

ΠΙΝΑΚΕΣ

Δημήτρης Κ. Ιακωβίδης

Στόχος αυτής της ενότητας είναι η εισαγωγή στις έννοιες που αφορούν την κατασκευή και τη χρήση μονοδιάστατων, διδιάστατων ή πολυδιάστατων πινάκων. Περιγράφονται οι ιδιαιτερότητες που υπεισέρχονται στην κατασκευή πινάκων αντικειμένων σε σχέση με την κατασκευή πινάκων πρωταρχικών τύπων, και η χρήση πινάκων σε μεθόδους κλάσεων. Εξηγείται ο μηχανισμός με τον οποίο γίνεται η εισαγωγή και η διαχείριση των ορισμάτων των μεθόδων με αναφορά και με τιμή.

6.1 Ορισμός μονοδιάστατων πινάκων

Ένας πίνακας (array) είναι ένα πλήθος στοιχείων π.χ. ακεραίων τιμών, που υπακούουν σε μια διάταξη. Μια μεταβλητή τύπου πίνακα έχει διαφορετικές θέσεις για την αποθήκευση αυτών των στοιχείων. Για κάθε θέση του πίνακα υπάρχει ένας δείκτης (index), μέσω του οποίου μπορούμε να αναφερθούμε στο αντίστοιχο στοιχείο του πίνακα. Έστω ότι επιθυμούμε να κατασκευάσουμε έναν πίνακα τριών (3) ακεραίων, τότε γράφουμε:

```
int[] a = new int[3]; (1)
```

Αυτή η έκφραση ισοδυναμεί με δύο εκφράσεις:

```
int[] a; (2)
```

που είναι η δήλωση της μεταβλητής *a* ως μεταβλητής τύπου (μονοδιάστατου) πίνακα, και

```
a = new int[3]; (3)
```

που πραγματοποιεί τη δημιουργία του πίνακα των τριών στοιχείων, δηλαδή δεσμεύει μνήμη για τους τρεις ακεραίους. Για να δώσουμε τιμές στο πρώτο, στο δεύτερο και στο τρίτο στοιχείο του αρκεί να γράψουμε αντίστοιχα:

```
a[0] = 5;
```

```
a[1] = 8; (4)
```

```
a[2] = 12;
```

Δηλαδή, το πρώτο στοιχείο του πίνακα αυτού έχει δείκτη θέσης 0 και το τελευταίο στοιχείο του έχει δείκτη 2 (δηλαδή μήκος πίνακα μείον 1).

Εναλλακτικά, αν επιθυμούμε να κάνουμε όλα τα παραπάνω γρήγορα, την πρώτη φορά που δηλώνουμε έναν πίνακα (αρχικοποίηση), μπορούμε να γράψουμε:

```
int[] a = {5, 8, 12};
```

 (5)

Στη γλώσσα Java, οι πίνακες είναι αντικείμενα μιας (υποθετικής) κλάσης πινάκων. Έτσι, έχουν και αυτοί μια πολύ χρήσιμη μεταβλητή υπόστασης `length` που μας δίνει το μήκος τους ως ακέραια τιμή. Για παράδειγμα η έκφραση,

```
int x = a.length;
```

 (6)

αναθέτει στη μεταβλητή `x` την τιμή 3.

Για να διατρέξουμε όλα ή μέρος των στοιχείων ενός πίνακα χρησιμοποιούμε βρόχους, όπως είναι ο βρόχος `for`. Στο Σχ. 1 εικονίζεται ένα παράδειγμα με το οποίο γεμίζουμε τα στοιχεία ενός πίνακα με τιμές από το 1 έως το 10.

```
1 class Array1D
2 {
3     public static void main(String args[])
4     {
5         // Create the array
6         int[] a = new int[10];
7
8         // Fill in the elements of the array
9         for (int i = 0; i < a.length; i++)
10        {
11            a[i] = i + 1;
12        }
13
14        // Print all the elements of the array
15        for (int i = 0; i < a.length; i++)
16        {
17            System.out.println(a[i]);
18        }
19    }
20 }
```

Σχήμα 1. Δημιουργία, γέμισμα και εκτύπωση μονοδιάστατου πίνακα.

