

2011 - 2012

Programare Orientata spre Obiecte (*Object-Oriented Programming*)

a.k.a. Programare Obiect-Orientata

Titular curs: Eduard-Cristian Popovici

Suport curs: <http://discipline.elcom.pub.ro/POO-Java/>

3. Programarea la nivel socket cu Java

3.1. Clase pentru fluxuri de intrare-iesire (IO)



Clase pentru fluxuri IO

Programele pot avea nevoie de a:

- **prelua** informatii **de la surse** externe
- **trimite** informatii **catre destinatii** externe

Sursa / destinatia poate fi:

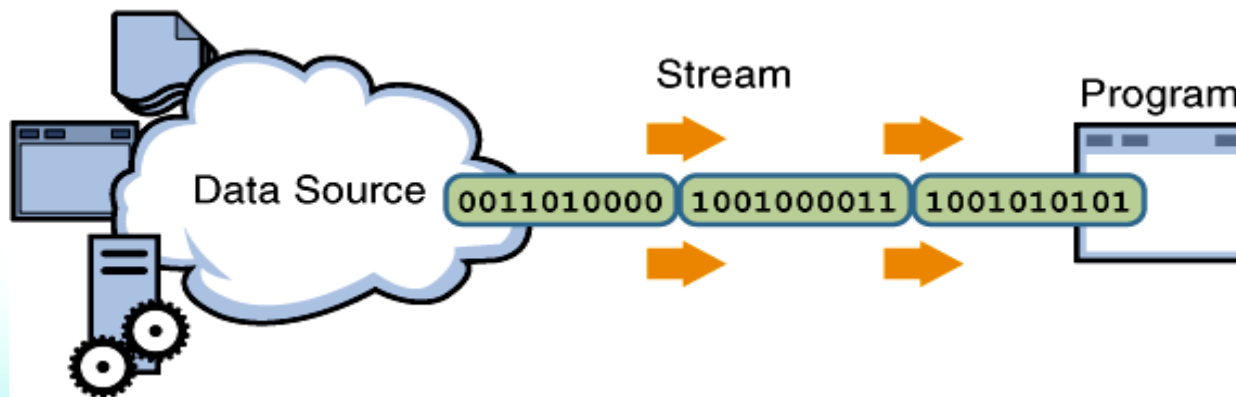
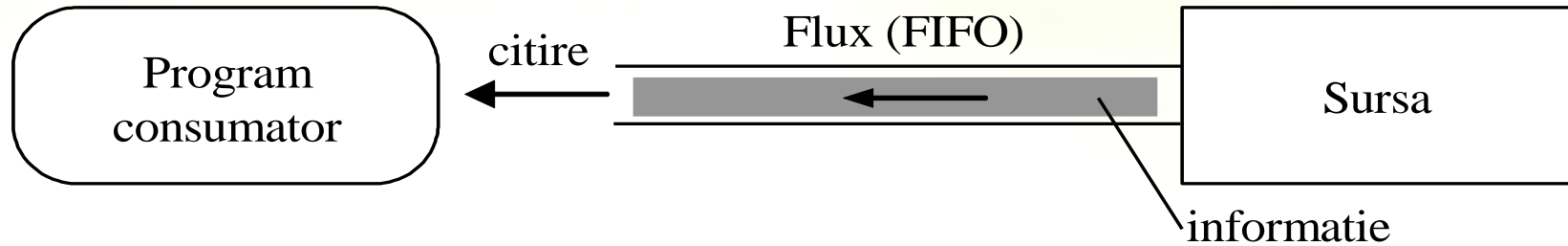
- **fisier pe disc**
- **retea (socket)**
- **memorie (program)**
- **dispozitiv IO (ecran, tastatura)**



Clase pentru fluxuri IO

Pentru preluarea informatiilor programul

- **deschide** un flux de intrare de la o sursa de informatii si
- **citeste** informatiile **secvential**

