

# Εισαγωγή στην Κληρονομικότητα

Δημήτρης Κ. Ιακωβίδης

Στόχος αυτής της ενότητας είναι η εισαγωγή στην κληρονομικότητα (inheritance), δηλαδή στη δυνατότητα των κλάσεων να κληρονομούν μεταβλητές υπόστασης και μεθόδους από άλλες κλάσεις.

## 10.1 Υποκλάσεις

Έστω ότι επιθυμούμε να κατασκευάσουμε μια νέα κλάση B η οποία έχει ίδιες μεταβλητές υπόστασης και μεθόδους με μια κλάση A, αλλά έχει επιπλέον και κάποιες νέες μεταβλητές υπόστασης και μεθόδους. Για να το αντιμετωπίσουμε αυτό το θέμα στον αντικειμενοστρεφή προγραμματισμό μπορούμε να δηλώσουμε ότι η B «επεκτείνει» (extends) την A, ώστε αυτομάτως η B να κληρονομήσει (inherit) τα στοιχεία της A. Έτσι δε χρειάζεται να υλοποιήσουμε ξανά τα στοιχεία της A, και μάλιστα στη Java δε χρειάζεται ούτε να έχουμε ούτε τον κώδικα Java της κλάσης A.

Για παράδειγμα στο Σχ. 1 εικονίζεται η κλάση Date που αναπαριστά μια ημερομηνία. Στο Σχ. 2 η κλάση DateTime επεκτείνει την κλάση Date με την ώρα. Η DateTime με τη χρήση της έκφρασης “extends” στον ορισμό της, κληρονομεί όλα τα στοιχεία της Date, και την επεκτείνει με τρεις επιπλέον νέες μεταβλητές. Η DateTime ονομάζεται **υποκλάση (subclass)** ή **παράγωγη κλάση (derived class)** της κλάσης Date. Η κλάση Date ονομάζεται **υπερκλάση (superclass)** της DateTime. Μια κλάση στη Java μπορεί να έχει μόνο μια υπερκλάση (μονή κληρονομικότητα). Σε άλλες αντικειμενοστρεφείς γλώσσες προγραμματισμού, όπως είναι η C++, υποστηρίζεται και πολλαπλή κληρονομικότητα, δηλαδή, υπάρχει δυνατότητα μια κλάση να διαθέτει περισσότερες υπερκλάσεις.

Αναφορικά με την υλοποίηση της κληρονομικότητας η οποία ενδεικτικά περιγράφεται στο Σχ. 2, βοηθάει να θυμάστε την έκφραση:

**“Η DateTime είναι και Date”**

Όμως η Date δεν είναι και DateTime εφόσον η DateTime περιέχει περισσότερα στοιχεία από την Date, δηλαδή την ώρα, τα λεπτά και τα δευτερόλεπτα.

```
1 class Date
2 {
3     int day, month, year;
4 }
```

**Σχήμα 1.** Μια πολύ απλή κλάση που αναπαριστά μια ημερομηνία.

```
1 class DateTime extends Date
2 {
3     int hour, minute, second;
4
5     public static void main(String[] args)
6     {
7         DateTime dt = new DateTime();
8
9         // DateTime inherits day, month, year
10        dt.day = 20;
11        dt.month = 10;
12        dt.year = 2010;
13
14        // DateTime has also hour, minute, second
15        dt.hour = 8;
16        dt.minute = 12;
17        dt.second = 25;
18
19        System.out.println(dt.day + "/" + dt.month +
20        "/" + dt.year + " " + dt.hour + ":" + dt.minute
21        + ":" + dt.second);
22    }
```

**Σχήμα 2.** Η κλάση DateTime είναι μια υποκλάση της κλάσης Date του Σχ.1. Δηλαδή, την επεκτείνει με τρεις νέες μεταβλητές υπόστασης, τις hour, minute, second, και τη μέθοδο main.

**Προσοχή!** Οι ιδιωτικές (private) μεταβλητές υπόστασης και μέθοδοι μιας κλάσης δεν είναι άμεσα προσπελάσιμες από τις υποκλάσεις της. Είναι προσπελάσιμες μόνο μέσω μεθόδων προσπέλασης και μεταλλαγής.

## 10.2 Μέθοδοι δημιουργοί υποκλάσεων

Στις μεθόδους δημιουργούς υποκλάσεων πρέπει πάντα στην πρώτη γραμμή τους να καλείται η μέθοδος δημιουργός της υπερκλάσης τους. Αυτό γίνεται με τη χρήση της έκφρασης **super**, η οποία είναι ανάλογη της έκφρασης **this** και αναφέρεται στην υπερκλάση (superclass). Δηλαδή, όταν χρησιμοποιείται η **super** ακολουθούμενη

από τελεία, π.χ. **super.day**, αναπαριστά ένα αντικείμενο της υπερκλάσης της κλάσης μας, ενώ όταν χρησιμοποιείται ακολουθούμενη από παρενθέσεις, π.χ., **super()**, αναπαριστά μια μέθοδο δημιουργό της υπερκλάσης της κλάσης μας. Στα Σχ.3-4 εικονίζονται τα παραδείγματα των Σχ.1-2 εμπλουτισμένα με μεθόδους δημιουργούς.

```
1  class Date
2  {
3      int day, month, year;
4
5      Date(int d, int m, int y)
6      {
7          day = d;
8          month = m;
9          year = y;
10     }
11 }
```

Σχήμα 3. Η κλάση του Σχ. 1 εμπλουτισμένη με μέθοδο δημιουργό.

```
1  class DateTime extends Date
2  {
3      int hour, minute, second;
4
5      DateTime(int d, int mo, int y, int h, int m, int s)
6      {
7          super(d, mo, y); // should be always first!
8
9          hour = h;
10         minute = m;
11         second = s;
12     }
13
14     public static void main(String[] args)
15     {
16         DateTime dt = new DateTime(8,4,2014,6,19,0);
17         System.out.println(dt.day + "/" + dt.month +
18         "/" + dt.year + " " + dt.hour + ":" + dt.minute
19         + ":" + dt.second);
20     }
```

Σχήμα 4. Η κλάση του Σχ.2 εμπλουτισμένη με μέθοδο δημιουργό.

