

ΤΕΧΝΙΚΕΣ ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΑΦΟΥΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ

Υπάρχουσες κλάσεις και αντικείμενα στην Java

Strings

Wrapper Classes

Δομές

ΚΛΑΣΕΙΣ ΚΑΙ ΑΝΤΙΚΕΙΜΕΝΑ

Κλάση

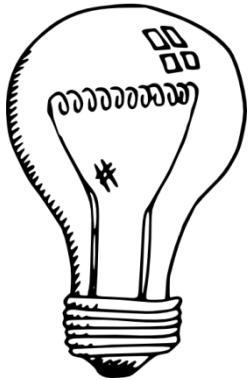
- Μια **κλάση** είναι μία αφηρημένη περιγραφή αντικειμένων με κοινά **χαρακτηριστικά** και κοινή **συμπεριφορά**.
 - Ένα **καλούπι/πρότυπο** που παράγει αντικείμενα
- Ένα **αντικείμενο** είναι ένα **στιγμιότυπο** μίας κλάσης.
- Η κλάση ορίζει τον **τύπο** του αντικειμένου.
 - Τα **χαρακτηριστικά** του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

Πρακτικά στον κώδικα

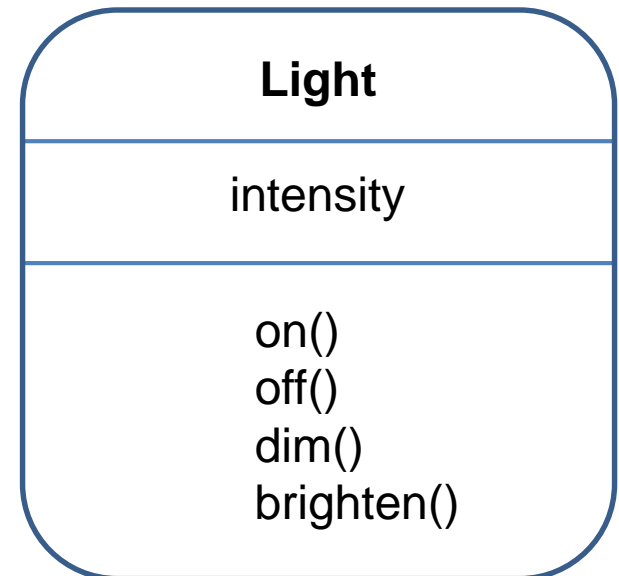
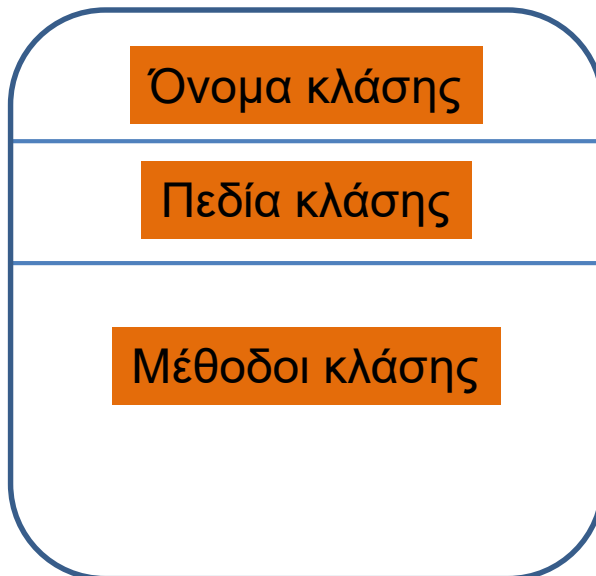
- Μία κλάση **K** ορίζεται από
 - Κάποιες **μεταβλητές** τις οποίες ονομάζουμε **πεδία**
 - Κάποιες **συναρτήσεις** που τις ονομάζουμε **μεθόδους**.
 - Οι μέθοδοι «**βλέπουν**» τα πεδία της κλάσης
- Ένα **αντικείμενο** ορίζεται ως μια **μεταβλητή τύπου K**
 - Το αντικείμενο έχει συγκεκριμένες **τιμές** 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



Έχουμε μια κλάση η οποία υλοποιεί το φως σε ένα δωμάτιο, το οποίο έχει και ένα dimmer



