ΓΕΝΙΚΟΣ ΟΡΙΣΜΟΣ ΜΕΘΟΔΟΥ

```
[<προσδ. μεθόδου>] <επιστρ. τύπος>
      <όνομα μεθ.> ([<λίστα παραμ.>])
{
[<δηλώσεις τοπικών μεταβλητών>]
<προτάσεις java>
}
```

Δήλωση ή κεφαλίδα μεθόδου

Σώμα μεθόδου

Τύποι μεθόδων

- κλάσης
- στιγμιοτύπων

ΠΡΟΣΔΙΟΡΙΣΤΕΣ ΜΕΘΟΔΟΥ

- public, private, protected, static (όπως για τις μεταβλητές)
- abstract (σε αφαιρετική κλάση)
- final (μέθοδος που δεν μπορεί να επικαλυφθεί στις υποκλάσεις της κλάσης-σχετίζεται με την έννοια της κληρονομικότητας)

ΚΛΗΣΗ ΜΕΘΟΔΟΥ- ΠΕΡΑΣΜΑ/ΑΠΟΣΤΟΛΗ ΜΗΝΥΜΑΤΩΝ

- Ο μόνος τρόπος επικοινωνίας αντικειμένων (δηλ. κλήσης μιας μεθόδου για εκτέλεση)
- Σύνταξη

<όνομα αντικειμ.>.<όνομα μεθόδου>
(<λίστα παραμέτρων>)

Π.χ.
```
c.area( ) ;
c.increase_radius(0.5);
Circle.bigger(c1, c2);
```

όπου c στιγμιότυπο και Circle κλάση.

ΠΑΡΑΔΕΙΓΜΑ (1)

Μέθοδος στιγμιοτύπου

```
public class Circle {
   public double x, y, r ;
   public Circle bigger(Circle c)
      {if (c.r > r)
          return c;
       else return this;}
 }
```

Αν c1, c2 δύο στιγμιότυπα της Circle με r 2.0 και 5.0 αντίστοιχα και c μια μεταβλητή τύπου Circle, τότε οι

c = c1.bigger(c2) ή c = c2.bigger(c1)

δίνουν στη c την τιμή c2 (: η c γίνεται αναφορά στο c2).

ΠΑΡΑΔΕΙΓΜΑ (2)

Μέθοδος κλάσης

```
public class Circle {
   public double x, y, r ;
   public static Circle bigger(Circle a, Circle b)
      {if (a.r > b.r)
          return a;
       else return b;}
}
```

Αν c1, c2 δύο στιγμιότυπα της Circle με r 2.0 και 5.0 αντίστοιχα και c μια μεταβλητή τύπου Circle, τότε η

```
c = Circle.bigger(c1, c2)
```

δίνουν στη c την τιμή c2 (: η c γίνεται αναφορά στο c2).

ΔΗΜΙΟΥΡΓΙΑ ΑΝΕΝΕΡΓΟΥ ΣΤΙΓΜΙΟΤΥΠΟΥ

Circle c = new Circle ( ) ;

- Μέθοδος δημιουργίας στιγμιότυπου
- Ιδιαίτερος τύπος αποστολής μηνύματος

- Εξ' ορισμού ή αφανής δημιουργός (constructor)

- Μεταβλητή τύπου Circle (δηλ. αποτελεί αναφορά σε στιγμιότυπα της κλάσης Circle)

ΔΗΜΙΟΥΡΓΙΑ ΕΝΕΡΓΟΥ ΣΤΙΓΜΙΟΤΥΠΟΥ (ΔΗΜΙΟΥΡΓΟΣ)

```
public class Circle {
   private double x, y, r ;
   public Circle (double x, double y, double r) {
       this.x = x ;
       this.y = y ;
       this.r = r ;
   }
        .
        .
        .
}
```

(ίδιο όνομα)

Αναφορά στο δημιουργούμενο αντικείμενο

Δημιουργός (constructor)

```
Circle c = new Circle (10.0, 20.0, 2.0) ;
```

ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΔΗΜΙΟΥΡΓΟΥ

- Ίδιο όνομα με την κλάση, προσδιοριστή public
- Ειδική κατηγορία συνάρτησης (όχι μέθοδος). Καλείται ανεξάρτητα από την ύπαρξη στιγμιοτύπου της κλάσης
- Δεν ορίζεται επιστρεφόμενη τιμή (όχι void, όχι return)
- Όχι άμεση κλήση σ' ένα πρόγραμμα (μόνο έμμεση, κατά τη δημιουργία αντικειμένου)
- Χρήση της λέξης κλειδί 'this' (όταν τα ονόματα των ορισμάτων είναι ίδια με αυτά των μεταβλητών της κλάσης)
- Επιστρέφει έμμεσα την τιμή αναφοράς του 'this'

ΔΗΜΙΟΥΡΓΙΑ ΣΤΙΓΜΙΟΤΥΠΟΥ ΧΩΡΙΣ ΔΗΜΙΟΥΡΓΟ

- Δήλωση μεταβλητών ως public
- Αρχικοποίηση μέσω εντολών της μορφής

  <στιγμιότυπο>.<μεταβλητή>

```
public class Circle {
   public double x, y, r ;
}

Circle c = new Circle () ;
c.x = 10.0; c.y = 20.0; c.r = 2.0;
```

ΠΟΛΛΑΠΛΟΙ ΔΗΜΙΟΥΡΓΟΙ

```
public class Circle {
    private double x, y, r ;
    public Circle (double x, double y, double r) {
        this.x = x ; this.y = y ; this.r = r ;}
    public Circle (double r) {
        x = 0.0 ; y = 0.0 ; this.r = r ;}
    public Circle () {
        x = 0.0 ; y = 0.0 ; r = 1.0 ;}
    public Circle (Circle c) {
        x = c.x ; y = c.y ; r = c.r ;}
}
```

Υπερφόρτωση Μεθόδων (method overloading)

(αναγνώριση μεθόδου όχι μόνο από το όνομα, αλλά και τον τύπο των ορισμάτων)

ΤΕΛΕΣΤΗΣ 'THIS'- ΕΠΑΝΑΧΡΗΣΙΜΟΠΟΙΗΣΗ

```
public class Circle {
    private double x, y, r ;
    public Circle (double x, double y, double r) {
        this.x = x ; this.y = y ; this.r = r ;}
    public Circle (double r) {this (0.0 , 0.0 , r );}
    public Circle () {this (0.0, 0.0, 1.0) ;}
    public Circle (Circle c) {this (c.x, c.y, c.r) ;}
}
```

Ο τελεστής 'this' στο σώμα μιας μεθόδου μεταφέρει τον έλεγχο στη μέθοδο με το ίδιο όνομα και αντίστοιχα ορίσματα.