

## Δομές Δεδομένων & Ανάλυση Αλγορίθμων 3ο Εξάμηνο

### • Αρχεία και Ρεύματα Bytes (Files an Byte Streams)

Δημοσθένης Σταμάτης

<http://www.iee.ihu.gr/~demos>

Τμήμα Μηχανικών Πληροφορικής & Ηλεκτρονικών Συστημάτων



#### Class Hierarchy

- o class java.lang.Object
  - o class java.io.File (implements java.lang.Comparable, java.io.Serializable)
  - o class java.io.FileDescriptor
  - o class java.io.InputStream
    - o class java.io.ByteArrayInputStream
    - o class java.io.FileInputStream
    - o class java.io.FilterInputStream
      - o class java.io.BufferedReader
      - o class java.io.DataInputStream (implements java.io.DataInput)
      - o class java.io.LineNumberInputStream
      - o class java.io.PushbackInputStream
    - o class java.io.ObjectInputStream (implements java.io.ObjectInput, java.io.ObjectStreamConstants)
    - o class java.io.PipedInputStream
    - o class java.io.SequenceInputStream
    - o class java.io.StringBufferInputStream
  - o class java.io.ObjectInputStream.GetField
  - o class java.io.ObjectOutputStream.PutField
  - o class java.io.ObjectStreamClass (implements java.io.Serializable)
  - o class java.io.ObjectStreamField (implements java.lang.Comparable)
  - o class java.io.OutputStream
    - o class java.io.ByteArrayOutputStream



### Ιεραρχία κλάσεων για την επεξεργασία αρχείων (package java.io)

#### **InputStream (abstract class)**

FileInputStream

FilterInputStream

    DataInputStream (υλοποιεί το interface DataInput)

    BufferedInputStream

    LineNumberInputStream

ObjectInputStream

#### **OutputStream (abstract class)**

FileOutputStream

FilterOutputStream

    DataOutputStream (υλοποιεί το DataOutput)

    BufferedOutputStream

ObjectOutputStream

### Ιεραρχία κλάσεων για την επεξεργασία αρχείων (package java.io)

#### **Reader (abstract class)**

BufferedReader

InputStreamReader

    FileReader

#### **Writer (abstract class)**

BufferedWriter

OutputStreamWriter

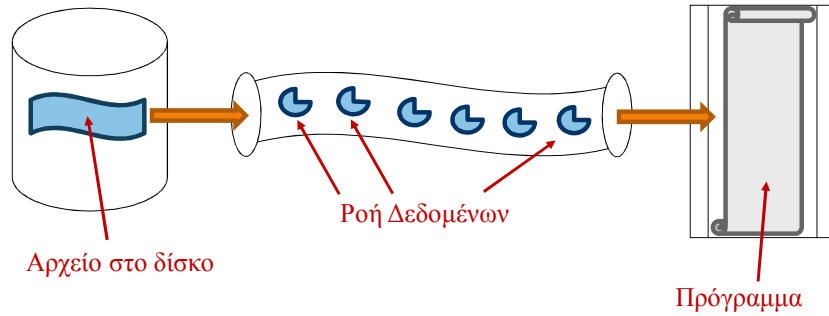
    FileWriter

**RandomAccessFile** (υλοποιεί τα interfaces DataInput & DataOutput)

**File**

### Ρεύμα Εισόδου (Input Stream)

Είσοδος δεδομένων από ένα εξωτερικό μέσο (π.χ. αρχείο στο δίσκο) προς το πρόγραμμα

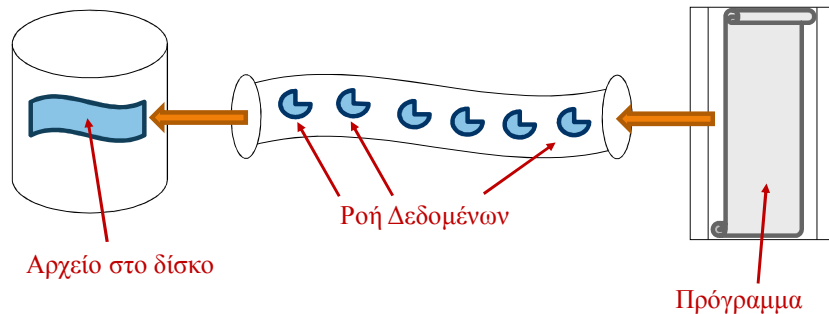


ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ

Γ' ΕΞΑΜΗΝΟ

### Ρεύμα Εισόδου (Input Stream)

Έξοδος δεδομένων από το πρόγραμμα προς ένα εξωτερικό μέσο (π.χ. αρχείο στο δίσκο)



ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ

Γ' ΕΞΑΜΗΝΟ

**Ρεύμα Εισόδου (input stream): Διαδικασία Ανάγνωσης****ΒΗΜΑ 1:** Δημιούργησε ένα ρεύμα εισόδου με βάση το αρχείο;**ΒΗΜΑ 2:** Εφόσον <υπάρχουν δεδομένα>

Διάβασε επόμενο δεδομένο;

[Επεξεργάσου το δεδομένο;]

