

ΣΥΝΘΕΤΕΣ ΚΛΑΣΕΙΣ ΚΑΙ ΜΕΘΟΔΟΙ

Δημήτρης Κ. Ιακωβίδης

Στόχος αυτής της ενότητας είναι η εφαρμογή των γνώσεων που αποκτήθηκαν στα προηγούμενα μαθήματα για την κατασκευή σύνθετων κλάσεων και μεθόδων. Ως σύνθετες νοούνται οι κλάσεις στις οποίες γίνεται ευρεία χρήση αντικειμένων άλλων κλάσεων, ως μεταβλητές υπόστασης, ως ορίσματα ή ως τύποι επιστροφής μεθόδων κλπ.

7.1 Κλάσεις με μεταβλητές υπόστασης τύπου κλάσης

Στη κλάση Human (Σχ.1) η μεταβλητή name είναι τύπου κλάσης (String). Ωστόσο η μεταβλητή αυτή δεν απαιτούσε ιδιαίτερο χειρισμό, γιατί η δημιουργία ενός αντικειμένου τύπου String μπορεί να γίνει με απλή ανάθεση τιμής όπως γίνεται στις μεταβλητές πρωταρχικού τύπου, δηλαδή χωρίς την έκφραση new.

```
1  class Human
2  {
3      String name;
4      int height;
5
6      Human(String n, int h)
7      {
8          name = n;
9          height = h;
10     }
11
12     public String toString()
13     {
14         return name + " " + height;
15     }
16 }
```

Σχήμα 1. Κλάση Java που περιγράφει σύντομα τι είναι “ανθρωπάκι”.

Το θέμα που πραγματεύεται αυτή η ενότητα είναι η δημιουργία κλάσεων με μεταβλητές υπόστασης τύπου κλάσης που **χρειάζεται η έκφραση new** για τη δημιουργία αντικειμένων.

Έστω μια κλάση που αναπαριστά μια ευθεία. Όπως γνωρίζουμε από τη γεωμετρία, μια ευθεία ορίζεται από δύο σημεία. Για το λόγο αυτό, προκειμένου να υλοποιηθεί μια κλάση `StraightLine` που αναπαριστά μια ευθεία, θα χρησιμοποιήσουμε ως μεταβλητές υπόστασης δύο μεταβλητές τύπου `Point`, όπου `Point` μια κλάση που αναπαριστά ένα σημείο στο επίπεδο. Η κλάση `Point` και η κλάση `StraightLine` εικονίζονται στα Σχ.2 και 3 αντίστοιχα.

```
1 class Point
2 {
3     int x, y;
4
5     Point(int x, int y)
6     {
7         this.x = x;
8         this.y = y;
9     }
10
11    public String toString()
12    {
13        return "(" + x + "," + y + ")";
14    }
15 }
```

Σχήμα 2. Μια κλάση που αναπαριστά ένα σημείο στο επίπεδο με συντεταγμένες x, y.

```
1 class StraightLine
2 {
3     // Instance variables of class type Point
4     private Point a, b;
5
6     // Constructor based on coordinates
7     public StraightLine(int x1, int y1, int x2, int y2)
8     {
9         this.a = new Point(x1, y1);
10        this.b = new Point(x2, y2);
11    }
12
13    // Constructor based on point objects
14    public StraightLine(Point a, Point b)
15    {
16        this.a = new Point(a.x, a.y);
17        this.b = new Point(b.x, b.y);
18    }
19
20    // Calculates the length of the line
21    public double length()
22    {
23        // Returns the Euclidean distance of the points
24        double d1 = a.x-b.x;
25        double d2 = a.y-b.y;
```

```

26         return Math.sqrt(d1*d1+d2*d2);
27     }
28
29     // Accessor method - returns an independent point
30     public Point getPointA()
31     {
32         Point first = new Point(a.x, a.y);
33         return first;
34
35         // if you use return a; the point will be dependent
36     }
37
38     // Accessor method - returns an independent point
39     public Point getPointB()
40     {
41         Point second = new Point(b.x, b.y);
42         return second;
43     }
44
45     // Constructs a new Straightline with inverted
46     // points, i.e. exchanges point a with point b
47     public StraightLine inverted()
48     {
49         StraightLine r = new StraightLine(b, a);
50         return r;
51     }
52 }
53
54 public String toString()
55 {
56     return "Line from " + a.toString() + " to " +
57         b.toString();
58 }
59
60 public static void main(String[] args)
61 {
62     // Construct a line from coordinates
63     StraightLine e1 = new StraightLine(1,5,8,2);
64
65     // Construct a line from two point objects
66     Point p1 = new Point(2, 3);
67     Point p2 = new Point(5, 6);
68     StraightLine e2 = new StraightLine(p1, p2);
69
70     // Get the length of a line
71     System.out.println("The first line has a length
72     of " + e1.length() + " points");
73
74     // Get the first point
75     Point c = e1.getPointA();
76
77     // Get an inverted straight line from e2
78     StraightLine e3 = e2.inverted();
79 }
80 }

