

Κων/νου Στυλιάδη

Έτοιμες  
Ασκήσεις σε  
Clipper

Φλώρινα, Οκτώβριος 1997

```
/* Πρόγραμμα clip01.prg - το πρώτο μας πρόγραμμα στον Clipper */  
CLEAR SCREEN /* καθάρισμα της οθόνης */
```

```
@ 10, 15 SAY 'Mary'
```

```
QUIT /* τελειώνει το πρόγραμμα και επιστρέφουμε στο Dos */
```

```
/* Πρόγραμμα clip02.prg - εύρεση του μεγίστου από τρεις ακεραίους */  
CLEAR SCREEN
```

```
A := 3
```

```
B := 8
```

```
C := 6
```

```
IF A>B .AND. A>C  
    ? 'MAX EINAI O A'
```

```
ENDIF
```

```
IF B>A .AND. B>C  
    ? 'MAX EINAI O B'
```

```
ENDIF
```

```
IF C>A .AND. C>B  
    ? 'MAX EINAI O C'
```

```
ENDIF
```

```
QUIT
```

```
/* άλλος τρόπος λύσης της ίδιας άσκησης */  
CLEAR SCREEN
```

```
A := 3
```

```
B := 8
```

```
C := 6
```

```
MAX := A
```

```
IF B>MAX  
    MAX := B
```

```
ENDIF
```

```
IF C>MAX  
    MAX := C
```

```
ENDIF
```

```
? 'MAX EINAI O : ' + MAX
```

```
QUIT
```

*/\* Πρόγραμμα clip03.prg - εκμάθηση του βρόχου do while - εύρεση του αθροίσματος των 10 πρώτων αριθμών \*/*

```
CLEAR SCREEN
SUM := 0
X := 1
DO WHILE X <= 10
    SUM := SUM + X
    X := X + 1
ENDDO
? 'Το άθροισμα είναι : ' + SUM
```

*/\* Πρόγραμμα clip04.prg - εύρεση αθροίσματος αριθμών έως ότου αυτό ξεπεράσει το 100 \*/*

```
CLEAR SCREEN
SUM := 0           // άθροισμα
I := 0            // μετρητής
X := 0           // αριθμός
ONOMA := SPACE(15) // όνομα (αλφαριθμητικό)
DO WHILE SUM < 100
    @ 5,5 SAY 'Δώσε αριθμό : ' GET X PICTURE '999'
    @ 6,5 SAY 'Δώσε όνομα : ' GET ONOMA PICTURE '@A'
    READ
    SUM := SUM + X
    I := I + 1
ENDDO
@ 5,5 TO 10,40      // σχεδιάζει ένα ορθογώνιο πλαίσιο
@ 7,8 SAY 'Το άθροισμα είναι : ' GET SUM PICTURE '9999'
@ 8,8 SAY 'Οι αριθμοί ήταν σε πλήθος : ' GET I PICTURE '999'
QUIT
```

*/\* Πρόγραμμα clip05.prg - υπόλοιπο ακεραίας διαίρεσης - εκμάθηση της εντολής do case \*/*

```
CLEAR SCREEN
YPOL := 0
A := 0
@ 5,5 SAY 'Δώσε έναν αριθμό : ' GET A PICTURE '99'
READ
YPOL := A % 3 // εύρεση ακεραίου υπολοίπου
DO CASE
    CASE YPOL == 0
        ? 'Το υπόλοιπο είναι 0'
    CASE YPOL == 1
        ? 'Το υπόλοιπο είναι 1'
    CASE YPOL == 2
        ? 'Το υπόλοιπο είναι 2'
    OTHERWISE
        ? 'Απίθανη περίπτωση'
ENDCASE
QUIT
```

*/\* Πρόγραμμα clip06.prg - υπολογισμός του λογαριασμού της ΔΕΗ - οι πρώτες 100 μονάδες χρεώνονται προς 15 δραχ., οι επόμενες 100 προς 16 δραχ. και από 201 μονάδες και πάνω χρεώνονται προς 17 δραχ. - το πάγιο είναι 1.500 δραχ., το ποσό για την ΕΡΤ 5% και ο ΦΠΑ 18% \*/*

CLEAR SCREEN

KATANALOSH := 0

@ 5, 8 SAY 'Δώσε την κατανάλωση : ' GET KATANALOSH;  
PICTURE '9999'

READ

IF KATANALOSH <= 100  
LOGAR := KATANALOSH \* 15  
ENDIF

IF KATANALOSH > 100 .AND. KATANALOSH < 200  
LOGAR := 1500 + (KATANALOSH - 100)\* 16  
ENDIF

IF KATANALOSH > 200  
LOGAR := 1500 + 1600 + (KATANALOSH - 200)\* 17  
ENDIF

DEH := LOGAR  
LOGAR := LOGAR + 1500  
ERT := LOGAR \* 0.05  
LOGAR := LOGAR + ERT  
FPA := LOGAR \* 0.18  
LOGAR := LOGAR + FPA

@ 07, 10 SAY 'Για τη ΔΕΗ πληρώνετε : ' GET DEH PICTURE '999999'

@ 08, 10 SAY 'Το πάγιο είναι : 1500 '

@ 09, 10 SAY 'Για την ΕΡΤ πληρώνετε : ' GET ERT PICTURE '9999'

@ 10, 10 SAY 'Ο ΦΠΑ είναι : ' GET FPA PICTURE '9999'

@ 11, 10 SAY 'Τελικό ποσό πληρωμής : ' GET LOGAR PICTURE '999999'

