

LIBRERIE

Definizione: una LIBRERIA è, col suo di verso,

- una collezione di funzioni/procedure
- tipicamente accomunate da qualcosa:
 - tipologie operazioni
 - descrizione e manipolazione di entità
- utilizzabile da diversi programmi
- utilizzabile, previa compilazione, in diversi ambienti

Una LIBRERIA NON è un programma!

⇒ NON HA UNA FUNZIONE MAIN

Avete già usato librerie: iostream

fstream
cmath
...

CREAZIONE LIBRERIE

Una libreria è costituito da:

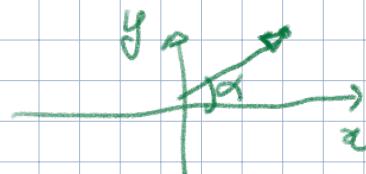
- File HEADER : contiene:
 - dichiarazione dei tipi di dato non di sistema usati / definiti dalla libreria
 - dichiarazione delle funzioni / procedure della libreria
- File SORGENTE: Contiene le definizioni delle funzioni della libreria
- File OGGETTO: contiene i record di attivazione di ogn. funzione/procedura definiti nelle librerie generate dal compilatore in compilazione

ESEMPIO : LIBRERIA VETTORI

Questa libreria contiene funzionalità per fare operazioni fra vettori in \mathbb{R}^2 .

OPERAZIONI:

- somma $\vec{V}_1 + \vec{V}_2$
- sottrazione $\vec{V}_1 - \vec{V}_2$
- prodotto scalare $\vec{V}_1 \cdot \vec{V}_2$
- modulo $|V|$
- angolo (rispetto asse)
- componente x da $|V|$ e α
y da $|V|$ e α
- test ortogonalità
- ...



- leggi-foto/vr \vec{V}
- leggi-file \vec{V}
- stampa-video \vec{V}
- stampa-file \vec{V}

HEADER: vector.h ← estensione
NOTE di BREDA

#include ... di tutte le librerie che servono:

- se la libreria che stiamo creando dovrà fare operazioni trigonometriche etc... <cmath>
 - se stampa/legge video/testiere <iostream>
 - " " " file <fstream>
- ... using namespace std;

#define ... : eventuali costanti utili

ESEMPIO: DHP h

perché lavoriamo con array di 2 elementi

Poi : tutte le DICHIARAZIONI delle FUNZIONI / PROCEDURE
della LIBRERIA

/* res = v1 + v2 */

(void somma (double v1[], double v2[], double res[])

↳ DOCUMENTA la funzione, spiegherà il significato dei
parametri

Una funzione può restituire 1 valore di un certo tipo

⇒ non può restituire una dupla di valori ...

... usiamo parametro array per ESPORTARE IL RISULTATO

void sottr (...)

double prodScalare (double v1[], double v2[]);

↳ il prodotto scalare $\vec{v}_1 \cdot \vec{v}_2$ è un valore

...
...

FILE SORGENTE : vettoriRh.C ← estensione

NOHE LIBRERIA

#include "vettoriRh.h"

Ovvero, copia/incolla tutto il contenuto di vettoriRh.h

⇒ DICHIARAZIONE DI TIPI LIBRERIA

DICHIARAZIONE DI FUNZIONI LIBRERIA

INCLUDE DI ALTRERUE ESTERNE (e.g. cmath)

void somma (double v1[], double v2[], double res[])

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

res[i] = v1[i] + v2[i];

}

...

double prodScalre (double v1[], double v2[]) {

double accu = 0.;

for (int i=0; i<Dim; i++) accu += v1[i] * v2[i];

return accu;

}

double modulo (double v[]) {

return sqrt (prodScalre (v, v));

}

...

FILE OGGETTO : vettoriR2.o ← estensione

name libreria

! Questo file è GENERATO dal COMPILATORE

g++ -c vettoriR2.C

↓
se non ci sono errori sintattici

vettoriR2.o

L'opzione -c dice al compilatore di creare un file contenente la descrizione dei record di attivazione di tutte le funzioni DEFINITE nel file .C che passiamo.

Il prodotto è un file oggetto CHE NON È ESEGIBILE

Uso di librerie

ad es. programme.C

Un programma (nuovo file con un main) che volesse usare la libreria DEVE:

- ① includere il file header delle librerie

#include "vettoriR2.h" ! si trova nella stessa cartella del programma,
altrimenti:

/ path relativo
\ " assoluto

- ② Essere compilato con

g++ -c programme.C



programma.o

programma.o : file
oggetto contenente le
descrizione dei record

di attivazione delle
funzioni/procedure
DEFINITE in programme.C

A questo punto abbiamo:

vettoriRl.o

: contiene tutti i record di attivazione
programma.o di tutte le funzioni

③ Collego tutto per creare un singolo file eseguibile
che conterrà:

- descrizione record main

- descrizione dei record delle funzioni usate

[- informazioni: accensione]

g++ programma.o vettoriRl.o -o programma.exe

elenco file oggetto necessari

nome del file

ESEGIBILE prodotto

della compilazione

[linking dei .o]