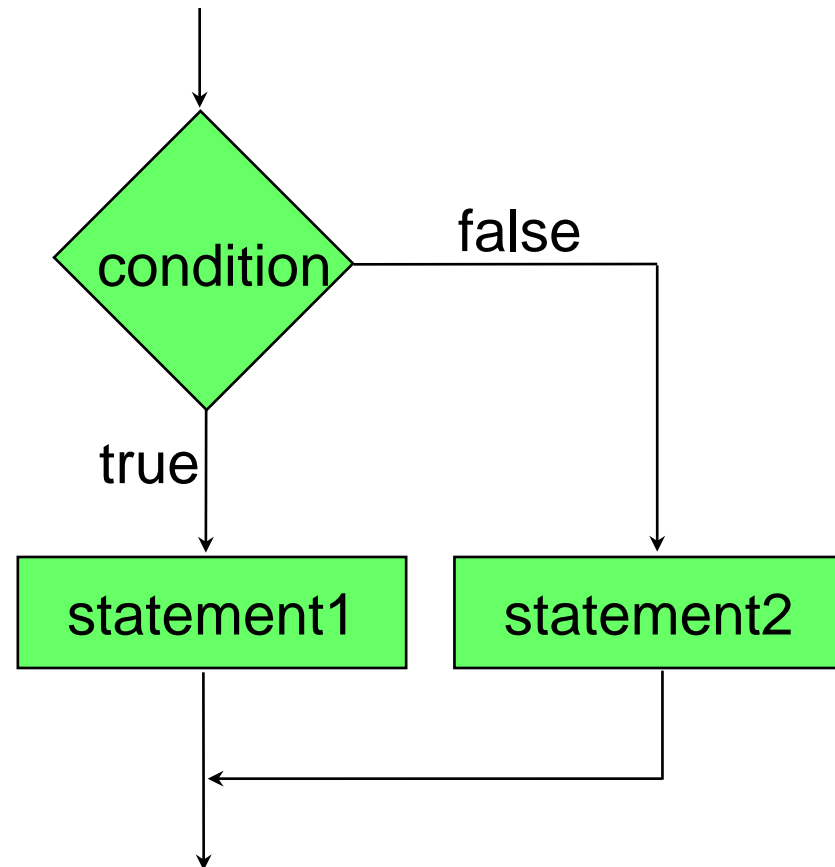


ΤΕΧΝΙΚΕΣ ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΑΦΟΥΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ

Εισαγωγή στη Java III

To if-else statement

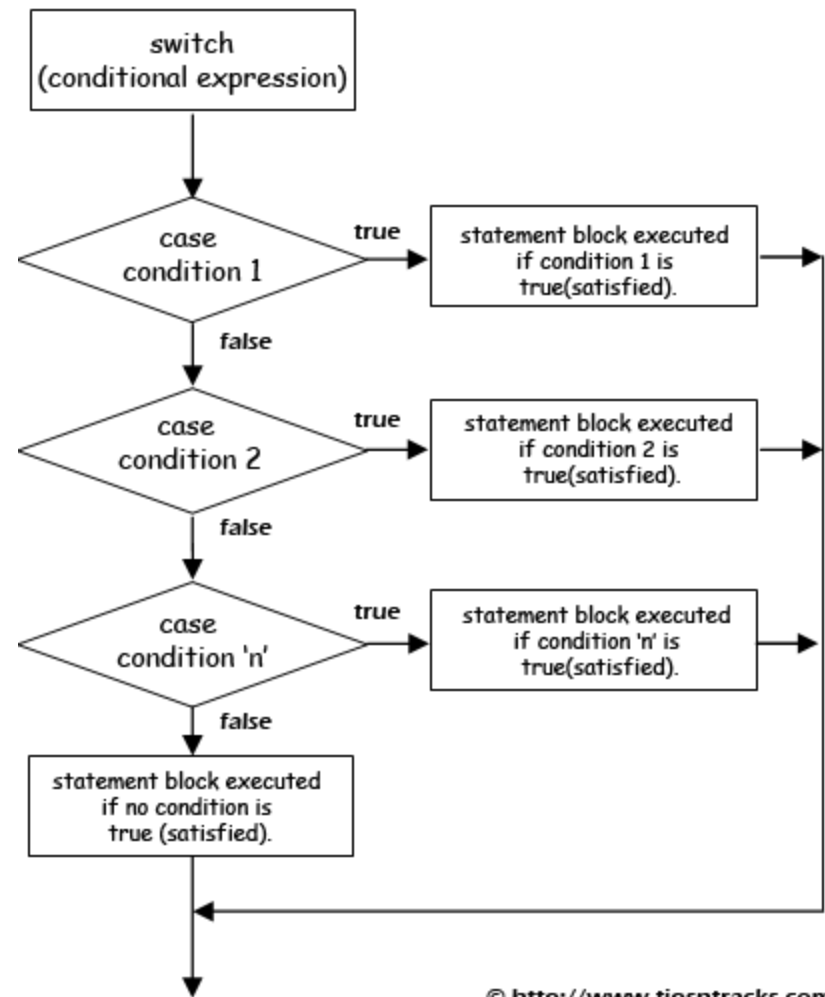
- Το if-else statement δουλεύει καλά όταν στο condition θέλουμε να περιγράψουμε μια επιλογή με **δύο** πιθανά ενδεχόμενα.
- Τι γίνεται αν η συνθήκη μας έχει πολλά ενδεχόμενα?