QUIT

*/\* Πρόγραμμα clip07.prg - υπολογισμός μισθοδοσίας - οι κρατήσεις είναι 20% των μικτών αποδοχών, ο φόρος είναι 5% στα μικτά-κρατήσεις και οι υπερωρίες πληρώνονται με το 20% του ημερομισθίου \*/*

```
CLEAR SCREEN
SET BELL ON
SET CONFIRM OFF
SET DATE TO BRITISH
SET COLOR TO 'R/W', 'RB/G'
```

```
EPONYMO := SPACE(20)
ONOMA := SPACE(10)
ADDRESS := SPACE(20)
TK := SPACE(6)
POLI := SPACE(10)
HM_GEN := CTOD(' / / ')
FYLO := .T.
HMEROM := 0
HM_ERG := 0
YPER := 0
MIKTA := 0
KATHARA := 0
KRATISIS := 0
FOROS := 0
```

```
@ 05, 8 SAY 'Επώνυμο      :' GET EPONYMO PICTURE '@A'
@ 06, 8 SAY 'Όνομα        :' GET ONOMA PICTURE '@A'
@ 07, 8 SAY 'Διεύθυνση     :' GET ADDRESS PICTURE '@X'
@ 08, 8 SAY 'TK            :' GET TK PICTURE '999 99'
@ 09, 8 SAY 'Πόλη          :' GET POLI PICTURE '@A'
@ 10, 8 SAY 'Ημ. Γένν.      :' GET HM_GEN
@ 11, 8 SAY 'Φύλο           :' GET FYLO
@ 12, 8 SAY 'Ημερομίσθιο : ' GET HMEROM PICTURE '99999'
@ 13, 8 SAY 'Ημέρ. Εργασ.: ' GET HM_ERG PICTURE '99'
@ 14, 8 SAY 'Υπερωρίες    :' GET YPER PICTURE '999'
READ
```

```
MIKTA := HMEROM * HM_ERG + YPER * HMEROM * 0.2
KRATISIS := MIKTA * 0.2
FOROS := (MIKTA - KRATISIS) * 0.05
KATHARA := MIKTA - KRATISIS - FOROS
```

```
@ 15, 8 SAY 'Μικτά          :' GET MIKTA PICTURE '999999'
@ 16, 8 SAY 'Κρατήσεις     :' GET KRATISIS PICTURE '999999'
@ 17, 8 SAY 'Φόρος           :' GET FOROS PICTURE '999999'
@ 18, 8 SAY 'Καθαρά         :' GET KATHARA PICTURE '999999'
```

```
QUIT
```

```

/* Πρόγραμμα clip08.prg - εμφάνιση οριζόντιων και κατακόρυφων μενού */
CLEAR SCREEN
SET BELL ON
SET CONFIRM OFF
SET DATE TO BRITISH
SET COLOR TO 'B/BG', 'R/W'
@ 0,0 TO 4,70 DOUBLE /* σχεδιάζει ένα οριζόντιο πλαίσιο */
DO WHILE .T.
    SET WRAP ON /* επιτρέπει την ανακύκλωση */
    SET MESSAGE TO 20 CENTRE
    @ 2,2 PROMPT 'ΠΕΛΑΤΕΣ' MESSAGE 'Στοιχεία Πελατών'
    @ 2,20 PROMPT 'ΠΡΟΜΗΘΕΥΤΕΣ' MESSAGE;
        'Στοιχεία Προμηθευτών'
    @ 2,40 PROMPT 'ΑΠΟΘΗΚΗ' MESSAGE 'Στοιχεία Αποθήκης'
    @ 2,60 PROMPT 'ΕΞΟΔΟΣ' MESSAGE 'Έξοδος στο DOS'
    MENU TO MENU_CHOICE
    DO CASE
        CASE MENU_CHOICE == 1
            PELATĒS() /* καλεί τη διαδικασία των πελατών */
        CASE MENU_CHOICE == 2
            PROMĪTH() /* καλεί τη διαδικασία των προμηθευτών */
        CASE MENU_CHOICE == 3
            APOTHĪKI() /* καλεί τη διαδικασία της αποθήκης */
        CASE MENU_CHOICE == 4
            QUIT /* έξοδος στο DOS */
        CASE MENU_CHOICE == 0
            QUIT
    ENDCASE
ENDDO

```

```

PROCEDURE PELATES()
@ 3, 2 TO 9, 20 DOUBLE      /* σχεδιάζει ένα κατακόρυφο πλαίσιο */
SET WRAP ON
SET CONFIRM OFF
@ 4, 2 PROMPT 'ΚΑΤΑΧΩΡΗΣΗ ΠΕΛΑΤΗ';
      MESSAGE 'Καταχώρηση Νέου Πελάτη'
@ 5, 2 PROMPT 'ΕΜΦΑΝΙΣΗ ΠΕΛΑΤΗ';
      MESSAGE 'Εμφάνιση Υπάρχοντος Πελάτη'
@ 6, 2 PROMPT 'ΔΙΑΓΡΑΦΗ ΠΕΛΑΤΗ';
      MESSAGE 'Διαγραφή Υπάρχοντος Πελάτη'
@ 7, 2 PROMPT 'ΕΚΤΥΠΩΣΗ ΠΕΛΑΤΗ';
      MESSAGE 'Εκτύπωση Υπάρχοντος Πελάτη'
@ 8, 2 PROMPT 'ΕΠΙΣΤΡΟΦΗ';
      MESSAGE 'Επιστροφή στο Κυρίως Μενού'
MENU TO PELATES_CHOICE

DO CASE
CASE PELATES_CHOICE == 1
  ΚΑΤΑΧ_ΡΕΛΑΤΕΣ()
  /* καλεί τη διαδικασία καταχώρησης νέων πελατών */
CASE PELATES_CHOICE == 2
  ΕΜΦ_ΡΕΛΑΤΕΣ()
  /* καλεί τη διαδικασία εμφάνισης πελατών */
CASE MENU_CHOICE == 3
  ΔΕΛ_ΡΕΛΑΤΕΣ()
  /* καλεί τη διαδικασία διαγραφής πελατών */
CASE MENU_CHOICE == 4
  ΕΚΤ_ΡΕΛΑΤΕΣ()
  /* καλεί τη διαδικασία εκτύπωσης πελατών */
CASE MENU_CHOICE == 5
  @ 3, 2 CLEAR TO 9, 20
  /* καθάρισμα του κατακόρυφου πλαισίου */
  RETURN
  /* επιστροφή στο κυρίως μενού */
ENDCASE

```

*/\* οι διαδικασίες PROMITH() και ΑΡΟΘΗΚΙ() είναι εντελώς παρόμοιες με τη διαδικασία PELATES() \*/*

```
/* Πρόγραμμα clip09.prg - εκμάθηση των εντολών exit και loop */
CLEAR SCREEN
```

```
NUM := 0
```

```
DO WHILE .T.
    @ 3, 5 SAY 'Δώσε έναν αριθμό : ' GET NUM PICTURE '999'
    READ
    IF NUM > 0 .AND. NUM < 100
        EXIT      /* έξοδος από τον βρόχο */
    ELSE
        LOOP      /* πηγαίνει στην αρχή του βρόχου */
    ENDIF
ENDDO

QUIT
```

```
/* Πρόγραμμα clip10.prg - κλήση συνάρτησης (function) που υπολογίζει τον
μέσο όρο τριών αριθμών */
CLEAR SCREEN
```

```
A1 := 0
A2 := 0
A3 := 0
```

```
@ 5, 8 SAY 'Δώσε τον πρώτο αριθμό : ' GET A1 PICTURE '9999'
@ 6, 8 SAY 'Δώσε τον δεύτερο αριθμό : ' GET A2 PICTURE '9999'
@ 7, 8 SAY 'Δώσε τον τρίτο αριθμό : ' GET A3 PICTURE '9999'
```

```
READ
```

```
MO := F_MO(A1, A2, A3)      /* κλήση της συνάρτησης */
@ 9, 8 SAY 'Ο Μέσος Όρος είναι : ' GET MO PICTURE '9999'
```

```
QUIT
```

```
FUNCTION F_MO(B1, B2, B3)
LOCAL MESOS      /* τοπική μεταβλητή της συνάρτησης */
    MESOS := INT((B1 + B2 + B3)/3 + 0.5)
RETURN MESOS     /* επιστροφή τιμής στο καλούν πρόγραμμα */
```



```
/* Πρόγραμμα clip11.prg - κλήση διαδικασίας (procedure) που τυπώνει ένα ό-  
νομα όσες φορές είναι η τιμή ενός αριθμού */  
CLEAR SCREEN
```

```
NAME := SPACE(15)  
NUM := 0
```

```
@ 5,8 SAY 'Δώσε όνομα : ' GET NAME PICTURE '@A'  
@ 6,8 SAY 'Δώσε αριθμό : ' GET NUM PICTURE '99'  
READ
```

```
APROC(NAME, NUM) /* κλήση της διαδικασίας */
```

```
QUIT
```

```
PROCEDURE APROC(ONOMA, A)  
LOCAL I /* τοπική μεταβλητή της διαδικασίας */  
FOR I := 1 TO A  
    QOUT(ONOMA) /* ίδια με την εντολή ? */  
NEXT I  
RETURN /* επιστροφή στο καλούν πρόγραμμα */
```

*/\* Πρόγραμμα clip12.prg - το πρόγραμμα αυτό θα επεξεργάζεται τους βαθμούς ενός μαθητή και θα εμφανίζει ένα μενού με τρεις επιλογές :*