6.2 Ορισμός διδιάστατων ή πολυδιάστατων πινάκων

Ένας πίνακας μπορεί να έχει περισσότερες από μια διαστάσεις, π.χ. ο παρακάτω πίνακας είναι ένας διδιάστατος πίνακας (όπως εκείνοι που γνωρίζουμε από τα μαθηματικά):

$$a = \begin{pmatrix} 5 & 2 \\ 3 & 4 \\ 1 & 8 \end{pmatrix}$$

Ένας τέτοιος πίνακας, 3 γραμμών (rows) και 2 στηλών (columns) δηλώνεται ως:

int[][] a; (7)

και δημιουργείται ως:

a = new int[3][2]; (8)

ενώ τα στοιχεία του λαμβάνουν τιμές

a[0][0] = 5;

a[0][1] = 2; (9)

...

a[2][1] = 8;

Εναλλακτικά, μπορεί να αρχικοποιηθεί με τις ίδιες τιμές χρησιμοποιώντας την έκφραση:

int[][] a = { {5, 2}, {3, 4}, {1, 8} }; (10)

Παρατηρείστε πως η παραπάνω έκφραση μοιάζει σα να δηλώνουμε έναν μονοδιάστατο πίνακα αποτελούμενο από στοιχεία {5, 2}, {3, 4} και {1, 8}, δηλαδή μικρούς μονοδιάστατους. Αυτή είναι και η πραγματικότητα!

Η έκφραση `a[0]` αναπαριστά την πρώτη γραμμή του πίνακα, η έκφραση `a[1]` τη δεύτερη κ.ο.κ.

Αν επιθυμούμε να διαβάσουμε το μέγεθος ενός διδιάστατου πίνακα αρκεί να χρησιμοποιήσουμε την έκφραση:

a.length (11)

για τις γραμμές, και μια από τις εκφράσεις:

a[0].length; ή a[1].length; ή a[2].length; (12)

για τις στήλες του πίνακα (αφού κάθε γραμμή του πίνακα στην περίπτωση αυτή έχει το ίδιο μήκος).

Με παρόμοιο τρόπο μπορούμε να δημιουργήσουμε και να χειριστούμε και πίνακες μεγαλύτερης διάστασης π.χ. τρισδιάστατους:

`int[][][] a = new int[4][3][2];` (13)

Ένας τέτοιος πίνακας μπορεί να θεωρηθεί ως ένας μονοδιάστατος πίνακας τεσσάρων στοιχείων, όπου κάθε στοιχείο του είναι ένας δισδιάστατος πίνακας τριών γραμμών και δύο στηλών.

Στο Σχ. 2 εικονίζεται ένα παράδειγμα, αντίστοιχο του Σχ. 1, που γεμίζει με αυξανόμενες τιμές τα στοιχεία ενός διδιάστατου πίνακα τριών γραμμών και δύο στηλών. Παρατηρείστε ότι για να διατρέξουμε το διδιάστατο πίνακα χρησιμοποιούμε δύο βρόχους εμφωλιασμένους μεταξύ τους, όπου ο εξωτερικός διατρέχει τις γραμμές και ο εσωτερικός της στήλης. Η μεταβλητή *c* είναι βοηθητική για την παραγωγή των αυξανόμενων τιμών.

```
1  class Array2D
2  {
3      public static void main(String args[])
4      {
5          // Create the array
6          int[][] a = new int[3][2];
7
8          // Fill in the elements of the array
9          int c = 1;
10         for (int i = 0; i < a.length; i++)
11         {
12             for (int j = 0; j < a[i].length; j++)
13             {
14                 a[i][j] = c;
15                 c++;
16             }
17         }
18
19         // Print all the elements of the array
20         for (int i = 0; i < a.length; i++)
21         {
22             for (int j = 0; j < a[i].length; j++)
23             {
24                 System.out.print(a[i][j] + "\\t");
25             }
26             System.out.println();
27         }
28     }
29 }
```

Σχήμα 2. Δημιουργία, γέμισμα και εκτύπωση διδιάστατου πίνακα.

