **int\*** lastClosedOrder**,** **int\*** currentReadyOrder**,** **int** lastReadyOrder**)** **{**

**if** **(\***currentReadyOrder **==** lastReadyOrder**)** **{**

printf**("No orders to close.\n");**

**return;**

**}**

**char** endDate**[**10**];**

OrderDetails**\*** order **=** calculateFinalSum**(**readyOrderArray**,** **\***currentReadyOrder**);**

getCurrentDate**(**endDate**);**

strcpy**(**order**->**dateDelivered**,** endDate**);**

addOrder**(**closedOrderArray**,** lastClosedOrder**,** order**);**

**(\***currentReadyOrder**)++;**

**}**

* **Εμφάνιση των προς εξόφληση παραγγελιών**

Για την λειτουργικότητα «Εμφάνιση των προς εξόφληση παραγγελιών» δώστε: την λεκτική περιγραφή, τον πηγαίο κώδικα της συνάρτησης που την υλοποιεί και ένα screenshot από την εκτέλεση της λειτουργικότητας.

* **Λεκτική περιγραφή**

Δώστε τη λεκτική περιγραφή της συνάρτησης που υλοποιεί τη λειτουργικότητα «Εμφάνιση των προς εξόφληση παραγγελιών».

-Πάρε τον πίνακα προς εξόφληση παραγγελίας

-Τύπωσε ολες τις παραγγελίες του πίνακα

* **Πηγαίος Κώδικας**

Δώστε τον πηγαίο κώδικα της συνάρτησης που υλοποιεί τη λειτουργικότητα «Εμφάνιση των προς εξόφληση παραγγελιών».

**case** SHOW\_READY\_ORDERS**:**

printf**("Ready Orders:\n");**

printArray**(**readyOrderArray**,** currentReadyOrder**,** lastReadyOrder**);**

**break;**

**void** printOrder**(**OrderDetails**\*** order**)** **{**

printf**("%-15s\t",** order**->**name**);**

printf**("%-10d\t",** order**->**sBottles**);**

printf**("%-10d\t",** order**->**bBottles**);**

printf**("%-10.2f",** order**->**startingCost**);**

printf**("%-10.2f\t",** order**->**finalCost**);**

printf**("%-4d\t",** order**->**discount**);**

printf**("%-10s\t",** order**->**startDate**);**

printf**("%-10s\t",** order**->**deadline**);**

printf**("%-10s\n",** order**->**dateDelivered**);**

**}**

**void** printArray**(**OrderDetails**\*** orderArray**[],** **int** currentOrder**,** **int** lastOrder**)** **{**

**if** **(**currentOrder **==** lastOrder**)** **{**

printf**("No orders to show.\n\n");**

**return;**

**}**

**for** **(int** i **=** currentOrder**;** i **<** lastOrder**;** i**++)** **{**

printf**("Order %d: ",** i **+** 1**);**

printOrder**(**orderArray**[**i**]);**

**}**

**}**

* **Screenshot εκτέλεσης της λειτουργικότητας**

Δώστε Screenshot της εκτέλεσης της λειτουργικότητας «Εμφάνιση των προς εξόφληση παραγγελιών».

* **Αποθήκευση των εκτελεσμένων παραγγελιών σε ξεχωριστό αρχείο**

Για την λειτουργικότητα «Αποθήκευση των εκτελεσμένων παραγγελιών σε ξεχωριστό αρχείο» δώστε: την λεκτική περιγραφή, τον πηγαίο κώδικα της συνάρτησης που την υλοποιεί και ένα screenshot που να δείχνει τα περιεχόμενα του αρχείου στο οποίο αποθηκεύσατε τις παραγγελίες.

* **Λεκτική περιγραφή**

Δώστε τη λεκτική περιγραφή της συνάρτησης που υλοποιεί τη λειτουργικότητα «Αποθήκευση των εκτελεσμένων παραγγελιών σε ξεχωριστό αρχείο».

-Πάρε πίνακα

-Άνοιξε η δημιούργησε αρχείο

-Αποθήκευσαι τον πίνακα στο αρχειίο

* **Πηγαίος Κώδικας**

Δώστε τον πηγαίο κώδικα της συνάρτησης που υλοποιεί τη λειτουργικότητα «Αποθήκευση των εκτελεσμένων παραγγελιών σε ξεχωριστό αρχείο».

**case** SAVE\_CLOSED\_ORDERS**:**

saveOrdersToFile**(**closedOrderArray**,** buffer**,** lastClosedOrder**,** **"closed-order.txt");**

**break;**

**void** saveOrdersToFile**(**OrderDetails **\***orderArray**[**MAX\_ORDERS**],** **int** currentOrder**,** **int** lastOrder**,** **const** **char** **\***filename**)** **{**

FILE **\***file **=** fopen**(**filename**,** **"w");**

**if** **(**file **==** NULL**)** **{**

printf**("Error opening file for writing.\n");**

**return;**

**}**

fprintf**(**file**,** **"Total Orders: %d\n",** lastOrder**);**

fprintf**(**file**,** **"Current Orders: %d\n",** currentOrder**);**

**for** **(int** i **=** currentOrder**;** i **<** currentOrder **+** lastOrder**;** **++**i**)** **{**

**int** index **=** i **%** MAX\_ORDERS**;**

fprintf**(**file**,** **"%-15s\t",** orderArray**[**index**]->**name**);**

fprintf**(**file**,** **"%-10d\t",** orderArray**[**index**]->**sBottles**);**

fprintf**(**file**,** **"%-10d\t",** orderArray**[**index**]->**bBottles**);**

fprintf**(**file**,** **"%-10.2f",** orderArray**[**index**]->**startingCost**);**

fprintf**(**file**,** **"%-10.2f\t",** orderArray**[**index**]->**finalCost**);**

fprintf**(**file**,** **"%-4d\t",** orderArray**[**index**]->**discount**);**

fprintf**(**file**,** **"%-10s\t",** orderArray**[**index**]->**startDate**);**

fprintf**(**file**,** **"%-10s\t",** orderArray**[**index**]->**deadline**);**

fprintf**(**file**,** **"%-10s\n",** orderArray**[**index**]->**dateDelivered**);**

**}**

fclose**(**file**);**

**}**

* **Screenshot εκτέλεσης της λειτουργικότητας**

Δώστε ένα screenshot που να δείχνει τα περιεχόμενα του αρχείου στο οποίο αποθηκεύσατε τις παραγγελίες.

* **Σχόλια, Παρατηρήσεις για το παραδοτέο**

Καταγράψτε τα σχόλια, τις παρατηρήσεις σας και ότι άλλο θεωρείται σκόπιμο για το παραδοτέο αυτό. Για τις λειτουργικότητες που δεν υλοποιήσατε μπορείτε να περιγράψετε τα προβλήματα που συναντήσατε.