- καταχώρηση στοιχείων μαθητή
- υπολογισμός βαθμολογίας και εμφάνιση αποτελέσματος
- έξοδος

- τα στοιχεία που θα δώσουμε για τον μαθητή είναι τα εξής : επώνυμο, όνομα, διεύθυνση, ΤΚ, πόλη, ημερ. γέννησης, αρ. μητρώου, φύλο, απουσίες και οι βαθμοί για τα εξής μαθήματα :

αρχαία από πρωτότυπο και αρχαία από μετάφραση (ανά δύο ένας βαθμός),  
 νέα ελληνικά και έκθεση (ανά δύο ένας βαθμός),  
 φυσική και χημεία (ανά δύο ένας βαθμός),  
 πληροφορική και τεχνολογία (ανά δύο ένας βαθμός),  
 μουσική και καλλιτεχνικά (ανά δύο ένας βαθμός),  
 ιστορία (μόνο του ένας βαθμός),  
 γυμναστική (μόνο του ένας βαθμός) και  
 οικιακή οικονομία (μόνο του ένας βαθμός)

- τα δέκα πρώτα μαθήματα δίνουν έναν βαθμό που προκύπτει από τον μέσο όρο των μαθημάτων ανά δύο, δηλ. αν έχει ο μαθητής 10 στη φυσική και 11 στη χημεία, τότε λαμβάνεται υπόψη ο βαθμός 11 για τα δύο αυτά μαθήματα - δηλ. έχουμε σύνολο  $5+3 = 8$  βαθμούς

- το πρόγραμμα θα κάνει έλεγχο κατά την εισαγωγή των δεδομένων αν οι βαθμοί είναι στο διάστημα 0-20 και οι απουσίες στο διάστημα 0-300

- αφού υπολογίσουμε τον μέσο όρο από τα 8 μαθήματα, για να βγάλουμε το αποτέλεσμα, δηλ. αν ο μαθητής προάγεται, παραπέμπεται ή απορρίπτεται, πρέπει να έχουμε υπόψη μας τα εξής :

αν έχει παραπάνω από 150 απουσίες, απορρίπτεται

αν έχει από 100-150 απουσίες, αλλά μέσο όρο σ' όλα τα μαθήματα  $< 16$ ,

τότε παραπέμπεται σ' όλα τα μαθήματα

αν έχει σε περισσότερα από 5 μαθήματα  $< 10$ , τότε απορρίπτεται

αν έχει μέσο όρο  $< 10$  σ' ένα έως τέσσερα μαθήματα και

γενικό μέσο όρο  $\geq 13$ , τότε προάγεται

αν έχει μέσο όρο  $< 10$  σ' ένα έως τέσσερα μαθήματα και

γενικό μέσο όρο  $< 13$ , τότε παραπέμπεται στα αντίστοιχα μαθήματα

αν δεν ισχύει κάτι από τα παραπάνω, τότε προάγεται \*/

```

/* κυρίως πρόγραμμα */
CLEAR SCREEN
SET BELL ON
SET CONFIRM OFF
SET DATE TO BRITISH
EPONYMO=SPACE(20)
ONOMA=SPACE(10)
ADDRESS=SPACE(20)
TK=SPACE(6)
POLI=SPACE(10)
HMEROM=CTOD(" / / ")
AR_MITR=0
FYLO=.T.
ARXAIΑ_PROT=0
ARXAIΑ_MET=0
NEA_ELL=0
EKTHESI=0
PHYSIKI=0
XHMEIA=0
PLHROF=0
TEXNOL=0
MOUSIKI=0
KALLIT=0
ISTORIA=0
GYMN=0
OIK_OIK=0
APOUSIES=0
APOR=0 // μετρητής των μαθημάτων που είναι κάτω από τη βάση (10)

@ 0, 0 TO 3, 53 DOUBLE
SET WRAP ON
DO WHILE .T.
    @ 1, 1 PROMPT "ΚΑΤΑΧΩΡΗΣΗ ΣΤΟΙΧΕΙΩΝ ΜΑΘΗΤΗ"
    @ 2, 1 PROMPT "ΥΠΟΛΟΓΙΣΜΟΣ ΒΑΘΜ. & ΑΠΟΤΕΛΕΣΜΑ"
    @ 3, 1 PROMPT "ΕΞΟΔΟΣ"
    MENU TO CHOICE
    DO CASE
        CASE CHOICE == 1
            ΚΑΤΑΧΟΡΙΣΙ()
            INKEY(0)
        CASE CHOICE == 2
            ΥΠΟΛΟΓΙΣΜΟΣ()
            INKEY(0)
        /* με τη συνάρτηση INKEY(0), το πρόγραμμα περιμένει μέχρι να
        πατήσουμε κάποιο πλήκτρο για να συνεχίσει */
        CASE CHOICE == 3 .OR. CHOICE == 0
            QUIT
    ENDCASE
ENDDO

```

*/\* η διαδικασία καταχώρησης των στοιχείων του μαθητή \*/*

PROCEDURE ΚΑΤΑΧΟΡΙΣΙ()

CLEAR SCREEN

```
@ 1, 1 SAY "ΕΠΩΝΥΜΟ      :" GET EPONYMO PICTURE '@A'
@ 2, 1 SAY "ΟΝΟΜΑ        :" GET ONOMA PICTURE '@A'
@ 3, 1 SAY "ΔΙΕΥΘΥΝΣΗ    :" GET ADDRESS PICTURE '@X'
@ 4, 1 SAY "ΤΚ           :" GET TK PICTURE '999 99'
@ 5, 1 SAY "ΠΟΛΗ         :" GET POLI PICTURE '@A'
@ 6, 1 SAY "ΗΜΕΡ.ΓΕΝΝ.   :" GET ΗΜΕΡΟΜ
@ 7, 1 SAY "ΦΥΛΟ         :" GET FYLO
@ 8, 1 SAY "ΑΡ. ΜΗΤΡΩΟΥ  :" GET AR_MITR PICTURE '9999'
@ 9, 1 SAY "ΑΡΧΑΙΑ-ΠΡΩΤ.  :" GET ARΧΑΙΑ_PROT PICTURE '99';
    RANGE 0, 20
@ 10, 1 SAY "ΑΡΧΑΙΑ-MET.  :" GET ARΧΑΙΑ_MET PICTURE '99';
    RANGE 0, 20
@ 11, 1 SAY "ΝΕΑ ΕΛΛΗΝ.   :" GET NEA_ELL PICTURE '99';
    RANGE 0, 20
@ 12, 1 SAY "ΕΚΘΕΣΗ      :" GET EKTHESI PICTURE '99';
    RANGE 0, 20
@ 13, 1 SAY "ΦΥΣΙΚΗ      :" GET PHYSIKI PICTURE '99';
    RANGE 0, 20
@ 14, 1 SAY "ΧΗΜΕΙΑ      :" GET ΧΗΜΕΙΑ PICTURE '99';
    RANGE 0, 20
@ 15, 1 SAY "ΠΛΗΡΟΦΟΡ.    :" GET PLHROF PICTURE '99';
    RANGE 0, 20
@ 16, 1 SAY "ΤΕΧΝΟΛΟΓΙΑ   :" GET TEXNOL PICTURE '99';
    RANGE 0, 20
@ 17, 1 SAY "ΜΟΥΣΙΚΗ      :" GET MOUSIKI PICTURE '99';
    RANGE 0, 20
@ 18, 1 SAY "ΚΑΛΛΙΤΕΧΝ.    :" GET KALLIT PICTURE '99';
    RANGE 0, 20
@ 19, 1 SAY "ΙΣΤΟΡΙΑ       :" GET ISTORIA PICTURE '99';
    RANGE 0, 20
@ 20, 1 SAY "ΓΥΜΝΑΣΤΙΚΗ     :" GET GYMN PICTURE '99';
    RANGE 0, 20
@ 21, 1 SAY "ΟΙΚΙΑΚΗ ΟΙΚ.   :" GET ΟΙΚ_OIK PICTURE '99';
    RANGE 0, 20
@ 22, 1 SAY "ΑΠΟΥΣΙΕΣ     :" GET APOUSIES PICTURE '999';
    RANGE 0, 300
```

