

Μάθημα 128

Υποπρογράμματα

Εμβέλεια μεταβλητών – σταθερών / Ορισμοί – παράδειγμα

Τι ονομάζεται εμβέλεια (scope) και ποια η εμβέλεια (ισχύς) των μεταβλητών στη ΓΛΩΣΣΑ;

Το τμήμα του προγράμματος που ισχύουν οι μεταβλητές λέγεται εμβέλεια (scope) μεταβλητών.

Τι σημαίνει απεριόριστη εμβέλεια; Δημιουργεί προβλήματα;

Σύμφωνα με αυτή την αρχή όλες οι μεταβλητές και όλες οι σταθερές είναι γνωστές και μπορούν να χρησιμοποιούνται σε οποιοδήποτε τμήμα του προγράμματος, άσχετα που δηλώθηκαν. Όλες οι μεταβλητές και σταθερές είναι **καθολικές**.

Η απεριόριστη εμβέλεια καταστρατηγεί την αρχή της αυτονομίας των υποπρογραμμάτων, δημιουργεί πολλά προβλήματα και τελικά είναι αδύνατη για μεγάλα προγράμματα με πολλά υποπρογράμματα, αφού ο καθένας που γράφει κάποιο υποπρόγραμμα πρέπει να γνωρίζει τα ονόματα όλων των μεταβλητών που χρησιμοποιούνται στα υπόλοιπα υποπρογράμματα.

ΜΕΤΑΒΛΗΤΕΣ

ΧΑΡΑΚΤΗΡΕΣ: X *! έστω X καθολική μεταβλητή*

!=====

ΠΡΟΓΡΑΜΜΑ μάθημα_128_ΚΑΘΟΛΙΚΗ_ΜΕΤΑΒΛΗΤΗ

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: Y, ΕΤΟΣ

ΑΡΧΗ

X ← ' μετά Χριστόν'

ΔΙΑΒΑΣΕ ΕΤΟΣ *! π.χ. 2024*

ΚΑΛΕΣΕ ΑΙΩΝ(ΕΤΟΣ, Y)

ΓΡΑΨΕ Y, 'ος αιώνας ', X *! 21ος αιώνας προ Χριστού*

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

ΔΙΑΔΙΚΑΣΙΑ ΑΙΩΝ(ΕΤΟΣ, Y)

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: ΕΤΟΣ, Y

ΑΡΧΗ

Y ← ΕΤΟΣ **div** 100 + 1 *! 20 + 1*

X ← ' προ Χριστού'

ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ

Εμβέλεια μεταβλητών – σταθερών / Ορισμοί

Τι σημαίνει περιορισμένη εμβέλεια και ποιο το πλεονέκτημά της;

Η περιορισμένη εμβέλεια υποχρεώνει όλες τις μεταβλητές που χρησιμοποιούνται σε ένα τμήμα προγράμματος, να δηλώνονται σε αυτό το τμήμα.

Όλες οι μεταβλητές είναι **τοπικές**, ισχύουν δηλαδή για το υποπρόγραμμα στο οποίο δηλώθηκαν.

Στη ΓΛΩΣΣΑ έχουμε περιορισμένη εμβέλεια.

Τα πλεονεκτήματα της περιορισμένης εμβέλειας είναι η απόλυτη αυτονομία όλων των υποπρογραμμάτων και η δυνατότητα να χρησιμοποιείται οποιοδήποτε όνομα, χωρίς να ενδιαφέρει αν το ίδιο χρησιμοποιείται σε άλλο υποπρόγραμμα.

Τι σημαίνει μερικώς περιορισμένη εμβέλεια; Προσφέρει πλεονεκτήματα;

Σύμφωνα με αυτή την αρχή άλλες μεταβλητές είναι τοπικές και άλλες καθολικές.

Κάθε γλώσσα προγραμματισμού έχει τους δικούς της κανόνες και μηχανισμούς για τον τρόπο και τις προϋποθέσεις που ορίζονται οι μεταβλητές ως τοπικές ή καθολικές.

Η μερικώς περιορισμένη εμβέλεια προσφέρει μερικά πλεονεκτήματα στον πεπειραμένο προγραμματιστή, αλλά για τον αρχάριο περιπλέκει το πρόγραμμα δυσκολεύοντας την ανάπτυξή του.

