

ΕΣΩΤΕΡΙΚΕΣ ΚΛΑΣΕΙΣ, ΓΕΝΙΚΕΥΜΕΝΕΣ ΚΛΑΣΕΙΣ ΚΑΙ ΛΙΣΤΕΣ

Δημήτρης Κ. Ιακωβίδης

Στόχος αυτής της ενότητας είναι η γνωριμία με τις έννοιες των **εσωτερικών κλάσεων** (inner classes), των **γενικευμένων κλάσεων** (generic classes) και των **λιστών** (lists) αντικειμένων. Οι τελευταίες στη Java υλοποιούνται με τη χρήση γενικευμένων κλάσεων και αποτελούν αντιπροσωπευτικό παράδειγμα για αυτές.

9.1 Εσωτερικές κλάσεις

Η Java δίνει τη δυνατότητα να ορίζουμε κλάσεις, εσωτερικά μέσα σε άλλες κλάσεις. Οι **εσωτερικές κλάσεις** (inner classes) αποσκοπούν κυρίως στην κατασκευή βοηθητικών αντικειμένων που χρησιμοποιούνται για «μικροδουλίτσες» μέσα στις κλάσεις που τις περιέχουν.

- Οι εσωτερικές κλάσεις έχουν πρόσβαση στα στοιχεία της εξωτερικής τους κλάσης (outer class) και αντιστρόφως, ακόμα και στα ιδιωτικά (private).
- Μια **βοηθητική εσωτερική κλάση** δηλώνεται συνήθως ως **private**.
- Σπανιότερα, μια εσωτερική κλάση μπορεί και να μη δηλωθεί ως **private** και τα στοιχεία της να είναι προσπελάσιμα και από άλλες κλάσεις.

Προσοχή! Οι εσωτερικές κλάσεις δεν είναι υποκλάσεις.

Στο παράδειγμα του Σχ. 1 περιγράφεται μια κλάση που αναπαριστά έναν τραπεζικό λογαριασμό (bank account), όπως ένα βιβλιάριο τραπεζικού λογαριασμού. Ο τραπεζικός λογαριασμός μπορεί να περιλαμβάνει πολλές συναλλαγές (transactions). Για μεγαλύτερη ακρίβεια στις συναλλαγές, αντί για τον τύπο **double**, του οποίου η ακρίβεια είναι πολύ περιορισμένη, χρησιμοποιείται ένας βοηθητικός τύπος που κατασκευάσαμε ειδικά για το συγκεκριμένο παράδειγμα. Ο τύπος αυτός περιγράφεται στην **εσωτερική κλάση AccurateNumber**, και επιτρέπει την αναπαράσταση χρηματικών ποσών με μεγαλύτερη ακρίβεια από τον double, χρησιμοποιώντας δύο μεγάλους ακέραιους (**long**). Η κλάση AccurateNumber δηλώθηκε ως εσωτερική γιατί πρόκειται να χρησιμοποιηθεί μόνο από τη BankAccount και από καμιά άλλη. Η χρήση της AccurateNumber θα κάνει πιο εύκολη την υλοποίηση και κατανόηση του κώδικα της κλάσης του τραπεζικού λογαριασμού BankAccount.

```

1  public class BankAccount
2  {
3      AccurateNumber[] transaction;
4      int currentTransaction;
5
6      public BankAccount(int numberOfTransactions)
7      {
8          transaction = new
9              AccurateNumber[numberOfTransactions];
10
11             currentTransaction = 0;
12         }
13
14         public void deposit(long amountEuros,
15                             long amountCents)
16         {
17             transaction[currentTransaction] = new
18                 AccurateNumber();
19             transaction[currentTransaction].euros =
20                 amountEuros;
21             transaction[currentTransaction].cents =
22                 amountCents;
23
24             currentTransaction++;
25         }
26
27         public String toString()
28         {
29             String s = "";
30             for(int i = 0; i < currentTransaction; i++)
31             {
32                 s += transaction[i].euros + "." +
33                     transaction[i].cents + "\n";
34             }
35             return s;
36         }
37
38         // Inner class
39         private class AccurateNumber
40         {
41             long euros, cents;
42         }
43
44         public static void main(String[] args)
45         {
46             BankAccount b = new BankAccount(100);
47             b.deposit(1000,20);
48             b.deposit(100,21);
49
50             System.out.println(b);
51
52         }
53     }

```

Σχήμα 1. Κλάση που αναπαριστά τραπεζικό λογαριασμό. Περιέχει την εσωτερική κλάση `AccurateNumber`, η οποία αναπαριστά χρηματικά ποσά με μεγάλη ακρίβεια.

