



## Επιλογή και επανάληψη



Η ύλη που αναπτύσσεται σε αυτό το κεφάλαιο είναι συναφής με την ύλη που αναπτύσσεται στο 2<sup>ο</sup> κεφάλαιο. Όπου υπάρχουν διαφορές αναφέρονται ρητά. Προσέξτε ιδιαίτερα, πάντως, ότι στο κεφάλαιο αυτό οι δεσμευμένες λέξεις γράφονται ολόκληρες με κεφαλαία, γεγονός που συνδέεται με την υλοποίηση προγραμμάτων. Κατά την υλοποίηση αλγόριθμου να έχετε υπόψη σας ότι όλες οι δεσμευμένες λέξεις ξεκινούν με κεφαλαίο γράμμα αλλά συνεχίζουν με πεζά.

### Λογική έκφραση ή συνθήκη

Για τη σύνταξη μιας λογικής έκφρασης ή συνθήκης χρησιμοποιούνται σταθερές, μεταβλητές, αριθμητικές παραστάσεις, συγκριτικοί και λογικοί τελεστές. Οι χρησιμοποιούμενοι συγκριτικοί τελεστές παρουσιάζονται στον επόμενο πίνακα:

Συγκριτικός Τελεστής	Ελεγχόμενη σχέση
=	Ισότητα
<>	Ανισότητα
>	Μεγαλύτερο από
>=	Μεγαλύτερο ή ίσο
<	Μικρότερο από
<=	Μικρότερο ή ίσο

Η σύνταξη μιας λογικής έκφρασης είναι της μορφής

Έκφραση\_1 *συγκριτικός τελεστής* Έκφραση\_2

Η τιμή της Έκφρασης\_1 συγκρίνεται με την τιμή της Έκφρασης\_2. Το αποτέλεσμα είναι μια λογική τιμή **ΑΛΗΘΗΣ** ή **ΨΕΥΔΗΣ**.

Οι τιμές των δυο εκφράσεων μπορούν να είναι αριθμοί, χαρακτήρες, αλφαριθμητικά ή λογικές. Πώς συγκρίνουμε, λοιπόν, τις τιμές αυτές;

- Η σύγκριση μεταξύ αριθμών γίνεται με προφανή τρόπο.
- Η σύγκριση ατομικών χαρακτήρων στηρίζεται στην αλφαβητική σειρά, π.χ. το 'α' θεωρείται μικρότερο από το 'β'.
- Η σύγκριση αλφαριθμητικών δεδομένων βασίζεται στη σύγκριση χαρακτήρα προς χαρακτήρα σε κάθε θέση μέχρι να βρεθεί κάποια διαφορά σε κάποιον χαρακτήρα.
- Η σύγκριση λογικών έχει νόημα μόνο στην περίπτωση του ίσου (=) και του διάφορου (<>).



Όταν σε μια έκφραση συνδυάζονται αριθμητικοί και συγκριτικοί τελεστές, εκτελούνται πρώτες οι αριθμητικές πράξεις.



## Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον 8<sup>ο</sup> κεφάλαιο



### Σύνθετες λογικές συνθήκες

Μια *σύνθετη λογική συνθήκη* σχηματίζεται χρησιμοποιώντας απλές λογικές συνθήκες και τους λογικούς τελεστές που παρουσιάζονται στον ακόλουθο πίνακα, κατά φθίνουσα σειρά προτεραιότητας.

Λογικός Τελεστής	Σύμβολο
Άρνηση	όχι
Σύζευξη	και
Διάζευξη	ή

Αν A και B είναι δυο απλές λογικές συνθήκες που η κάθε μια από αυτές μπορεί να είναι **ΑΛΗΘΗΣ** ή **ΨΕΥΔΗΣ** τότε η δράση των λογικών τελεστών φαίνεται στον παρακάτω πίνακα:

A	B	A ή B	A και B	όχι A
Αληθής	Αληθής	Αληθής	Αληθής	Ψευδής
Αληθής	Ψευδής	Αληθής	Ψευδής	Ψευδής
Ψευδής	Αληθής	Αληθής	Ψευδής	Αληθής
Ψευδής	Ψευδής	Ψευδής	Ψευδής	Αληθής



Η ιεραρχία των λογικών τελεστών είναι μικρότερη από των αριθμητικών.