Δημιουργία αντικειμένων

- Δημιουργία αντικειμένου:

```
<Όνομα Κλάσης> myObject = new <Όνομα Κλάσης> ([Ορίσματα]);
```

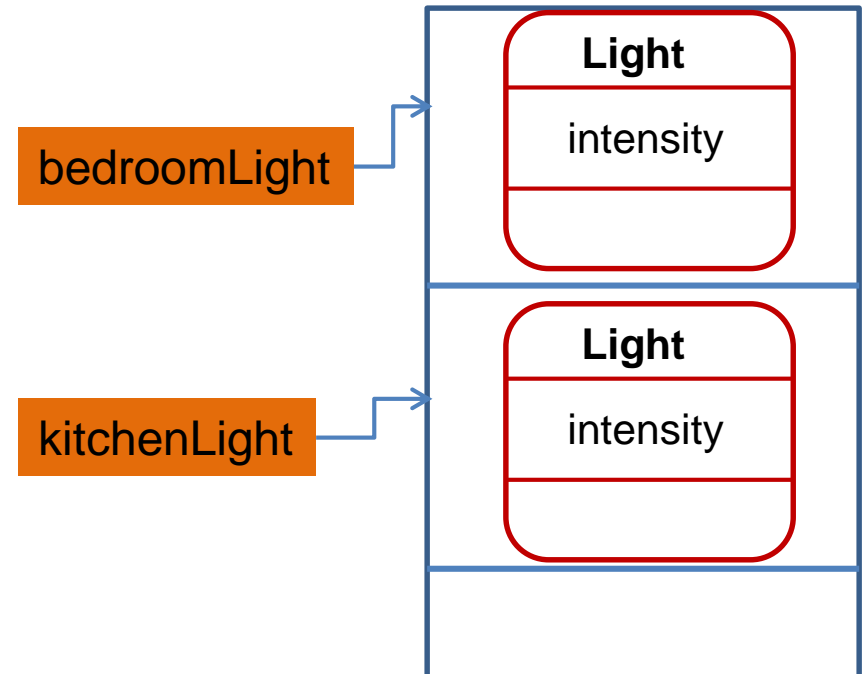
- Η λέξη κλειδί **new** δημιουργεί ένα καινούριο αντικείμενο
 - Δεσμεύει τον απαραίτητο χώρο στην μνήμη.
- Παράδειγμα που έχουμε δει ήδη:
`Scanner input = new Scanner(System.in)`
- Η λίστα των ορισμάτων μπορεί να είναι και κενή

Δημιουργία αντικειμένων

```
Light bedroomLight = new Light();
```

```
Light kitchenLight = new Light();
```

- Με την εντολή **new** δημιουργούμε ένα καινούριο αντικείμενο της κλάσης και του δίνουμε ένα **όνομα**.
- Η **new** δεσμεύει **χώρο μνήμης** για το αντικείμενο
 - Μας επιστρέφει την **διεύθυνση** του χώρου που δεσμεύτηκε.
- Η **μεταβλητή** που ορίζουμε κρατάει την διεύθυνση που “**δείχνει**” σε αυτό τον χώρο μνήμης



Κλήση μεθόδων

- Η πρόσβαση που έχουμε στα αντικείμενα είναι (κατά κύριο λόγο) μέσα από τις μεθόδους τους.
- Η κλήση μιας μεθόδου
 - `<όνομα αντικειμένου>.<όνομα μεθόδου>`
- Π.χ.

```
Light bedroomLight = new Light();  
bedroomLight.on();  
bedroomLight.brighten();  
bedroomLight.dim();  
bedroomLight.off();
```

