

## Πίνακες

Αν και έχουμε ήδη αναφέρει τους σημαντικότερους τύπους δεδομένων στη Java σε προηγούμενη παράγραφο, αφήσαμε σκόπιμα την αναφορά στους πίνακες. Ένας λόγος είναι ότι σε αντίθεση με τη C/C++, οι πίνακες στη Java υλοποιούνται ως αντικείμενα.

Αυτό έχει πολλά οφέλη, όπως π.χ. το ότι δεν χρειάζεται να αποδεσμεύουμε τη μνήμη που χρησιμοποιεί ένας πίνακας (την αποδέσμευση την αναλαμβάνει το σύστημα περισυλλογής σκουπιδιών της Java), ή το ότι είναι μεταβλητού μεγέθους από κατασκευής (κάτι που δεν ισχύει για τη C/C++).

Οι πίνακες γενικά χρησιμοποιούνται για την οργάνωση και καταχώρηση όμοιων αντικειμένων. Μπορούν να είναι μίας ή πολλών διαστάσεων και ο τρόπος προσπέλασης των στοιχείων είναι παρόμοιος με αυτόν της C/C++.

Μπορούμε να ορίσουμε ένα μονοδιάστατο πίνακα πολύ απλά ως εξής:

```
type table[];
```

Ενώ αν θέλουμε να τον δημιουργήσουμε κιάλας, χρησιμοποιούμε την new:

```
table = new type[size];
```

Ή μπορούμε να πραγματοποιήσουμε και τα δύο στάδια με μία εντολή:

```
type table[] = new type[size];
```

Ως type θεωρούμε τον τύπο δεδομένων των αντικειμένων του πίνακα και μπορεί να είναι είτε ένας από τους απλούς τύπους (bool, byte, short, int, long, float, double, char, String) είτε το όνομα μιας κλάσης αντικειμένων (της Java ή δική μας). Το size απεικονίζει το μέγεθος του πίνακα table.

Όπως και στη C/C++, μπορούμε να προσπελλάσουμε τα στοιχεία του πίνακα με την σύνταξη table[i], όπου i η θέση του στοιχείου που μας ενδιαφέρει. Εδώ πρέπει να σημειωθεί ότι σε αντιστοιχία με τη C/C++ η Java πραγματοποιεί την αρίθμηση των πινάκων από το μηδέν (0) ως το size-1.

Δηλαδή αν έχουμε έναν πίνακα A με 10 στοιχεία το πρώτο στοιχείο είναι το A[0] και το τελευταίο το A[9].

Ακολουθεί ένα παράδειγμα για καλύτερη κατανόηση:

```
int data[] = new int[10];
```

```
int i;
```

```
System.out.println("Size of array data: " + data.length);
```

```
for (i = 0; i < data.length; i++)
```

```
{
```

```
data[i] = i*i;
System.out.println("data[" + i + "] = " + data[i]);
}
```

Το αποτέλεσμα του κώδικα αυτού θα είναι:

Size of array data: 10

data[0] = 0

data[1] = 1

data[2] = 4

data[3] = 9

data[4] = 16

data[5] = 25

data[6] = 36

data[7] = 49

data[8] = 64

data[9] = 81

Στο συγκεκριμένο παράδειγμα χρησιμοποιήσαμε τη μέθοδο `length` που επιστρέφει το τρέχον μέγεθος του πίνακα. Σε περίπτωση που θέλουμε να αρχικοποιήσουμε ένα πίνακα, δηλαδή να ορίσουμε αρχικές τιμές για τα στοιχεία του, αυτό μπορούμε να το πετύχουμε με την εξής συντακτική δομή:

```
int dataset[] = { 22, 3, 54, 43, 199, 20, 20, 67, 7, 80 };
```

Αξίζει να σημειωθεί ότι δεν είναι απαραίτητη η χρήση της `new` κατά την δήλωση του πίνακα `dataset`. Εννοείται η χρήση της στη δημιουργία του αντικειμένου του δεξιού μέρους της ισότητας.

## Πολυδιάστατοι πίνακες

Δε θα ήταν πλήρης η υποστήριξη των πινάκων σε μια γλώσσα προγραμματισμού όπως η Java, αν αυτή δεν υποστήριζε πολυδιάστατους πίνακες.

Για παράδειγμα, ένας δισδιάστατος πίνακας μπορεί να δηλωθεί ως εξής:

```
int twodim[][] = new int[4][4];
```

```
int i, j, counter = 1;
```

```
for (i = 0; i < twodim.length; i++)  
{  
  for (j = 0; j < twodim[i].length; j++)  
  {  
    twodim[i][j] = counter;  
    counter++;  
    System.out.print(twodim[i][j] + " ");  
  }  
  System.out.println();  
}
```

Το αποτέλεσμα αυτού του κώδικα θα είναι:

1 2 3 4

5 6 7 8

9 10 11 12

13 14 15 16