

Namespace e librerie di inclusione

Il linguaggio C++ offre la possibilità di dividere lo spazio globale di definizione delle variabili (e delle funzioni) in diverse parti, dette **namespace** (spazio dei nomi), ad ognuna delle quali è associato un nome univoco definito dal programmatore. Questa possibilità ha permesso agli sviluppatori di programmi software di sfruttare l'utilizzo di *risorse globali*, ossia definite globalmente e visibili a tutte le funzioni, senza entrare in conflitto con altre librerie che impiegano lo stesso nome di variabile.

In pratica la *namespace* è un raggruppamento di dati e di funzioni che hanno tra loro un nesso logico per il fatto di riferirsi alla gestione di una determinata entità utile alla programmazione.

La definizione di un *namespace* utilizza una sintassi come nell'esempio seguente:

```
namespace spaziglobale1 {  
    // esempi di variabili e funzioni globali  
    int totale;  
    int Calcolo (int a, int b);  
}
```

Le componenti di un *namespace* possono essere richiamate antepoendo alla componente (variabile o funzione) il nome del *namespace* di appartenenza separato con i doppi due punti (::).

Per esempio:

I caratteri :: si chiamano **scope resolution operator** (operatore di risoluzione della visibilità).

```
cout << spaziglobale1::totale << endl;
```

Per un impiego permanente di tutte le componenti di un *namespace* si deve utilizzare il comando **using namespace**. Per esempio:

```
using namespace spaziglobale1;  
.....  
cout << totale << endl;
```

Quest'ultima istruzione è del tutto equivalente alla precedente, perché la clausola *using* rende visibile la variabile senza l'utilizzo dell'operatore ::.

L'esempio seguente mostra un semplice uso pratico del *namespace*.

ESEMPIO

Definire un *namespace* per gestire un rettangolo e le operazioni di calcolo su di esso.

Il *namespace* definisce la parte dati costituita dalle dimensioni del rettangolo e la parte funzioni per il calcolo dell'area e del perimetro.

```
namespace Rettangolo {  
    // dati  
    int base, altezza;  
    // funzioni  
    int Area (int, int);  
    int Perimetro (int, int);  
}
```

Il *namespace* contiene solo il nome delle funzioni e i parametri necessari.

```

int Rettangolo::Area (int b, int h)
{
    return b * h;
}

int Rettangolo::Perimetro (int b, int h)
{
    return (b + h) * 2;
}

```

Il dettaglio del codice delle funzioni, per maggiore chiarezza di lettura, è scritto dopo la definizione del *namespace*.

L'uso dell'operatore con i doppi due punti, ::, specifica il riferimento delle funzioni al *namespace* di appartenenza.

Questo file, che è un normale file di testo, viene salvato su disco con il nome *rettangolo* nella sottodirectory **include** all'interno della directory di installazione del compilatore C++, in modo che possa essere ritrovato durante la compilazione.

Quando viene richiamato nel programma, il suo nome è scritto tra due parentesi angolari < >. In alternativa può essere registrato nella directory di lavoro dell'utente; in questo secondo caso viene incluso indicandone il nome tra virgolette: `#include "rettangolo"`.

Di fatto esso costituisce a tutti gli effetti una libreria disponibile per il programmatore, che può essere inclusa all'interno del programma di calcolo con la direttiva **#include**, proprio come abbiamo fatto negli esempi precedenti con la libreria *iostream*.

```

#include <iostream>
#include <rettangolo>
using namespace std;

int main()
{
    cout << "Base: ";
    cin >> Rettangolo::base;
    cout << "Altezza: ";
    cin >> Rettangolo::altezza;
    cout << Rettangolo::Area(Rettangolo::base, Rettangolo::altezza) << endl;
    cout << Rettangolo::Perimetro(Rettangolo::base, Rettangolo::altezza) << endl;

    return 0;
}

```

Il programmatore può utilizzare le variabili e le funzioni che si riferiscono al rettangolo, senza dichiararle nel programma. Il riferimento ad esse è specificato dall'operatore *Rettangolo::* che indica il nome del *namespace*.

Volendo evitare di dover scrivere ogni volta questo riferimento, si deve specificare la clausola

```
using namespace Rettangolo;
```

Con questa dichiarazione il programma precedente può essere riscritto nel seguente modo:

```

#include <iostream>
#include <rettangolo>
using namespace std;
using namespace Rettangolo;

int main()
{
    cout << "Base: ";
    cin >> base;
    cout << "Altezza: ";
    cin >> altezza;
    cout << Area(base, altezza) << endl;
    cout << Perimetro(base, altezza) << endl;

    return 0;
}

```

L'esempio precedente, pur essendo molto semplice, fornisce un'idea dell'importanza del *namespace* e delle librerie disponibili per il programmatore, il quale può utilizzare nei suoi programmi dati e funzioni senza conoscerne l'effettiva implementazione con il codice C++. In questo modo si realizza l'**information hiding** (nascondere le informazioni), con il vantaggio di poter modificare in tempi successivi le funzioni (e in generale le librerie) per renderle più efficienti, senza dover cambiare i programmi che le utilizzano.

Per esempio, il programma precedente utilizza le variabili *base*, *altezza* e le funzioni *Area*, *Perimetro*, ma la loro implementazione non è presente nel programma perché è definita nel *namespace Rettangolo*. In pratica le funzioni rappresentano le **interfacce** che il programmatore utilizza per effettuare le elaborazioni sul rettangolo.

Il nome del *namespace* delle principali librerie del linguaggio C++ è **std** (*standard*).