

Input/Output in JAVA

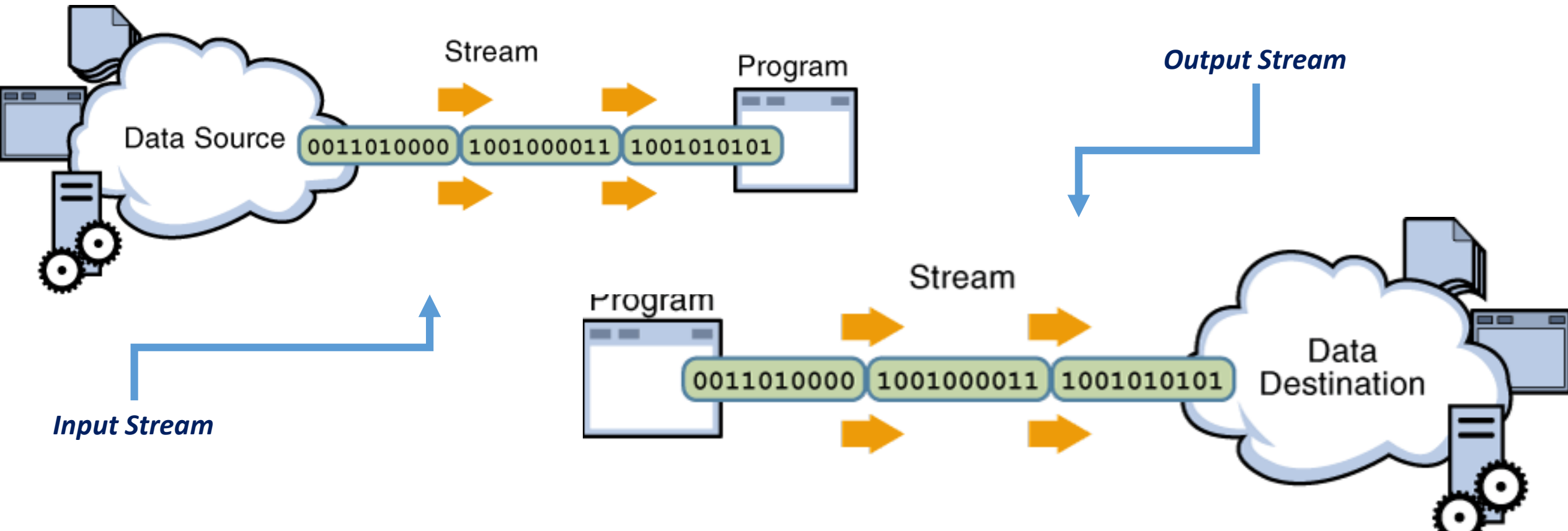
Stampa sul monitor e Lettura da tastiera

LETTURA DA TASTIERA

- In Java esistono fondamentalmente due modi per leggere un dato da tastiera:
 1. Attraverso gli **stream**;
 2. Utilizzando la classe **Scanner** ;
- Il primo metodo è sicuramente quello più generale perché consente di leggere dati (byte o caratteri) da una Sorgente (in questo caso la tastiera, ma puoi essere un file) ad una destinazione in questo caso il programma in esecuzione (ma può essere un altro file su cui scrive il programma stesso)

GLI STREAM IN JAVA

- In JAVA uno stream è **un canale di comunicazione**
 1. **monodirezionale** (o è di input o è di output);
 2. **Ad uso generale** (è riferito in generale ad una sorgente e ad una destinazione);
 3. Che **trasferisce byte (o caratteri)**



3 importanti stream (*in, out, err*)

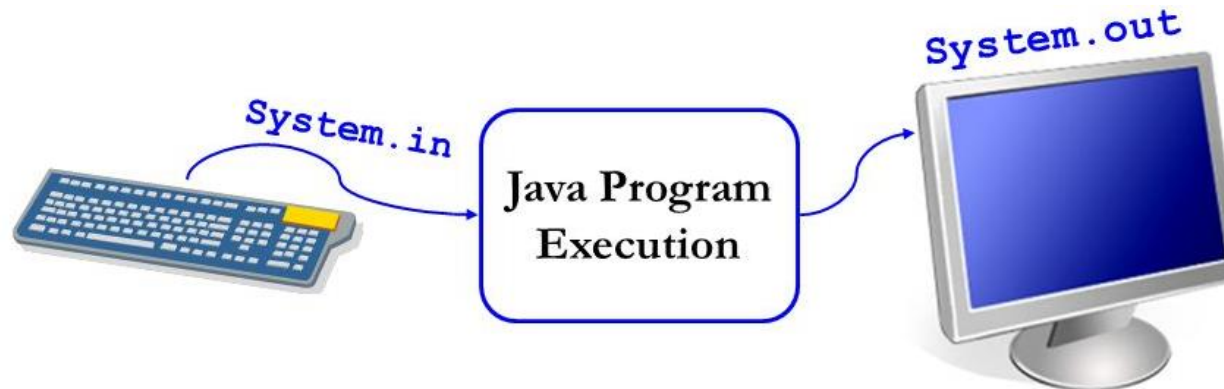
Questi stream si traducono poi in una serie di **classi di base**; queste sono alla base della tassonomia delle classi di stream realizzate da Java

Il **package java.io** distingue tra stream di byte (file binari) e stream di caratteri (file di testo)