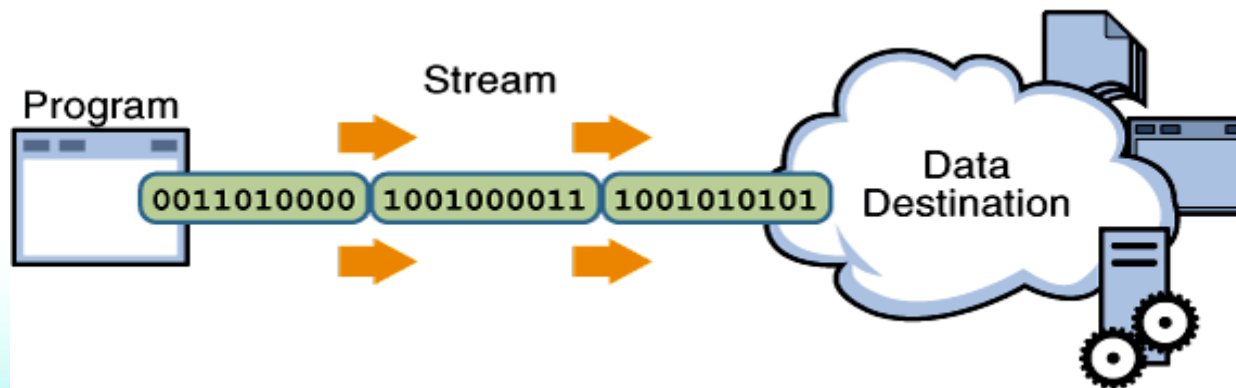
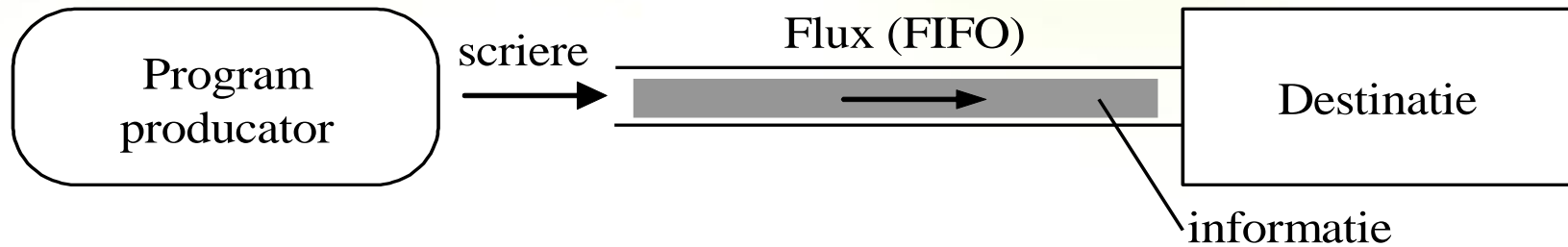




Clase pentru fluxuri IO

Pentru trimiterea informatiilor programul

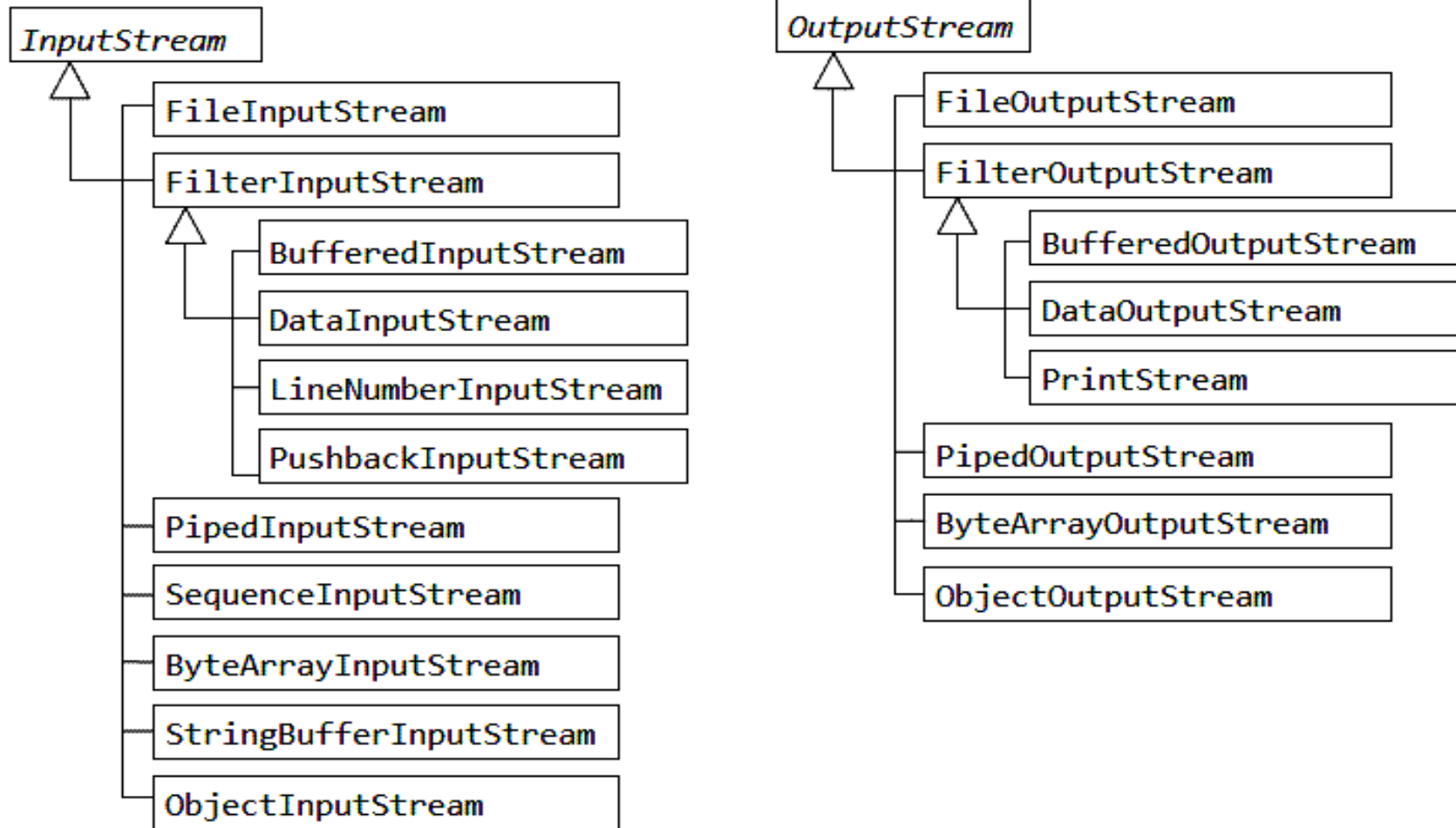
- **deschide** un **flux de iesire** catre o **destinatie** a informatiilor si
- **scrie** informatiile **secvential**