Switch statement

ΣΥΝΤΑΚΤΙΚΟ:

```
switch (<condition expression>) {  
  case <condition 1>:  
    code statements 1  
    break;  
  case <condition 2>:  
    code statements 2  
    break;  
  case <condition 3>:  
    code statements 3  
    break;  
  default:  
    default statements  
    break;  
}
```



Παράδειγμα

- Ένα πρόγραμμα που να εύχεται καλημέρα σε τρεις διαφορετικές γλώσσες ανάλογα με την επιλογή του χρήστη.

```
import java.util.Scanner;

class SwitchTest{
    public static void main(String args[])
    {
        Scanner input = new Scanner(System.in);
        String option = input.next();

        switch(option) {
            case "GR":
            case "gr":
                System.out.println("kalimera");
                break;
            case "EN":
            case "en":
                System.out.println("good morning");
                break;
            case "FR":
            case "fr":
                System.out.println("bonjour");
                break;
            default:
                System.out.println("I do not speak this language. " +
                    "Greek, English, French only");
        }
    }
}
```

ΚΛΑΣΕΙΣ ΚΑΙ ΑΝΤΙΚΕΙΜΕΝΑ

Κλάση

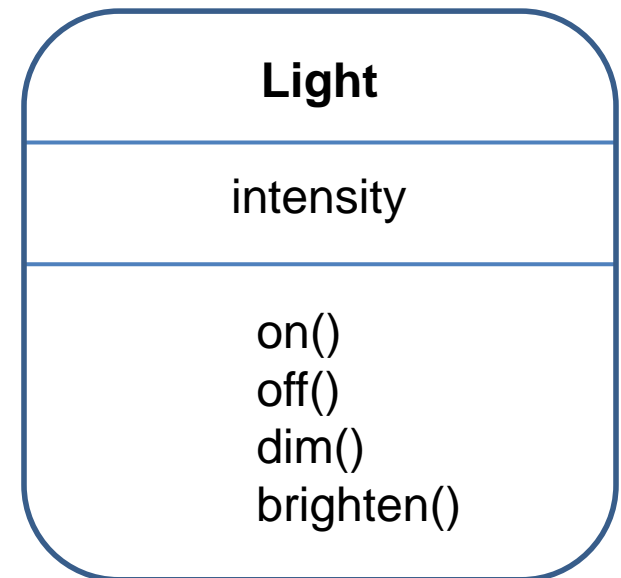
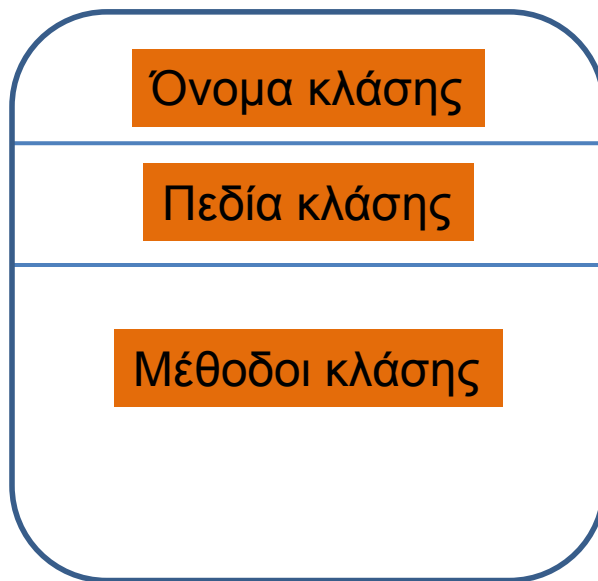
- Μια **κλάση** είναι μία αφηρημένη περιγραφή αντικειμένων με κοινά **χαρακτηριστικά** και κοινή **συμπεριφορά**.
 - Ένα **καλούπι/πρότυπο** που παράγει αντικείμενα
- Ένα **αντικείμενο** είναι ένα **στιγμιότυπο** μίας κλάσης.
- Η κλάση ορίζει τον **τύπο** του αντικειμένου.
 - Τα **χαρακτηριστικά** του αντικειμένου
 - Τις **ενέργειες** που μπορεί να επιτελέσει.