```

Σχήμα 3. Μια κλάση που αναπαριστά μια ευθεία, η οποία ορίζεται από δύο σημεία του επιπέδου a και b.

Η κλάση του Σχ.3 περιέχει τη μέθοδο `length` η οποία επιστρέφει το μήκος της ευθείας ως τιμή πρωταρχικού τύπου `double`. Περιέχει όμως και ένα πλήθος μεθόδων που επιστρέφουν αντικείμενα διαφορετικών κλάσεων. Αυτές είναι οι μέθοδοι `getPointA` και `getPointB` που επιστρέφουν τα σημεία `a` και `b`, τη μέθοδο `inverted` που επιστρέφει μια νέα ευθεία με τα σημεία της ανεστραμμένα, και μια μέθοδο `toString` που επιστρέφει την αλφαριθμητική αναπαράσταση της κλάσης `StraightLine`.

Παρατηρείστε ότι για να επιστρέφει μια μέθοδος αντικείμενο μιας κλάσης πρέπει το αντικείμενο αυτό να δημιουργηθεί μέσα στη μέθοδο. Επίσης, κατά την κλήση της μεθόδου από τη μέθοδο `main` δε χρειάζεται περαιτέρω δημιουργία αντικειμένου με τη χρήση της λέξης κλειδί `new`.

7.2 Εξαρτημένα αντικείμενα

Αν στην κλάση του Σχ.3 οι μέθοδοι `getPointA` και `getPointB` δεν περιελάμβαναν την πρόταση `new`, για τη δημιουργία νέου αντικειμένου, αλλά απλά επέστρεφαν τα σημεία `a` (`return a;`) και `b` (`return b;`) αντίστοιχα, τότε τα επιστρεφόμενα αντικείμενα θα ήταν εξαρτημένα από τα πρώτα. Δηλαδή, αν κάνει κανείς αλλαγή στο επιστρεφόμενο αντικείμενο `c` (βλ. `main`) τότε θα άλλαζε και το σημείο `a` της ευθείας `e1`.

Αυτό θα συνέβαινε γιατί οι μέθοδοι `getPointA` και `getPointB` θα επέστρεφαν αναφορές προς τα υπάρχοντα αντικείμενα `a` και `b` αντίστοιχα και δε θα δημιουργούσαν νέα.

Ασκήσεις

1. Να υλοποιηθούν τα παραπάνω παραδείγματα.
2. Να υλοποιηθεί η παραλλαγή που περιγράφει η ενότητα 7.2. Να επαληθευτεί ότι αλλάζοντας το αντικείμενο `c` θα αλλάξει και το πρώτο σημείο (`a`) της ευθείας `e1`.
3. Να κατασκευαστεί κλάση που αναπαριστά έναν κύκλο, έχοντας ως πρότυπο την κλάση `StraightLine`. Ένας κύκλος ορίζεται από το κέντρο του και την ακτίνα του.

4. Σε ένα δοκιμαστικό πρόγραμμα να κατασκευάσετε ένα πολύγωνο ως πίνακα από αντικείμενα της StraightLine.