

Fisiere text in C++

Fisier text=succesiune de caractere ASCII ce se incheie cu caracterul **EOF**.

Accesul la caracterele scrise intr-un fisier text se poate face doar **secvential**, si poate fi :

- Acces pentru citire
- Acces pentru scriere

Toate utilitarele necesare pentru accesarea fisierelor text din programe C++ se afla in biblioteca **fstream.h**

Pentru a putea accesa un fisier text dintr-un program C++ trebuie declarata o **variabila**, astfel :

- Pentru a avea acces la afisare in fisier : **ofstream g;**
- Pentru a avea acces la citirea din fisier : **ifstream f;**

Cand ne hotarâm sa accesam fisierul, mai intai trebuie sa-l **"deschidem"**. De fapt, creem legatura logica intre variabila declarata pentru accesarea fisierului si zona de pe disc unde se gaseste fisierul. Dupa crearea acestei legaturi, ori de cate ori dorim sa accesam fisierul vom accesa de fapt variabila asociata lui.

Ex.

```
f.open("intrare.txt");  
g.open("iesire.txt");  
f.open("c:\\teza\\unu.in");  
ifstream f("intrare.txt");  
ofstream g("iesire.txt");
```

} cele 2 fisiere se cauta in folderul de lucru

} fisierul se cauta pe calea specificata

} initializare variabila de acces, la declarare

Cand ne hotarâm sa nu mai accesam fisierul va trebui sa-l **"inchidem"**. De fapt, distrugem legatura logica intre variabila declarata pentru accesarea fisierului si zona de pe disc unde se gaseste fisierul.

Ex.

```
f.close(); g.close();
```

Afisarea in fisier

Se face la fel ca si afisarea pe ecran numai ca metoda **cout** se inlocuieste cu numele variabilei de acces la fisier.

Ex.

```
g<<17<<' ';<br>g<<"ana are mere"<<endl;
```

Aplicatie : Sa se afiseze in fisierul text **unu.out**, toate numerele de la 20 la 1, in ordine descrescatoare si separate prin cate un spatiu.

```
#include <fstream.h>  
int main() {  
    int i;  
    ofstream g("unu.out");  
    for (i=20; i>=1; i--)  
        g<<i<<' ';<br>    g.close();  
    return 0;  
}
```

Citirea din fisier

- Operatorul >> permite citirea datelor in variabila de orice tip, fara insa a trata caracterele albe (spatii, salturi la linie noua).
Citirea se face similar celei de la tastatura numai ca se inlocuieste metoda **cin** cu variabila de acces la fisier.

Ex1: Se considera fisierul text **unu.in** care contine pe o singura linie si separate prin spatii 10 numere intregi de maxim 4 cifre fiecare. Sa se afiseze pe ecran suma acestor numere.

```
#include <fstream.h>
#include <iostream.h>
int main() {
    int x; long s=0;
    ifstream f("unu.in");
    int i;
    for(i=1;i<=10;i++) {
        f>>x; s=s+x; }
    f.close();
    cout<<s;
    return 0;
}
```

Ex2: Se considera fisierul text **doi.in** care contine: pe prima linie numarul natural nenul **n** ($n \leq 50$) iar pe linia a doua, separate prin spatiu, n numere naturale cu maxim 9 cifre fiecare. Sa se afiseze pe ecran cel mai mare dintre cele **n** numere.

```
#include <fstream.h>
#include <iostream.h>
int main() {
    long x, max; int n,i;
    ifstream f("doi.in");
    f>>n>>max;
    for(i=2;i<=n;i++) {
        f>>x;
        if (x>max) max=x; }
    f.close();
    cout<<max;
    return 0;
}
```

Ex3: Se considera fisierul text **trei.in** care contine pe mai multe linii numere intregi in cadrul fiecarei linii fiind separate prin spatiu. Sa se afiseze, in fisierul text **trei.out**, toate numerele continute in **unu.in** separate printr-un spatiu.

```
#include <fstream.h>
int main() {
    int x;
    ifstream f("trei.in");
    ofstream g("trei.out");
    while (f>>x)
        g<<x<<" ";
    g.close();f.close();
    return 0;
}
```

Explicatie : expresia **f>>x** va avea valoarea **1** (adevarat) daca citirea s-a produs respectiv valoarea **0** (fals) daca citirea nu s-a produs – deoarece fisierul s-a terminat.

}