Πρακτικά στον κώδικα

- Μία κλάση **K** ορίζεται από
 - Κάποιες **μεταβλητές** τις οποίες ονομάζουμε **πεδία**
 - Κάποιες **συναρτήσεις** που τις ονομάζουμε **μεθόδους**.
 - Οι μέθοδοι «**βλέπουν**» τα πεδία της κλάσης
- Ένα **αντικείμενο** ορίζεται ως μια **μεταβλητή τύπου K**
 - Το αντικείμενο έχει συγκεκριμένες **τιμές** στα πεδία.
 - Στο πρόγραμμα έχουμε (συνήθως) **πρόσβαση** μόνο τις **μεθόδους**.
 - Μέσω των μεθόδων έχουμε πρόσβαση στα πεδία
 - Αν υπάρχουν κάποια **πεδία** στα οποία έχουμε πρόσβαση αυτά τα λέμε **properties**.

} μέλη
της
κλάσης

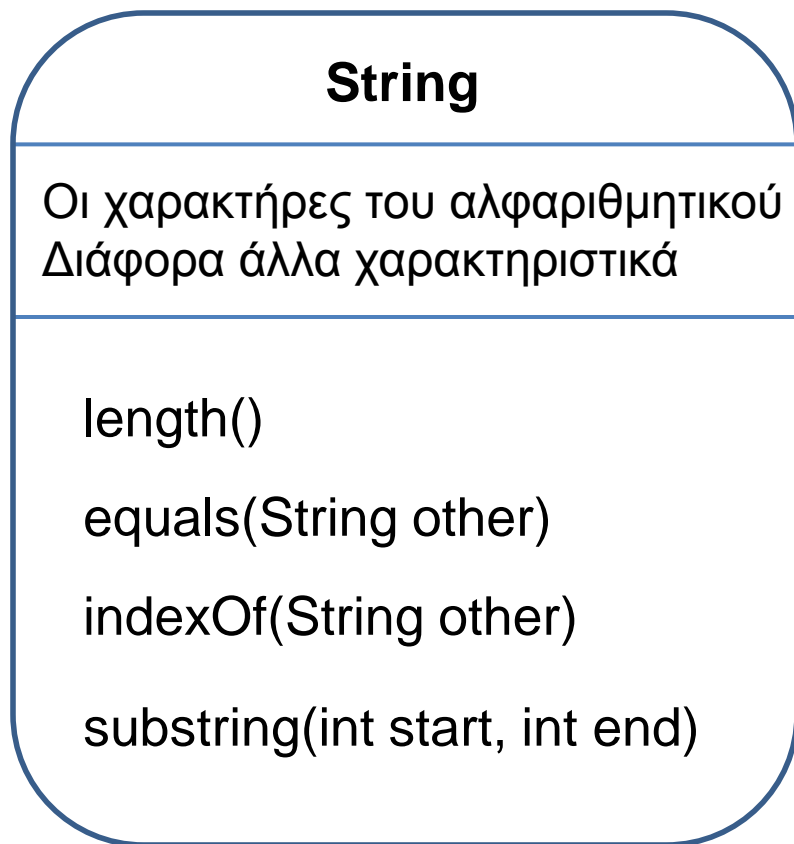
Γενική μορφή της κλάσης



ΥΠΑΡΧΟΥΣΕΣ ΚΛΑΣΕΙΣ

Strings

- Έχουμε ήδη χρησιμοποιήσει κλάσεις και αντικείμενα όταν χρησιμοποιούμε Strings



Η ακριβής αναπαράσταση του αλφαριθμητικού δεν έχει και τόσο σημασία εφόσον εμείς χρησιμοποιούμε μόνο τις μεθόδους.

