

Εισαγωγή στον Αντικειμενοστρεφή Προγραμματισμό – Διάλεξη #16

Διάλεξη #16:

Είσοδος / Έξοδος

[Input / Output]

Εισαγωγή στον Αντικειμενοστρεφή Προγραμματισμό, Αντώνιος Συμβώνης, ΣΕΜΦΕ, ΕΜΠ, Slide 1

Μηχανισμοί εισόδου/εξόδου [I/O mechanisms]

- Όλα τα προγράμματα χρειάζονται είσοδο/έξοδο για να επικοινωνούν με το περιβάλλον τους.
- Η είσοδος/έξοδος γίνεται μέσω κειμένου, γραφικών, αισθητήρων, εξωτερικών συσκευών, κλπ.
- Για προσωπικούς υπολογιστές: οθόνη, πληκτρολόγιο, ποντίκι, δίκτυο, σύστημα αρχείων.
- Για εμφυτευμένα [embedded] υπολογιστικά συστήματα: αισθητήρες, άλλοι μηχανισμοί.

Εισαγωγή στον Αντικειμενοστρεφή Προγραμματισμό, Αντώνιος Συμβώνης, ΣΕΜΦΕ, ΕΜΠ, Slide 2

Μηχανισμοί εξόδου

- Αποτελέσματα μεθόδων (BlueJ)
- Κείμενο σε οθόνη, εκτυπωτή, κλπ.
- Στοιχεία [data] σε αρχείο
- Στοιχεία προς άλλα προγράμματα, δίκτυο, κλπ.
- Παράθυρα, γραφικά (GUI)

Εισαγωγή στον Αντικειμενοστρεφή Προγραμματισμό, Αντώνιος Συμβώνης, ΣΕΜΦΕ, ΕΜΠ, Slide 3

Μηχανισμοί εισόδου

- Παράμετροι μεθόδων (BlueJ)
- Παράμετροι γραμμής εντολών [command line]
- Κείμενο από πληκτρολόγιο
- Στοιχεία [data] από αρχεία
- Στοιχεία από προγράμματα, το δίκτυο, κλπ.
- Είσοδος από ΓΠΕ [GUI] (δράσεις ποντικιού – πίεση πλήκτρων, επιλογή μενού, κλπ.)

Εισαγωγή στον Αντικειμενοστρεφή Προγραμματισμό, Αντώνιος Συμβώνης, ΣΕΜΦΕ, ΕΜΠ, Slide 4

Μηχανισμοί εισόδου/εξόδου [I/O] της Java

- Στην Java η είσοδος/έξοδος γίνεται μέσω ενός συνόλου κλάσεων
- Για I/O: καμία νέα δομή της γλώσσας (μόνο νέες κλάσεις)
- Οι κλάσεις παρέχουν αρκετές διαπροσωπίες προς **ρεύματα [streams]** και άλλες «επινοήσεις» εισόδου/εξόδου [IO concepts].

Εισαγωγή στον Αντικειμενοστρεφή Προγραμματισμό, Αντώνιος Συμβώνης, ΣΕΜΦΕ, ΕΜΠ, Slide 5

Ρεύματα [Streams]

The diagram shows two scenarios of data flow. In the first, a box labeled 'Πρόγραμμα' (Program) has an arrow pointing left to it labeled 'διαβάζει [reads]' (reads), which originates from a cloud labeled 'Πηγή [source]' (source). In the second, a box labeled 'Πρόγραμμα' has an arrow pointing right from it labeled 'γράφει [writes]' (writes), which terminates at a cloud labeled 'Προορισμός [destination]' (destination). A wavy line labeled 'Ρεύμα [stream]' (stream) connects the source to the destination. At the bottom, two figures are shown carrying a rope between them, representing the stream.