Clase pentru fluxuri IO

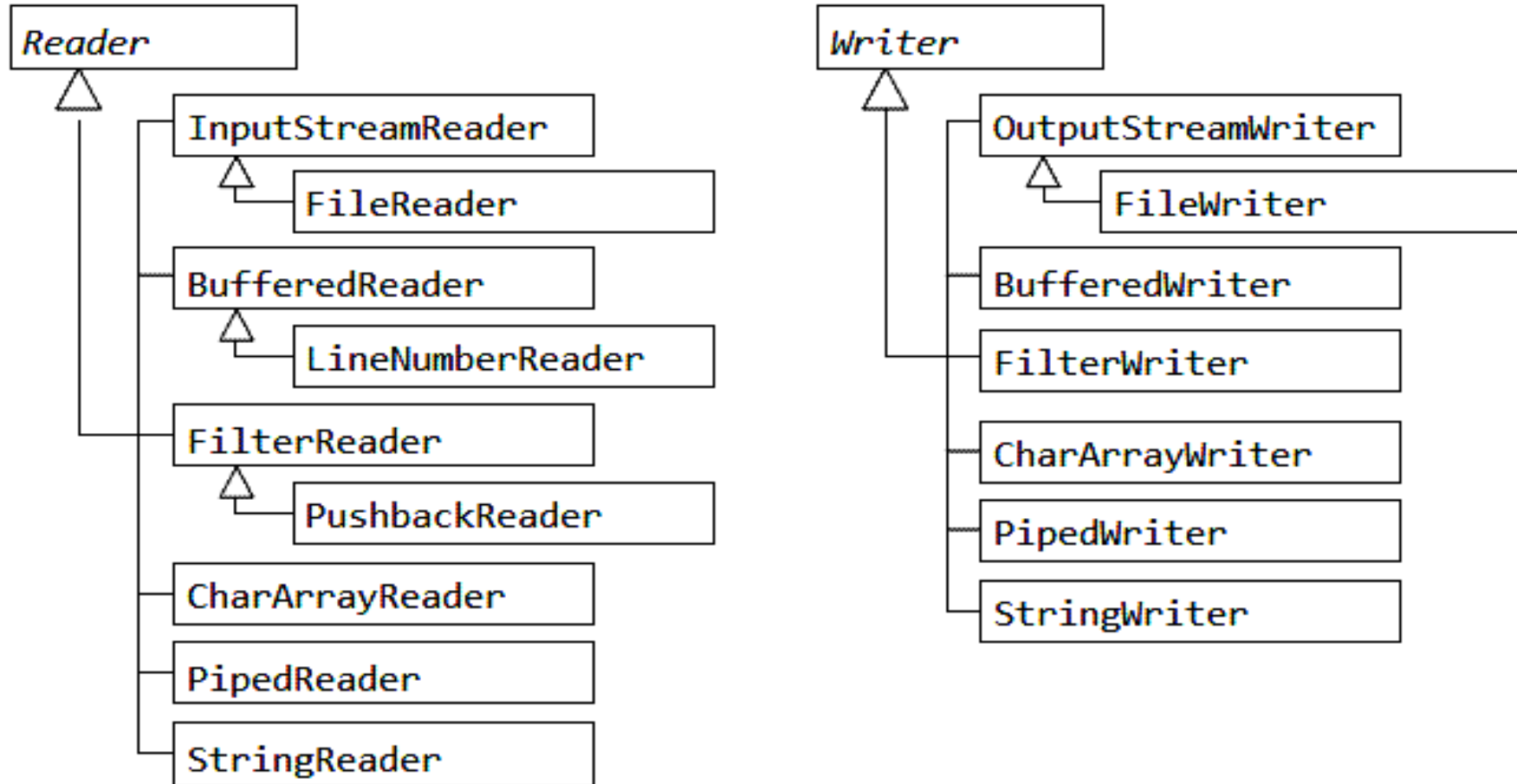
Fluxuri de octeti





Clase pentru fluxuri IO

Fluxuri de caractere





Clase pentru fluxuri IO

In functie de specializarea pe care o implementeaza, subclasele claselor abstracte se impart in alte doua categorii:

- **terminale** (*data sink*), care nu au ca sursa / destinatie alte fluxuri, ci:
 - fisierele
 - memoria (tablourile)
 - reseaua (socketurile)