String αντικείμενα

- Ένα String αντικείμενο είναι μια μεταβλητή τύπου String.
 - Τρεις διαφορετικοί τρόποι να δώσουμε τιμή σε ένα String object

```
import java.util.Scanner;

class StringExample{
    public static void main(String[] args){
        Scanner input = new Scanner(System.in);

        String x = input.next();
        String z = new String("java");
        String y = "java";
    }
}
```

String μέθοδοι

- Έχοντας τα String αντικείμενα μπορούμε να καλέσουμε τις μεθόδους τους

```
import java.util.Scanner;

class StringExample{
    public static void main(String[] args){
        Scanner input = new Scanner(System.in);

        String x = input.next();
        String z = new String("java");
        String y = "java";

        int offset = y.indexOf("va");
        int end = y.length();
        String z = y.substring(offset, end);
    }
}
```

Τα Strings είναι αμετάβλητα (**immutable**) αντικείμενα
Η τελευταία ανάθεση δημιουργεί ένα **καινούριο** αντικείμενο

Ισότητα String

Τι θα εκτυπωθεί?

(μια λογική συνθήκη τυπώνει true/false ανάλογα αν είναι αληθής/ψευδής)

```
import java.util.Scanner;

class StringExample{
    public static void main(String[] args){
        Scanner input = new Scanner(System.in);

        String x = input.next();
        String z = new String("java");
        String y = "java";

        System.out.println("1. " + (x == "java"));
        System.out.println("2. " + (y == "java"));
        System.out.println("3. " + (z == "java"));
        System.out.println("4. " + x.equals("java"));
        System.out.println("5. " + y.equals("java"));
        System.out.println("6. " + z.equals("java"));
    }
}
```

1. false

2. true

3. false

4. true

5. true

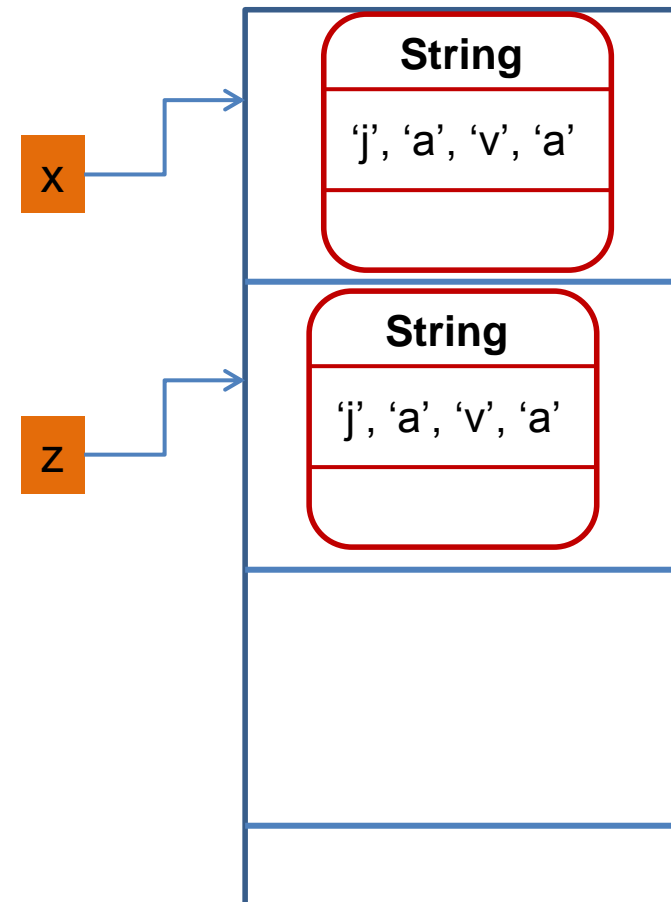
6. true

Για την σύγκριση Strings **ΠΑΝΤΑ** χρησιμοποιούμε την μέθοδο **equals**.