**Προσοχή!** Αν δεν προσθέσετε την έκφραση **super** που διακρίνεται στη γραμμή 7 του Σχ. 4, τότε η Java καλεί αυτομάτως τη **super()**, δηλαδή τη μέθοδο δημιουργό της **Date** που δεν έχει ορίσματα. Όμως, αν παρατηρήσετε στο Σχ. 3, δεν υπάρχει τέτοια

μέθοδος δημιουργός μέσα στην Date, με αποτέλεσμα να μη μπορεί να γίνει μεταγλώττιση του προγράμματος.

### 10.3 Υπέρβαση μεθόδων

Μια υποκλάση κληρονομεί όλες τις μεθόδους της υπερκλάσης της. Αν η υποκλάση έχει μια μέθοδο που ορίζεται ακριβώς με τον ίδιο τρόπο (δηλαδή έχει την ίδια επικεφαλίδα) με μια μέθοδο της υπερκλάσης της, τότε η μέθοδος αυτή **υπερβαίνει (overrides)** τη μέθοδο της υπερκλάσης. Αυτό σημαίνει ότι για την υποκλάση καταργείται η αντίστοιχη μέθοδος της υπερκλάσης και θα ισχύει μόνο η δική της, η οποία ορίστηκε τελευταία.

Στα Σχ.5-6 οι κλάσεις των Σχ.3-4 έχουν εμπλουτιστεί με μέθοδο toString. Η μέθοδος toString της DateTime υπερβαίνει τη μέθοδο toString της Date. Έτσι για την DateTime θα ισχύει η δική της toString, και όχι η toString της Date.

Ωστόσο, εάν το επιθυμούμε, εξακολουθούμε να μπορούμε να καλούμε τη μέθοδο toString της Date μέσα από τη DateTime. Αυτό επιτυγχάνεται με τη χρήση της έκφρασης **super** ως αντικείμενο κλήσης (δηλαδή με τελεία δεξιά). Στο Σχ.6 η κλήση super.toString() καλεί την toString της κλάσης Date.

```
1  class Date
2  {
3      int day, month, year;
4
5      Date(int d, int m, int y)
6      {
7          day = d;
8          month = m;
9          year = y;
10     }
11
12     public String toString()
13     {
14         return day + "/" + month + "/" + year;
15     }
16 }
```

Σχήμα 5. Η κλάση του Σχ. 3 εμπλουτισμένη με μέθοδο toString.



```

1  class DateTime extends Date
2  {
3      int hour, minute, second;
4
5      DateTime(int d, int mo, int y, int h, int m, int s)
6      {
7          super(d, mo, y); // should be always first!
8
9          hour = h;
10         minute = m;
11         second = s;
12     }
13
14     // This toString overrides Date's toString
15     public String toString()
16     {
17         // The following line calls toString of Date
18         return super.toString() + " " + hour + ":" +
19         minute + ":" + second;
20     }
21
22     public static void main(String[] args)
23     {
24         DateTime dt = new DateTime(8,4,2014,6,19,0);
25         System.out.println(dt);
26     }

```

Σχήμα 6. Η κλάση του Σχ. 4 εμπλουτισμένη με μέθοδο toString.

### Ασκήσεις

1. Να υλοποιήσετε και να δοκιμάσετε τα παραδείγματα των Σχ.5-6. Κατόπιν, να κατασκευάσετε την κλάση AccurateDateTime ως υποκλάση της κλάσης DateTime, η οποία να επεκτείνει την DateTime με χιλιοστά του δευτερολέπτου (millisecond). Στην AccurateDateTime να προστεθούν:

α) Μια μέθοδος δημιουργός που να δέχεται ως ορίσματα την ημέρα, το μήνα, το έτος, την ώρα, τα λεπτά, τα δευτερόλεπτα και τα χιλιοστά του δευτερολέπτου.

β) Μια μέθοδος toString που να συμπεριλαμβάνει και τα χιλιοστά του δευτερολέπτου.

2. Να κατασκευαστεί μια κλάση DateEvent που να επεκτείνει τη Date με το όνομα ενός συμβάντος, π.χ. μιας εορτής που αντιστοιχεί σε μια ημερομηνία. Να δοκιμάσετε την κλάση κατασκευάζοντας ένα αντικείμενο e1 της DateEvent για την

25<sup>η</sup> Μαρτίου με όνομα συμβάντος το «Επέτειος 1821», και ένα αντικείμενο e2 για την 28<sup>η</sup> Οκτωβρίου του 2014 με όνομα συμβάντος το «Επέτειος 1940».

3. Να κατασκευαστεί μια κλάση Point2D που αναπαριστά ένα σημείο στο επίπεδο (δηλ. στις δύο διαστάσεις). Κατόπιν να κατασκευαστεί μια υποκλάση της, με όνομα Point3D που αναπαριστά ένα σημείο στο χώρο (δηλ. στις τρεις διαστάσεις). Να προστεθούν μέθοδοι δημιουργοί και toString.

4. Να κατασκευαστεί μια κλάση Product που αναπαριστά ένα προϊόν. Να κατασκευαστεί μια υποκλάση της Product η οποία να διαθέτει και ημερομηνία λήξεως. Η ημερομηνία να αναπαρασταθεί με αντικείμενο της κλάσης Date.