

Αντικειμενοστρεφής Προγραμματισμός

Παναγιώτης Αδαμίδης
adamidis@ihu.gr

Πίνακες (Arrays)



Πίνακες

- Declaration: `<type> <array_name> [] ;`
ή `<type> [] <array_name> ;` ♥
- Space allocation: `<array_name> = new <type>[size];`
- Παράδειγμα: `int monthDays[];`
`monthDays = new int[12];`
ή
`int[] monthDays = new int[12];`
- Δείκτες: έναρξη αρίθμησης από **0**



Πίνακες: Παράδειγμα (1)

```
class Array1{
    public static void main(String args[]){
        int monthDays[];
        monthDays=new int [12];
        monthDays[0]=31;
        monthDays[1]=28;
        .....
        monthDays[11]=31;
        System.out.println("April has " +
            monthDays[3] + "days.");
    }
}
```



Πίνακες: Παράδειγμα (2)

```
class Array1{
    public static void main(String args[]){
        int monthDays[] =
            {31,28,31,30,31,30,31,31,30,31,30,31};
        System.out.println("April has " +
            monthDays[3] + "days");
    }
}
```



Πίνακες: Παράδειγμα Εύρεση καλύτερης τιμής (1)

```
import java.io.*;
public class BestPrice{
    public static void main(String[] args) throws
    IOException {
        final int DATA_LENGTH=100;
        double[] data = new double [DATA_LENGTH];
        int dataSize=0;
        //Αντικείμενο για εισαγωγή δεδομένων από
        // το πληκτρολόγιο
        BufferedReader br=new BufferedReader(new
            InputStreamReader(System.in));
        boolean done=false;
        double price;
```



Πίνακες: Παράδειγμα Εύρεση καλύτερης τιμής (2)

```
while (!done) {
    System.out.println("Enter price, 0 to quit:");
    price=Double.parseDouble(br.readLine());
    if (price == 0) done=true; //end of input
    else if ( dataSize<data.length ) {
        data[dataSize]=price;
        dataSize++;
    }
    else { // array is full
        System.out.println ("The array is full");
        done=true;
    }
} // end of while
```



Πολυδιάστατοι Πίνακες (Multi-dimensional Arrays)

- `double[][] matrix = new double [4][4]`
ή
- `double matrix [][] = new double [4][];`
- `matrix[0] = new double [4];`
- `matrix[1] = new double [4];`
- `matrix[2] = new double [4];`
- `matrix[3] = {0,1,2,3};`



Εύρεση μεγαλύτερης τιμής πίνακα ακεραίων δύο διαστάσεων

```
import java.util.Random;
class Array2DMax {
    public static int array2DMax (int b[][]) {
        int max = b[0][0];
        for (int i=0; i<b.length; i++)
            for (int k=0; k<b[0].length; k++) //Τι θα ήταν καλύτερο;
                if (max < b[i][k]) max=b[i][k];
        return max;
    }
    public static void main (String[] args) {
        int[][] a = new int[7][9];
        Random rnd = new Random();
        for (int i=0; i<a.length; i++) //Τι θα ήταν καλύτερο;
            for (int k=0; k<a[0].length; k++)
                a[i][k] = rnd.nextInt(1000);
        System.out.println("Max value of the array is: "+array2DMax(a));
    }
}
```



Προσοχή: Δημιουργία αντικειμένων πριν την πρόσβαση

- Τι πρόβλημα υπάρχει στον παρακάτω κώδικα;

```
class Test {
    int x;
}
class ArrayTest1 {
    public static void main(String args[]) {
        int i;
        Test TestArray[] = new Test[100];
        for (i = 0; i < 100; i++)
            TestArray[i].x = i;
        System.out.println("We made it!");
    }
}
```



Αρνητικά σημεία

- Πρέπει να οριστεί ένα μέγιστο μέγεθος κατά την δήλωση
 - ♦ Συνήθως επιλέγουμε το μέγιστο δυνατό μέγεθος
- Οι δείκτες μπορεί να είναι μόνο ακέραιοι
- Η πρόσθεση νέων στοιχείων μπορεί να είναι υπολογιστικά χρονοβόρα
 - ♦ Ιδιαίτερα εάν τα στοιχεία είναι σε κάποια σειρά
- Τα στοιχεία πρέπει να είναι του ίδιου τύπου
 - ♦ Λύση: υπαρξη/χρήση κλάσης Object



Θετικά σημεία

- Μπορούμε να πάρουμε άμεσα την τιμή οιοδήποτε στοιχείου
 - ♦ Εάν γνωρίζουμε την θέση του
- Εύκολη επεξεργασία με ένα βρόχο
 - ♦ Ίδια επεξεργασία σε διαφορετικά δεδομένα
- Αποτελεσματική χρήση μνήμης
 - ♦ Εάν ο πίνακας είναι «πυκνός»
- Παρέχει μία φυσική σειρά διάταξης των στοιχείων