String Interning

```
String x = input.next();  
String z = new String("java");
```

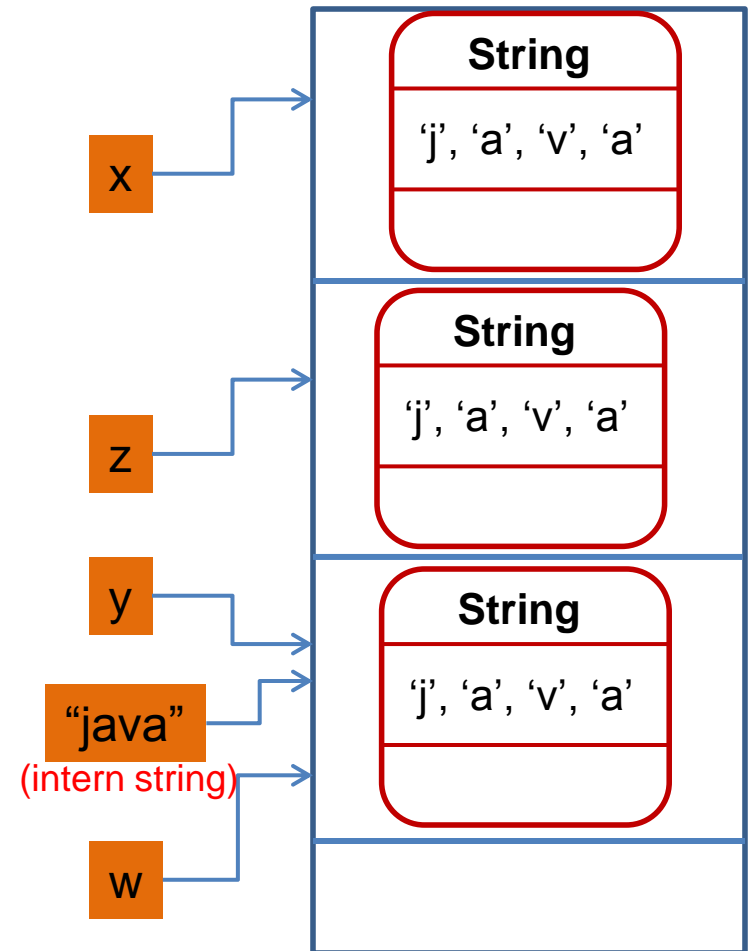
- Γιατί συμβαίνει αυτό?
- Όταν δημιουργούμε ένα String αντικείμενο δεσμεύουμε χώρο στη μνήμη για το αντικείμενο
- Η μεταβλητή που ορίζουμε «δείχνει» σε αυτό το χώρο μνήμης



String Interning

```
String x = input.next();  
String z = new String("java");  
String y = "java";  
String w = "java";
```

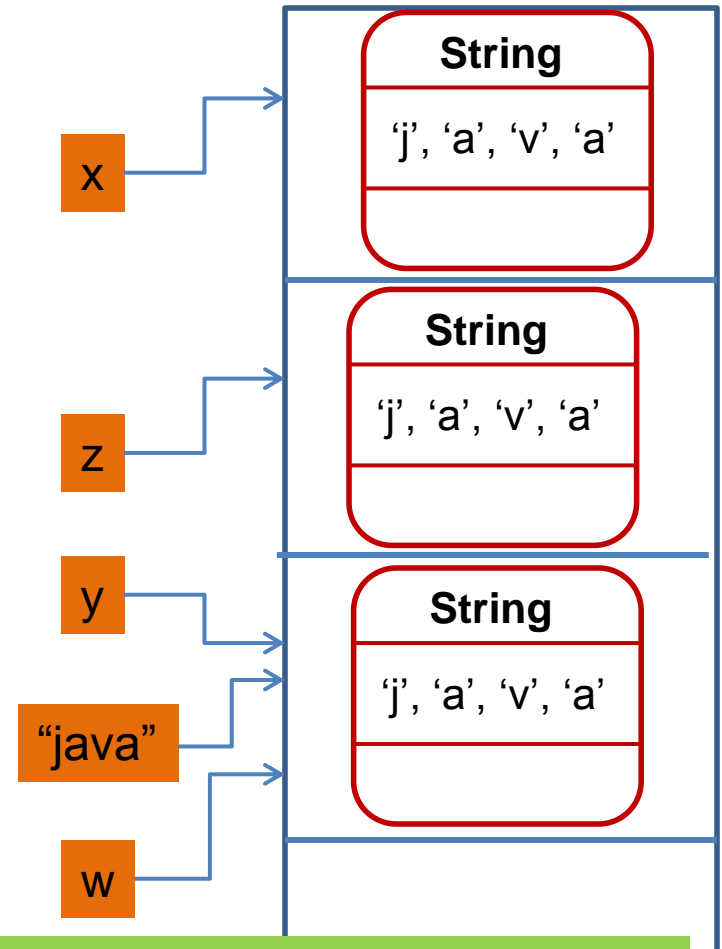
- Το JVM για κάθε **string value** που εμφανίζεται δημιουργείται ένα **αντικείμενο**, το οποίο ονομάζεται **intern string**, και το οποίο κρατάει αυτή την τιμή.
- Άρα δημιουργείται ένα αντικείμενο για την τιμή **"java"**
- Η εντολή **String y = "java";** κάνει το **y** να δείχνει στη θέση που είναι αποθηκευμένη η τιμή **"java"**