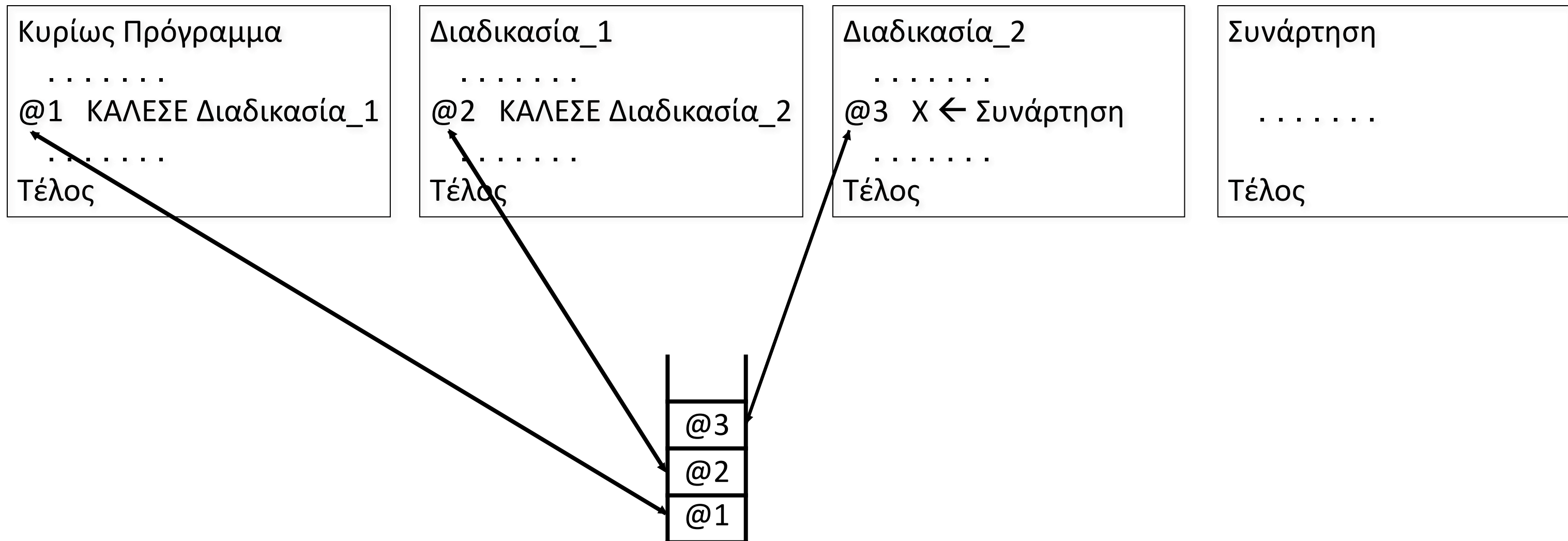
Στοιβά χρόνου εκτέλεσης / Ορισμός – παράδειγμα

Τι ονομάζεται στοιβά χρόνου εκτέλεσης;

Όταν μία διαδικασία ή συνάρτηση καλείται από το κύριο πρόγραμμα, τότε η αμέσως επόμενη διεύθυνση του κύριου προγράμματος, που ονομάζεται διεύθυνση επιστροφής, αποθηκεύεται από το μεταφραστή σε μία στοιβά που ονομάζεται στοιβά χρόνου εκτέλεσης.

Μετά την εκτέλεση της διαδικασίας ή της συνάρτησης η διεύθυνση επιστροφής απωθείται από τη στοιβά και έτσι ο έλεγχος του προγράμματος μεταφέρεται και πάλι στο κύριο πρόγραμμα.

Η τεχνική αυτή εφαρμόζεται και γενικότερα, δηλαδή οποτεδήποτε μία διαδικασία ή συνάρτηση καλεί μία διαδικασία ή συνάρτηση.



Μενού διαδικασιών – Ουρά – Στοίβα – Συγχώνευση / ΣΑΒΒΑΣ

Να γραφεί πρόγραμμα σε γλώσσα, το οποίο θα εμφανίζει τις επιλογές:

1. ΩΘΗΣΗ
2. ΑΠΩΘΗΣΗ
3. ΕΙΣΑΓΩΓΗ
4. ΕΞΑΓΩΓΗ
5. ΕΞΟΔΟΣ

μέχρι να επιλεγεί το 5.