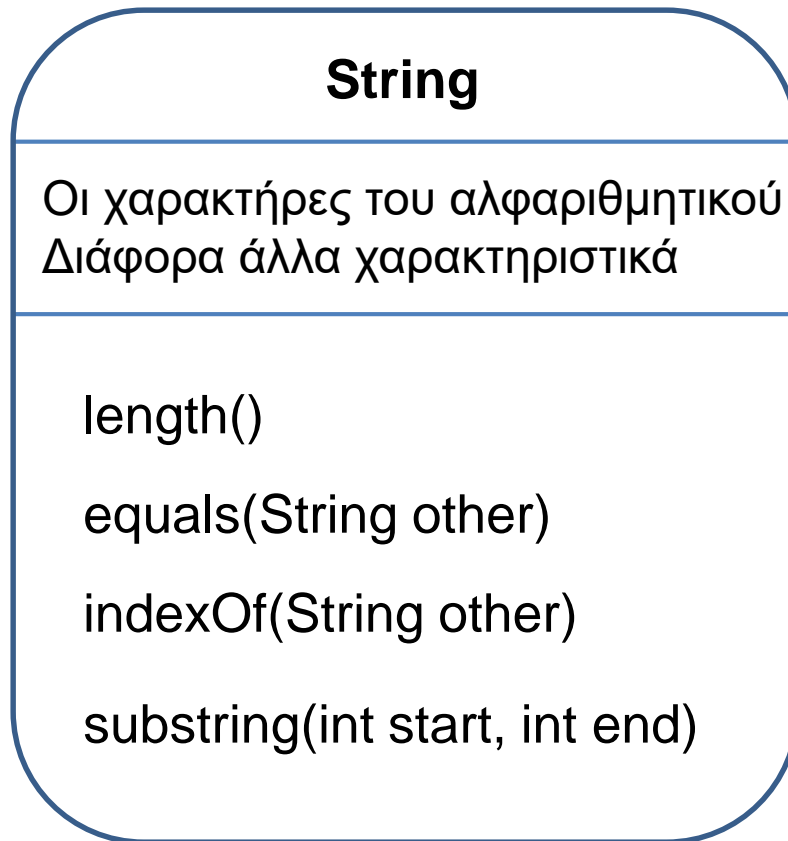

Κλήση μεθόδων

- Για να καλέσουμε μια μέθοδο μιας κλάσης θα πρέπει να δημιουργήσουμε ένα **αντικείμενο** της κλάσης
- Εξαίρεση: οι **στατικές** μέθοδοι
 - Μπορούν να κληθούν χρησιμοποιώντας το **όνομα της κλάσης**
- Παράδειγμα:
 - Η μέθοδος **main** που έχουμε δει καλείται χωρίς να έχουμε δημιουργήσει αντικείμενο
 - Γι αυτό και πρέπει να οριστεί ως **static**.
 - **System.out.println("hello") ;** καλεί την στατική μέθοδο **println** του αντικειμένου **System.out**

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

- Υπάρχουν πολλές χρήσιμες μέθοδοι της κλάσης String.
 - `length()`: μήκος του String
 - `equals(String x)`: τσεκάρει για ισότητα του String που καλεί την μέθοδο με το String x
 - `trim()`: αφαιρεί κενά στην αρχή και το τέλος του string.
 - `split(char delim)`: σπάει το string σε πίνακα από strings με βάσει τον χαρακτήρα delim.
 - `indexOf(String s)`: Επιστρέφει την θέση της πρώτης εμφάνισης του s μέσα στο String που καλεί την μέθοδο
 - `substring(int start, int end)`: Επιστρέφει το υπο-string μέσα στο String που καλεί την μέθοδο μεταξύ των θέσεων start και end
 - Κλπ.

Παράδειγμα

```
class StringExample{
    public static void main(String[] args){
        String x = new String("introduction to java programming");
        String y = "java";

        int offset = x.indexOf(y);
        int end = x.length();
        x = x.substring(offset,end);
        System.out.println(x);
    }
}
```

Τα Strings είναι **αμετάβλητα** (**immutable**) αντικείμενα
Η τελευταία ανάθεση δημιουργεί ένα **καινούριο**
αντικείμενο και το αναθέτει στην μεταβλητή x