READ

RETURN

*/\* η διαδικασία υπολογισμού του βαθμού προαγωγής καθώς και του αποτελέσματος, δηλ., αν ο μαθητής προάγεται, παραπέμπεται ή απορρίπτεται \*/*  
 PROCEDURE ΥΠΟΛΟΓΙΣΜΟΣ()  
 CLEAR SCREEN

*/\* η συνάρτηση INT() επιστρέφει το ακέραιο μέρος ενός αριθμού χωρίς να κάνει στρογγυλοποίηση - εδώ προσθέτουμε το 0.5 για να στρογγυλευθεί ο μέσος όρος δύο βαθμών στον πλησιέστερο ακέραιο \*/*  
 ARΧΑΙΑ = INT((ARΧΑΙΑ\_PROT + ARΧΑΙΑ\_MET)/2+0.5)  
 IF ARΧΑΙΑ < 10  
     APOR = APOR + 1  
 ENDIF

NEA = INT((NEA\_ELL + EKTHESI)/2+0.5)  
 IF NEA < 10  
     APOR = APOR + 1  
 ENDIF

PHS\_XHM = INT((PHYSIKI + ΧΗΜΕΙΑ)/2+0.5)  
 IF PHS\_XHM < 10  
     APOR = APOR + 1  
 ENDIF

ΤΕΧΝΟΛΟΓΙΑ = INT((PLHROF + TEXNOL)/2+0.5)  
 IF ΤΕΧΝΟΛΟΓΙΑ < 10  
     APOR = APOR + 1  
 ENDIF

ΚΑΛΛΙΤΕΧΝ = INT((MOUSIKI + ΚΑΛΛΙΤ)/2+0.5)  
 IF ΚΑΛΛΙΤΕΧΝ < 10  
     APOR = APOR + 1  
 ENDIF

IF ΙΣΤΟΡΙΑ < 10  
     APOR = APOR + 1  
 ENDIF

IF GYMN < 10  
     APOR = APOR + 1  
 ENDIF

IF ΟΙΚ\_ΟΙΚ < 10  
     APOR = APOR + 1  
 ENDIF

MO = INT(ARΧΑΙΑ + NEA + PHS\_XHM + ΤΕΧΝΟΛΟΓΙΑ +  
 ΚΑΛΛΙΤΕΧΝ + ΙΣΤΟΡΙΑ + GYMN + ΟΙΚ\_ΟΙΚ)/8 + 0.5)  
*/\* εδώ γίνεται ο υπολογισμός του γενικού βαθμού προαγωγής - ο μέσος όρος από 8 μαθήματα \*/*

```
IF APOUSIES >= 150
    @ 2, 2 SAY "Απορρίπτεται σ' όλα τα μαθήματα λόγω απουσιών"
    RETURN
ENDIF

IF APOR >= 5
    @ 2, 2 SAY "Απορρίπτεται σε περισσότερα από 4 μαθήματα"
    RETURN
ENDIF

IF (APOUSIES > 100 .AND. APOUSIES < 150) .AND. MO < 16
    @ 2, 2 SAY "Παραπέμπεται σ' όλα τα μαθήματα λόγω απουσιών"
    RETURN
ENDIF

IF (APOR >= 1 .AND. APOR <= 4) .AND. MO >= 13
    @ 2, 2 SAY "Προάγεται"
    RETURN
ENDIF

IF (APOR >= 1 .AND. APOR <= 4) .AND. MO < 13
    @ 2, 2 SAY "Παραπέμπεται"
    RETURN
ENDIF

@ 2, 2 SAY "Προάγεται"

RETURN
```

/\* Πρόγραμμα clip13.prg - να γίνει πρόγραμμα σε Clipper που να δημιουργεί τη μορφή ενός τιμολογίου, όπως φαίνεται στο παρακάτω σχήμα :

**ΤΙΜΟΛΟΓΙΟ**  
**ΣΤΟΙΧΕΙΑ ΠΕΛΑΤΗ**

Αρ.Παραστ. :                      Επώνυμο :                      Όνομα :  
Ημερ.: / /                      Διεύθυνση :                      ΤΚ :                      Πόλη :                      ΑΦΜ :

A/A	Κωδικός	Περιγραφή	Τιμή	Ποσότητα	Αξία χωρίς ΦΠΑ
	ΦΠΑ %	Αξία ΦΠΑ	Τιμή με ΦΠΑ	Έκπτωση %	Τελική αξία
001	1520	Πόρτα FIAT	20.000	3	60.000
	18	10.800	70.800	10	63.720
002	1309	Καπό BMW	50.000	2	100.000
	16	16.000	116.000	5	110.200

...

ΣΥΝΟΛΟ					
	26.800	186.800		12.880	160.000
					173.920

θα γίνουν 5 καταχωρήσεις για ανταλλακτικά αυτοκινήτων και στο κάτω μέρος της οθόνης θα εμφανιστούν τα σύνολα για τις στήλες όπως φαίνονται παραπάνω \*/

SET BELL ON  
SET CONFIRM OFF  
SET DATE TO BRITISH  
CLEAR SCREEN

```
arith_paras=0
eponimo=SPACE(14)
onoma=SPACE(10)
imerom=CTOD(" / / ")
address=SPACE(10)
tk=SPACE(6)
poli=SPACE(12)
afm=SPACE(8)
a_a=0
kodikos=0
perigrabi=SPACE(20)
timi=0
posotita=0
axia_no_fpa=0
fpa=0
axia_fpa=0
timi_fpa=0
ekptosi=0
poso_ekptosis=0
teliki_axia=0
sum_no_fpa=0
sum_fpa=0
sum_me_fpa=0
```

```

sum_ekptosi=0
sum_teliki_axia=0
@ 0,0 SAY PADC("ΤΙΜΟΛΟΓΙΟ", 80)
@ 1,0 SAY PADC(REPLICATE(' ', 9), 80)
@ 2,0 SAY PADC("ΣΤΟΙΧΕΙΑ ΠΕΛΑΤΗ", 80)
@ 3,0 SAY PADC(REPLICATE(' ', 15), 80)
@ 4,0 SAY "Αριθμ. Παρασπ.: " GET arith_paras PICTURE '9999'
@ 4,14 SAY "Επώνυμο : " GET eponimo PICTURE '@A'
@ 4,40 SAY "Όνομα : " GET onoma PICTURE '@A'
@ 5,0 SAY "Ημερ.: " GET imerom
IF imerom = CTOD(" / / ")
    imerom = DATE()
    @ 5,7 SAY imerom
ENDIF
@ 5,15 SAY "Διεύθυνση : " GET address PICTURE '@X'
@ 5,32 SAY "TK : " GET tk PICTURE '999 99'
@ 5,45 SAY "Πόλη : " GET poli PICTURE '@A'
@ 5,65 SAY "ΑΦΜ : " GET afm PICTURE '99999999'
@ 8,0 SAY "Α/Α Κωδικός Περιγραφή          Τιμή Ποσότητα Αξία;
        χωρίς ΦΠΑ"
@ 9,8 SAY "ΦΠΑ %          Αξία ΦΠΑ  Τιμή με ΦΠΑ  Έκπτωση %  Ποσό;
        έκπτωσης  Τελική αξία"
for i:=1 to 5
    @ 10 + 2*(i-1), 0 SAY i picture '9999'
    @ 10 + 2*(i-1), 6 GET kodikos PICTURE '9999'
    @ 10 + 2*(i-1), 15 GET perigrafi PICTURE '@A'
    @ 10 + 2*(i-1), 40 GET timi PICTURE '99999999'
    @ 10 + 2*(i-1), 50 GET posotita PICTURE '999'
    axia_no_fpa = timi * posotita
    @ 10 + 2*(i-1), 60 SAY axia_no_fpa
    @ 11 + 2*(i-1), 6 GET fpa PICTURE '99'
    axia_fpa = int(axia_no_fpa * fpa/100 + 0.5)
    @ 11 + 2*(i-1), 16 SAY axia_fpa
    timi_fpa = axia_fpa + axia_no_fpa
    @ 11 + 2*(i-1), 30 SAY timi_fpa
    @ 11 + 2*(i-1), 40 GET ekptosi PICTURE '99'
    poso_ekptosis = int(timi_fpa * ekptosi/100 + 0.5)
    @ 11 + 2*(i-1), 0 SAY poso_ekptosis
    teliki_axia = timi_fpa - poso_ekptosis
    @ 11 + 2*(i-1), 5 SAY teliki_axia
    sum_no_fpa = sum_no_fpa + axia_no_fpa
    sum_fpa = sum_fpa + axia_fpa
    sum_me_fpa = sum_me_fpa + timi_fpa
    sum_ekptosi = sum_ekptosi + poso_ekptosis
    sum_teliki_axia = sum_teliki_axia + teliki_axia
read
next i
quit

