

Metodi statici

Le funzioni membro statiche sono particolari metodi che possono essere invocati senza la necessità di istanziare un oggetto della classe di appartenenza. Per questo motivo, spesso si dice che i metodi statici appartengono alla classe e non alle sue istanze.

Quest'ultima affermazione trova riscontro nell'effetto che il qualificatore `static` ha sul ciclo di vita (*storage duration*) dell'entità cui viene applicata, cioè membri di classe o in generale, variabili locali o globali. Un'entità statica in C++ viene infatti istanziata all'inizio del programma e persiste fino alla sua terminazione. I membri statici di una classe pertanto esistono nella memoria del nostro programma anche se non è stato istanziato nessun oggetto, e salvo casi particolari, la loro locazione di memoria è unicamente definita e condivisa da tutte le istanze della classe. Questa particolarità ha come effetto il fatto che all'interno dei metodi statici si può far riferimento a soli membri statici di classe

Il listato seguente riporta un'implementazione alternativa della classe *SimpleMath* che usa il qualificatore `static`:

```
// simplemath.h
class SimpleMath
{
public:
    static int sum(int a, int b);
};

// simplemath.cpp
int SimpleMath::sum(int a, int b)
{
    return (a + b);
}
// main.cpp
#include <iostream>
#include "simplemath.h"
int main()
{
    std::cout << SimpleMath::sum(2, 3) << std::endl;
    return 0;
}
```

Copy

I dati membro o le funzioni membro statiche di una classe sono accessibili mediante l'uso dell'operatore di scope `::` oppure tramite l'operatore punto o freccia applicato ad un oggetto. Si noti che la il qualificatore `static` appare nella dichiarazione della funzione membro, ma non nella sua definizione. Nulla vieta che i metodi statici siano `inline`, ma vista la loro peculiarità non possono essere `const`.