Αμετάβλητα αντικείμενα

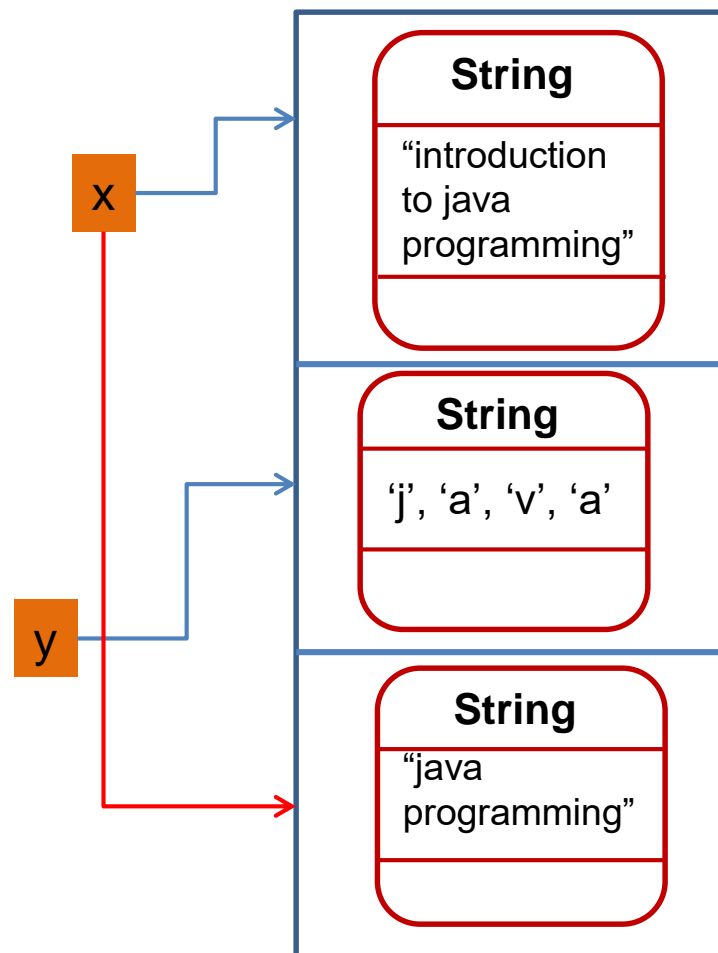
- Τα **αμετάβλητα** αντικείμενα (**immutable objects**) είναι αντικείμενα των οποίων η **εσωτερική κατάσταση** (ουσιαστικά τα πεδία τους) δεν μπορεί να μεταβληθεί.
- Τα **Strings** είναι **αμετάβλητα** αντικείμενα
 - Αυτό σημαίνει ότι δεν μπορούμε να αλλάξουμε τα περιεχόμενα ενός αντικειμένου String
 - Π.χ., δεν μπορούμε να αλλάξουμε ένα χαρακτήρα ενός String
 - Ότι αλλαγή κάνουμε έχει αποτέλεσμα να δημιουργείται ένα καινούριο String και να εκχωρείται στην μεταβλητή μας.

Αμετάβλητα αντικείμενα

```
String x = new String("introduction to java programming");  
String y = "java";  
x = x.substring(offset, end);
```

Τα Strings είναι **αμετάβλητα** (**immutable**) αντικείμενα

Η τελευταία ανάθεση δημιουργεί ένα **καινούριο** αντικείμενο και το αναθέτει στην μεταβλητή x



String σταθερές

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

```
import java.util.Scanner;

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

        int offset = "java programming".indexOf("pro");
        int end = "java programming".length();
        String z = "java programming".substring(offset,end);
        System.out.println(z);

    }
}
```

Scanner

- Δημιουργία αντικειμένου Scanner
 - `Scanner input = new Scanner(System.in) ;`
- Μέθοδοι της Scanner:
 - `next ()` : επιστρέφει το επόμενο String από την είσοδο (όλοι οι χαρακτήρες από το σημείο που σταμάτησε την προηγούμενη φορά μέχρι να βρει white space: κενό, tab, αλλαγή γραμμής)
 - `nextInt ()` : διαβάζει το επόμενο String και το μετατρέπει σε int και επιστρέφει ένα int αριθμό.
 - `nextDouble ()` : διαβάζει το επόμενο String και το μετατρέπει σε double και επιστρέφει τον double αριθμό.
 - `nextLine ()` : Διαβάζει ότι υπάρχει μέχρι να βρει newline και το επιστρέφει ως String.

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
- Κάποιες από τις μεθόδους είναι **στατικές**
 - Μπορούμε να τις καλέσουμε **χωρίς να έχουμε αντικείμενο**.
 - Έχουμε επίσης πρόσβαση σε κάποια **στατικά πεδία**.

Παράδειγμα

```
class WrapperTest{
    public static void main(String args[])
    {
        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);
    }
}
```