

Appunti Lezione 01

1 Gestione di file input/output

Codice di riferimento: `ioexample.C`

La classe `fstream` permette di gestire file in input e output. I file sono associati alle variabili `in` e `fout` alle righe 9 e 16. Il blocco di codice tra le righe 9 e 21 si occupa dell'apertura dei file con eventuale controllo degli errori.

Per leggere un valore dal file di input si usa il comando `>>`, che è *overloaded* per la classe `fstream`:

```
in >> variabile;
```

Per scrivere un valore sul file di output si usa il comando `<<`, anch'esso *overloaded* per la classe `fstream`:

```
fout << variabile;
```

Dopo aver finito di usare i file, è buona norma chiuderli con il comando `close()`.

Il codice `ioexample.C` prende in input il nome del file di input e il nome del file di output, dopodiché legge coppie di numeri contenute nel file di input e le trascrive nel file di output.

2 Utilizzo delle macro di ROOT

Codice di riferimento: `ioexample.C`

Le macro di ROOT vengono convenzionalmente salvate con estensione `.C` e possono essere eseguite direttamente nell'interprete di ROOT. Per caricare il file `ioexample.C` nell'interprete di ROOT, si usa il comando `.L`, quindi si può eseguire la macro passandole come argomento i nomi dei file di input e di output come richiesto dalla funzione alla riga 7:

```
root[ ] .L ioexample.C
```

```
root[ ] ioexample("input.dat","output.dat")
```

Dove i file `input.dat` e `output.dat` si trovano nella stessa cartella della macro. In alternativa, si possono specificare i percorsi assoluti o relativi dei file. Convenzionalmente si usa l'estensione `.dat` per i file contenenti dati numerici separati da spazi.

3 Codici

3.1 ioexample.C

```
1 #include <iostream>
2 #include <fstream>
3 #include <string>
4
5 using namespace std;
6
7 void ioexample(string fimpName, string foupName)
8 {
9     ifstream in(fimpName);
10    if (!in)
11    {
12        cout << "Il file " << fimpName << " non esiste." << endl;
13        return;
14    }
15
16    ofstream fout(foupName);
17    if (!fout)
18    {
19        cout << "Non posso aprire il file di output " << foupName << endl;
20        return;
21    }
22
23    double number1, number2;
24    int count = 0;
25
26    while (in >> number1 >> number2)
27    {
28        count++;
29
30        cout << count << " ) ";
31        cout << "Number 1 = " << number1 << ";\t Number 2 = " << number2 << endl;
32
33        fout << count << " ) ";
34        fout << "Number 1 = " << number1 << ";\t Number 2 = " << number2 << endl;
35    }
36
37    in.close();
38    fout.close();
39 }
```