- Εάν επιλεγεί 1, θα ζητά ένα όνομα το οποίο θα εισάγει σε στοίβα που θα υλοποιηθεί με τον πίνακα S[10].
- Εάν επιλεγεί 2, θα εμφανίζει και στη συνέχεια θα "διαγράφει" το τελευταίο όνομα που εισήχθη στη στοίβα.
- Εάν επιλεγεί 3, θα ζητά ένα όνομα το οποίο θα εισάγει σε ουρά που θα υλοποιηθεί με τον πίνακα Q[10].
- Εάν επιλεγεί 4, εμφανίζει και στη συνέχεια θα " διαγράφει" το όνομα που βρίσκεται πρώτο στην ουρά.
- Εάν επιλεγεί 5, τα ονόματα που "απέμειναν" στη στοίβα, θα απωθηθούν προς τον πίνακα A[10], ενώ τα ονόματα που "απέμειναν" στην ουρά, θα εισαχθούν τον πίνακα B[10], οι οποίοι πίνακες A και B στη συνέχεια θα ταξινομηθούν αλφαβητικά, θα συγχωνευτούν στον πίνακα Γ[20], τα στοιχεία του οποίου θα εμφανιστούν και το πρόγραμμα θα τερματιστεί.

Οι λειτουργίες ΩΘΗΣΗ, ΑΠΩΘΗΣΗ, ΕΙΣΑΓΩΓΗ, ΕΞΑΓΩΓΗ, ΤΑΞΙΝΟΜΗΣΗ, ΣΥΓΧΩΝΕΥΣΗ να υλοποιηθούν με διαδικασίες. Η εντολή ΓΡΑΨΕ, όπου χρειαστεί, να χρησιμοποιηθεί μόνο από το κυρίως πρόγραμμα.

Άσκηση / κώδικας 1

ΠΡΟΓΡΑΜΜΑ μάθημα_128_ΣΑΒΒΑΣ

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: top, front, rear, i, sizeA, sizeB

ΧΑΡΑΚΤΗΡΕΣ: ΕΠ, S[10], Q[10], A[10], B[10], Γ[20], ON, item

ΛΟΓΙΚΕΣ: done

ΑΡΧΗ

top ← 0

front ← 0

rear ← 0

ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ

ΓΡΑΨΕ '1. ΩΘΗΣΗ'

ΓΡΑΨΕ '2. ΑΠΩΘΗΣΗ'

ΓΡΑΨΕ '3. ΕΙΣΑΓΩΓΗ'

ΓΡΑΨΕ '4. ΕΞΑΓΩΓΗ'

ΓΡΑΨΕ '5. ΕΞΟΔΟΣ'

ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ

ΔΙΑΒΑΣΕ ΕΠ

ΜΕΧΡΙΣ_ΟΤΟΥ ΕΠ='1' Η ΕΠ='2' Η ΕΠ='3' Η ΕΠ='4' Η ΕΠ='5'

ΕΠΙΛΕΞΕ ΕΠ

ΠΕΡΙΠΤΩΣΗ '1'

ΔΙΑΒΑΣΕ ON

ΚΑΛΕΣΕ ΩΘΗΣΗ(S, top, ON, done)

ΑΝ done = ΨΕΥΔΗΣ ΤΟΤΕ

ΓΡΑΨΕ 'Στοίβα γεμάτη, δεν έγινε ώθηση'

ΤΕΛΟΣ_ΑΝ

ΠΕΡΙΠΤΩΣΗ '2'

ΚΑΛΕΣΕ ΑΠΩΘΗΣΗ(S, top, item, done)

ΑΝ done = ΨΕΥΔΗΣ ΤΟΤΕ

ΓΡΑΨΕ 'Στοίβα άδεια, δεν έγινε απώθηση'

ΑΛΛΙΩΣ

ΓΡΑΨΕ item

ΤΕΛΟΣ_ΑΝ

ΠΕΡΙΠΤΩΣΗ '3'

ΔΙΑΒΑΣΕ ON

ΚΑΛΕΣΕ ΕΙΣΑΓΩΓΗ(Q, front, rear, ON, done)

ΑΝ done = ΨΕΥΔΗΣ ΤΟΤΕ

ΓΡΑΨΕ 'Ουρά γεμάτη, δεν έγινε εισαγωγή'

ΤΕΛΟΣ_ΑΝ

ΠΕΡΙΠΤΩΣΗ '4'

ΚΑΛΕΣΕ ΕΞΑΓΩΓΗ(Q, front, rear, item, done)

