



ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
Εθνικό και Καποδιστριακό
Πανεπιστήμιο Αθηνών

ΠΛΗΡΟΦΟΡΙΚΗ II

Ενότητα 12: Ροές και είσοδος/έξοδος αρχείων

Μιχάλης Δρακόπουλος

Σχολή Θετικών επιστημών

Τμήμα Μαθηματικών

ΠΛΗΡΟΦΟΡΙΚΗ ΙΙ (Java) Ενότητα 12

Ροές και είσοδος/έξοδος αρχείων

Τα *δεδομένα εισόδου* ενός προγράμματος, εκτός από το να δίνονται από το χρήστη με τη χρήση του πληκτρολογίου και/ή παραθυρικού περιβάλλοντος, μπορούν να δίνονται στο πρόγραμμα μέσω ενός αρχείου στο οποίο βρίσκονται αποθηκευμένα. Ομοίως, τα *αποτελέσματα* ενός προγράμματος (δεδομένα εξόδου), εκτός από το να εμφανίζονται στην οθόνη (στο terminal window ή σε κάποιο παραθυρικό περιβάλλον), μπορούν να αποθηκεύονται σε κάποιο αρχείο, το οποίο μπορεί μετέπειτα να διαβάσει ο χρήστης π.χ. με έναν κειμενογράφο.

Η είσοδος/έξοδος δεδομένων από/σε αρχεία στη Java γίνεται μέσω *ροών (streams)* → αντικείμενα συγκεκριμένων κλάσεων για είσοδο/έξοδο.

→ Ροές χρησιμοποιούνται και για την έξοδο στην οθόνη ή την είσοδο από το πληκτρολόγιο. Π.χ., η `System.out` που χρησιμοποιείται για την έξοδο στο terminal window (π.χ. `System.out.println()`) είναι μια ροή εξόδου, ενώ τα αντικείμενα της `Scanner` που χρησιμοποιούνται για είσοδο από το πληκτρολόγιο είναι ροές εισόδου.

Δηλαδή:

```

streams
├── input streams (π.χ., πληκτρολόγιο ή αρχείο)
└── output streams (π.χ., οθόνη ή αρχείο)

```

Έξοδος σε αρχείο κειμένου

Χρήση της κλάσης `PrintWriter` του πακέτου `java.io`.

Άρα: `import java.io.*;`

- Διαδικασία εγγραφής σε αρχείο, σε 3 βήματα:
- ◆ Δημιουργία / άνοιγμα αρχείου (στην ουσία, δημιουργία ροής εξόδου):

```
PrintWriter outputStream = new PrintWriter(new
    FileOutputStream("όνομα αρχείου"));
```

- ◆ Εγγραφή στο αρχείο:

```
outputStream.println("...");
```

- ◆ Κλείσιμο αρχείου:

```
outputStream.close();
```

Κάποιες παρατηρήσεις:

- ◆ Ο κατασκευαστής της `PrintWriter` δεν μπορεί να δεχθεί σαν όρισμα `string` με το όνομα του αρχείου, γι αυτό και χρησιμοποιείται ένα αντικείμενο της κλάσης `FileOutputStream`, της οποίας ο κατασκευαστής μπορεί να δεχθεί όνομα αρχείου.
- ◆ Το αρχείο, αν δεν υπάρχει, δημιουργείται. Αν υπάρχει ήδη, τότε δημιουργείται εκ νέου, άρα το αρχικό αρχείο διαγράφεται (χάνεται).
- ◆ Για εγγραφή δεδομένων στο τέλος υπάρχοντος αρχείου (χωρίς δηλαδή να χαθούν τα δεδομένα που έχει ήδη το αρχείο):

```
PrintWriter outputStream = new PrintWriter(new
    FileOutputStream("όνομα αρχείου", true));
```

Αν σαν δεύτερο όρισμα εισόδου στον κατασκευαστή της `FileOutputStream` δοθεί `boolean` μεταβλητή με τιμή `false`, τότε η ροή εξόδου λειτουργεί όπως πριν, δηλαδή σαν να μην υπήρχε δεύτερο όρισμα, άρα διαγράφεται το αρχείο σε περίπτωση που υπάρχει ήδη.