**ΒΗΜΑ 3:** Κλείσε το ρεύμα εισόδου;**Ρεύμα Εξόδου (output stream): Διαδικασία Εγγραφής****ΒΗΜΑ 1:** Δημιούργησε ένα ρεύμα εξόδου με βάση το αρχείο;**ΒΗΜΑ 2:** Εφόσον <υπάρχουν δεδομένα>

[Προετοίμασε επόμενο δεδομένο;]

Γράψε δεδομένο;

**ΒΗΜΑ 3:** Κλείσε το ρεύμα εξόδου;**Ρεύμα Εισόδου (input stream): Διαδικασία Ανάγνωσης****ΒΗΜΑ 1:** Δημιούργησε ένα ρεύμα εισόδου με βάση το αρχείο;**ΒΗΜΑ 2:** Εφόσον <υπάρχουν δεδομένα>

Διάβασε επόμενο δεδομένο;

[Επεξεργάσου το δεδομένο;]

**ΒΗΜΑ 3:** Κλείσε το ρεύμα εισόδου;**Ρεύμα Εξόδου (output stream): Διαδικασία Εγγραφής****ΒΗΜΑ 1:** Δημιούργησε ένα ρεύμα εξόδου με βάση το αρχείο;**ΒΗΜΑ 2:** Εφόσον <υπάρχουν δεδομένα>

[Προετοίμασε επόμενο δεδομένο;]

Γράψε δεδομένο;

**ΒΗΜΑ 3:** Κλείσε το ρεύμα εξόδου;

Βασικές Μέθοδοι της κλάσης `InputStream`

Μέθοδος	Περιγραφή
<code>int read( )</code>	Επιστρέφει έναν ακέραιο αριθμό που αντιστοιχεί στο επόμενο διαθέσιμο byte του stream εισόδου. Επιστρέφει -1 όταν συναντήσει το τέλος του αρχείου.
<code>int read(byte buffer[ ])</code>	Προσπαθεί να διαβάσει από το stream εισόδου αριθμό από bytes ίσο με το μήκος <code>buffer.length</code> του πίνακα <code>buffer</code> και τους τοποθετεί σε αυτόν. Επιστρέφει έναν ακέραιο αριθμό που αντιστοιχεί στον αριθμό των bytes που διάβασε ή, -1 αν συναντήσει το τέλος του αρχείου.
<code>int available( )</code>	Επιστρέφει τον αριθμό των bytes του stream εισόδου που είναι διαθέσιμα.
<code>long skip(long numBytes)</code>	Προσπαθεί να παρακάμψει τα επόμενα <code>numBytes</code> bytes του stream εισόδου επιστρέφοντας τον αριθμό όσων αγνόησε.
<code>void close( )</code>	Κλείνει το stream εισόδου. Μετά το κλείσιμο προσπάθειες ανάγνωσης δεδομένων οδηγούν σε εξαίρεση <code>IOException</code>

ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ

Γ' ΕΞΑΜΗΝΟ

Βασικές Μέθοδοι της κλάσης `OutputStream`

Μέθοδος	Περιγραφή
<code>void write(int oneByte )</code>	Γράφει ένα μόνο byte στο stream εξόδου.
<code>void write(byte buffer[ ])</code>	Γράφει τα bytes που είναι αποθηκευμένα στον πίνακα <code>buffer</code> στο stream εξόδου.
<code>void close( )</code>	Κλείνει το stream εξόδου. Μετά το κλείσιμο προσπάθειες εγγραφής δεδομένων οδηγούν σε εξαίρεση <code>IOException</code>

ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ

Γ' ΕΞΑΜΗΝΟ

**Περιγραφή της read της FileInputStream**

```
public int read(byte[] b) throws IOException
```

Διαβάζει έως b.length bytes δεδομένων από το ρεύμα εισόδου (input stream) και τους τοποθετεί σε έναν πίνακα από bytes. Η μέθοδος αυτή μπαίνει σε κατάσταση αναμονής (blocks) μέχρις ότου υπάρξουν διαθέσιμα δεδομένα.

**Παράμετροι (Parameters):**

b – ο πίνακας στον οποίον τοποθετούνται τα δεδομένα που διαβάζονται.

**Επιστρέφει (Returns):**

Τον ολικό αριθμό των Bytes που διαβάστηκαν στον πίνακα ή -1 αν δεν υπάρχουν άλλα δεδομένα γιατί η διαδικασία ανάγνωσης έφτασε στο τέλος του αρχείου (end of the file).

**Προκαλεί (Throws):**

IOException – εάν προκύψει λάθος εισόδου/εξόδου (I/O error).

