



C: Control structures

Vasilios Papaliakos, <vpapaliakos@gmail.com>

Hit the space bar or swipe left for next slide

Σκοποί μαθήματος

Στο τέλος του μαθήματος, θα μπορείτε:

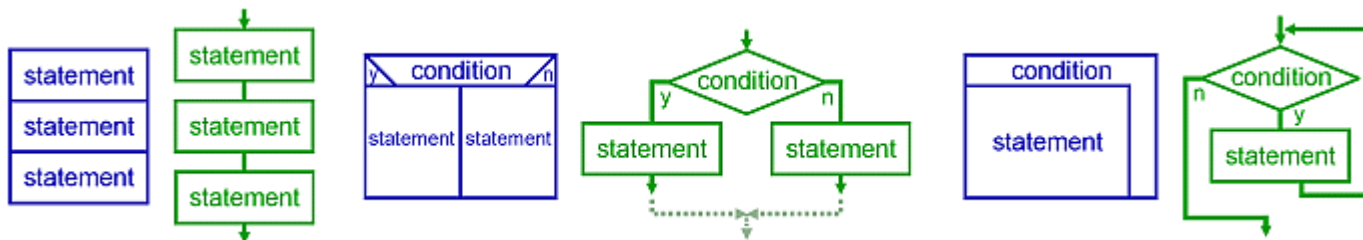
- να εξηγείτε την έννοια του δομημένου προγραμματισμού
- να ξεχωρίζετε τις 3 βασικές δομές
- να χρησιμοποιείτε τις δομές ελέγχου και επανάληψης

Δομές ελέγχου - Δομημένος προγραμματισμός

- Δομές ελέγχου ροής προγράμματος
 - Οι εντολές σε ένα πρόγραμμα εκτελούνται κατά βάση “σειριακά”, δηλαδή με τη σειρά που είναι γραμμένες.
 - Υπάρχουν όμως περιπτώσεις που απαιτούν διαφορετικά.
 - Τότε έχουμε μια “μεταφορά του ελέγχου” (της ροής του προγράμματος), δηλαδή αλλάζουμε τη σειρά και ορίζουμε την επόμενη εντολή με άλλο τρόπο
- Δομημένος Προγραμματισμός (Structured Programming)

Κάθε πρόγραμμα φτιάχνεται με το **συνδυασμό τριών δομών ελέγχου** (control structures):

- Τη “σειριακή δομή” (*sequence*)
- Τη “δομή επιλογής” (*selection*)
- Τη “δομή επανάληψης” (*repetition*)



Δομή επιλογής (1/2): Εντολή if

- Βασική σύνταξη:

```
if (έκφραση)
    εντολή x (ή block εντολών)
else
    εντολή y (ή block εντολών)
```

- Παραδείγματα:

```
/* simple if */
if (age >= 18)
    printf("You can vote.");
```

```
/* if - else */
if (age >= 18)
    printf("You can vote.");
else /* age < 18 */
    printf("You cannot vote.");
```

```
/* with blocks */
bool can_vote = false;

if (age >= 18) {
    printf("You can vote.");
    can_vote = true;
} else {
    printf("You cannot vote.");
    can_vote = false;
}
```

```
/* else if */
if (grade >= 90)
    printf("excellent");
else if (grade >= 75)
    printf("good");
else if (grade >= 50)
    printf("pass");
else
```

```
printf("fail");
```

- Σημεία προσοχής:
 - μετά τα *if* και *else*, είτε μία μονή εντολή ή ένα *block* εντολών {}
 - στο *if* και στο *else* δεν υπάρχει ; στο τέλος
 - οι εσοχές στον κώδικα βοηθούν στην αντίληψη / επίλυση λαθών
- Συντετμημένη σύνταξη:
 - Χρήση του τριαδικού τελεστή υπό συνθήκη:

```
(συνθήκη)?εντολή_αν_αληθής:εντολή_αν_ψευδής;
```

- Παράδειγμα:

```
max = (a > b)? a : b;
```

Δομή επιλογής (2/2): Εντολή switch

- Βασική σύνταξη:

```
switch (έκφραση) /* int ή char μόνο - όχι float */
{
    case σταθερή_τιμή_1: εντολή_1 (ή block εντολών
                        break;
    case σταθερή_τιμή_2: εντολή_2 (ή block εντολών
                        break;
    ...
    default:           εντολή_n (ή block εντολών)
                        break;
}
```

