
I file in Java

File

I **file** sono un insieme di dati memorizzati su memoria di massa. Sono gli "archivi elettronici".

Se il file è una successione di record con la stessa struttura si dice **file di record o strutturati**, se è una successione di righe di caratteri si parla di **file di testo**.

Il modo con cui un programma utilizza i dati memorizzati rappresenta il **tipo di accesso** al file, che può essere:

- **Sequenziale**
- **Diretto o random**

File di testo o binari

Il SO tratta i file in 2 modi, come:

- **File di testo:** ovvero come successione di caratteri, distinguendo una riga da un'altra grazie ad un carattere di fine riga. ('\n')
- **File binari:** successione di byte senza distinguere una riga da un'altra. (per i file di record, per poter interpretare i dati si deve conoscere la struttura)

Per poter trattare i dati memorizzati su un file, questi si devono trasferire dalla memoria di massa alla memoria centrale (operazione di *input*) o viceversa (operazione di *output*). Il trasferimento fisico è relativo non ad un singolo byte, ma ad un insieme di byte: **blocco fisico** (che, in generale, contiene 1 o più record - righe). Dal punto di vista del programmatore il trasferimento è relativo ad un solo record - riga.

File

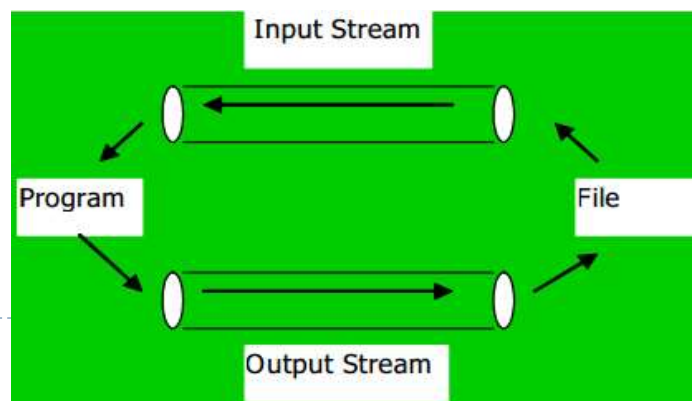
operazioni

- **Apertura:**
 - si crea un collegamento tra MM e MC associando al **nome logico** del file (riconosciuto all'interno del programma) il **nome fisico** (riconosciuto a livello di SO)
 - Si riserva in MC un buffer per le operazioni di I/O
 - Si aggiornano le tabelle di gestione dei file per specificare le informazioni necessarie per l'individuazione dei dati su MM
 - **lettura:** si trasferiscono i dati dalla MM al buffer relativo al file
 - **scrittura:** si trasferiscono i dati dal buffer relativo al file alla MM
 - **chiusura:** si aggiornano le tabelle di gestione file
-

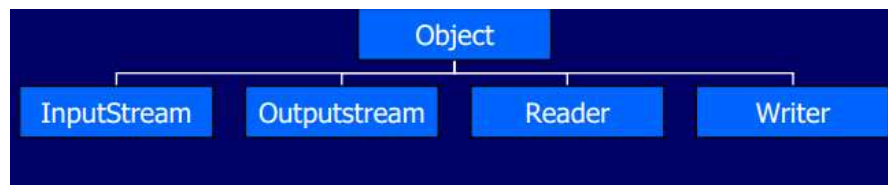
Stream

Uno **stream** o **flusso** è una sequenza ordinata di dati:

- ▶ **monodirezionale** (o di input, o di output)
- ▶ adatto a trasferire byte (o anche caratteri)
- ▶ **Buffer**: deposito di dati da (su) cui si può leggere (scrivere)
- ▶ **Canale**: connessione con entità in grado di eseguire operazioni di I/O; i canali includono i buffer, i file e i socket
- ▶ **Stream di Input**: i dati sono presi da una sorgente (file o tastiera o rete) e trasferiti al programma
- ▶ **Stream di Output**: i dati sono generati dal programma e trasferiti ad una destinazione (file o video o rete)



Stream e file



stream di byte

stream di caratteri

I flussi permettono operazioni di I/O

Il relativo package è **java.io**, in cui ci sono 4 classi astratte di base.

I flussi si possono concatenare uno dopo l'altro, così da sfruttare il fatto che ogni stream vede i dati e li tratta in modo diverso con metodo propri.

In un flusso non è necessario conoscere la fonte per leggere un flusso di input come non è necessario conoscere il destinatario per poter inviare (scrivere) un flusso, pertanto un **file** può essere visto come un particolare flusso di dati scritto su memoria di massa.

I File in Java possono essere:

- ▶ **binari** (flusso di byte a 8 bit)
- ▶ di **testo** (flusso di caratteri ASCII)
- ▶ di tipi primitivi
- ▶ di oggetti

Questi tipi possono solo essere gestiti in modo sequenziale