

23. Scheda di lavoro (gestione file di testo in linguaggio Java)

11/12/2018

In Java per poter leggere/scrivere dai/sui file si usano gli stream (flussi)

<p>Uno stream è un canale di comunicazione :</p> <ul style="list-style-type: none"> • monodirezionale (o è input o è output) • che trasferisce byte (8 bit) o anche caratteri (16 bit, Unicode) 	
---	--

<p>In particolare uno stream può essere associato alla tastiera,</p>	<p>ma stesso modo potrebbe essere associato ad un file.</p>
--	---

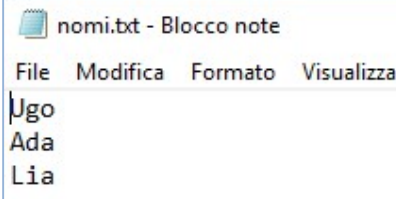
Il package **java.io** distingue tra stream di byte (file binari) e stream di caratteri (file testuali) ma mette a disposizione classi che non sono molto comode da utilizzare per le gestione dei file di testo: per questo **Java 5.0** ha introdotto la classe **Scanner** nel package **util**.

Letture di un file con java.util.Scanner, associando allo scanner non la tastiera, ma il mio file di cui indico il percorso.

```

1 package pkg4binfo;
2 import java.io.*;
3 import java.util.Scanner;
4 public class Main {
5     public static void main(String[] args) {
6         FileReader reader = new FileReader("E:\\dati\\nomi.txt");
7         Scanner in = new Scanner(reader);
8
9         //leggo il file riga per riga, fino a quando ci sono righe
10        while (in.hasNextLine()) {
11            //leggo una riga del file
12            String line = in.nextLine();
13            //elaboro la linea che ho appena letto
14            System.out.println(line);
15        }
16
17        //chiudo il file
18        reader.close();
19    } //fine main
20 }
    
```

Ottingo però un errore, perché il file potrebbe non esistere, e quindi verrebbe lanciata l'eccezione **FileNotFoundException** a gestione obbligatoria: inserisco un try ... catch



```

nomi.txt - Blocco note
File Modifica Formato Visualizza
Ugo
Ada
Lia

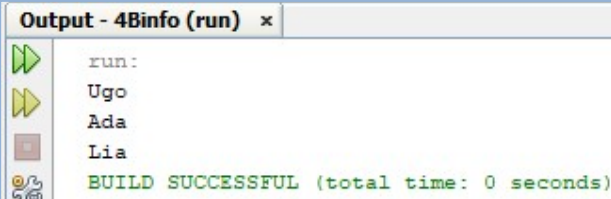
```

```

1 package pkg4binfo;
2 import java.io.*;
3 import java.util.Scanner;
4 public class Main {
5     public static void main(String[] args) {
6         try {
7             FileReader reader = new FileReader("E:\\dati\\nomi.txt");
8             Scanner in = new Scanner(reader);
9
10            //leggo il file riga per riga, fino a quando ci sono righe
11            while(in.hasNextLine()){
12                //leggo una riga del file
13                String line = in.nextLine();
14                //elaboro la linea che ho appena letto
15                System.out.println(line);
16            }
17
18            //chiudo il file
19            in.close();
20
21        } catch (FileNotFoundException e) {
22            System.out.println("Il file non è stato trovato!");
23            //e.printStackTrace();
24        }
25    }
26 } //fine main

```

In caso positivo ottengo a video la lista dei nomi del file

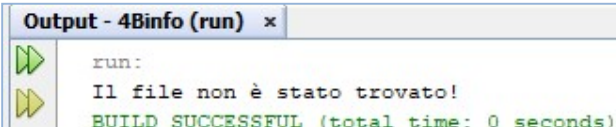


```

Output - 4Binfo (run) x
run:
Ugo
Ada
Lia
BUILD SUCCESSFUL (total time: 0 seconds)

```

Altrimenti un messaggio di errore




```

Output - 4Binfo (run) x
run:
Il file non è stato trovato!
BUILD SUCCESSFUL (total time: 0 seconds)

```

Oppure chiedo esplicitamente di non gestire l'eventuale errore

```
1 package pkg4binfo;
2 import java.io.*;
3 import java.util.Scanner;
4 public class Main {
5     public static void main(String[] args) throws FileNotFoundException {
6
7         FileReader reader = new FileReader("E:\\dati\\nomi.txt");
8         Scanner in = new Scanner(reader);
9
10        //leggo il file riga per riga, fino a quando ci sono righe
11        while(in.hasNextLine()){
12            //leggo una riga del file
13            String line = in.nextLine();
14            //elaboro la linea che ho appena letto
15            System.out.println(line);
16        }
17
18        //chiudo il file
19        in.close();
20
21    } //fine main
22 }
```



E in caso tutto sia ok ottengo lo stesso output di prima

```
Output - 4Binfo (run) x
run:
Ugo
Ada
Lia
BUILD SUCCESSFUL (total time: 0 seconds)
```

Ma posso anche invece ottenere

```
Output - 4Binfo (run) x
run:
Exception in thread "main" java.io.FileNotFoundException: E:\dati\nomi.txt (Impossibile trovare il file specificato)
    at java.io.FileInputStream.open0(Native Method)
    at java.io.FileInputStream.open(FileInputStream.java:195)
    at java.io.FileInputStream.<init>(FileInputStream.java:138)
    at java.io.FileInputStream.<init>(FileInputStream.java:93)
    at java.io.FileReader.<init>(FileReader.java:58)
    at pkg4binfo.Main.main(Main.java:7)
Java Result: 1
BUILD SUCCESSFUL (total time: 0 seconds)
```