

Informatica

C++

Prof. Bortolai Gabriele



Librerie

L'ultima volta abbiamo migliorato l'aspetto del nostro programma utilizzando le funzioni.

Le domande a cui vorremmo rispondere oggi sono:

1. Si può far di meglio rispetto la volta scorsa?
2. Esiste un modo per svolgere le operazioni più frequenti senza che si debba riscrivere il codice ogni volta da capo?

La risposta ad entrambe le domande é ovviamente si, basta usare le librerie.



Librerie

Per utilizzare le librerie bisogna spezzare il file con all'interno il `mai()` in altri tre file, più precisamente:

1. `nome.h`
2. `nome.cpp`
3. `programma.cpp`

Nel primo file bisogna inserire le definizioni delle funzioni che si erano definite nel vecchio programma.

Nel secondo file bisogna per prima cosa includere il file `nome.h` e successivamente definire in modo completo la funzione e le operazioni che deve svolgere.

Nel terzo file bisogna includere il file `nome.h` e scrivere il codice richiamando le funzioni scritte in `nome.cpp` quando necessario.

Librerie

Compilazione

Avendo spezzato il programma in tre file diversi bisogna compilare in modo diverso.

Per prima cosa va compilato il file nome.cpp utilizzando il seguente comando:

```
g++ -c nome.cpp
```

Se la compilazione va a buon fine si otterrà un file nome.o

Dopo di che possiamo compilare il nostro programma linkando la libreria, é possibile fare ciò utilizzano il seguente comando:

```
g++ -o programma nome.o programma.cpp
```

C++

Librerie

A destra é possibile veder il vecchio programma, iniziamo a costruire una libreria che quando viene utilizzata restituisce la dimensione di un file a due colonne.

Chiamo:

1. size.h l'header file;
2. size.cpp il source file;

```
#include <iostream>
#include <fstream>

using namespace std;

//Functions
int size (string);
double* media (int n, double [n]);

int main(){

    string filename= "dati.dat";
    int n;

    n=size(filename);

    ifstream ifile;
    ifile.open(filename.c_str());

    ofstream ofile("risultati.dat");

    double* x= new double [n];
    double* y= new double [n];

    for(int i=0;i<n;i++){

        ifile>>x[i]>>y[i];

    }

    double* med1=media(10,x);
    double* med2=media(10,y);

    for(int i=0;i<n/5;i++){

        ofile<<med1[i]<<" "<<med2[i]<<endl;

    }

    ifile.close();
    ofile.close();
    return 0;

}

//Define functions
int size (string filename){

    double d,f;
    int n=0;

    ifstream ifile;
    ifile.open(filename.c_str());

    while (ifile>>d>>f){

        n++;

    }

}
```

C++

Librerie

A destra é possibile veder il vecchio programma, iniziamo a costruire una libreria che quando viene utilizzata restituisce la dimensione di un file a due colonne.

Chiamo:

1. size.h l'header file;
2. size.cpp il source file;

```
double d,f;
int n=0;

ifstream ifile;
ifile.open(filename.c_str());

while (ifile>>d>>f){

    n++;

}

ifile.clear();
ifile.seekg(ios::beg);

return n;
}

double* media (int n, double v[n]){

double sum1,sum2;
double* mid = new double [n];

for(int i=0;i<n;i++){
    sum1=0;

    for(int k=5*i;k<5*i+5;k++){

        sum1=sum1+v[k];

    }

    mid[i]=sum1/5;
}

return mid;

}
```

C++

Header file

Nell'header file inseriamo la definizione della funzione del preambolo, segue il codice di programma:

```
#include <iostream>

using namespace std;

int size (string);
```

C++

Source file

In questo file scriviamo le operazioni che deve svolgere la nostra funzione, ovvero quanti dati sono scritti all'interno del file dati.dat, segue il codice:

Qui includo l'header file

```
#include "size.h"
#include <fstream>

using namespace std;

int size (string filename){

    double d,f;
    int n=0;

    ifstream ifile;
    ifile.open(filename.c_str());

    while (ifile>>d>>f){

        n++;

    }

    ifile.clear();
    ifile.seekg(ios::beg);

    return n;
}
```


C++

Librerie

Adesso possiamo compilare il file size.cpp, segue il comando e il risultato:

```
gabrielebortolai@MacBook-Air-di-Gabriele Settimo % g++ -c size.cpp
gabrielebortolai@MacBook-Air-di-Gabriele Settimo %
```

Se la compilazione va a buon fine otterrete anche il file size.o:

```
gabrielebortolai@MacBook-Air-di-Gabriele Settimo % ls
#alloc.h#      allocx.cpp      function2.cpp   risultati.dat   write.cpp~
#allocx.cpp#    allocx.o        function2.cpp~  size.cpp        write.h
#allocy.cpp#    allocy.cpp      media.cpp       size.cpp~       write.h~
#function2.cpp# allocy.cpp~     media.cpp~     size.h          write.o
alloc.cpp~     allocy.o        media.h         size.h~
alloc.h        dati.dat        media.h~        size.o
alloc.h~       function2       media.o         write.cpp
gabrielebortolai@MacBook-Air-di-Gabriele Settimo %
```

C++

Libreire

Infine compiliamo il programma, segue il comando:

```
[gabrielebortolai@MacBook-Air-di-Gabriele Settimo % g++ -o function2 size.o function2.cpp]
```

Nel caso in cui abbiate più librerie potete aggiungere in i diversi file.o dopo size.o

C++

Librerie

Qui includo gli header file

```
#include <iostream>
#include <fstream>
#include "size.h"
#include "alloc.h"
#include "media.h"
#include "write.h"
```

```
using namespace std;
```

```
int main(){
```

```
    string filename= "dati.dat";
    int n;
```

```
    n=size(filename);
```

Qui faccio la
chiamata alla
funzione per
contare i dati

```
    double* x=allocx(n,filename);
    double* y=allocy(n,filename);
```

```
    ofstream ofile("risultati.dat");
```

```
    double* med1=media(10,x);
    double* med2=media(10,y);
```

```
    write(10,med1,med2);
```

```
    return 0;
```

```
}
```

Se eseguite la stessa cosa anche per le altre funzioni tipo: l'allocazione e lettura del file dati.dat, la scrittura su file dei risultati e il calcolo dei valori medi, ciò che otterrete è il seguente codice estremamente ordinato e semplice da leggere.

Inoltre le librerie le potrete usare anche per altre volte nel caso in cui dobbiate eseguire le stesse operazioni, basterà includerle nel main ().



Librerie

Per compito, su base volontaria, vi lascio da fare le librerie per le altre funzioni ovvero:

1. Allocazione dinamica;
2. Scrittura su file;
3. Calcolo dei valori medi;

Il programma di partenza é il medesimo usato per le funzioni.

Su Classroom allegherò le soluzioni e tutti i file necessari per compilare.