6.3 Ακανόνιστοι πίνακες

Το γεγονός ότι ένας διδιάστατος πίνακας είναι ουσιαστικά ένας μονοδιάστατος πίνακας με στοιχεία μονοδιάστατους πίνακες, μας κάνει να υποπτευθούμε ότι

μπορούμε να έχουμε και πίνακες των οποίων κάθε γραμμή έχει διαφορετικό μήκος. Ένας τέτοιος πίνακας ονομάζεται **ακανόνιστος (ragged array)**. Η διαφορά των ακανόνιστων διδιάστατων πινάκων από τους απλούς είναι στο ότι πρέπει να δημιουργηθεί κάθε γραμμή του πίνακα ξεχωριστά. Στο Σχ. 3 εικονίζεται η δημιουργία και η χρήση ενός ακανόνιστου πίνακα με γραμμές που έχουν ένα, δύο και τρία στοιχεία αντίστοιχα, δηλαδή:

$$a = \begin{pmatrix} 1 \\ 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}$$

Προσοχή! Τα στοιχεία που λείπουν σε σχέση με τον αντίστοιχο κανονικό διδιάστατο πίνακα δεν έχουν μηδενικές τιμές, απλά δεν υπάρχουν.

```
1  class Array2D
2  {
3      public static void main(String args[])
4      {
5          // Create the array
6          int[][] a = new int[3][];
7          a[0] = new int[1];
8          a[1] = new int[2];
9          a[2] = new int[3];
10
11         // Fill in the elements of the array
12         int c = 1;
13         for (int i = 0; i < a.length; i++)
14         {
15             for (int j = 0; j < a[i].length; j++)
16             {
17                 a[i][j] = c;
18                 c++;
19             }
20         }
21
22         // Print all the elements of the array
23         for (int i = 0; i < a.length; i++)
24         {
25             for (int j = 0; j < a[i].length; j++)
26             {
27                 System.out.print(a[i][j] + "\t");
28             }
29             System.out.println();
30         }
31     }
32 }
```

Σχήμα 3. Δημιουργία, γέμισμα και εκτύπωση ακανόνιστου διδιάστατου πίνακα.

6.4 Πίνακες αντικειμένων

Με τον ίδιο τρόπο που δημιουργούμε πίνακες πρωταρχικών τύπων όπως ο `int`, μπορούμε να κατασκευάσουμε και πίνακες αντικειμένων, όπως π.χ. τα αντικείμενα της κλάσης `Human` (Σχ. 4). Η διαφορά είναι ότι κάθε ένα από τα στοιχεία του πίνακα αυτού πρέπει να δημιουργηθεί ξεχωριστά (όπως ξεχωριστά δημιουργούνται και τα αντικείμενα των μονοδιάστατων πινάκων του Σχ. 3). Στο Σχ. 5 εικονίζεται η δημιουργία και χρήση ενός πίνακα αποτελούμενο από αντικείμενα της κλάσης `Human`.

```
1  class Human
2  {
3      String name;
4      int height;
5
6      Human(String n, int h)
7      {
8          name = n;
9          height = h;
10     }
11
12     public String toString()
13     {
14         return name + " " + height;
15     }
16 }
```

Σχήμα 4. Κλάση Java που περιγράφει σύντομα τι είναι 'ανθρωπάκι'.

```
1  class HumanArray
2  {
3      public static void main(String args[])
4      {
5          // Create the array
6          Human[] a = new Human[3];
7
8          // Create each object of the array
9          a[0] = new Human("Nikos", 20);
10         a[1] = new Human("Anna", 18);
11         a[2] = new Human("John", 25);
12
13         // Print all the elements of the array
14         for (int i = 0; i < a.length; i++)
15         {
16             System.out.println(a[i]);
17         }
18     }
19 }
```

Σχήμα 5. Δημιουργία, γέμισμα και εκτύπωση πίνακα αντικειμένων τύπου `Human`.

6.5 Οι πίνακες ως ορίσματα μεθόδων

Οι πίνακες μπορούν να χρησιμοποιηθούν ως ορίσματα μεθόδων, με τον ίδιο τρόπο που χρησιμοποιείται οποιαδήποτε άλλη μεταβλητή. Στο παράδειγμα του Σχ. 6 η μέθοδος `fill` χρησιμοποιείται για να γεμίσει τον πίνακα `a` με τιμές `value`.