**Επαναορίζει (Overrides):**

Τη read της κλάσης InputStream

**Δημιουργία αρχείου από bytes**

```
import java.io.*;

public class WriteBytesToFile {

    public static void main(String[] arguments) {

        int[] data = { 71, 73, 70, 56, 57, 97, 16, 0, 15, 0, 128, 0, 0,
            255, 255, 255, 0, 0, 0, 44, 0, 0, 0, 0, 15, 0, 15, 0, 0, 2, 33,
            132, 127, 161, 200, 185, 205, 84, 128, 241, 81, 35, 175,
            155, 26, 228, 254, 105, 33, 102, 121, 165, 201, 145, 169,
            154, 142, 172, 116, 162, 240, 90, 197, 5, 0, 59 };

        try {
            FileOutputStream file = new FileOutputStream("File_with_Bytes");
            for (int i = 0; i < data.length; i++)
                file.write(data[i]);
            file.close();
        } catch (IOException error) {
            System.out.println("Error -- " + error);
        }
    }
}
```

Δημιουργία stream εξόδου στο αρχείο

Εγγραφή ενός byte στο αρχείο

Κλείσιμο του αρχείου

**Διάβασμα bytes από αρχείο**

```
import java.io.*;
public class ReadBytesFromFile {
    public static void main(String[] arguments) {
        try {
            FileInputStream file = new FileInputStream("File_with_Bytes");
            boolean eof = false;
            int count = 0;
            while (!eof) {
                int input = file.read();
                System.out.print(input + " ");
                if (input == -1) eof = true;
                else count++;
            }
            file.close();
            System.out.println("\nBytes read: " + count);
        } catch (IOException e) {
            System.out.println("Error -- " + e);
        }
    }
}
```

Δημιουργία stream εξόδου προς το αρχείο

Είσοδος ενός byte από το αρχείο

Τέλος του αρχείου

Κλείσιμο του αρχείου

**Ενδεικτικές Μέθοδοι της κλάσης DataInputStream**

Μέθοδος	Περιγραφή
boolean readBoolean( )	Διαβάζει το επόμενο byte από το ρεύμα εισόδου. Επιστρέφει true εάν η τιμή του είναι μη μηδενική και false εάν είναι μηδέν.
int readInt( )	Διαβάζει τα επόμενα τέσσερα bytes από το ρεύμα εισόδου και επιστρέφει τον ακέραιο αριθμό τύπου int που αντιστοιχεί σε αυτά.
float readFloat( )	Διαβάζει τα επόμενα τέσσερα bytes από το ρεύμα εισόδου και επιστρέφει τον πραγματικό αριθμό τύπου float που αντιστοιχεί σε αυτά.

Ενδεικτικές Μέθοδοι της κλάσης `DataOutputStream`

Μέθοδος	Περιγραφή
<code>void writeInt(int n)</code>	Γράφει έναν ακέραιο αριθμό στο ρεύμα εξόδου σαν μία σειρά από τέσσερα bytes, αρχίζοντας από το περισσότερο σημαντικό byte.
<code>void writeFloat(float n)</code>	Μετατρέπει τον πραγματικό αριθμό n σε ακέραιο χρησιμοποιώντας τη μέθοδο <code>floatToIntBits</code> της κλάσης <code>Float</code> , και στη συνέχεια γράφει τον αριθμό αυτό στο ρεύμα εξόδου σαν μία σειρά από τέσσερα bytes, αρχίζοντας από το περισσότερο σημαντικό.
<code>void close( )</code>	Κλείνει το stream εξόδου. Μετά το κλείσιμο προσπάθειες εγγραφής δεδομένων οδηγούν σε εξαίρεση <code>IOException</code>

ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ

Γ' ΕΞΑΜΗΝΟ

```
import java.io.*;
public class ReadIntsFromFile {
    public static void main(String arguments[]) {
        try {
            FileInputStream file = new FileInputStream("File_with_Integers");
            BufferedInputStream buff = new BufferedInputStream(file);
            DataInputStream data = new DataInputStream(buff);
            try {
                while (true) {
                    int in = data.readInt();
                    System.out.print(in + " ");
                }
            } catch (EOFException eof) {
                buff.close();
            }
        } catch (IOException e) {
            System.out.println("Error -- " + e.toString());
        }
    }
}
```

ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ

Γ' ΕΞΑΜΗΝΟ



```
import java.io.*;

public class WriteIntsToFile {

    public static void main(String arguments[]) {
        int[] matrix = new int[40];
        for (int i=0; i<matrix.length; i++)
            matrix[i]=i*20;

        try {
            FileOutputStream file = new FileOutputStream("File_with_Integers.dat");
            DataOutputStream data = new DataOutputStream(file);
            for (int i = 0; i<matrix.length; i++)
                data.writeInt(matrix[i]);
            data.close();
        } catch (IOException e) {
            System.out.println("Error -- " + e.toString());
        }
    }
}
```

ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ

Γ' ΕΞΑΜΗΝΟ

```
import java.io.*;

public class WriteIntsToFile {

    public static void main(String arguments[]) {
        int[] matrix = new int[40];
        for (int i=0; i<matrix.length; i++)
            matrix[i]=i*20;

        try {
            FileOutputStream file = new FileOutputStream("File_with_Integers.dat");
            BufferedOutputStream buff = new BufferedOutputStream(file);
            DataOutputStream data = new DataOutputStream(buff);
            for (int i = 0; i<matrix.length; i++)
                data.writeInt(matrix[i]);
            data.close();
        } catch (IOException e) {
            System.out.println("Error -- " + e.toString());
        }
    }
}
```

ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ

Γ' ΕΞΑΜΗΝΟ