- ◆ Αφού ολοκληρωθεί η εγγραφή δεδομένων στο αρχείο, θα πρέπει πάντα να εκτελείται η εντολή που κλείνει το αρχείο, ώστε να διατηρηθεί η γενικότερη ευστάθεια του (λειτουργικού) συστήματος.
- ◆ Συνήθως, η δημιουργία της ροής της `PrintWriter` γίνεται σε δύο στάδια, με χρήση `try-block` ώστε να μπορεί να γίνει χειρισμός πιθανής εξαίρεσης. Δηλαδή:

```
εκτός try:    PrintWriter outputStream = null;
μέσα σε try: try {
                outputStream = new PrintWriter(new
                    FileOutputStream("όνομα"));
            }
            catch (FileNotFoundException e) {
                ...
            }
```

Η εξαίρεση αυτή, που είναι ένα είδος της γενικότερης κατηγορίας `IOException`, θα συμβεί εάν το αρχείο για κάποιο λόγο δεν δημιουργήθηκε. Θα μπορούσαμε να κάνουμε `catch (IOException e)`, αλλά προσπαθούμε να κάνουμε `catch` όσο πιο εξειδικευμένη εξαίρεση μπορούμε γιατί έτσι παίρνουμε πιο σαφή πληροφόρηση για το τι ακριβώς συνέβη.

Παράδειγμα εξόδου σε αρχείο:

```
import java.io.*; // για χρήση αρχείων
import java.util.*; // για χρήση της Scanner
public class FileOutputStreamDemo
{
    public static void main(String [] args)
    {
        PrintWriter outputStream = null;
        try {
            outputStream = new PrintWriter(new
                FileOutputStream("out.txt"));
        }
        catch (FileNotFoundException e) {
            System.out.println("Error opening the file out.txt");
            System.exit(0);
        }
        System.out.println("Enter a line of text:");
        String line = null;
        Scanner input = new Scanner(System.in);
        line = input.nextLine();
        outputStream.println(line);
        outputStream.close();
    }
}
```

Είσοδος από αρχείο κειμένου

Χρήση της κλάσης `BufferedReader` του πακέτου `java.io`. Άρα: `import java.io.*;`

- Διαδικασία ανάγνωσης από αρχείο, σε 3 βήματα:

- ◆ Άνοιγμα αρχείου (στην ουσία, δημιουργία ροής εισόδου):

```
BufferedReader inputStream = new BufferedReader(new
    FileReader("όνομα αρχείου"));
```

- ◆ Ανάγνωση από το αρχείο:

```
String line = inputStream.readLine();
```

Αν φτάσει στο τέλος του αρχείου, η `readLine()` επιστρέφει την τιμή `null`.

Για ανάγνωση *χαρακτήρα*: `inputStream.read();`

Επιστρέφει τιμή `int` (-1 εάν φτάσει στο τέλος του αρχείου), άρα χρειάζεται μετατροπή σε `char`:

```
char c = (char) inputStream.read();
```

- ◆ Κλείσιμο αρχείου:

```
inputStream.close();
```

Παράδειγμα εισόδου από αρχείο:

```
import java.io.*; // για χρήση αρχείων

public class FileInputDemo
{
    public static void main(String [] args)
    {
        try {
            BufferedReader inputStream = new BufferedReader(new
                FileReader("data.txt"));
            String line = null;
            line = inputStream.readLine();
            System.out.println("The first line in data.txt is:");
            System.out.println(line);
            inputStream.close();
        }
        catch (FileNotFoundException e1) {
            System.out.println("Cannot find or open the file data.txt!");
        }
        catch (IOException e2) {
            System.out.println("Error reading from file data.txt.");
        }
    }
}
```

Το πρώτο catch συλλαμβάνει τυχόν exception που θα συμβεί κατά τη δημιουργία του αρχείου, ενώ το δεύτερο κατά την ανάγνωση από το αρχείο.

Εισαγωγή ονόματος αρχείου και διαδρομής (path)

Το όνομα αρχείου δίνεται σαν ένα string. Άρα, μπορεί να έχει δοθεί νωρίτερα στο πρόγραμμα:

```
String fileName = "data.txt";
```

ή να ζητείται από τον χρήστη:

```
Scanner input = new Scanner(System.in);
System.out.println("Enter file name:");
String fileName = input.next();
```

και στη συνέχεια να χρησιμοποιείται το όνομα της string μεταβλητής που κρατάει το όνομα του αρχείου:

```
BufferedReader inputStream = new BufferedReader(new
    FileReader(fileName));
```

Εισαγωγή path:

π.χ., η αναπαράσταση της διαδρομής (path) υποκαταλόγων σε διάφορα λειτουργικά συστήματα είναι ως εξής:

- a) σε Unix/Linux: /user/john/home/work1/data.txt
- b) σε Windows: D:\homework\hw1\data.txt

Οι αντίστοιχες εισαγωγές των διαδρομών στη Java θα ήταν αντίστοιχα ως εξής:

- a) `... new FileReader("/user/john/home/work1/data.txt");`
- b) `... new FileReader("D:\\homework\\hw1\\data.txt");`
 ή
`... new FileReader("D:/homework/hw1/data.txt");`

Προσοχή: Αν στα Windows το path/όνομα αρχείου ζητηθεί από το χρήστη, θα πρέπει να δοθεί είτε με το δεύτερο τρόπο (της (b) περίπτωσης παραπάνω) είτε με τον πρώτο τρόπο χρησιμοποιώντας μία κάθετο αντί για δύο, δηλαδή όπως είναι η κανονική αναπαράσταση του path στα Windows:

```
Enter file name:
D:\homework\hw1\data.txt
```

και όχι με διπλή κάθετο.