ΑΝ done = ΨΕΥΔΗΣ ΤΟΤΕ

ΓΡΑΨΕ 'Ουρά άδεια, δεν έγινε εξαγωγή'

ΑΛΛΙΩΣ

ΓΡΑΨΕ item

ΤΕΛΟΣ_ΑΝ

ΠΕΡΙΠΤΩΣΗ ΑΛΛΙΩΣ

i ← 0

ΟΣΟ top > 0 ΕΠΑΝΑΛΑΒΕ

i ← i + 1

ΚΑΛΕΣΕ ΑΠΩΘΗΣΗ(S, top, item, done)

A[i] ← item

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

sizeA ← i

i ← 0

ΟΣΟ front <= rear ΚΑΙ front <> 0 ΕΠΑΝΑΛΑΒΕ

i ← i + 1

ΚΑΛΕΣΕ ΕΞΑΓΩΓΗ(Q, front, rear, item, done)

B[i] ← item

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

sizeB ← i

ΚΑΛΕΣΕ ΤΑΞΙΝΟΜΗΣΗ(A, sizeA)

ΚΑΛΕΣΕ ΤΑΞΙΝΟΜΗΣΗ(B, sizeB)

ΚΑΛΕΣΕ ΣΥΓΧΩΝΕΥΣΗ(A, B, Γ, sizeA, sizeB)

ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ sizeA + sizeB

ΓΡΑΨΕ Γ[i]

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ_ΕΠΙΛΟΓΩΝ

ΜΕΧΡΙΣ_ΟΤΟΥ ΕΠ = '5'

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

Άσκηση / κώδικας 2

ΔΙΑΔΙΚΑΣΙΑ ΩΘΗΣΗ(stack, top, item, done)

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: top

ΧΑΡΑΚΤΗΡΕΣ: stack[10], item

ΛΟΓΙΚΕΣ: done

ΑΡΧΗ

ΑΝ top < 10 **ΤΟΤΕ**

top ← top + 1

stack[top] ← item

done ← **ΑΛΗΘΗΣ**

ΑΛΛΙΩΣ

done ← **ΨΕΥΔΗΣ**

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ

/=====

ΔΙΑΔΙΚΑΣΙΑ ΑΠΩΘΗΣΗ(stack, top, item, done)

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: top

ΧΑΡΑΚΤΗΡΕΣ: stack[10], item

ΛΟΓΙΚΕΣ: done

ΑΡΧΗ

ΑΝ top > 0 **ΤΟΤΕ**

item ← stack[top]

top ← top - 1

done ← **ΑΛΗΘΗΣ**

ΑΛΛΙΩΣ

done ← **ΨΕΥΔΗΣ**

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ

ΔΙΑΔΙΚΑΣΙΑ ΕΙΣΑΓΩΓΗ(queue, front, rear, item, done)

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: front, rear

ΧΑΡΑΚΤΗΡΕΣ: queue[10], item

ΛΟΓΙΚΕΣ: done

ΑΡΧΗ

ΑΝ rear = 10 **ΤΟΤΕ**

done ← **ΨΕΥΔΗΣ**

ΑΛΛΙΩΣ

rear ← rear + 1

queue[rear] ← item

done ← **ΑΛΗΘΗΣ**

ΑΝ front = 0 **ΤΟΤΕ**

front ← 1

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ

/=====

ΔΙΑΔΙΚΑΣΙΑ ΕΞΑΓΩΓΗ(queue, front, rear, item, done)

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: front, rear

ΧΑΡΑΚΤΗΡΕΣ: queue[10], item

ΛΟΓΙΚΕΣ: done

ΑΡΧΗ

ΑΝ front = 0 **ΤΟΤΕ**

done ← **ΨΕΥΔΗΣ**

ΑΛΛΙΩΣ

item ← queue[front]

front ← front + 1

done ← **ΑΛΗΘΗΣ**

ΑΝ front > rear **ΤΟΤΕ**

front ← 0

rear ← 0

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ

Άσκηση / κώδικας 3

ΔΙΑΔΙΚΑΣΙΑ ΤΑΞΙΝΟΜΗΣΗ(T, size)

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: i, j, size

ΧΑΡΑΚΤΗΡΕΣ: T[10], temp

ΑΡΧΗ

ΓΙΑ i ΑΠΟ 2 ΜΕΧΡΙ size

ΓΙΑ j ΑΠΟ size ΜΕΧΡΙ i ΜΕ_ΒΗΜΑ -1

ΑΝ T[j - 1] > T[j] ΤΟΤΕ

temp ← T[j - 1]

