

Αντικειμενοστρεφής Προγραμματισμός



Πίνακες (Arrays)



Παναγιώτης Αδαμίδης

adamidis@ihu.gr

**Τμήμα Μηχανικών Πληροφορικής &
Ηλεκτρονικών Συστημάτων**

Πίνακες

- Declaration: `<type> <array_name>[];`
ή `<type>[] <array_name>;` ♥
- Declaration + 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**



???

- Should array indices start at 0 or 1?
- My compromise of 0.5 was rejected without, I thought, proper consideration.

Stan Kelly-Bootle (1929-2014)

Βρετανός συγγραφέας, ακαδημαϊκός, τραγουδιστής-τραγουδοποιός
και επιστήμονας υπολογιστών.



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
```



Πίνακες: Παράδειγμα

Εύρεση καλύτερης τιμής (3)

```
if (dataSize==0) return; // no data
double lowest = data[0];
for(int i=1; i<dataSize; i++)
    if (data[i] < lowest) lowest=data[i];
// Εμφάνιση αποτελεσμάτων
for(int i=0; i<dataSize; i++){
    System.out.print(data[i]);
    if (data[i]==lowest)
        System.out.print("← lowest price");
    System.out.println();
}
} //main
} //class BestPrice
```



Συνήθη Λάθη

- [illegible]



Πίνακες και τελεστής ανάθεσης

```
class ArrayCopyTest {  
    static void print (double[] u, String id){  
        for(int i=0; i<u.length; i++)  
            System.out.println(id+"["+i+"]="+u[i]);  
        System.out.println();  
    }  
    public static void main(String args[]){  
        double[] x={2.2,4.4};  print(x,"x");  
        double[] y={1.1,3.3,5.5};  print(y,"y");  
        y=x;  
        print (y,"y");  
        x[0]=8.8;  
        print(x,"x");  
        print(y,"y");  
    }  
}
```



Πίνακες και τελεστής ανάθεσης (2)

ΕΞΟΔΟΣ:

$x[0]=2.2$

$x[1]=4.4$

$y[0]=1.1$

$y[1]=3.3$

$y[2]=5.5$

$y[0]=2.2$

$y[1]=4.4$

$x[0]=8.8$

$x[1]=4.4$

$y[0]=8.8$

$y[1]=4.4$



Αντιγραφή Πινάκων