Ο πίνακας που περνάει ως όρισμα μιας μεθόδου πρέπει να έχει προηγουμένως δημιουργηθεί.

```
1  class ArrayInput
2  {
3      static void fill(int[] a, int value)
4      {
5          // Fill in the elements of the array
6          for (int i = 0; i < a.length; i++)
7          {
8              a[i] = value;
9          }
10     }
11
12     public static void main(String args[])
13     {
14         // Create the array
15         int[] x = new int[10];
16
17         // Call method fill to fill array with values 1
18         fill(x, 1);
19
20         // Print all the elements of the array
21         for (int i = 0; i < x.length; i++)
22         {
23             System.out.println(x[i]);
24         }
25     }
26 }
```

Σχήμα 6. Εισαγωγή πίνακα ως όρισμα σε μέθοδο.

Αν η μέθοδος τροποποιεί τις τιμές του πίνακα που της δίνεται ως όρισμα, τότε οι τιμές του πίνακα τροποποιούνται όχι μόνο για όσο διαρκεί η εκτέλεση της μεθόδου, αλλά και στο υπόλοιπο πρόγραμμα. Αυτό σημαίνει ότι οι πίνακες περνούν ως ορίσματα στις μεθόδους **με αναφορά (by reference)**. Περνάει δηλαδή μια αναφορά στη θέση μνήμης που βρίσκεται ο πραγματικός πίνακας. Το ίδιο συμβαίνει και σε οποιοδήποτε αντικείμενο π.χ. ακόμα και της κλάσης `Human`, όταν περνά ως όρισμα σε μια μέθοδο. Δε συμβαίνει όμως το ίδιο στις μεταβλητές πρωταρχικού τύπου, όπως π.χ. είναι η μεταβλητή `value` που είναι τύπου `int`. Στην περίπτωση αυτή, περνάει η τιμή της μεταβλητής `value` π.χ. η τιμή `1`. Αν προσπαθήσει κανείς να

αλλάζει την τιμή της value μέσα στη μέθοδο fill, τότε δε θα αλλάξει η τιμή της value μέσα στη μέθοδο main. Για το λόγο αυτό λέγεται πως οι μεταβλητές πρωταρχικού τύπου περνούν **με τιμή (by value)**.

6.6 Οι πίνακες ως τύποι επιστροφής μεθόδων

Οι πίνακες επιστρέφονται από μεθόδους με τον ίδιο τρόπο που επιστρέφεται και οποιαδήποτε άλλη μεταβλητή πρωταρχικού τύπου. Η διαφορά είναι ότι πριν επιστραφεί ο πίνακας πρέπει να έχει πρώτα δημιουργηθεί μέσα στη μέθοδο (με τη χρήση της new).

Στο παράδειγμα του Σχ. 7 η μέθοδος array επιστρέφει ένα νέο πίνακα ακεραίων μήκους numberOfElements, και με τιμές value σε όλα τα στοιχεία του.

```
1  class ArrayOutput
2  {
3      static int[] fill(int numberOfElements, int value)
4      {
5          // Create the array
6          int[] a = new int[numberOfElements];
7
8          // Fill in the elements of the array
9          for (int i = 0; i < a.length; i++)
10         {
11             a[i] = value;
12         }
13
14         // Return the array
15         return a;
16     }
17
18     public static void main(String args[])
19     {
20         // Declare the array
21         int[] x;
22
23         // Call method to get a 5 element array of ones
24         x = fill(5, 1);
25
26         // Print all the elements of the array
27         for (int i = 0; i < x.length; i++)
28         {
29             System.out.println(x[i]);
30         }
31     }
32 }
```

Σχήμα 7. Επιστροφή πίνακα από μέθοδο.

6.7 Ορίσματα από τη γραμμή εντολών

Τι σημαίνει **όρισμα από τη γραμμή εντολών (command line argument)**; Όταν μεταγλωττίζουμε ένα πρόγραμμα Java με όνομα Test.java, στη γραμμή εντολών γράφουμε:

```
javac Test.java
```

Στην προκειμένη περίπτωση το όνομα του προγράμματος Test.java είναι το πρώτο όρισμα του προγράμματος javac που εκτελείται από τη γραμμή εντολών.