T[j - 1] ← T[j]

T[j] ← temp

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ

ΔΙΑΔΙΚΑΣΙΑ ΣΥΓΧΩΝΕΥΣΗ(A, B, Γ, M, N)

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: δ, δ1, δ2, M, N, i

ΧΑΡΑΚΤΗΡΕΣ: A[10], B[10], Γ[20]

ΑΡΧΗ

δ1 ← 1

δ2 ← 1

δ ← 1

ΟΣΟ δ1 ≤ M ΚΑΙ δ2 ≤ N ΕΠΑΝΑΛΑΒΕ

ΑΝ A[δ1] < B[δ2] ΤΟΤΕ

Γ[δ] ← A[δ1]

δ1 ← δ1 + 1

ΑΛΛΙΩΣ

Γ[δ] ← B[δ2]

δ2 ← δ2 + 1

ΤΕΛΟΣ_ΑΝ

δ ← δ + 1

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΑΝ δ1 ≤ M ΤΟΤΕ

ΓΙΑ i ΑΠΟ δ1 ΜΕΧΡΙ M

Γ[δ] ← A[i]

δ ← δ + 1

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΑΛΛΙΩΣ

ΓΙΑ i ΑΠΟ δ2 ΜΕΧΡΙ N

Γ[δ] ← B[i]

δ ← δ + 1

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΔΙΑΔΙΚΑΣΙΑΣ

Άσκηση / Έσοδα – έξοδα – κέρδη

Μια επιχείρηση έχει δέκα υποκαταστήματα. Για στατιστικούς λόγους καταχωρούνται σε δισδιάστατο πίνακα $EΣ[10, 12]$ τα έσοδα των υποκαταστημάτων ανά μήνα και σε αντίστοιχο πίνακα $EΞ[10, 12]$ τα έξοδα.

Να γραφεί πρόγραμμα σε ΓΛΩΣΣΑ, το οποίο:

- α. i) Να περιλαμβάνει κατάλληλο τμήμα δηλώσεων.
ii) Να διαβάζει και να καταχωρεί σε πίνακα $ΟΝ[10]$ τα ονόματα των δέκα (10) υποκαταστημάτων.
- β. Για κάθε κατάστημα να εμφανίζει το όνομά του, να διαβάζει και να καταχωρεί στις κατάλληλες θέσεις των πινάκων $EΣ$ και $EΞ$ τα έσοδα και τα έξοδα ανά μήνα για ένα έτος.
- γ. Με κλήση του υποπρογράμματος $ΥΠΟΛΟΓΙΣΜΟΣ$, που περιγράφεται στο ερώτημα Δ5, να υπολογίζει τις τιμές του πίνακα $ΤΑΜΕΙΟ[10]$.
- δ. Αξιοποιώντας τα στοιχεία του πίνακα $ΤΑΜΕΙΟ[10]$ να εμφανίζει ταξινομημένα σε φθίνουσα σειρά κέρδους τα ονόματα των υποκαταστημάτων που έχουν κέρδος καθώς και τα αντίστοιχα κέρδη τους. Όταν η διαφορά έσοδα μείον έξοδα είναι μεγαλύτερη του μηδενός, τότε θεωρούμε ότι το υποκατάστημα έχει κέρδος. Σε περίπτωση που δύο ή περισσότερα υποκαταστήματα έχουν ίδιο κέρδος να ταξινομηθούν με αλφαβητική σειρά.
- ε. Να κατασκευάσετε το υποπρόγραμμα $ΥΠΟΛΟΓΙΣΜΟΣ$, το οποίο να δέχεται ως είσοδο τους πίνακες εσόδων $EΣ[10, 12]$ και εξόδων $EΞ[10, 12]$ και να επιστρέφει στο κυρίως πρόγραμμα μονοδιάστατο πίνακα $ΤΑΜΕΙΟ[10]$, ο οποίος να έχει σε κάθε θέση του τη διαφορά ετήσιων εσόδων μείον ετήσιων εξόδων για κάθε υποκατάστημα.

Ενότητα 5

Θεωρία

§5.6 / σελίδα 227

Ασκήσεις

Την άσκηση στην προηγούμενη σελίδα