2. Βασικά Στοιχεία της Γλώσσας

2.2. Η δομή επιλογής if … else
2.3. Οι δομές επανάληψης for και while

2.2. Η δομή επιλογής if … else

Η δομή επιλογής ή εντολή διακλάδωσης στην Java δεν διαφέρει ιδιαίτερα από τις άλλες γλώσσες προγραμματισμού. Η εντολή if χρησιμοποιείται όταν θέλουμε να εκτελέσουμε ορισμένες εντολές ανάλογα με την ικανοποίηση ή όχι κάποιας συνθήκης. Η σύνταξή της είναι η εξής:

```
if (συνθήκη)
{
Εντολές
}
else
{
Εντολές
}
```

Όταν έχουμε πολλές περιπτώσεις μπορούμε να συνδυάσουμε πολλές εντολές if μεταξύ τους όπως φαίνεται παρακάτω:

```
if (number < 0)
   digits = -1;
else if (number < 10)
   digits = 1;
else if (number < 100)
   digits = 2;
else if (number < 1000)
   digits = 3;
else
   digits = 4;
```

Επίσης μπορούμε να έχουμε εμφωλευμένες δομές επιλογής όπως φαίνεται στο παρακάτω παράδειγμα:

```
if (a != 0) {
    if (b > 0) {
        if (c < 0) {
            f = 1;
            }
        else {
            f = 2;
            if (b == 100)
                g = 9;
        }
    }
}
```

2.3. Οι δομές επανάληψης for και while

Στην Java έχουμε τις δομές επανάληψης for και while, οι οποίες έχουν την εξής σύνταξη:

```
<αρχικοποίηση>
while ( συνθήκη ) {
`<εντολές>
`<εντολές_βήματος>
}
```

```
for ( αρχικοποίηση; συνθήκη; εντολές_βήματος ) {
   <εντολές>
}
```

Οι παραπάνω δομές είναι απολύτως ισοδύναμες.

Στο παρακάτω παράδειγμα υπολογίζουμε το άθροισμα όλων των αριθμών από το 1 έως και το 1000.

```
sum = 0;
for (i = 1; i <= 1000; i=i+1 ) {
    sum = sum + i;
}
```

Ο αντίστοιχος αλγόριθμος στην ψευδογλώσσα του μαθήματος “Εισαγωγή στην Επιστήμη των Υπολογιστών” της Β’ Λυκείου:

sum ← 0
i ← 1
Όσο i <= 1000 **Επανάλαβε**
    sum = sum + i;
    i = i + 1;
Τέλος Επανάληψης

Στην Java αντί να γράφουμε την εντολή i = i + 1 έχει επικρατήσει ο γνωστός τελεστής μεταύξησης **i++** , ομοίως υπάρχει και αντίστοιχος τελεστής μείωσης κατά 1, **i--.**

Εδώ θα πρέπει να εξηγήσουμε τη διαφορά που παρουσιάζει στο αποτέλεσμα η χρήση των διπλών συμβόλων αύξησης/μείωσης αριθμών (++ / --) όταν τοποθετούνται πριν από τη μεταβλητή και μετά από τη μεταβλητή.

Όταν τοποθετούνται πριν από τη μεταβλητή γίνεται πρώτα η αύξηση (ή μείωση αντίστοιχα) της τιμής της και μετά εκτελείται η εντολή που την περιέχει,λαμβάνοντας υπόψη τη νέα της τιμή.

Αντίθετα, όταν τοποθετούνται μετά από τη μεταβλητή εκτελείται πρώτα η εντολή που την περιέχει και μετά γίνεται η αύξηση (ή μείωση αντίστοιχα) της τιμής της.

Ας δούμε ένα πιο συγκεκριμένο παράδειγμα για να καταλάβουμε τη διαφορά τους:

```
int var=3; //Ορίζουμε μια μεταβλητή var με αρχική τιμή 3
System.out.println(++var); // αύξησε την var κατά 1 και εμφάνισε την τιμή της var
System.out.println(var); // εμφάνισε την τιμή της var
```

Το αποτέλεσμα που θα εμφανιστεί στην οθόνη μας είναι ο αριθμός 4 δύο φορές. Αυτό οφείλεται στο γεγονός ότι στην εντολή System.out.println(++var); η μεταβλητή var πρώτα αυξάνει την τιμή της από 3 σε 4 και μετά εκτελείται η εντολή εκτύπωσης

Αντίθετα, με τη χρήση του τελεστή μετά τη μεταβλητή θα έχουμε:

```
int var=3;
System.out.println(var++); // εμφάνισε και μετά αύξησε την var κατά 1
System.out.println(var); // εμφάνισε την τιμή της var
```

Το αποτέλεσμα που θα εμφανιστεί στην οθόνη μας είναι ο αριθμός 3 και μετά ο αριθμός 4. Αυτό οφείλεται στο γεγονός ότι, στην εντολή System.out.println(var++); πρώτα εκτελείται η εντολή εκτύπωσης και εμφανίζεται στην οθόνη η τρέχουσα τιμή της μεταβλητής var, δηλαδή ο αριθμός 3 και μετά η μεταβλητή var αυξάνει την τιμή της από 3 σε 4. Στην αμέσως επόμενη γραμμή ζητάμε να δούμε την τρέχουσα τιμή της μεταβλητής που φυσικά τώρα είναι ο αριθμός 4