```



*/\* Πρόγραμμα Clip14.prg - να γίνει πρόγραμμα σε Clipper που να διαβάζει ένα string από το πληκτρολόγιο και να μας εμφανίζει πόσοι από τους χαρακτήρες του είναι ψηφία, πόσοι είναι γράμματα και πόσοι είναι σύμβολα. Ακόμα, από τους χαρακτήρες που είναι γράμματα, πόσοι είναι κεφαλαία και πόσοι είναι μικρά γράμματα. Όλοι οι χαρακτήρες θα πρέπει να είναι γραμμένοι στα αγγλικά \*/*

```

CLEAR SCREEN
STRING=SPACE(20)
@ 10, 10 SAY 'Δώσε ένα string : ' GET STRING
READ
DIGIT := 0
ALPHA := 0
UPPER := 0
LOWER := 0
SYMBOL := 0
FOR X:=1 TO LEN(ALLTRIM(STRING))
  CHAR := SUBSTR(STRING, X, 1)
  IF ISDIGIT(CHAR)
    DIGIT := DIGIT + 1
  ENDIF
  IF ISALPHA(CHAR) .AND. ISUPPER(CHAR)
    ALPHA := ALPHA + 1
    UPPER := UPPER + 1
  ENDIF
  IF ISALPHA(CHAR) .AND. ISLOWER(CHAR)
    ALPHA := ALPHA + 1
    LOWER := LOWER + 1
  ENDIF
  IF !ISDIGIT(CHAR) .AND. !ISALPHA(CHAR)
    SYMBOL := SYMBOL + 1
  ENDIF
NEXT X
? 'Υπάρχουν ' + DIGIT + ' ψηφία'
? 'Υπάρχουν ' + ALPHA + ' γράμματα'
? 'Υπάρχουν ' + UPPER + ' κεφαλαία γράμματα'
? 'Υπάρχουν ' + LOWER + ' μικρά γράμματα'
? 'Υπάρχουν ' + SYMBOL + ' σύμβολα'
QUIT

```

*/\* η συνάρτηση LEN(STRING) υπολογίζει το μήκος ενός αλφαριθμητικού, εδώ του STRING, η συνάρτηση SUBSTR(STRING, X, 1) αποσπάει από το STRING τον χαρακτήρα που είναι στη θέση X, η συνάρτηση ISALPHA(STRING) επιστρέφει αληθή τιμή .T. αν ο πρώτος χαρακτήρας του STRING είναι γράμμα, η συνάρτηση ISDIGIT(STRING) επιστρέφει αληθή τιμή .T. αν ο πρώτος χαρακτήρας του STRING είναι αριθμός (ψηφίο), η συνάρτηση ISUPPER(STRING) επιστρέφει αληθή τιμή .T. αν ο πρώτος χαρακτήρας του STRING είναι κεφαλαίο γράμμα και η συνάρτηση ISLOWER(STRING) επιστρέφει αληθή τιμή .T. αν ο πρώτος χαρακτήρας του STRING είναι πεζό (μικρό) γράμμα \*/*

*/\* Πρόγραμμα Clip15.prg - να γίνει πρόγραμμα σε Clipper που να διαβάζει το επώνυμο και το όνομα ενός ατόμου, τα οποία είναι χωρισμένα με έναν κενό χαρακτήρα, από μια μεταβλητή και να τα ξεχωρίζει και να τα καταχωρεί σε δύο διαφορετικές μεταβλητές \*/*

```
CLEAR SCREEN
EPONYMIA=SPACE(30)
@ 10,10 SAY 'ΔΩΣΕ ΟΝΟΜΑ ΚΑΙ ΕΠΩΝΥΜΟ ΜΕ ΚΕΝΟ ΑΝΑΜΕΣΑ :';
GET EPONYMIA
READ
EPONYMIA = ALLTRIM(EPONYMIA)
/* αφαιρεί τα αρχικά και τελικά κενά */
THESI = AT(' ', EPONYMIA)
/* εντοπίζει τη θέση του ' ' μέσα στην EPONYMIA */
MIKOS = LEN(EPONYMIA)
ONOMA = LEFT(EPONYMIA, THESI-1)
EPONYMO = RIGHT(EPONYMIA, MIKOS-THESI)
```

*/\* Πρόγραμμα Clip16.prg - να γίνει πρόγραμμα σε Clipper που να διαβάζει συνέχεια διάφορους χαρακτήρες από το πληκτρολόγιο και να τελειώνει όταν δοθούν οι χαρακτήρες 'CLIPPER' με οποιαδήποτε σειρά \*/*

```
LEXI=SPACE(7)
METRITIS_P=0
DO WHILE LEXI != 'CLIPPER'
  CH=CHR(INKEY(0)) /* διαβάζουμε έναν χαρακτήρα */
  DO CASE
    CASE CH=='C'
      STUFF(LEXI, 1, 1, 'C')
    CASE CH=='L'
      STUFF(LEXI, 2, 1, 'L')
    CASE CH=='I'
      STUFF(LEXI, 3, 1, 'I')
    CASE CH=='P'
      IF METRITIS_P = 0
        STUFF(LEXI, 4, 1, 'P')
        METRITIS_P = METRITIS_P + 1
      ELSEIF METRITIS_P = 1
        STUFF(LEXI, 5, 1, 'P')
        METRITIS_P = METRITIS_P + 1
      ENDIF
    CASE CH=='E'
      STUFF(LEXI, 6, 1, 'E')
    CASE CH=='R'
      STUFF(LEXI, 7, 1, 'R')
  ENDCASE
ENDDO
```

*/\* η συνάρτηση STUFF(LEXI, 3, 1, 'I') τοποθετεί στην αλφαριθμητική μεταβλητή LEXI στη θέση 3 τον χαρακτήρα 'I' αφού προηγουμένως έχει διαγράψει έναν χαρακτήρα από τη θέση 3 - κάνει δηλ. αντικατάσταση χαρακτήρων μέσα σε μεταβλητή \*/*