Εισαγωγή στον Αντικειμενοστρεφή Προγραμματισμό, Αντώνιος Συμβώνης, ΣΕΜΦΕ, ΕΜΠ, Slide 6

Εισαγωγή στον Αντικειμενοστρεφή Προγραμματισμό – Διάλεξη #16

Ιδιωματισμοί Java [java idioms]

- Εγγραφή


```

            Άνοιξε το ρεύμα [stream]
            while(περισσότερα στοιχεία)
                γράψε στοιχεία στο ρεύμα
            Κλείσε το ρεύμα
            
```
- Ανάγνωση


```

            Άνοιξε το ρεύμα [stream]
            while(περισσότερα στοιχεία)
                διάβασε στοιχεία από το ρεύμα
            Κλείσε το ρεύμα
            
```

Εισαγωγή στον Αντικειμενοστρεφή Προγραμματισμό, Αντώνιος Συμβώνης, ΣΕΜΦΕ, ΕΜΠ, Slide 7

Χαρακτήρες και Bytes

Ρεύμα χαρακτήρων

Εγγραφή και ανάγνωση κειμένου

Ρεύμα byte

Εγγραφή και ανάγνωση δυαδικών στοιχείων [binary data] (αριθμοί, εικόνες, ήχος,...)

Εισαγωγή στον Αντικειμενοστρεφή Προγραμματισμό, Αντώνιος Συμβώνης, ΣΕΜΦΕ, ΕΜΠ, Slide 8

Ρεύματα και αναγνώστες/εγγραφείς

- Ρεύματα [streams]
 - Διαβάζουν/γράφουν bytes
 - Μπορεί να χρησιμοποιηθούν για οποιοδήποτε τύπο δεδομένων
 - Χρησιμοποιούνται για αποθήκευση στοιχείων σε σειριακή μορφή ["serialisation"]
- Αναγνώστης/εγγραφέας [reader/writer]
 - Διαβάζει/γράφει χαρακτήρες
 - Χρησιμοποιούνται για εισοδο/έξοδο κειμένου
 - Προτιμούνται για εισοδο/έξοδο κειμένου

Εισαγωγή στον Αντικειμενοστρεφή Προγραμματισμό, Αντώνιος Συμβώνης, ΣΕΜΦΕ, ΕΜΠ, Slide 9

Η ιεραρχία Stream (ελλιπής)

```

    OutputStream
    ├── FilterOutputStream
    │   ├── PrintStream
    │   └── BufferedOutputStream
    ├── FileOutputStream
    └── PipedOutputStream
    
```

Εισαγωγή στον Αντικειμενοστρεφή Προγραμματισμό, Αντώνιος Συμβώνης, ΣΕΜΦΕ, ΕΜΠ, Slide 10

System.in και System.out

```

System.out.println("trust no-one");

class System
{
    public static final PrintStream err;
    public static final InputStream in;
    public static final PrintStream out;
    ...
}
    
```

Η κλάση System (για ιστορικούς λόγους) χρησιμοποιεί τις PrintStream και InputStream για I/O. Ο προτιμητέος τρόπος για εισοδο/έξοδο κειμένου είναι μέσω των «Reader/Writer» κλάσεων.

Εισαγωγή στον Αντικειμενοστρεφή Προγραμματισμό, Αντώνιος Συμβώνης, ΣΕΜΦΕ, ΕΜΠ, Slide 11

Μεθοδοί της PrintStream

```

public void print(boolean b)
public void print(char c)
public void print(double d)
public void print(float f)
public void print(int i)
public void print(long l)
public void print(String s)
public void print(Object o)

public void println(boolean b)
public void println(char c)
public void println(double d)
public void println(float f)
public void println(int i)
public void println(long l)
public void println(String s)
public void println(Object o)
    
```


Εισαγωγή στον Αντικειμενοστρεφή Προγραμματισμό, Αντώνιος Συμβώνης, ΣΕΜΦΕ, ΕΜΠ, Slide 12

Εισαγωγή στον Αντικειμενοστρεφή Προγραμματισμό – Διάλεξη #16

Ανάγνωση

Παράδειγμα

```
BufferedReader reader =  
    new BufferedReader(new InputStreamReader(System.in));  
try  
{  
    String inputLine = reader.readLine();  
}  
catch (IOException exc)  
{  
    // an IO error occurred  
}
```