9.2 Γενικευμένες κλάσεις

Οι **γενικευμένες κλάσεις** (generic classes) δίνουν τη δυνατότητα να ορίζουμε ως παραμέτρους τους τύπους των μεταβλητών υπόστασής τους, ή/και μεθόδων τους. Ο παραμετρικά οριζόμενος τύπος εισάγεται δίπλα από το όνομα της κλάσης με τη χρήση οξυγώνιων παρενθέσεων <>. Στο Σχ.2 εικονίζεται αντιπροσωπευτικό παράδειγμα τέτοιας κλάσης. Η κλάση GenericClass είναι τόσο γενική που μπορεί να χρησιμοποιηθεί εξίσου, είτε με ακέραιους είτε με αλφαριθμητικά στη θέση του τύπου T. Παράδειγμα χρήσης της εικονίζεται στο Σχ. 3. Πρέπει να γνωρίζουμε ότι:

- Ο τύπος T πρέπει να είναι τύπος κλάσης. Αν επιθυμούμε να έχουμε πρωταρχικό τύπο, π.χ. int, θα πρέπει να χρησιμοποιήσουμε την αντίστοιχη περιβάλλουσα κλάση, π.χ. την κλάση Integer αντί για τον τύπο int.
- Μια γενικευμένη κλάση μπορεί να έχει περισσότερες από μια παραμέτρους, π.χ. GenericClass<T1,T2>.

```
1 public class GenericClass<T>
2 {
3     T x;
4
5     public void print()
6     {
7         System.out.println(x);
8     }
9 }
```

Σχήμα 2. Γενικευμένη κλάση με παράμετρο T.

```
1 public class GenericTest
2 {
3     public static void main(String[] args)
4     {
5         GenericClass<Integer> a = new
6             GenericClass<Integer>();
7         a.x = 1230;
8         a.print();
9
10        GenericClass<String> b = new
11            GenericClass<String>();
12        b.x = "Hello Generics";
13        b.print();
14    }
15 }
```

Σχήμα 3. Παράδειγμα χρήσης της γενικευμένης κλάσης του Σχ. 2.

9.3 Λίστες

Οι λίστες είναι δομές δεδομένων στις οποίες αποθηκεύονται ακολουθίες δεδομένων. Λειτουργούν όπως οι πίνακες με τη διαφορά ότι επιτρέπεται η επέκτασή τους με νέα δεδομένα ή η περικοπή τους (δηλαδή επιτρέπεται δυναμικά η αύξηση ή η μείωση του μήκους τους) ανά πάσα στιγμή χωρίς να χάνονται τα δεδομένα τους (ενώ στην περίπτωση των πινάκων χάνονται αν δε διαχειριστούμε οι ίδιοι τα δεδομένα τους). Στη Java οι λίστες υλοποιούνται μέσω της γενικευμένης κλάσης `ArrayList` ή της κλάσης `Vector`, του πακέτου `java.util`.

Στο Σχ. 4 εικονίζεται ένα παράδειγμα χρήσης της κλάσης `ArrayList` (γραμμές 18-55) το οποίο αντιπαραβάλλεται με αυτό της χρήσης των πινάκων (γραμμές 7-16). Οι πίνακες μπορούν να αλλάζουν μέγεθος δυναμικά μέσα στο πρόγραμμα με επαναδέσμευσή τους (χρησιμοποιώντας `new`), αλλά τα περιεχόμενά τους χάνονται στη διαδικασία αυτή.