*/\* Πρόγραμμα menu.prg - το πρόγραμμα αυτό δημιουργεί ένα μενού πελατών και παραγγελιών, από το οποίο καλούμε προγράμματα για καταχώρηση, διόρθωση και διαγραφή εγγραφών πελατών και των παραγγελιών που κάνουν οι πελάτες - τα αρχείο των πελατών έχει όνομα PELATES.DBF και χρησιμοποιεί το αρχείο ευρετηρίου CODE\_PEL.NTX, ενώ το αρχείο των παραγγελιών έχει όνομα ORDER.DBF και χρησιμοποιεί το αρχείο ευρετηρίου CODE\_ORD.NTX - και τα δύο αρχεία δημιουργήθηκαν από το πρόγραμμα DBU \*/*

```
CLEAR SCREEN
SET BELL ON
SET CONFIRM OFF
SET DATE TO BRITISH
```

```
@ 0, 0 to 7, 70 double
SET WRAP ON
```

```
DO WHILE .T.
    @ 02, 02 PROMPT "ΠΕΛΑΤΕΣ"
    @ 02, 35 PROMPT "ΠΑΡΑΓΓΕΛΙΕΣ"
    @ 02, 60 PROMPT "ΕΞΟΔΟΣ"
    MENU TO menu_choice
    do case
        case menu_choice == 1
            pelates() /* καλεί τη διαδικασία pelates() */
        case menu_choice == 2
            order() /* καλεί τη διαδικασία order() */
        case menu_choice == 3 .or. menu_choice == 0
            exit
    endcase
ENDDO
quit
```

```
PROCEDURE PELATES()
```

@ 4, 3 to 10, 29 double

SET WRAP ON

@ 05, 05 PROMPT "ΚΑΤΑΧΩΡΗΣΗ"

@ 06, 05 PROMPT "ΠΡΟΒΟΛΗ-ΔΙΟΡΘΩΣΗ"

@ 07, 05 PROMPT "ΔΙΑΓΡΑΦΗ"

@ 08, 05 PROMPT "ΕΞΟΔΟΣ"

MENU TO pel\_choice

do case

case pel\_choice == 1

do kat\_pel

*/\* καλεί το πρόγραμμα kat\_pel.prg \*/*

case pel\_choice == 2

do pro\_pel

*/\* καλεί το πρόγραμμα pro\_pel.prg \*/*

case pel\_choice == 3

do del\_pel

*/\* καλεί το πρόγραμμα del\_pel.prg \*/*

case pel\_choice == 4

return

endcase

return

PROCEDURE ORDER()

@ 4, 34 to 10, 51 double

SET WRAP ON

@ 05, 35 PROMPT "ΚΑΤΑΧΩΡΗΣΗ"

@ 06, 35 PROMPT "ΠΡΟΒΟΛΗ-ΔΙΟΡΘΩΣΗ"

@ 07, 35 PROMPT "ΔΙΑΓΡΑΦΗ "

@ 08, 35 PROMPT "ΕΞΟΔΟΣ"

MENU TO order\_choice

do case

case order\_choice == 1

do kat\_order

case order\_choice == 2

do pro\_order

case order\_choice == 3

do del\_order

case order\_choice == 4

return

endcase

return

*/\* Πρόγραμμα kat\_pel.prg - το πρόγραμμα αυτό καταχωρεί στοιχεία για τους νέους πελάτες στο αρχείο PELATES.DBF και χρησιμοποιεί το αρχείο ευρετηρίου (index) CODE\_PEL.NTX με κλειδί τον κωδικό πελάτη (CODE) \*/*

```
CLEAR SCREEN
SET BELL ON
SET CONFIRM OFF
SET DATE TO BRITISH
```

```
USE PELATES
INDEX ON CODE TO CODE_PEL
```

```
do while .t.
```

```
    m_code = 0
    m_eponymo = SPACE(20)
    m_onoma = SPACE(10)
    m_address = SPACE(20)
    m_phone = SPACE(9)
    m_poli = SPACE(10)
    m_tk = SPACE(6)
    m_hmer = CTOD(" / / ")
    m_fylo = .T.
    m_ypoloipo = 0
```

```
@01,01 SAY PADC("ΚΑΤΑΧΩΡΗΣΗ ΣΤΟΙΧΕΙΩΝ ΠΕΛΑΤΩΝ",80)
@ 02, 01 SAY PADC(REPLICATE('_', 28), 80)
```

```
@ 04, 01 SAY "Κωδικός : " GET m_code PICTURE '9999'
READ
```

```
if m_code = 9999
    exit /* με κωδικό = 9999 βγαίνουμε από το πρόγραμμα */
endif
```

```
seek m_code /* αναζήτηση του κωδικού */
if found() /* αν βρεθεί ο κωδικός */
    tone(300,50) /* ηχητικό σήμα */
    @ 19,20 SAY "Ο κωδικός υπάρχει ήδη"
    loop /* πάμε στην αρχή του βρόχου */
endif
```

```

/* από δω και κάτω αρχίζει η καταχώρηση των στοιχείων του πελάτη */
@ 05, 01 SAY "Επώνυμο   :" GET m_eronymo PICTURE '@A'
@ 06, 01 SAY "Όνομα     :" GET m_onoma PICTURE '@A'
@ 07, 01 SAY "Διεύθυνση  :" GET m_address PICTURE '@X'
@ 08, 01 SAY "Τηλέφωνο   :" GET m_phone PICTURE '@X'
@ 09, 01 SAY "Πόλη       :" GET m_poli PICTURE '@A'
@ 10, 01 SAY "TK         :" GET m_tk PICTURE '999 99'
@ 11, 01 SAY "Ημ/νία     :" GET m_hmer
@ 12, 01 SAY "Φύλο       :" GET m_fylo
@ 13, 01 SAY "Υπόλοιπο   :" GET m_ypoloipo PICTURE '999999999'
READ

wait /* ο Η/Υ περιμένει μέχρι να πατήσουμε ένα πλήκτρο */

append blank
/* προσθέτει μια κενή εγγραφή στο τέλος του PELATES.DBF */

replace code with m_code
replace eronymo with m_eronymo
replace onoma with m_onoma
replace address with m_address
replace phone with m_phone
replace poli with m_poli
replace tk with m_tk
replace hmer with m_hmer
replace fylo with m_fylo
replace ypoloipo with m_ypoloipo
/* με την εντολή replace δίνουμε τιμές στα πεδία της τρέχουσας εγγρα-
φής του αρχείου PELATES.DBF */
enddo

close /* κλείνουν τα αρχεία */
clear screen
return /* επιστροφή στο menu.prg */