```
public static void arraycopy  
(Object scr, int scrPos, Object dst, int dstPos, int length)
```

Το προηγούμενο πρόγραμμα με χρήση της:
`System.arraycopy(x,0,y,0,x.length);`

ΕΞΟΔΟΣ:

x[0]=2.2

x[1]=4.4

y[0]=1.1

y[1]=3.3

y[2]=5.5

y[0]=2.2

y[1]=4.4

y[2]=5.5

x[0] =8.8

x[1]=4.4

y[0]=2.2

y[1]=4.4

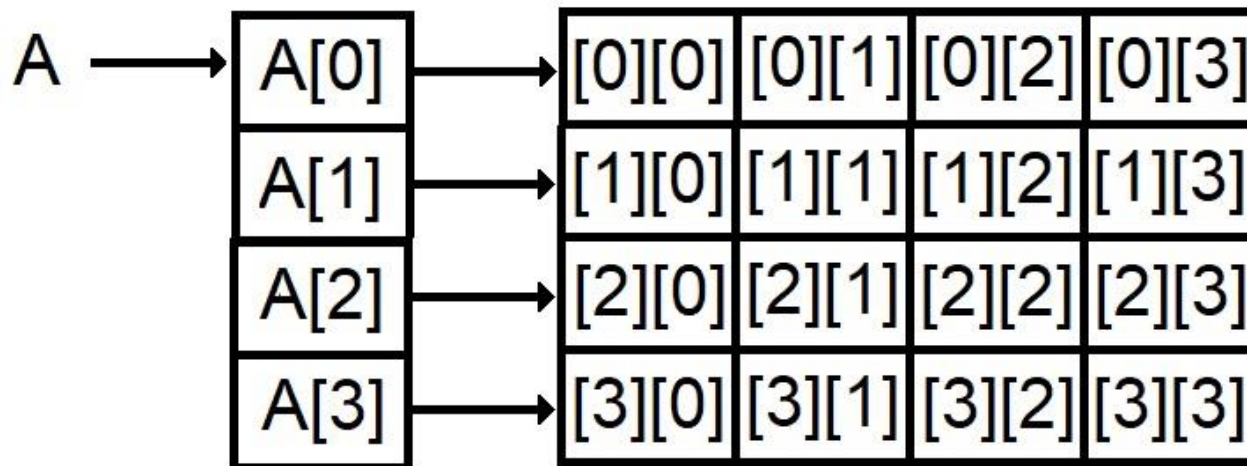
y[2]=5.5



Πολυδιάστατοι Πίνακες (Multi-dimensional Arrays)

1

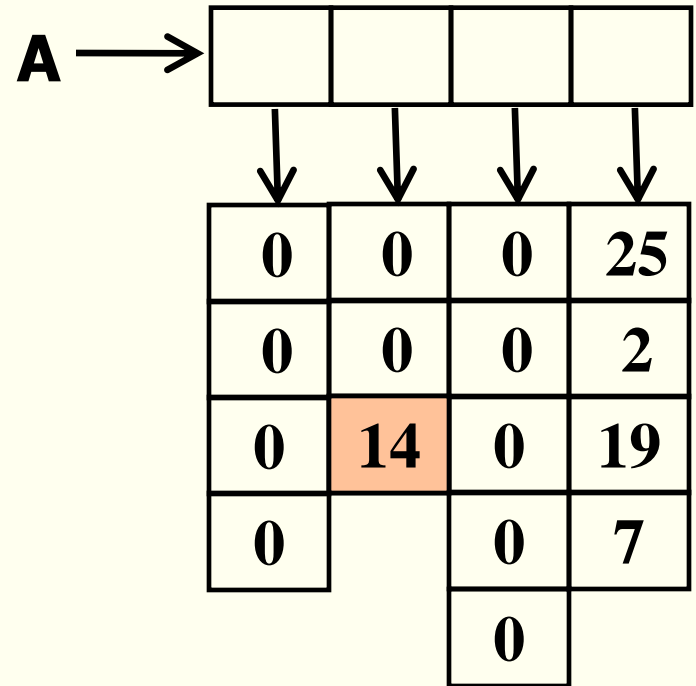
- Αντιμετωπίζονται ως μονοδιάστατοι πίνακες που αποτελούνται από πίνακες
- Γενική μορφή δήλωσης:
`type[][]...[] arrayName;`
- Χειρισμός δυσδιάστατων πινάκων ως μονοδιάστατων των οποίων τα στοιχεία είναι μονοδιάστατοι πίνακες.
- Παράδειγμα: `int[][] A = new int[4][4];`



Πολυδιάστατοι Πίνακες (Multi-dimensional Arrays)

2

- 'H
- `double[][] A;`
- `A = new double [4][];`
- `A[0] = new double [4];`
- `A[1] = new double [3];`
- `A[2] = new double [5];`
- `A[3] = {25,2,19,7};`
- `A[1][2] = 14;`



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++)
            for (int k=0; k<a[0].length; k++) //Τι θα ήταν καλύτερο;
                a[i][k] = rnd.nextInt(1000);
        System.out.println("Max value of the array is: "+array2DMax(a));
    }
}
```



Πίνακας Αντικειμένων (Class Student)

1

```
public class Student {  
    private String name;  
    private int AM;  
    // Constructors  
    // get/set  
    // toString  
  
} //class
```



Πίνακας Αντικειμένων (Class Student)

1a

Παράδειγμα Constructor:

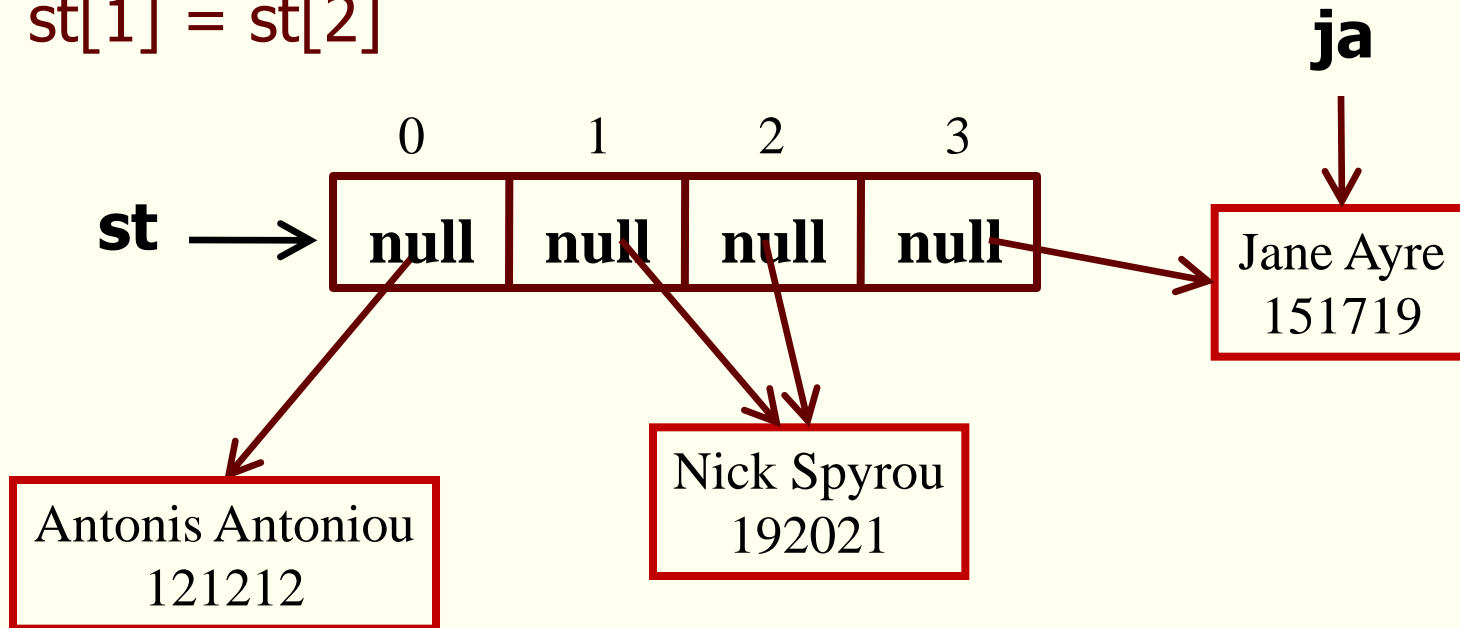
```
public Student (String n, int ar) {  
    name=n;  
    AM=ar;  
}
```



Πίνακας Αντικειμένων (Class Student)

2

- Student st[] = new Student[4];
- st[0]= new Student("Antonis Antoniou", 121212);
- Student ja = new Student("Jane Ayre", 151719);
- st[3] = ja;
- st[2]= new Student("Nick Spyrou", 192021);
- st[1] = st[2]



Προσοχή: Δημιουργία αντικειμένων πριν την πρόσβαση σε αυτά

- Τι πρόβλημα υπάρχει στον παρακάτω κώδικα;

```
class Test {  
    int x;  
}
```

```
TestArray[i] = new Test();
```

```
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!");  
    }  
}
```



Πίνακες: 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: ύπαρξη/χρήση κλάσης Object



Πίνακες: Θετικά σημεία

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



Ερωτήσεις;