La tastiera è normalmente collegata al campo statico (*in*) presente nella classe di sistema **System**. Si utilizzerà questo campo in per costruire il canale di comunicazione tra la tastiera e il programma. Quindi il programma vede solo un flusso di dati che esce da un canale di comunicazione. Sotto queste ipotesi la tastiera e' vista da un programma Java come **System.in**

Il monitor normalmente collegata al campo statico (*out*) presente nella classe di sistema **System**. Cioè la monitor è visto da un programma Java come **System.out**

Un altro campo statico della classe System è **err** che normalmente è collegato (o coincide) con la console in cui si esegue il programma



Lettura di un intero da tastiera

La gestione dell'input avviene mediante la classe **InputStream** fornita da System. Tale classe tuttavia dispone solo di alcune funzionalità di base di gestione dell'input. Il metodo `read()` in essa definita, infatti consente di leggere un solo carattere alla volta.

Se vogliamo sviluppare un programma dobbiamo utilizzare funzioni più evolute, come per esempio la lettura di un'intera riga di testo quella che l'utente digita con la pressione del tasto INVIO. Il metodo **`readLine()`** della classe **BufferedReader** legge sequenze di caratteri che restituisce sotto forma di stringhe

Per leggere un numero intero, un numero in virgola mobile, una stringa o qualsiasi tipo di dato, è necessario quindi eseguire una conversione di tipo.

In pratica per leggere un intero si procederà così:

1. creiamo un canale di tipo **InputStreamReader** (utilizzando **System.in**);
2. creiamo un canale tramite **BufferedReader** che serve a prendere quello che esce dal canale **InputStreamReader** (che sono byte) e che trasformerà in stringhe;
3. Infine, e' EVENTUALMENTE NECESSARIO TRAFORMARE l'uscita del canale di **BufferedReader** (che e' di tipo stringa) in un intero o in un float etc...

In pratica per leggere un intero si procederà così:

1. creiamo un canale di tipo **InputStreamReader** (utilizzando System.in) e lo chiamiamo **input**;

InputStreamReader **input** = new InputStreamReader (**System.in**);

2. creiamo un canale tramite **BufferedReader** utilizzando input (il canale di prima) e lo chiamiamo **tastiera** che serve a prendere quello che esce dal canale *InputStreamReader*, cioè *da input* (che sono byte) e che trasformerà in stringhe;

BufferedReader **tastiera** = new BufferedReader (input);

3. Utilizzando il metodo **readLine()** posseduto dall'oggetto di tipo BufferedReader (cioè da **tastiera**) posso leggere stringhe dalla tastiera fisica e mettere quanto acquisito in una variabile **str** (che deve quindi essere di tipo stringa)

str = tastiera.readLine();

4. Se quanto letto deve essere trattato come intero, e' EVENTUALMENTE NECESSARIO TRASFORMARE l'uscita del canale di **BufferedReader** (cioè da tastiera che e' di tipo stringa) in un intero **nlntero**.

nlntero = Integer.valueOf (str).intValue()

Come se non bastasse.....

- La lettura da stream può generare eccezioni quindi deve essere racchiusa tra il blocco di gestione delle eccezioni **try{}catch{}**

Esempio:

```
/*  
Scrivere un programma che  
acquisisce un nome e un  
numero che rappresenta il  
saldo di un conto corrente e  
lo visualizza in output  
*/
```

```
import java.io.*;  
public class LeggiNomeNumero {  
  
    public static void main(String[] args) {  
        String nome, str;  
        int saldo;  
  
        InputStreamReader input = new InputStreamReader (System.in);  
        BufferedReader tastiera = new BufferedReader (input);  
  
        try {  
            //acquisico il nome  
            System.out.print("Digita il nome: ");  
            nome = tastiera.readLine();  
  
            //acquisico il saldo  
            System.out.print("\nDigita il relativo importo: ");  
            str = tastiera.readLine();  
            saldo = Integer.valueOf (str).intValue();  
  
            //stampa il risultato  
            System.out.print("\nIl Dott. " + nome +  
                             " ha un saldo pari a: " + saldo);  
        } catch (IOException e) {  
            System.err.println("\n si e' verificato un errore");  
        } //try  
    } //main  
} //class
```

/*

Il blocco try/catch viene utilizzato per gestire eventuali errori che si possono verificare in fase di lettura da un flusso di informazioni (Stream). In questo specifico caso è difficile che si verificano problemi da lettura da tastiera: se questa smettesse di funzionare probabilmente non si vedrebbero più sul video i caratteri digitati.

Si pensi se la lettura avvenisse da un disco rimovibile l'espulsione anticipata del dischetto farebbe mancare le informazioni che si stanno leggendo; in tal caso viene individuato da JAVA un errore del tipo IOException. La gestione degli errori sarà ripresa più avanti

*/

```
try {  
    //acquisico il nome  
    System.out.print("Digita il nome: ");  
    nome = tastiera.readLine();  
  
    //acquisico il saldo  
    System.out.print("\nDigita il relativo importo: ");  
    str = tastiera.readLine();  
    saldo = Integer.valueOf (str).intValue();  
  
    //stampa il risultato  
    System.out.print("\nIl Dott. " + nome +  
        " ha un saldo pari a: " + saldo);  
} catch (IOException e) {  
    System.err.println("\n si e' verificato un errore");  
} //try
```