```

*/\* Πρόγραμμα pro\_pel.prg - το πρόγραμμα αυτό εμφανίζει τα στοιχεία των πελατών του αρχείου PELATES.DBF και χρησιμοποιεί το αρχείο ευρετηρίου (index) CODE\_PEL.NTX με κλειδί τον κωδικό πελάτη (CODE) \*/*

```
CLEAR SCREEN
SET BELL ON
SET CONFIRM OFF
SET DATE TO BRITISH
```

```
USE PELATES
INDEX ON CODE TO CODE_PEL
```

```
do while .t.
```

```
    m_code = 0
    m_eponymo = SPACE(20)
    m_onoma = SPACE(10)
    m_address = SPACE(20)
    m_phone = SPACE(9)
    m_poli = SPACE(10)
    m_tk = SPACE(6)
    m_hmer = CTOD(" / / ")
    m_fylo = .T.
    m_ypoloipo = 0
    @ 01, 01 SAY PADC("ΠΡΟΒΟΛΗ ΣΤΟΙΧΕΙΩΝ ΠΕΛΑΤΩΝ", 80)
    @ 02, 01 SAY PADC(REPLICATE('_', 25), 80)
```

```
    @ 04, 01 SAY "Κωδικός : " GET m_code PICTURE '9999'
    READ
```

```
    if m_code = 9999
        exit /* με κωδικό = 9999 βγαίνουμε από το πρόγραμμα */
    endif
```

```
    seek m_code /* αναζήτηση του κωδικού */
    if .not. found() /* αν δεν βρεθεί ο κωδικός */
        tone(300, 50) /* ηχητικό σήμα */
        @ 19, 20 SAY "Ο κωδικός δεν υπάρχει"
        loop /* πάμε στην αρχή του βρόχου */
    endif
```

```
    m_eponymo = eponymo
    m_onoma = onoma
    m_address = address
    m_phone = phone
    m_poli = poli
    m_tk = tk
    m_hmer = hmer
    m_fylo = fylo
    m_ypoloipo = ypoloipo
```

```
@ 05, 01 SAY "Επώνυμο : " GET m_εponymo PICTURE '@A'  
@ 06, 01 SAY "Όνομα : " GET m_ονομα PICTURE '@A'  
@ 07, 01 SAY "Διεύθυνση: " GET m_αδδress PICTURE '@X'  
@ 08, 01 SAY "Τηλέφωνο: " GET m_ρhone PICTURE '@X'  
@ 09, 01 SAY "Πόλη : " GET m_ρoli PICTURE '@A'  
@ 10, 01 SAY "TK : " GET m_τk PICTURE '999 99'  
@ 11, 01 SAY "Ημ/νία : " GET m_ηmer  
@ 12, 01 SAY "Φύλο : " GET m_ρylo  
@ 13, 01 SAY "Υπόλοιπο : " GET m_ρπολοίρo PICTURE '999999999'  
READ
```

```
wait /* ο Η/Υ περιμένει μέχρι να πατήσουμε ένα πλήκτρο */
```

```
replace code with m_code  
replace εponymo with m_εponymo  
replace ονομα with m_ονομα  
replace αδδress with m_αδδress  
replace ρhone with m_ρhone  
replace ρoli with m_ρoli  
replace τk with m_τk  
replace ηmer with m_ηmer  
replace ρylo with m_ρylo  
replace ρπολοίρo with m_ρπολοίρo
```

```
enddo
```

```
close /* κλείνουν τα αρχεία */  
clear screen  
return /* επιστροφή στο menu.prg */
```



*/\* Πρόγραμμα del\_pel.prg - το πρόγραμμα αυτό διαγράφει μια εγγραφή ενός πελάτη από το αρχείο PELATES.DBF και χρησιμοποιεί το αρχείο ευρετηρίου (index) CODE\_PEL.NTX με κλειδί τον κωδικό πελάτη (CODE) \*/*

```
CLEAR SCREEN
SET BELL ON
SET CONFIRM OFF
SET DATE TO BRITISH
```

```
USE PELATES
INDEX ON CODE TO CODE_PEL
do while .t.
```

```
    m_code = 0
    m_eponymo = SPACE(20)
    m_onoma = SPACE(10)
    m_address = SPACE(20)
    m_phone = SPACE(9)
    m_poli = SPACE(10)
    m_tk = SPACE(6)
    m_hmer = CTOD("/")
    m_fylo = .T.
    m_ypoloipo = 0
```

```
    @ 01, 01 SAY PADC("ΔΙΑΓΡΑΦΗ ΠΕΛΑΤΗ", 80)
```

```
    @ 02, 01 SAY PADC(REPLICATE('_', 15), 80)
```

```
    @ 04, 01 SAY "Κωδικός : " GET m_code PICTURE '9999'
    READ
```

```
    if m_code = 9999
        exit /* με κωδικό = 9999 βγαίνουμε από το πρόγραμμα */
    endif
```

```
    seek m_code /* αναζήτηση του κωδικού */
    if .not. found() /* αν δεν βρεθεί ο κωδικός */
        tone(300,50) /* ηχητικό σήμα */
        @ 19, 20 SAY "Ο κωδικός δεν υπάρχει"
        loop /* πάμε στην αρχή του βρόχου */
    endif
```

```
    m_eponymo = eponymo
    m_onoma = onoma
    m_address = address
    m_phone = phone
    m_poli = poli
    m_tk = tk
    m_hmer = hmer
    m_fylo = fylo
    m_ypoloipo = ypoloipo
```

```

@ 05, 01 SAY "Επώνυμο : " GET m_eronymo PICTURE '@A'
@ 06, 01 SAY "Όνομα : " GET m_onoma PICTURE '@A'
@ 07, 01 SAY "Διεύθυνση: " GET m_address PICTURE '@X'
@ 08, 01 SAY "Τηλέφωνο: " GET m_phone PICTURE '@X'
@ 09, 01 SAY "Πόλη : " GET m_poli PICTURE '@A'
@ 10, 01 SAY "TK : " GET m_tk PICTURE '999 99'
@ 11, 01 SAY "Ημ/νία : " GET m_hmer
@ 12, 01 SAY "Φύλο : " GET m_fylo
@ 13, 01 SAY "Υπόλοιπο : " GET m_ypoloipo PICTURE '999999999'
wait /* ο Η/Υ περιμένει μέχρι να πατήσουμε ένα πλήκτρο */

@ 20, 01 SAY "Θέλετε να διαγραφεί ; "
apantisi = INKEY(0)
/* η συνάρτηση INKEY(0) περιμένει μέχρι να πατήσουμε ένα πλήκτρο
και καταχωρεί την ascii τιμή του πλήκτρου στη μεταβλητή apantisi */
if apantisi = ASC("N") .or. apantisi = ASC("n") .or.;
    apantisi = ASC("N") .or. apantisi = ASC("v")
    replace code with 0
    replace eronymo with " "
    replace onoma with " "
    replace address with " "
    replace phone with " "
    replace poli with " "
    replace tk with " "
    replace hmer with CTOD(" / / ")
    replace fylo with .F.
    replace ypoloipo with 0
    delete /* σημειώνει για διαγραφή την τρέχουσα εγγραφή */
endif
enddo

close /* κλείνουν τα αρχεία */
clear screen
return /* επιστροφή στο menu.prg */