- Παράδειγμα:

```
int x = 2;

switch (x)
{
    case 1: printf("Choice is 1");
            break;
    case 2: printf("Choice is 2");
            break;
    case 3: printf("Choice is 3");
            break;
    default: printf("Choice other than 1, 2 and 3");
             break;
}
```

- Σημεία προσοχής:
 - στο switch δεν υπάρχει ; στο τέλος

- η έκφραση (συνθήκη ελέγχου) είναι απλή: αφορά μία μόνο **διακριτή** μεταβλητή
- η εντολή *break* είναι προαιρετική
- όσο δεν υπάρχει *break*, οι εντολές που ακολουθούν θα εκτελεστούν και αυτές
- οι τιμές σύγκρισης στις εντολές *case* θα πρέπει να είναι σταθερές - όχι μεταβλητές
- με τη *switch* συγκρίνουμε περιπτώσεις **ισότητας** διακριτών τιμών μόνο

Δομή επανάληψης (1/3): Εντολή while

- Βασική σύνταξη:

```
while (έκφραση)  
    εντολή x (ή block εντολών)
```

- Παράδειγμα:

```
int i = 0;  
  
while (i < 10) {  
    i++;  
    printf("%5d", i);  
}
```

Output:

```
1 2 3 4 5 6 7 8 9 10
```

- Σημεία προσοχής:
 - στο *while* δεν υπάρχει ; στο τέλος
 - μια εντολή *while* μπορεί να μην εκτελεστεί ποτέ (αν η συνθήκη είναι από την αρχή ψευδής)
 - ή μπορεί να εκτελείται για πάντα (ατέρμονας βρόγχος)
 - η εντολή *break* μπορεί να χρησιμοποιηθεί (πχ. με ένα *if*)

Δομή επανάληψης (2/3): Εντολή do-while

- Βασική σύνταξη:

```
do
    εντολή x (ή block εντολών)
while (έκφραση);
```

- Παράδειγμα:

```
int choice = 0;

printf("Menu\n");
printf("1. Continue\n");
printf("2. Exit\n");
do {
    printf("Your choice: ");
    scanf("%d", &choice);
} while(choice != 1 && choice != 2);
```

- Σημεία προσοχής:
 - στο *do* δεν υπάρχει ; στο τέλος, αλλά στο *while* υπάρχει!
 - μια εντολή *do-while* θα εκτελεστεί **τουλάχιστον μία φορά**

Δομή επανάληψης (3/3): Εντολή for

- Βασική σύνταξη:

```
for (ανάθεση_αρχικής_τιμής; συνθήκη_ελέγχου; βήμα_μεταβολής_τιμής)  
    εντολή x (ή block εντολών)
```

- Παράδειγμα:

```
int i;  
  
for (i = 1; i <= 10; i++) {  
    printf("%5d", i);  
    /* more commands */  
}
```

Output:

```
1 2 3 4 5 6 7 8 9 10
```

- Σημεία προσοχής:
 - στο for δεν υπάρχει ; στο τέλος
 - οι 3 εκφράσεις μέσα στην παρένθεση (ανάθεση_αρχικής_τιμής; συνθήκη_ελέγχου; βήμα_μεταβολής_τιμής) είναι προαιρετικές (πχ. η εντολή "for (; ;)"θα δημιουργήσει έναν ατέρμονα βρόγχο)

Σύνθετοι βρόγχοι και ειδικές εντολές

- Σύνθετοι βρόγχοι:

- Σε πολλές περιπτώσεις, απαιτείται η δημιουργία πιο σύνθετων δομών που περιλαμβάνονται μέσα σε άλλες δομές
- Παράδειγμα:

```
int i, j;
for (i = 0; i < 5; i++) {
    for (j = 0; j < 5; j++) {
        if (i == j)
            printf(" 1");
        else
            printf(" 0");
    }
    printf("\n"); /* add a new line for every new i */
}
```

Output:

```
1 0 0 0 0
0 1 0 0 0
0 0 1 0 0
0 0 0 1 0
0 0 0 0 1
```

- Ειδικές εντολές:

break

- η εντολή μπορεί να χρησιμοποιηθεί για να σταματήσει ένας βρόγχος επανάληψης
- χρησιμοποιείται επίσης στη δομή select-case

continue

- η εντολή μπορεί να χρησιμοποιηθεί για να παραληφθεί η τρέχουσα επανάληψη σε ένα βρόγχο
- το πρόγραμμα θα συνεχίσει με την επόμενη επανάληψη
- σχεδόν πάντα χρησιμοποιείται σε συνδυασμό με ένα *if*