### Εντολή AN

Η εντολή **AN** υλοποιεί την δομή επιλογής και εμφανίζεται με τρεις διαφορετικές μορφές.

#### 1. **AN ... ΤΟΤΕ ... ΤΕΛΟΣ\_AN**

Η εντολή αυτή χρησιμοποιείται όταν μας ενδιαφέρει να εκτελεστούν κάποιες εντολές μόνο όταν ισχύει μια συνθήκη.

Η γενική μορφή της εντολής έχει ως εξής:

**AN συνθήκη ΤΟΤΕ**  
Εντολές  
**ΤΕΛΟΣ\_AN**

Η εντολή **AN** ελέγχει τη συνθήκη (η οποία μπορεί να είναι απλή ή σύνθετη) και αν αυτή ισχύει (αν είναι αληθής), τότε εκτελούνται οι εντολές που περιλαμβάνονται μεταξύ των δεσμευμένων λέξεων **ΤΟΤΕ** και **ΤΕΛΟΣ\_AN**. Αν η συνθήκη δεν ισχύει (αν είναι ψευδής) εκτελούνται οι εντολές που ακολουθούν το **ΤΕΛΟΣ\_AN**.

#### 2. **AN ... ΤΟΤΕ ... ΑΛΛΙΩΣ ... ΤΕΛΟΣ\_AN**



## Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον 8<sup>ο</sup> κεφάλαιο



Η εντολή αυτή χρησιμοποιείται όταν θέλουμε να εκτελεστούν κάποιες εντολές εφόσον ισχύει μια συνθήκη, και, ταυτόχρονα, θέλουμε να εκτελεστούν κάποιες άλλες συγκεκριμένες εντολές όταν δεν ισχύει η ίδια συνθήκη. Δηλαδή η εντολή αυτή καλύπτει την επιλογή μιας από δυο εναλλακτικές περιπτώσεις.

Η γενική μορφή της εντολής έχει ως εξής:

**ΑΝ συνθήκη ΤΟΤΕ**  
*Εντολής1*  
**ΑΛΛΙΩΣ**  
*Εντολής2*  
**ΤΕΛΟΣ\_ΑΝ**

Η εντολή **ΑΝ** ελέγχει τη συνθήκη (η οποία μπορεί αν είναι απλή ή σύνθετη) και αν αυτή ισχύει (αν είναι αληθής), τότε εκτελούνται οι εντολές που περιλαμβάνονται μεταξύ των δεσμευμένων λέξεων **ΤΟΤΕ** και **ΑΛΛΙΩΣ** (εντολής1). Στη συνέχεια εκτελούνται οι εντολές που ακολουθούν μετά το **ΤΕΛΟΣ\_ΑΝ**.

Αν η συνθήκη δεν ισχύει (αν είναι ψευδής) εκτελούνται οι εντολές που περιλαμβάνονται μεταξύ των δεσμευμένων λέξεων **ΑΛΛΙΩΣ** και **ΤΕΛΟΣ\_ΑΝ** (εντολής2). Στη συνέχεια εκτελούνται οι εντολές που ακολουθούν μετά το **ΤΕΛΟΣ\_ΑΝ**.

### 3. **ΑΝ ... ΤΟΤΕ ... ΑΛΛΙΩΣ\_ΑΝ ... ΤΕΛΟΣ\_ΑΝ**

Η εντολή αυτή χρησιμοποιείται όταν οι εναλλακτικές περιπτώσεις είναι περισσότερες από δυο.

