5. Αντικείμενα – Κλάσεις & Διασυνδέσεις

Δήλωση και δημιουργία αντικειμένου

ΔΗΛΩΣΗ

```
<όνομα_κλάσης> <όνομα_αντικειμένου>;

// ΠΑΡΑΔΕΙΓΜΑ:
MyFirstClass  anObject;
```

ΔΗΜΙΟΥΡΓΙΑ

```
<όνομα_αντικειμένου> = new <όνομα_κλάσης> ( [ορίσματα] );

// ΠΑΡΑΔΕΙΓΜΑ:
anObject = new MyFirstClass();
```

Παράδειγμα

```
public class SimplePoint {
    public int x = 0;
    public int y = 0;
}

public class SimpleRectangle {
    public int width = 0;
    public int height = 0;
    public SimplePoint origin = new SimplePoint();
}
```

*Η λέξη-κλειδί **public** πριν από τις μεταβλητές **x, y** δηλώνει πως αυτές μπορούν ελεύθερα να προσπελασθούν από οποιαδήποτε κλάση.*

Μέθοδος αρχικών συνθηκών (*"constructor"*)

```
public class Point {
   public int x = 0;
   public int y = 0;

   // Constructor
   public Point(int a, int b) { // δεν επιστρέφει τιμή αλλά δεν έχει void
      x = a; y = b;
   }
}
```

*Η **μέθοδος αρχικών συνθηκών** μιας κλάσης, συνήθως, δίνει αρχικές τιμές στις μεταβλητές της κλάσης, π.χ:* ***Point P= new Point (44, 32);***

Συμβουλές για μεθόδους αρχικών συνθηκών (*"constructors"*)

- Για κάθε νέα κλάση ***ορίζουμε πάντα μια μέθοδο αρχικών συνθηκων με κενό όρισμα η οποία δίνει αρχικές τιμές στις μεταβλητές που περιλαμβάνει η κλάση*** (μηδενίζει αριθμούς, ορίζει τις συμβολοσειρές ίσες με κενό, ορίζει αρχικό μέγεθος σε πίνακες κτλ.

- Σε περίπτωση που δεν το κάνουμε, η Java χρησιμοποιεί μια default συνάρτηση αρχικών συνθηκών που δεν φαίνεται τι ακριβώς κάνει.

- Στη συνέχεια, μπορείτε να ορίσετε όσες συναρτήσεις αρχικών συνθηκών θέλετε για την κλάση σας, ***αρκει να αρχικοποιούν όλες τις μεταβλητές της***.

Προσπέλαση των μελών αντικειμένου

```
<όνομα_αντικειμένου>.<όνομα_μέλους>
// μέλος μπορεί να είναι μια μεταβλητή ή μια μέθοδος
```

```
class PointsExample {
    int a;

    public static void main (String[] args) {
        Point origin = new Point (0,0);
        Point aPoint = new Point (22,63);
        origin.x = 10 + aPoint.x;
        origin.y = 30 + aPoint.y;
    }
}
```

Τροποποιητές πρόσβασης

Πρόσβαση	Προσδιοριστής	Κλάση	Υποκλάση	Πακέτο	Έξω κόσμος
private	**private**	☑			
protected	**protected**	☑	☑	☑	
public	**public**	☑	☑	☑	☑
package	-	☑	☑	☑	

*Όταν η κλάση **Β κληρονομεί** την κλάση **Α**, τότε οι **public** και οι **protected** μεταβλητές και μέθοδοι της **Α** κληρονομούνται ως **public** και **private** αντίστοιχα στην **Β**. Οι **private** της Α **δεν κληρονομούνται** στην **Β**.*

Στατικές Μεταβλητές

ΟΡΙΣΜΟΣ:

Στατική λέγεται μια μεταβλητή η οποία σχετίζεται με μια κλάση και όχι με τα επιμέρους στιγμιότυπα που προκύπτουν από την κλάση αυτήν.

ΔΗΛΩΣΗ

<τροπ_προσβ> static <τύπος> <όνομα1>, ... , <όνομαN> ;
public static int i;

*Όταν η τιμή μιας **στατικής μεταβλητής** αλλάξει μέσα σε κάποιο αντικείμενο της κλάσης που αυτή ορίζεται, τότε η τιμή της μεταβλητής αλλάζει και σε κάθε άλλο αντικείμενο της ίδιας κλάσης που έχει δημιουργηθεί.*

Μέθοδοι

Στιγμιότυπου

```
<τροπ_προσβ> <τύπος_επιστρεφόμενης_τιμής> <ονομα> ( <ορίσματα> ){
  // Σώμα μεθόδου
}
```

Κλάσης (στατική)

```
 <τροπ_προσβ> static <τύπος_επιστρεφόμενης_τιμής> <ονομα> ( <ορίσματα> ){
   // Σώμα μεθόδου
 }
```

Παραδείγματα

```
private float area( Point p ){
  // Σώμα μεθόδου
}
public static void main( String[] args ){
  // Σώμα μεθόδου
}
```

Προσπέλαση μεθόδων

Στιγμιότυπου

<όνομα_αντικειμένου>.<όνομα_μεθόδου> (<ορίσματα>);

```
Graphics g = new Graphics();
Point p = new Point (12, 82);
g.drawPoint(p);
```

Κλάσης (στατική)

<όνομα_κλάσης>.<όνομα_μεθόδου> (<ορίσματα>);

```
double x = 2.18234445532;
double sinx = Math.sin(x); // Math is a class name
```

this

- Η λέξη-κλειδί **this** χρησιμοποιείται για την προσπέλαση μελών (μεταβλητών και συναρτήσεων) του ιδίου αντικειμένου.