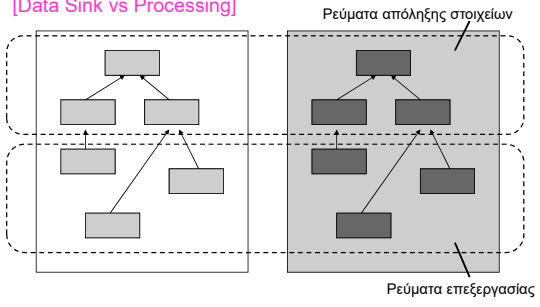


Τι σημαίνουν όλα αυτά;

Εισαγωγή στον Αντικειμενοστρεφή Προγραμματισμό, Αντώνιος Συμβώνης, ΣΕΜΦΕ, ΕΜΠ, Slide 13

Απόληξη στοιχείων και επεξεργασία

[Data Sink vs Processing]



Εισαγωγή στον Αντικειμενοστρεφή Προγραμματισμό, Αντώνιος Συμβώνης, ΣΕΜΦΕ, ΕΜΠ, Slide 14

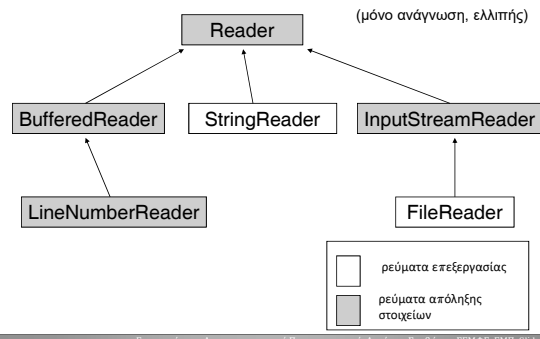
Data Sink vs Processing (2)

- Τα «ρεύματα απόληξης στοιχείων» [data sink streams] είναι υλοποιήσεις της διαπροσωπείας “stream” οι οποίες παρέχουν πρόσβαση σε διάφορες πηγές και προορισμούς στοιχείων (αλλά δεν προσθέτουν νέες λειτουργίες).
- Τα «ρεύματα επεξεργασίας» [processing streams] παρέχουν επιπλέον λειτουργίες σε σχέση με υπάρχοντα ρεύματα.

Εισαγωγή στον Αντικειμενοστρεφή Προγραμματισμό, Αντώνιος Συμβώνης, ΣΕΜΦΕ, ΕΜΠ, Slide 15

Η ιεραρχία αναγνωστών/εγγραφέων (ελλιπής)

(μόνο ανάγνωση, ελλιπής)



Εισαγωγή στον Αντικειμενοστρεφή Προγραμματισμό, Αντώνιος Συμβώνης, ΣΕΜΦΕ, ΕΜΠ, Slide 16

InputStream

- Αφηρημένη [abstract] κλάση
- Διαβάζει bytes δεδομένων από οποιοδήποτε ρεύμα εισόδου

Εισαγωγή στον Αντικειμενοστρεφή Προγραμματισμό, Αντώνιος Συμβώνης, ΣΕΜΦΕ, ΕΜΠ, Slide 17

InputStreamReader

- Τελική [concrete] κλάση
- Συνδέεται με έναν ρεύμα-υπόστρωμα τύπου InputStream
- Μετατρέπει bytes από το InputStream σε χαρακτήρες
- Κάθε λειτουργία ανάγνωσης μπορεί να προκαλέσει μία ή περισσότερες λειτουργίες ανάγνωσης του ρεύματος-υπόστρωμα InputStream

Εισαγωγή στον Αντικειμενοστρεφή Προγραμματισμό, Αντώνιος Συμβώνης, ΣΕΜΦΕ, ΕΜΠ, Slide 18