Η γενική μορφή της εντολής έχει ως εξής:

**ΑΝ συνθήκη1 ΤΟΤΕ**  
*Εντολής1*  
**ΑΛΛΙΩΣ\_ΑΝ συνθήκη2**  
*Εντολής2*  
**ΑΛΛΙΩΣ**  
*Εντολής3*  
**ΤΕΛΟΣ\_ΑΝ**

Η εντολή **ΑΝ** ελέγχει τη συνθήκη1 (η οποία μπορεί αν είναι απλή ή σύνθετη) και αν αυτή ισχύει (αν είναι αληθής), τότε εκτελούνται οι εντολές που περιλαμβάνονται μεταξύ των δεσμευμένων λέξεων **ΤΟΤΕ** και **ΑΛΛΙΩΣ\_ΑΝ συνθήκη2** (εντολής1). Στη συνέχεια εκτελούνται οι εντολές που ακολουθούν μετά το **ΤΕΛΟΣ\_ΑΝ**.

Αν η συνθήκη1 δεν ισχύει (αν είναι ψευδής) ελέγχεται η συνθήκη2 και αν ισχύει εκτελούνται οι εντολές που περιλαμβάνονται μεταξύ των δεσμευμένων λέξεων **ΑΛΛΙΩΣ\_ΑΝ συνθήκη2** και **ΑΛΛΙΩΣ** (εντολής2). Στη συνέχεια εκτελούνται οι εντολές που ακολουθούν μετά το **ΤΕΛΟΣ\_ΑΝ**.

Αν η συνθήκη2 δεν ισχύει εκτελούνται οι εντολές που περιλαμβάνονται μεταξύ των δεσμευμένων λέξεων **ΑΛΛΙΩΣ** και **ΤΕΛΟΣ\_ΑΝ**.

Ας σημειωθεί ότι μπορεί να υπάρχουν περισσότερες εντολές **ΑΛΛΙΩΣ\_ΑΝ**, ο τρόπος αντιμετώπισης, όμως, είναι παρόμοιος.



## Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον 8<sup>ο</sup> κεφάλαιο



Εναλλακτικά αντί για την εντολή **ΑΝ ... ΤΟΤΕ ... ΑΛΛΙΩΣ\_ΑΝ ... ΤΕΛΟΣ\_ΑΝ** θα μπορούσαν να χρησιμοποιηθούν πολλές εντολές **ΑΝ** ή μια μέσα στην άλλη, οι *εμφωλευμένες εντολές ΑΝ* όπως ονομάζονται. Προτείνεται, ωστόσο, η χρήση εμφωλευμένων **ΑΝ** να αποφεύγεται καθώς οδηγεί συνήθως σε πολύπλοκες δομές που αυξάνουν την πιθανότητα λάθους και τη δυσκολία κατανόησης του προγράμματος. Η χρήση εμφωλευμένων **ΑΝ** μπορεί να αποφευχθεί και με τη χρήση σύνθετων λογικών συνθηκών.

### Εντολή ΕΠΙΛΕΞΕ

Η εντολή αυτή χρησιμοποιείται αν οι περιπτώσεις επιλογής είναι πολλές. Η γενική μορφή της είναι η εξής:

```
ΕΠΙΛΕΞΕ έκφραση
  ΠΕΡΙΠΤΩΣΗ λίστα_τιμών1
  Εντολές1
  ΠΕΡΙΠΤΩΣΗ λίστα_τιμών2
  Εντολές2
  .....
  ΠΕΡΙΠΤΩΣΗ ΑΛΛΙΩΣ
  Εντολές_αλλιώς
ΤΕΛΟΣ_ΕΠΙΛΟΓΩΝ
```

Αρχικά υπολογίζεται η τιμή της έκφρασης και στη συνέχεια ανάλογα με το σε ποια λίστα τιμών βρίσκεται η τιμή αυτή, εκτελούνται οι αντίστοιχες εντολές. Στην περίπτωση που η τιμή της έκφρασης δεν υπάρχει σε καμία λίστα τιμών, εκτελούνται οι *εντολές\_αλλιώς*.