 - sirurile de caractere (String)
 - alte programe (prin conducte - *pipes*)
- **de prelucrare** (*processing*), care au ca sursa / destinatie alte fluxuri, si au ca rol **prelucrarea informatiilor**:
 - *buffer-are* (stocare temporara)
 - filtrare de diferite tipuri (conversie, contorizare, etc.)
 - tiparire

Clase pentru fluxuri IO

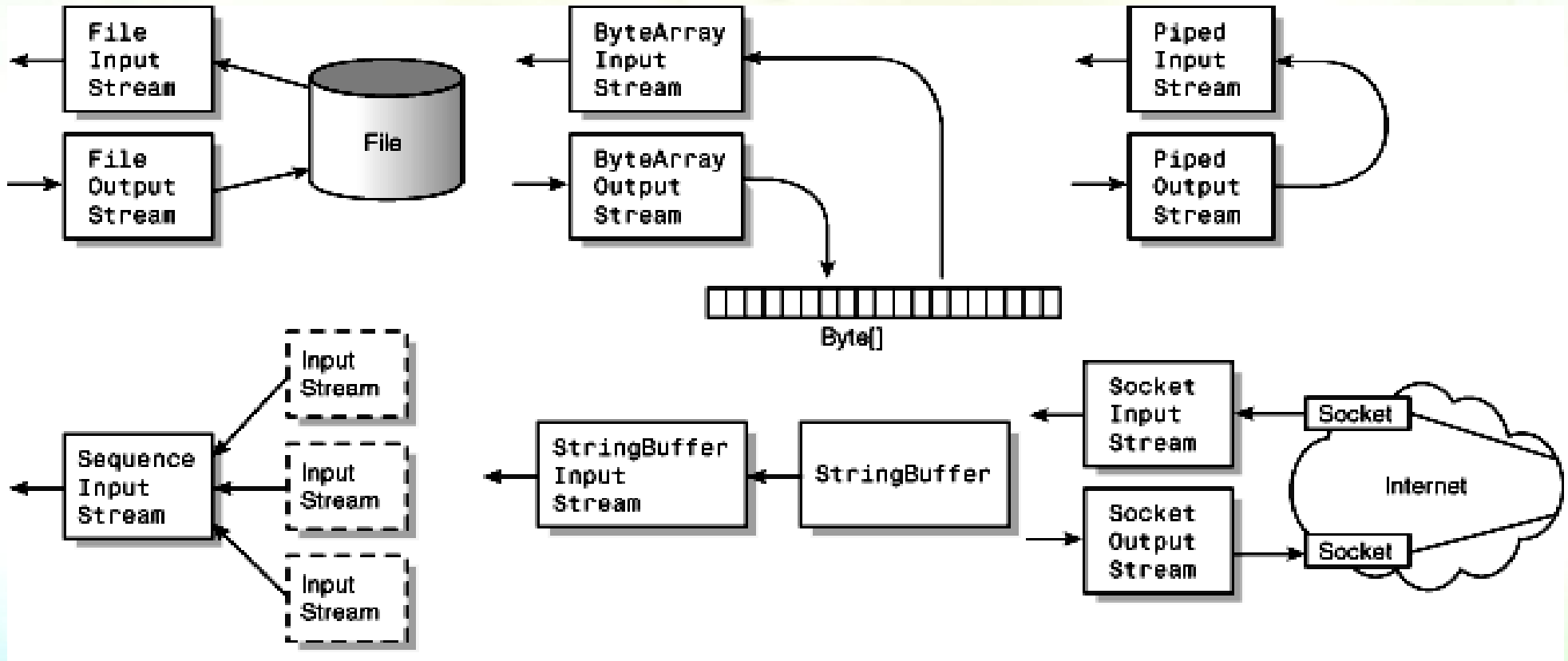
Tipuri de fluxuri terminale - NU au ca sursa / destinatie alte fluxuri