```

```
/* Πρόγραμμα kat_ord.prg - το πρόγραμμα αυτό καταχωρεί στοιχεία για τις
νέες παραγγελίες στο αρχείο ORDER.DBF και χρησιμοποιεί το αρχείο ευρε-
τηρίου (index) CODE_ORD.NTX με κλειδί τον κωδικό παραγγελίας (CODE_
PAR) */
```

```
CLEAR SCREEN
SET BELL ON
SET CONFIRM OFF
SET DATE TO BRITISH
```

```
use order
index on code_par to code_ord
use pelates new
index on code to code_pel
select order /* κάνουμε ενεργό το αρχείο παραγγελιών */
```

```
m_code=0
m_code_par=0
m_perigrafi=space(20)
m_posohta=0
m_timh=0
m_fpa=0
m_syn_axia=0
m_hmerom=CTOD("//")
@ 1, 20 say "ΚΑΤΑΧΩΡΗΣΗ ΠΑΡΑΓΓΕΛΙΩΝ"
```

```
do while .t.
  @ 2, 2 say "Κωδικός Παραγγελίας : " get m_code_par picture '9999'
  read
  seek m_code_par

  if m_code_par = 9999
    exit /* βγαίνουμε από τον βρόχο */
  endif

  if found()
    @ 18, 2 say "Ο Κωδικός Υπάρχει"
    tone(300, 15) /* ηχητικό σήμα */
    loop /* επιστροφή στην αρχή του βρόχου */
  endif
```

```

select pelates      /* κάνουμε ενεργό το αρχείο πελατών */
do while .t.
    @ 3, 2 say "Κωδικός Πελάτη : " get m_code picture '9999'
    read
    seek m_code

    if .not. found()
        @ 18, 2 say "Ο Κωδικός δεν υπάρχει"
        tone(300, 15)      /* ηχητικό σήμα */
        loop /* επιστροφή στην αρχή του βρόχου */
    endif

    if found()
        @ 3, 30 say "Επώνυμο : " get eronymo
        exit /* βγαίνουμε από τον βρόχο */
    endif
enddo /* τέλος εσωτερικού βρόχου */

select order /* κάνουμε ενεργό το αρχείο παραγγελιών */
@ 4, 2 say "Περιγραφή : " get m_perigrafi picture '@A'
@ 5, 2 say "Ποσότητα : " get m_posothta picture '999999'
@ 6, 2 say "Τιμή : " get m_timh picture '99999'
@ 7, 2 say "ΦΠΑ : " get m_fpa picture '99';
        valid m_fpa=4 .or. m_fpa=8 .or. m_fpa=18
@ 8, 2 say "Ημερομηνία : " get m_hmerom
read

/* υπολογισμός της συνολικής αξίας */
m_syn_axia=m_posothta * m_timh
m_syn_axia=m_syn_axia+m_syn_axia*m_fpa/100
m_syn_axia=int(m_syn_axia+0.5)
@ 9,2 say "Συνολική Αξία : " get m_syn_axia

append blank /* προσθήκη μιας κενής εγγραφής */
replace code_par with m_code_par
replace code with m_code
replace perigrafi with m_perigrafi
replace posothta with m_posothta
replace timh with m_timh
replace fpa with m_fpa
replace syn_axia with m_syn_axia
replace hmeron with m_hmerom
enddo

clear screen
close /* κλείνουν τα αρχεία */
return /* επιστροφή στο menu.prg */

```

```
/* Πρόγραμμα pro_ord.prg - το πρόγραμμα αυτό εμφανίζει τα στοιχεία των
παραγγελιών του αρχείου ORDER.DBF και χρησιμοποιεί το αρχείο ευρετηρίου
(index) CODE_ORD.NTX με κλειδί τον κωδικό παραγγελίας (CODE_
PAR) */
```

```
CLEAR SCREEN
SET BELL ON
SET CONFIRM OFF
SET DATE TO BRITISH
```

```
use order
index on code_par to code_ord
use pelates new
index on code to code_pel
select order
```

```
m_code=0
m_code_par=0
m_perigrafi=space(20)
m_posohta=0
m_timh=0
m_fpa=0
m_syn_axia=0
m_hmerom=CTOD("//")
@ 1, 20 say "ΠΡΟΒΟΛΗ-ΔΙΟΡΘΩΣΗ ΠΑΡΑΓΓΕΛΙΩΝ"
```

```
do while .t.
    @ 2, 2 SAY "Κωδικός Παραγγελίας : " get m_code_par picture '9999'
    read
    seek m_code_par

    if m_code_par=9999
        exit
    endif

    if .not. found()
        @ 18, 2 say "Ο Κωδικός δεν υπάρχει"
        tone(300,15)
        loop
    endif

    m_code=code
    m_perigrafi=perigrafi
    m_posohta=posohta
    m_timh=timh
    m_fpa=fpa
    m_hmerom=hmerom
    m_syn_axia=syn_axia
```

```
@ 3, 2 say "Κωδικός Πελ. : " get m_code picture 9999
@ 4, 2 say "Περιγραφή : " get m_perigrafi picture '@A'
@ 5, 2 say "Ποσότητα : " get m_posothta picture '999999'
@ 6, 2 say "Τιμή : " get m_timh picture '99999'
@ 7, 2 say "ΦΠΑ : " get m_fpa picture '99';
      valid m_fpa=4 .or. m_fpa=8 .or.m_fpa=18
@ 8, 2 say "Ημερομηνία : " get m_hmerom
@ 9, 2 say "Συνολική Αξία: " get m_syn_axia
read

replace code with m_code
replace code_par with m_code_par
replace perigrafi with m_perigrafi
replace posothta with m_posothta
replace timh with m_timh
replace fpa with m_fpa
replace syn_axia with m_syn_axia
replace hmerom with m_hmerom

enddo

clear screen
close      /* κλείνουν τα αρχεία */
return    /* επιστροφή στο menu.prg */
```

*/\* Πρόγραμμα del\_ord.prg - το πρόγραμμα αυτό διαγράφει μια εγγραφή παραγγελίας από το αρχείο ORDER.DBF και χρησιμοποιεί το αρχείο ευρετηρίου (index) CODE\_ORD.NTX με κλειδί τον κωδικό παραγγελίας (CODE\_PAR) \*/*

```
CLEAR SCREEN
SET BELL ON
SET CONFIRM OFF
SET DATE TO BRITISH
```

```
use order
index on code_par to code_ord
use pelates new
index on code to code_pel
select order
```

```
m_code=0
m_code_par=0
m_perigrafi=space(20)
m_posothta=0
m_timh=0
m_fpa=0
m_syn_axia=0
m_hmerom=CTOD("//")
@ 1, 20 say "ΔΙΑΓΡΑΦΗ ΠΑΡΑΓΓΕΛΙΩΝ"
```

```
do while .t.
    @ 2, 2 SAY "Κωδικός Παραγγελίας : " get m_code_par picture '9999'
    read
    seek m_code_par
    if m_code_par=9999
        exit
    endif
    if .not. found()
        @ 18, 2 say "Ο Κωδικός δεν υπάρχει "
        tone(300,15)
        loop
    endif
    m_code=code
    m_perigrafi=perigrafi
    m_posothta=posothta
    m_timh=timh
    m_fpa=fpa
    m_hmerom=hmerom
    m_syn_axia=syn_axia
    @ 3, 2 say "Κωδικός Πελ. : " get m_code picture '9999'
    @ 4, 2 say "Περιγραφή : " get m_perigrafi picture '@A'
    @ 5, 2 say "Ποσότητα : " get m_posothta picture '999999'
    @ 6, 2 say "Τιμή : " get m_timh picture '99999'
```

```

@ 7, 2 say "ΦΠΑ          : " get m_fpa picture '99';
      valid m_fpa=4 .or. m_fpa=8 .or.m_fpa=18
@ 8, 2 say "Ημερομηνία  : " get m_hmerom
@ 9, 2 say "Συνολική Αξία: " get m_syn_axia
@ 20,1 say "Να διαγραφεί : N/O "
answer=inkey(0)
if answer=asc('n') .or. answer=asc('N') .or. answer=asc('N');
      .or. answer=asc('v')
      replace code_par with 0
      replace code with 0
      replace perigrafi with space(20)
      replace posothta with 0
      replace timh with 0
      replace fpa with 0
      replace syn_axia with 0
      replace hmerom with ctod(" / / ")
      delete /* σημειώνει για διαγραφή την τρέχουσα εγγραφή */
endif
enddo
clear screen
close      /* κλείνουν τα αρχεία */
return    /* επιστροφή στο menu.prg */

```