Ενδεικτικές μέθοδοι (τις υπόλοιπες μπορείτε να τις εξερευνήσετε στο API της Java):

- `add(element)` – προσθέτει ένα νέο στοιχείο (`element`) στο τέλος της λίστας.
- `add(pos,element)` – προσθέτει ένα νέο στοιχείο στη θέση `pos` της λίστας.
- `set(pos,element)` – θέτει νέο στοιχείο στην υπάρχουσα θέση `pos` της λίστας.
- `remove(pos)` – διαγράφει όποιο στοιχείο βρίσκεται στη θέση `pos`.
- `remove(element)` – διαγράφει το στοιχείο `element` (την πρώτη εμφάνισή του)

```
1  import java.util.ArrayList;
2
3  class TestArrayList
4  {
5      public static void main(String[] args)
6      {
7          // How we do it with arrays
8          int[] a = {5, 1, 2};
9
10         // We can change length later (dynamic
11         // allocation)
12         a = new int[10];
13
14         // But the contents of array 'a' are lost after
15         // new!
16
17         // How we do it with lists
18         ArrayList<String> list = new
```

```

20                                     ArrayList<String>(1);
21
22                                     // Add new elements to the list
23                                     list.add("Maria");
24                                     list.add("George");
25                                     list.add("John");
26                                     list.add("Costas");
27
28                                     System.out.println("Before...");
29
30                                     // Classic 'for' loop to see what's in the list
31                                     for (int i = 0; i < list.size(); i++)
32                                     {
33                                         System.out.println(list.get(i));
34                                     }
35
36                                     // Change the second element to 'Anna'
37                                     list.set(2, "Anna");
38
39                                     // Change the first element to 'Ioanna'
40                                     list.add(1, "Ioanna");
41
42                                     // Remove the second element
43                                     list.remove(2);
44
45                                     // Remove the element 'Costas'!!!
46                                     list.remove("Costas");
47
48                                     System.out.println("After...");
49
50                                     // New 'for' loop to see what's in the list
51                                     for (String element:list)
52                                     {
53                                         System.out.println(element);
54                                     }
55     }
56 }

```

Σχήμα 4. Παράδειγμα χρήσης της γενικευμένης κλάσης ArrayList.

Ασκήσεις

1. Να υλοποιήσετε το παράδειγμα του Σχ. 1.
2. Να τροποποιήσετε το παράδειγμα του Σχ.1 ώστε να χρησιμοποιείται λίστα αντί για πίνακας.
3. Να κατασκευαστεί μια γενικευμένη κλάση NumericType που αναπαριστά έναν αριθμό ανεξαρτήτως αν αυτός είναι ακέραιος, δεκαδικός κλπ. Να δοκιμάσετε να συμπεριλάβετε μέθοδο προσπέλασης (getNumber) η οποία θα επιστρέφει τον αριθμό αυτό, και να τη χρησιμοποιήσετε.

4. Βάσει του παραδείγματος του Σχ. 1 να κατασκευάσετε μια κλάση που να αναπαριστά δευτεροβάθμιες εξισώσεις. Στην κλάση αυτή να συμπεριλάβετε μια εσωτερική κλάση για την αναπαράσταση των ριζών της.

5. Να κατασκευάσετε μια κλάση που αναπαριστά ένα βαθμολόγιο φοιτητών. Κάθε εγγραφή του βαθμολογίου θα πρέπει να είναι ένα ξεχωριστό αντικείμενο το οποίο περιέχει το ονοματεπώνυμο του φοιτητή, τον αριθμό μητρώου και τη βαθμολογία του. Το βαθμολόγιο θα πρέπει να έχει μια μεταβλητή υπόστασης (στιγμιότυπου) τύπου λίστας εγγραφών. Η κλάση θα πρέπει να είναι εκτελέσιμη και να προσφέρει μενού για εγγραφή νέου φοιτητή με το βαθμό του, διαγραφή φοιτητή και αλλαγή της βαθμολογίας του φοιτητή. Για κάθε μια από τις λειτουργίες του μενού θα πρέπει να υπάρχει ξεχωριστή αντίστοιχη μέθοδος στην κλάση.