String Interning

```
String x = input.next();  
String z = new String("java");  
String y = "java";  
String w = "java";  
System.out.println((y == "java"));
```

- Ο τελεστής `==` μεταξύ δύο αντικειμένων εξετάζει αν πρόκειται για την **ίδια θέση μνήμης**.
- Γι αυτό (`y == "java"`) επιστρέφει `true`.
- Όλα αυτά θα είναι πιο ξεκάθαρα όταν θα μιλήσουμε για **αναφορές**.



Για την σύγκριση Strings **ΠΑΝΤΑ** χρησιμοποιούμε την μέθοδο `equals`.

String σταθερές

- Οι String τιμές είναι κι αυτές αντικείμενα και μπορούμε να καλέσουμε τις μεθόδους τους

```
import java.util.Scanner;

class StringExample{
    public static void main(String[] args){
        Scanner input = new Scanner(System.in);

        String x = input.next();
        String z = new String("java");
        String y = "java";

        int offset = "java".indexOf("va");
        int end = "java".length();
        String z = "java".substring(offset, end);
    }
}
```

Wrapper classes

- Για κάθε βασικό τύπο η Java έχει και μία **wrapper class**:
 - **Integer** class
 - **Double** class
 - **Boolean** class
- Οι κλάσεις αυτές έχουν κάποιες μεθόδους και πεδία που μπορεί να μας είναι χρήσιμα
 - Κατά κύριο λόγο **μετατροπή** από και προς **string**
 - Τη **μέγιστη** και την **ελάχιστη** τιμή κάθε τύπου

Παράδειγμα

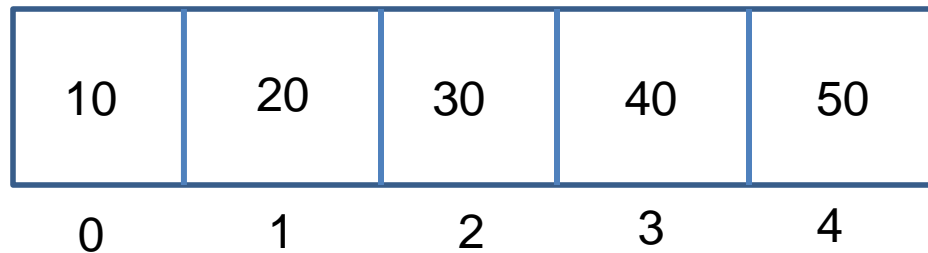
```
class WrapperTest{
    public static void main(String argsp[])
    {
        int i = Integer.valueOf("2");
        double d = Double.parseDouble("2.5");
        System.out.println(i*d);
        Integer x = 5;
        Double y = 2.5;
        String s = x.toString() + y.toString();
        System.out.println(s);
        System.out.println(Integer.MAX_VALUE);
    }
}
```