Εισαγωγή στον Αντικειμενοστρεφή Προγραμματισμό – Διάλεξη #16

BufferedReader

- Τελική [concrete] κλάση
- Συνδέεται με έναν αναγνώστη-υπόστρωμα (συνήθως τύπου `InputStreamReader` ή `FileReader`)
- Προσθέτει «προσωρινή μνήμη» [buffering] για καλύτερες επιδόσεις
- Αποφεύγει να πραγματοποιεί μια κλήση «ανάγνωσης» για κάθε χαρακτήρα

Εισαγωγή στον Αντικειμενοστρεφή Προγραμματισμό, Αντώνιος Συμβώνης, ΣΕΜΦΕ, ΕΜΠ, Slide 19

Ανάγνωση: παράδειγμα

```
BufferedReader reader =
new BufferedReader(new InputStreamReader(System.in));
try
{
    while(true)
    {
        String line = reader.readLine();
        System.out.println("the line was: " + line);
    }
}
catch(IOException exc)
{
    // an IO error occurred
}
```

Εισαγωγή στον Αντικειμενοστρεφή Προγραμματισμό, Αντώνιος Συμβώνης, ΣΕΜΦΕ, ΕΜΠ, Slide 20

Είσοδος/έξοδος αρχείου

[File I/O]

Εισαγωγή στον Αντικειμενοστρεφή Προγραμματισμό, Αντώνιος Συμβώνης, ΣΕΜΦΕ, ΕΜΠ, Slide 21

Τεμαχισμός [tokenising] της εισόδου

```
StringTokenizer tokenizer = new StringTokenizer(line);
while(tokenizer.hasMoreTokens())
    System.out.println(tokenizer.nextToken());
```

Γραμμή εισόδου:
It's a beautiful day!

Τεμάχια [tokens]:
It's a beautiful day!

Εισαγωγή στον Αντικειμενοστρεφή Προγραμματισμό, Αντώνιος Συμβώνης, ΣΕΜΦΕ, ΕΜΠ, Slide 22

Τεμαχισμός της εισόδου(2)

Παράδειγμα- ο κώδικας:

```
String s = "this is a test";
StringTokenizer st = new StringTokenizer(s);
while (st.hasMoreTokens()) {
    println(st.nextToken());
}
```

Τυπώνει την παρακάτω έξοδο:

```
this
is
a
test
```

Εισαγωγή στον Αντικειμενοστρεφή Προγραμματισμό, Αντώνιος Συμβώνης, ΣΕΜΦΕ, ΕΜΠ, Slide 23

Η διαπροσωπεία "Tokenizer"

Κατασκευαστές:

- StringTokenizer(String str)**
Constructs a string tokenizer for the specified string.
- StringTokenizer(String str, String delim)**
Constructs a string tokenizer for the specified string and a given delimiter.
- StringTokenizer(String str, String delim, boolean returnTokens)**
Constructs a string tokenizer for the specified string, a given delimiter and a flag indicating whether to return the delimiter tokens.

Εισαγωγή στον Αντικειμενοστρεφή Προγραμματισμό, Αντώνιος Συμβώνης, ΣΕΜΦΕ, ΕΜΠ, Slide 24

Εισαγωγή στον Αντικειμενοστρεφή Προγραμματισμό – Διάλεξη #16

Η διαπροσωπεία "Tokenizer"(2)

Μέθοδοι:

- boolean hasMoreTokens ()**
Tests if there are more tokens available from this tokenizer's string.
- String nextToken ()**
Returns the next token from this string tokenizer.
- int countTokens ()**
Calculates the number of times that this tokenizer's nextToken method can be called before it generates an exception.

Εισαγωγή στον Αντικειμενοστρεφή Προγραμματισμό, Αντώνιος Συμβώνης, ΣΕΜΦΕ, ΕΜΠ, Slide 25

Η διαπροσωπεία Reader

Αφηρημένη κλάση Reader:

- Reader ()**
Create a new character-stream reader whose critical sections will synchronize on the reader itself.
- int read ()** ← Οι χαρακτήρες επιστρέφονται ως int (-1 εάν έχουν εξαντηθεί) Πρέπει να μετατραπούν σε χαρακτήρες μέσω εντολής τύπου cast
Read a single character
- abstract void close ()**
Close the stream.