Οι λίστες τιμών μπορεί να αποτελούνται από συγκεκριμένες τιμές χωρισμένες με κόμμα, ή περιοχές τιμών που δηλώνονται ως εξής: **από** αρχική\_τιμή **έως** τελική\_τιμή. Παραδείγματα λιστών τιμών είναι οι εξής:

- ΠΕΡΙΠΤΩΣΗ 1, 3, 5
- ΠΕΡΙΠΤΩΣΗ ΑΠΟ 6 ΕΩΣ 10



Η δομή **ΕΠΙΛΕΞΕ** πρέπει να είναι έτσι φτιαγμένη ώστε για κάθε τιμή της έκφρασης να υπάρχει μια μόνο ομάδα εντολών που μπορεί να εκτελεστεί.

### Εντολές επανάληψης

Πολλές φορές μια ακολουθία εντολών πρέπει να επαναληφθεί πολλές φορές. Για το λόγο αυτό χρησιμοποιούμε τρεις επαναληπτικές δομές.

#### 1. Επαναληπτική δομή **ΟΣΟ ... ΕΠΑΝΑΛΑΒΕ**

Η επαναληπτική αυτή δομή είναι η γενικότερη από όλες, με αποτέλεσμα οι άλλες δυο δομές να μπορούν να εκφραστούν με αυτήν. Η σύνταξη αυτής της δομής είναι η εξής:

**ΟΣΟ συνθήκη ΕΠΑΝΑΛΑΒΕ**



## Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον 8<sup>ο</sup> κεφάλαιο



Εντολές

### ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ

Τα βήματα εκτέλεσης της δομής αυτής είναι τα εξής:

1<sup>ο</sup> Βήμα. Ελέγχεται η τιμή της συνθήκης. Αν είναι αληθής, πηγαίνουμε στο 2<sup>ο</sup> βήμα, αλλιώς πηγαίνουμε στο 1<sup>ο</sup> βήμα .

2<sup>ο</sup> Βήμα. Εκτελούνται όλες οι εντολές της ομάδας εντολών και στη συνέχεια πηγαίνουμε στο 1<sup>ο</sup> βήμα.

3<sup>ο</sup> Βήμα. Βγαίνουμε από την επαναληπτική δομή και εκτελείται η αμέσως επόμενη εντολή μετά το **ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**.



Εφόσον μετά από κάθε επανάληψη ελέγχεται εκ νέου η συνθήκη, πρέπει υποχρεωτικά μέσα στο βρόχο να υπάρχει μια εντολή, η οποία να μεταβάλλει την τιμή της μεταβλητής που ελέγχεται με τη συνθήκη. Διαφορετικά η επανάληψη δε θα τερματίζεται (ατέρμων βρόχος).

Η επαναληπτική δομή **ΟΣΟ ... ΕΠΑΝΑΛΑΒΕ** μπορεί να μην εκτελεστεί ούτε μια φορά εφόσον η συνθήκη είναι εξαρχής ψευδής.

### 2. Επαναληπτική δομή **ΑΡΧΗ\_ΕΠΑΝΑΛΗΨΗΣ ... ΜΕΧΡΙΣ\_ΟΤΟΥ**.

Η σύνταξη αυτής της δομής είναι η εξής:

#### ΑΡΧΗ\_ΕΠΑΝΑΛΗΨΗΣ

Εντολές

#### ΜΕΧΡΙΣ\_ΟΤΟΥ συνθήκη

Τα βήματα εκτέλεσης αυτής της δομής είναι τα εξής:

1<sup>ο</sup> βήμα. Εκτελούνται όλες οι εντολές της ομάδας εντολών

2<sup>ο</sup> βήμα. Ελέγχεται η συνθήκη τερματισμού. Αν είναι ψευδής πηγαίνουμε στο 1<sup>ο</sup> βήμα, ενώ αν είναι αληθής πηγαίνουμε στο 3<sup>ο</sup> βήμα.