Πίνακες

- Πολλές φορές έχουμε πολλές μεταβλητές του ίδιου τύπου που συσχετίζονται και θέλουμε να τις βάλουμε μαζί.
 - Τα ονόματα των φοιτητών σε μία τάξη
 - Οι βαθμοί ενός φοιτητή για όλα τα εργαστήρια.
- Για το σκοπό αυτό χρησιμοποιούμε τους πίνακες.
- Ορισμός πίνακα:
 - `int [] myArray1 = {10,20};` // αρχικοποιημένος πίνακας
 - `int myArray2[] = new int[2];`
 - Δημιουργούν δύο πίνακες 2 θέσεων (`length 2`) που κρατάνε ακέραιους
- Οι πίνακες ορίζονται με ένα μέγεθος (`length`) και αυτό δεν αλλάζει
- Στη Java ένας πίνακας είναι ένα αντικείμενο και έχει properties
 - `System.out.println(myArray2.length);`
 - Τυπώνει το μέγεθος του πίνακα.

Πρόσβαση των στοιχείων του πίνακα

- **Προσοχή!** Τα στοιχεία του πίνακα αριθμούνται από το `0...length-1` (**ΟΧΙ** `1...length`)
 - `int myArray[] = {10,20,30,40,50};`



- Για να προσπελάσουμε το **δεύτερο** στοιχείο του πίνακα
 - `myArray[1] += 5;`
 - `System.out.println(myArray[1]);`

Πίνακες

```
public class TestArrays1 {
    public static void main(String [] args){

        int arr0[]; // int[] arr0;

        int arr1 [] = {1, 2, 3, 4};
        for (int i = 0; i < arr1.length; i ++){
            System.out.println(arr1[i]);
        }

        int arr2[] = new int [10];
        for (int i = 0; i < arr2.length; i ++){
            arr2[i] = i+1;
        }
        arr0 = arr2;
    }
}
```

Διατρέχοντας ένα πίνακα

- Στην Java έχουμε δύο τρόπους να διατρέχουμε ένα πίνακα

Διατρέχουμε τα στοιχεία

```
for (<array type> element: array)
{
    ... do something with element...
}
```

```
int array[] = {1,3,5,7};
for (int element: array)
{
    System.out.println(element)
}
```

Διατρέχουμε τις θέσεις του πίνακα

```
for (int i = 0; i < array.length; i ++ )
{
    ... do something with array[i]...
}
```

```
int array[] = {1,3,5,7};
for (int i = 0; i < array.length; i ++ )
{
    System.out.println(array[i])
}
```


Παράδειγμα

- Τυπώστε όλα τα **στοιχεία** του πίνακα και όλα τα ζεύγη από **στοιχεία** στον πίνακα

```
class ScanArray
{
    public static void main(String [] args)
    {
        double [] array = {5.3, 3.4, 2.3, 1.2, 0.1};

        // Print all elements
        for (double element: array){
            System.out.println(element);
        }

        // Print all pairs of elements
        for (int i = 0; i < array.length; i ++){
            for (int j = i+1; j < array.length; j ++){
                System.out.println(array[i] + " " + array[j]);
            }
        }
    }
}
```

Πολυδιάστατοι πίνακες

- Μπορούμε να ορίσουμε και **πολυδιάστατους** πίνακες
 - `int myArray1[][] = {{10,20,30},{3,4,5}};`
 - `int myArray2[][] = new int[2][3];`
- Ένας δισδιάστατος πίνακας είναι ένας **πίνακας από πίνακες**.
 - `int myArray3[][] = new int[2][]`
 - `myArray3[0] = new int[3]`
 - `myArray3[1] = new int[3]`
- Ο πίνακας μπορεί να είναι ασύμμετρος
 - `myArray3[1] = new int[5]`
- Τι παίρνω για τα παρακάτω?
 - `System.out.println(myArray3.length);`
 - `System.out.println(myArray3[1].length);`

10	20	30
3	4	5

→	10	20	30
→	3	4	5

→	10	20	30		
→	3	4	5	6	7

Πίνακες

```
public class TestArrays2 {  
    public static void main(String [] args) {  
        int arr3[][] = {{1, 2, 3}, {3, 4, 5}};  
        int arr4[][] = new int [10][20];  
        arr4 = arr3;  
        System.out.println(arr3.length + " "  
            + arr3[0].length);  
        int arr5[][] = new int[2][];  
        arr5[0] = new int[3];  
        arr5[1] = new int[5];  
    }  
}
```

Τυπώνει "2 3"

Ασύμμετρος πίνακας