Η κλάση StringTokenizer

Για την ανάγνωση μεμονωμένων λέξεων από αρχείο και όχι ολόκληρων γραμμών, χρησιμοποιούμε την κλάση `StringTokenizer` του πακέτου `java.util`.

Βασικές μέθοδοι:

- `nextToken()` → επιστρέφει την επόμενη λέξη (String)
- `hasMoreTokens()` → επιστρέφει true αν υπάρχουν αδιάβαστες λέξεις
 επιστρέφει false αν έχουν διαβαστεί όλες οι λέξεις

Άρα, αφού διαβάσουμε μια γραμμή ενός αρχείου με την `BufferedReader`, μπορούμε να χρησιμοποιήσουμε την `StringTokenizer` στο string της γραμμής για να διαβάσουμε κάθε λέξη ξεχωριστά.

Συνήθης χρήση:

```
String str = ".....";  
StringTokenizer st = new StringTokenizer(str);  
while (st.hasMoreTokens())  
{  
    System.out.println(st.nextToken());  
}
```

Υπάρχει η δυνατότητα να καθορίσουμε *εμείς* ποια θα είναι τα διαχωριστικά των λέξεων (το default είναι το κενό):

Αυτό γίνεται με τη χρήση του ακόλουθου κατασκευαστή της StringTokenizer:

```
public StringTokenizer(String str, String delimiters)
```

π.χ., `StringTokenizer st = new StringTokenizer(str, " \n.,");`

οπότε, σε αυτή την περίπτωση διαχωριστικά είναι: το κενό, η αλλαγή γραμμής, η τελεία και το κόμμα.

Σημειώματα

Σημείωμα Αναφοράς

Copyright Εθνικών και Καποδιστριακών Πανεπιστημίων Αθηνών, Μιχάλης Δρακόπουλος, 2014.
Μιχάλης Δρακόπουλος. «Πληροφορική II. Ενότητα 12: Ροές και είσοδος/έξοδος αρχείων». Έκδοση:
1.0. Αθήνα 2014. Διαθέσιμο από τη δικτυακή διεύθυνση:
<http://opencourses.uoa.gr/courses/MATH106/>.

Σημείωμα Αδειοδότησης

Το παρόν υλικό διατίθεται με τους όρους της άδειας χρήσης Creative Commons Αναφορά, Μη Εμπορική Χρήση Παρόμοια Διανομή 4.0 [1] ή μεταγενέστερη, Διεθνής Έκδοση. Εξαιρούνται τα αυτοτελή έργα τρίτων π.χ. φωτογραφίες, διαγράμματα κ.λ.π., τα οποία εμπεριέχονται σε αυτό και τα οποία αναφέρονται μαζί με τους όρους χρήσης τους στο «Σημείωμα Χρήσης Έργων Τρίτων».



[1] <http://creativecommons.org/licenses/by-nc-sa/4.0/>

Ως **Μη Εμπορική** ορίζεται η χρήση:

- που δεν περιλαμβάνει άμεσο ή έμμεσο οικονομικό όφελος από την χρήση του έργου, για το διανομέα του έργου και αδειοδόχο
- που δεν περιλαμβάνει οικονομική συναλλαγή ως προϋπόθεση για τη χρήση ή πρόσβαση στο έργο
- που δεν προσπορίζει στο διανομέα του έργου και αδειοδόχο έμμεσο οικονομικό όφελος (π.χ. διαφημίσεις) από την προβολή του έργου σε διαδικτυακό τόπο

Ο δικαιούχος μπορεί να παρέχει στον αδειοδόχο ξεχωριστή άδεια να χρησιμοποιεί το έργο για εμπορική χρήση, εφόσον αυτό του ζητηθεί.

Διατήρηση Σημειωμάτων

- Οποιαδήποτε αναπαραγωγή ή διασκευή του υλικού θα πρέπει να συμπεριλαμβάνει:
- το Σημείωμα Αναφοράς
- το Σημείωμα Αδειοδότησης
- τη δήλωση Διατήρησης Σημειωμάτων
- το Σημείωμα Χρήσης Έργων Τρίτων (εφόσον υπάρχει)

μαζί με τους συνοδευόμενους υπερσυνδέσμους.

Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στο πλαίσιο του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο Πανεπιστήμιο Αθηνών**» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.