3<sup>ο</sup> βήμα. Βγαίνουμε από την επαναληπτική δομή και εκτελείται η αμέσως επόμενη εντολή με το **ΜΕΧΡΙΣ\_ΟΤΟΥ συνθήκη**.

Η επαναληπτική δομή **ΑΡΧΗ\_ΕΠΑΝΑΛΗΨΗΣ ... ΜΕΧΡΙΣ\_ΟΤΟΥ** εκτελείται υποχρεωτικά τουλάχιστον μια φορά.

### 3. Επαναληπτική δομή **ΓΙΑ ... ΑΠΟ ... ΜΕΧΡΙ**

Η δομή αυτή χρησιμοποιείται όταν είναι γνωστός από την αρχή ο αριθμός των επαναλήψεων. Η σύνταξή της γίνεται ως εξής:

**ΓΙΑ** μετρητής **ΑΠΟ** αρχική\_τιμή **ΜΕΧΡΙ** τελική\_τιμή **ΜΕ\_ΒΗΜΑ** βήμα

Εντολές

#### ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ



## Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον 8<sup>ο</sup> κεφάλαιο



Τα βήματα εκτέλεσης αυτή της δομής είναι τα εξής:

1<sup>ο</sup> βήμα. Η μεταβλητή *μετρητής* παίρνει την τιμή *αρχική\_τιμή*.

2<sup>ο</sup> βήμα. Ελέγχεται η συνθήκη *μετρητής*  $\leq$  *τελική\_τιμή* αν η μεταβλητή *βήμα* είναι θετικός αριθμός, ή ελέγχεται η συνθήκη *μετρητής*  $\geq$  *τελική\_τιμή* αν η μεταβλητή *βήμα* είναι αρνητικός αριθμός. Αν η συνθήκη είναι αληθής πηγαίνουμε στο 3<sup>ο</sup> βήμα, ενώ αν είναι ψευδής πηγαίνουμε στο 5<sup>ο</sup> βήμα.

3<sup>ο</sup> βήμα. Εκτελούνται όλες οι εντολές της ομάδας εντολών

4<sup>ο</sup> βήμα. Ο *μετρητής* αυξάνει κατά ποσότητα ίση με το *βήμα*. Επιστρέφουμε στο 2<sup>ο</sup> βήμα.

5<sup>ο</sup> βήμα. Βγαίνουμε από την επαναληπτική δομή και εκτελείται η αμέσως επόμενη εντολή μετά το **ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**.



Αν η τιμή της μεταβλητής *βήμα* ισούται με ένα, τότε μπορούμε να παραλείψουμε το τμήμα της εντολής **ΜΕ\_ΒΗΜΑ βήμα**.

### Εμφωλευμένοι βρόχοι

Εμφωλευμένους βρόχους έχουμε όταν ένας βρόχος βρίσκεται μέσα σε άλλον, για παράδειγμα όταν επεξεργαζόμαστε πίνακες δυο διαστάσεων. Για τη σωστή λειτουργία των προγραμμάτων πρέπει να ακολουθούνται οι παρακάτω κανόνες:

- Ο εσωτερικός βρόχος πρέπει να βρίσκεται ολόκληρος μέσα στον εξωτερικό. Ο βρόχος που ξεκινάει τελευταίος πρέπει να ολοκληρώνεται πρώτος.
- Η είσοδος σε κάθε βρόχο γίνεται υποχρεωτικά από την αρχή του.
- Δεν μπορεί να χρησιμοποιηθεί η ίδια μεταβλητή ως μετρητής δυο ή περισσότερων εμφωλευμένων βρόχων.