

Εντολές Επιλογής και Επανάληψης

Σε κάθε πρόγραμμα είναι απαραίτητο να ελέγχουμε τη ροή του αναλόγως κάποιες συνθήκες και να την ανακατευθύνουμε κατάλληλα. Υπάρχουν πολλοί τρόποι να κατευθύνουμε τη ροή του προγράμματος και ο κάθε ένας εξυπηρετεί συγκεκριμένο σκοπό.

Η εντολή if

Η εντολή if χρησιμοποιείται όταν θέλουμε να εκτελέσουμε κάποιες εντολές μόνο όταν ικανοποιείται κάποια συνθήκη:

```
if (συνθήκη)
```

```
    { εντολές; }
```

```
Else
```

```
    { εντολές; }
```

Μπορούν να συνδυαστούν πολλές εντολές if μεταξύ τους, όπως στο ακόλουθο παράδειγμα:

```
if (x == 1)
```

```
    { System.out.println("x is one."); }
```

```
else if (x == 2)
```

```
    { System.out.println("x is two."); }
```

```
else if (x == 3)
```

```
    { System.out.println("x is three."); }
```

```
else if (x == 4)
```

```
    { System.out.println("x is four."); }
```

```
else
```

```
    { System.out.println("x is not between 1-4."); }
```

Η εντολή switch

Η εντολή switch χρησιμοποιείται όταν έχουμε πολλαπλές επιλογές ή τιμές για μια μεταβλητή και θέλουμε να εκτελεστούν διαφορετικές εντολές για κάθε τιμή.

Με την switch, το προηγούμενο παράδειγμα θα πάρει τη μορφή:

```
switch (x) {  
  case 1:  
    System.out.println("x is one.");  
    break;  
  case 2:  
    System.out.println("x is two.");  
    break;  
  case 3:  
    System.out.println("x is three.");  
    break;  
  case 4:  
    System.out.println("x is four.");  
    break;  
  default:  
    System.out.println("x is not between 1-4.");  
    break; }  
}
```

Προσέξτε την χρήση της `break`. Χωρίς την `break` η εκτέλεση του προγράμματος θα συνεχίσει μέχρι την επόμενη `break` ή μέχρι το τέλος της εντολής `switch`.

Αυτό μπορούμε να το εκμεταλευτούμε αν θέλουμε κοινή αντιμετώπιση ορισμένων περιπτώσεων.

```
switch (x) {  
  case 1;  
    System.out.println("x is one.");  
  case 2;  
    System.out.println("x is two.");  
  case 3;  
    System.out.println("x is three.");  
}
```

```
case 4;
System.out.println("x is four.");
System.out.println("x is between one and four.");
break;
case 5;
System.out.println("x is five.");
case 6;
System.out.println("x is six.");
case 7;
System.out.println("x is seven.");
case 8;
System.out.println("x is eight.");
System.out.println("x is between five and eight.");
break;
default:
System.out.println("x is not between one and eight.");
break; }
```

Η εντολή for

Η εντολή for χρησιμοποιείται για επανάληψη (loop) υπό συνθήκη κάποιων εντολών.

Θεωρείται ίσως η πιο δυνατή και ευέλικτη εντολή τύπου for, σε οποιαδήποτε γλώσσα (έχει ακριβώς την ίδια λειτουργία με την αντίστοιχη της C/C++).

Η σύνταξή της είναι η εξής:

```
for (εντολή αρχικοποίηση; συνθήκη; εντολή επανάληψης)
{ εντολές; }
```

Η εντολή αρχικοποίησης (initialization) εκτελείται στην αρχή του loop και στη συνέχεια ελέγχεται αν είναι αληθής ή ψευδής η συνθήκη (condition). Αν είναι αληθής, εκτελούνται οι εντολές και στο τέλος εκτελείται η εντολή επανάληψης (iteration) και ελέγχεται και πάλι αν είναι αληθής η συνθήκη. Αν είναι αληθής εκτελούνται οι εντολές, κ.ο.κ.

Παράδειγμα:

```
for (int i = 1; i < 20; i += 3)
{ System.out.println(i); }
```

Το αποτέλεσμα θα είναι το εξής:

```
1
4
7
10
13
16
19
```

Κάθε μία από τις εντολές αρχικοποίησης, επανάληψης και η συνθήκη μπορεί να παραλειφθεί αλλάζοντας τη συμπεριφορά της εντολής for. Σημειώνουμε ότι χωρίζονται μεταξύ τους με το ερωτηματικό ';'.

Παραδείγματα:

```
int x = 10;

for (; x < 5; x++)

{ System.out.println(x); }

double y = 20000;

// (y = pi)

for (; y >= 10.0;)

{

// υπολογίζει την τετραγωνική του y και τοποθετεί

// το αποτέλεσμα πάλι στο y.

// Το for loop θα εκτελείται όσο το y είναι μεγαλύτερο από 10.0

System.out.println(y);

y = Math.sqrt(y);

}
```

```
for (;;)
{ // infinite loop wait_for_signal(); }
```

Η εντολή while

Η εντολή while χρησιμοποιείται αντί της for όταν δε μπορούμε να προβλέψουμε εύκολα πόσες φορές θέλουμε να εκτελεστούν οι εντολές, ή όταν δεν έχει σημασία ο αριθμός των επαναλήψεων αλλά η ικανοποίηση ή όχι της συνθήκης:

```
while (συνθήκη)
{ εντολές; }
```

Παράδειγμα:

```
bool exit_from_loop = false;
while (exit_from_loop = false)
{
// υποθετική ρουτίνα που διαβάζει δεδομένα από ένα αρχείο
read_bytes(file1);
// άλλη ρουτίνα που γράφει δεδομένα σε ένα άλλο αρχείο
write_bytes(file2);
if (file_empty(file1) == true)
exit_from_loop = true;
}
```

Η εντολή do/while

Η εντολή do/while είναι παρόμοια με την while, με μία διαφορά, οι εντολές εντός του loop θα εκτελεστούν τουλάχιστον μια φορά ανεξαρτήτως αν είναι αληθής η συνθήκη ή όχι.

```
do {
εντολές;
} while (συνθήκη);
```

Παράδειγμα:

```
int x = 10;
```

```
do {
```

```
System.out.println(x);
```

```
x++;
```

```
} while (x < 10)
```