Έστω ότι επιθυμούμε να κατασκευάσουμε ένα πρόγραμμα που να δέχεται ένα ή περισσότερα ορίσματα από τη γραμμή εντολών π.χ. το πρόγραμμα TestArguments το οποίο να εκτελείται από τη γραμμή εντολών ως εξής:

```
java TestArguments one two three
```

οι λέξεις one, two, three είναι το πρώτο, το δεύτερο και τρίτο όρισμα του προγράμματος TestArguments. Τα ορίσματα αυτά αποθηκεύονται αυτομάτως ως αλφαριθμητικά μέσα στη μεταβλητή args που χρησιμοποιούμε στον ορισμό της μεθόδου main. Η μεταβλητή args είναι ορισμένη ως πίνακας αλφαριθμητικών. Στο Σχ. 8 εικονίζεται ένα πρόγραμμα που διαβάζει αυτά τα ορίσματα και μετά τα τυπώνει.

Αν επιθυμούμε να ελέγξουμε αν το πλήθος των ορισμάτων που έδωσε ο χρήστης είναι σωστό, αρκεί να ελέγξουμε την τιμή της μεταβλητής args.length.

```
1  class TestArguments
2  {
3      public static void main(String[] args)
4      {
5          for (int i = 0; i < args.length; i++)
6          {
7              System.out.println(args[i]);
8          }
9      }
10 }
11 }
```

Σχήμα 8. Ανάγνωση και εκτύπωση ορισμάτων προγράμματος από τη γραμμή εντολών.

Ασκήσεις

1. Να υλοποιηθούν τα παραπάνω παραδείγματα.
2. Να κατασκευαστεί μονοδιάστατος πίνακας με 10 τυχαίες πραγματικές τιμές. Οι τυχαίες τιμές πρέπει να παράγονται αυτομάτως.
3. Να υλοποιήσετε αλγόριθμο που προσθέτει τα στοιχεία ενός μονοδιάστατου πίνακα.
4. Να υλοποιήσετε αλγόριθμο που να βρίσκει το μέγιστο στοιχείο ενός μονοδιάστατου πίνακα. Στο τέλος θα πρέπει να εκτυπώνετε την τιμή και τη θέση του μέγιστου στοιχείου.
5. Να κατασκευαστεί διδιάστατος πίνακας με τυχαίες τιμές. Οι τυχαίες τιμές πρέπει να παράγονται αυτομάτως.
6. Να υλοποιηθεί η πρόσθεση δύο διδιάστατων πινάκων, στοιχείο προς στοιχείο.
7. Να συγκρίνετε αν δύο διδιάστατοι πίνακες είναι ίσοι.
8. Να κατασκευάσετε μια κλάση Point που αναπαριστά ένα σημείο στο επίπεδο. Να κατασκευαστεί ένας πίνακας 1000 σημείων.
9. Να κατασκευαστεί μέθοδος που να επιστρέφει το άθροισμα των στοιχείων ενός πίνακα πραγματικών αριθμών που θα δέχεται ως όρισμα.

10. Να κατασκευαστεί μέθοδος που να δέχεται έναν πίνακα ακεραίων input και μια ακέραια τιμή value, ως ορίσματα. Η μέθοδος θα πρέπει να μηδενίζει όποιο στοιχείο του input έχει τιμή μεγαλύτερη από την τιμή της value.

11. Να κατασκευαστεί μέθοδος που να επιστρέφει έναν πίνακα τυχαίων ακεραίων αριθμών από το 0 έως το 100.

12. Να κατασκευαστεί πρόγραμμα που να δέχεται δύο πραγματικούς αριθμούς ως ορίσματα από τη γραμμή εντολών και να τους προσθέτει. Να γίνεται έλεγχος αν το πλήθος των ορισμάτων είναι σωστό. Σε περίπτωση που δεν είναι σωστό, να δίνονται εξηγήσεις για τη χρήση του προγράμματος.

13. Να τροποποιηθεί το πρόγραμμα της παραπάνω άσκησης για να πραγματοποιεί το άθροισμα των ορισμάτων της γραμμής εντολών, όσα και αν είναι αυτά.