Εισαγωγή στον Αντικειμενοστρεφή Προγραμματισμό, Αντώνιος Συμβώνης, ΣΕΜΦΕ, ΕΜΠ, Slide 26

Κλάση FileReader

class FileReader (extends InputStreamReader):

- FileReader (File file)**
- FileReader (FileDescriptor fd)**
- FileReader (String fileName)**

(Καμία νέα μέθοδος)

Εισαγωγή στον Αντικειμενοστρεφή Προγραμματισμό, Αντώνιος Συμβώνης, ΣΕΜΦΕ, ΕΜΠ, Slide 27

Παράδειγμα: copyFile

```
public void copyFile(String fromFile, String toFile)
    throws IOException
{
    File inputFile = new File(fromFile);
    File outputFile = new File(toFile);

    FileReader in = new FileReader(inputFile);
    FileWriter out = new FileWriter(outputFile);
    int ch;

    while ((ch = in.read()) != -1)
        out.write(ch);
    in.close();
    out.close();
}
```

Εισαγωγή στον Αντικειμενοστρεφή Προγραμματισμό, Αντώνιος Συμβώνης, ΣΕΜΦΕ, ΕΜΠ, Slide 28

Δημιουργία αναγνώστη αρχείου

Ο κώδικας με τον οποίο η μέθοδος `copyFile` δημιουργεί έναν αναγνώστη αρχείου [FileReader]:

```
inputFile = new File (fromFile);
FileReader in = new FileReader (inputFile);
```

Εισαγωγή στον Αντικειμενοστρεφή Προγραμματισμό, Αντώνιος Συμβώνης, ΣΕΜΦΕ, ΕΜΠ, Slide 29

Η δομή του αναγνώστη αρχείων

Διαβάζει χαρακτήρες από ένα αρχείο

FileReader

File

Αρχείο Κειμένου [text file]

Σύστημα αρχείων file system (δίσκος)

Δημιουργεί, ανοίγει ένα αρχείο, ελέγχει εάν ένα αρχείο υπάρχει, κλπ.

read (επιστρέφει ένα χαρακτήρα ή -1 εάν βρισκόμαστε στο τέλος του αρχείου)

Εισαγωγή στον Αντικειμενοστρεφή Προγραμματισμό, Αντώνιος Συμβώνης, ΣΕΜΦΕ, ΕΜΠ, Slide 30

Εισαγωγή στον Αντικειμενοστρεφή Προγραμματισμό – Διάλεξη #16

Μετατροπή [casting] σε χαρακτήρες

```
public void processFile(String fileName)
    throws IOException
{
    File inputFile = new File(fileName);
    FileReader in = new FileReader(inputFile);
    int ch;

    while ((ch = in.read()) != -1)
        processCharacter((char)ch);

    in.close();
}

public void processCharacter(char aChar)
{
    ...
}
```

Αναγκασία μετατροπή!

Εισαγωγή στον Αντικειμενοστρεφή Προγραμματισμό, Αντώνιος Συμβώνης, ΣΕΜΦΕ, ΕΜΠ, Slide 31

Ανάγνωση αριθμών

```
...
int number1;
int number2;
boolean invalidInput = false;

String line = reader.readLine();
try
{
    number1 = Integer.parseInt(line);
    line = reader.readLine();
    number2 = Integer.parseInt(line);
}
catch (NumberFormatException exc)
{
    // input was not a valid number
    invalidInput = true;
}
```

42
128

↑
Αρχείο
κειμένου

Εισαγωγή στον Αντικειμενοστρεφή Προγραμματισμό, Αντώνιος Συμβώνης, ΣΕΜΦΕ, ΕΜΠ, Slide 32