| Tip de Terminal | Utilizare | Fluxuri de caractere | Fluxuri de octeti |
|-------------------------|--|--------------------------|-------------------------------------|
| Memorie | <i>Accesul secvential la tablouri</i> | CharArrayReader | ByteArrayInputStream |
| | | CharArrayWriter | ByteArrayOutputStream |
| | <i>Accesul secvential la siruri de caractere</i> | StringReader | StringBufferInputStream |
| | | StringWriter | StringBufferOutputStream |
| Canal / conducta (pipe) | <i>Conducte intre programe</i> | PipedReader | PipedInputStream |
| | | PipedWriter | PipedOutputStream |
| Fisier | <i>Accesul la fisiere</i> | FileReader FileWriter | FileInputStream FileOutputStream |



Clase pentru fluxuri IO

Fluxuri terminale - NU au ca sursa / destinatie alte fluxuri





Clase pentru fluxuri IO

Tipuri de fluxuri de prelucrare - AU ca sursa / destinatie alte fluxuri

| Tip de Prelucrare | Utilizare | Fluxuri de Caractere | Fluxuri de octeti |
|---------------------------------|--|---------------------------------|------------------------------------|
| <i>Buffer-are</i> | <i>Stocare temporară</i> | <code>BufferedReader</code> | <code>BufferedInputStream</code> |
| | | <code>BufferedWriter</code> | <code>BufferedOutputStream</code> |
| Filtrare | <i>Prelucrare</i> | <code>FilterReader</code> | <code>FilterInputStream</code> |
| | | <code>FilterWriter</code> | <code>FilterOutputStream</code> |
| Conversie octet/caracter | <i>Bridge byte-char</i> | <code>InputStreamReader</code> | |
| | | <code>OutputStreamWriter</code> | |
| Concatenare | <i>Prelucrare</i> | | <code>SequenceInputStream</code> |
| Serializarea obiectelor | | | <code>ObjectInputStream</code> |
| | | | <code>ObjectOutputStream</code> |
| Conversia datelor | <i>Acces la tip date primitiv Java</i> | | <code>DataInputStream</code> |
| | | | <code>DataOutputStream</code> |
| Numararea | <i>Numarare linii</i> | <code>LineNumberReader</code> | <code>LineNumberInputStream</code> |
| Imprimare | <i>Tiparire</i> | <code>PrintWriter</code> | <code>PrintStream</code> |



Fluxuri de date formatate

**Citirea unui nume de la tastatura folosind inlantuirea (cascada),
DataInputStream, BufferedInputStream:**

```
DataInputStream in = new
    DataInputStream(new BufferedInputStream(System.in));
System.out.println("Introduceti numele: ");
String nume = in.readLine();
```

Afisarea argumentelor programului curent folosind un DataOutputStream in cascada cu System.out

```
DataOutputStream dos = new DataOutputStream (System.out);
dos.writeBytes("Argumentele programului: \n");
for (int i=0; i<args.length; i++) {
    dos.writeBytes(args[i] + " ");
}
dos.writeChar('\n');
dos.flush();
```




Cascada de fluxuri – inlantuirea fluxurilor

Citirea de la tastatura: pentru eficienta maxima, este recomandata inlantuirea (plasarea in cascada), astfel:

```
BufferedReader in = new BufferedReader(new InputStreamReader(System.in));
```

Citirea unui nume de la tastatura:

```
BufferedReader in = new BufferedReader(new InputStreamReader(System.in));
System.out.println("Introduceti numele: ");
String nume = in.readLine();
```