Παράδειγμα

```
public class AppThis{
   int a,b;


   public AppThis(int a, int b){
      this.a = a;
      this.b = b;
   }
}
```

Επικάλυψη και Υπέρβαση μεθόδων

- **Επικάλυψη (*overload*)**: Μια μέθοδος μπορεί να έχει διαφορετικές υλοποιήσεις με το ίδιο όνομα, αλλά με διαφορετικό αριθμό ή τύπο ορισμάτων.
- **Υπέρβαση (*override*)**: Μια μέθοδος που έχει κληρονομηθεί σε μια κλάση Β από την **υπερκλάση** της Α, μπορεί να έχει διαφορετική υλοποίηση στην Β.

Παραδείγμα

Επικάλυψη

```
void methodos(int y1, int y2){
    x1+=y1+y2+x3;
    System.out.println("X1="+x1+" X2="+x2);
    System.out.println("X3="+x3+" X4="+x4);
}

void methodos(int y1, int y2, int y3){
      x1+=y1*y2+x4;
      x2=y3*x3;
      System.out.println("X1="+x1+" X2="+x2);
      System.out.println("X3="+x3+" X4="+x4);
}
```

Παραδείγμα

Υπέρβαση

```
class AppFirst{
   int x1=12;
   void methodos( ){
      System.out.println( “X1 =” + x1);
   }
}


class AppSecond extends AppFirst{
   void methodos( ){
      System.out.println( “2*X1 =” + ( 2 * x1 ) );
   }
   public static void main(String[] args){
      AppSecond sec = new AppSecond();
      sec.methodos();
   }
}
```

*Όταν δεν αναφέρεται ο **τροποποιητής τύπου** στη δήλωση μιας μεταβλητής ή στον ορισμό μιας μεθόδου, τότε εννοείται η λέξη **package.***

super

- Η δεσμευμένη λέξη-κλειδί ***super*** χρησιμοποιείται για να **καλέσουμε συναρτήσεις της υπερκλάσης από μια θυγατρική κλάση.**
- Όταν δημιουργείται ένα αντικείμενο μίας κλάσης, η μέθοδος δημιουργού της κλάσης καλείται αυτόματα. **Επίσης καλούνται αυτόματα και οι μέθοδοι αρχικών συνθηκών όλων των υπερκλάσεων αυτής.**
- Για να καλέσουμε τη μέθοδο αρχικών συνθηκων της υπερκλάσης χρησιμοποιούμε τη λέξη ***super*** και σε παρενθέσεις τα επιθυμητά ορίσματα. **Είναι υποχρεωτικό να βρίσκεται στην πρώτη γραμμή εντολών του σώματος.**

  π.χ. *super(5);*

Παραδείγμα

Υπέρβαση δημιουργού

```
class Myclass{
   int p,m;
   Myclass( ){
      System.out.println( “My 1st constructor!”);
   }
   Myclass(int k, int l ){
      System.out.println( “My 2nd constructor!”);
      p=k; m=l;
   }
}

class Mynewclass extends Myclass{
   int x1;
   Mynewclass(int x){
      super(2,9);
      x1=x;
   }
   public static void main(String[] args){
      Mynewclass A = new Mynewclass(-7);
      System.out.println( “x1=”+x1);
   }
}
```

Η ***super(…)*** *πρέπει να είναι υποχρεωτικά η πρώτη εντολή στο σώμα της μεθόδου δημιουργού που την καλεί.*

super

- Εκτός από τις μεθόδους αρχικών συνθηκών, η δεσμευμένη λέξη-κλειδί ***super*** χρησιμοποιείται για να καλέσουμε και κανονικές συναρτήσεις της υπερκλάσης από μια θυγατρική κλάση.

  π.χ. *super.doubleSpace(10);*
- Σε αυτή τη περίπτωση η εντολή μπορεί να βρίσκεται σε οποιοδήποτε σημείο του κώδικα, όχι υποχρεωτικά στην πρώτη γραμμή.

Παραδείγματα

Υπέρβαση, super

```
class AppFirst{
   int x1=12;
   void methodos( ){
      System.out.println( “X1 =” + x1);
   }
}

class AppSecond extends AppFirst{
   void methodos( ){
      System.out.println(“New Function”);
      super.methodos( );
      System.out.println( “2*X1 =” + ( 2 * x1 ) );
   }
   public static void main(String[] args){
      AppSecond sec = new AppSecond();
      sec.methodos();
   }
}
```

Μετατροπή τύπου

- Μεταξύ δεδομένων (int, float, κλπ.)
- Μεταξύ αντικειμένων
- Τύπου δεδομένων σε αντικείμενο

```
\\Μετατροπή  από μικρότερη σε μεγαλύτερη ακρίβεια (προσθέτω μηδένικά)
int c = 10;
float a = (float) c;  // a = 10.000

\\Μετατροπή από μεγαλύτερη σε μικρότερη ακρίβεια (κόβω δεκαδικά)
float c = 5.999;
int a = (int) c; // a = 5

//  Μετατροπή String σε ακέραιο αριθμό
String aString = “234”;
int a = Integer.parseInt(aString);
```

Ο τροποποιητής final

Όταν μια μεταβλητή / μέθοδος / κλάση χαρακτηρίζεται από τον τροποποιητή **final,** δεν μπορεί να υποστεί υπέρβαση (να μεταβληθεί από μια υποκλάση) ή τροποποίηση (να αλλάξει τιμή).

```
public final int douzen = 12;


final void myFinalMethod (int a, int b){
  // java source
}


public final myFinalClass {
    void methodos( ){
        System.out.println( “2*X1 =” + ( 2 * x1 ) );
    }
}
```