

Prefazione

Vuoi insegnare? Sii breve.
– Cicerone

C++ sembra un linguaggio nuovo, nel senso che mi permette di esprimere le mie idee in modo più chiaro, più semplice e più diretto oggi di quanto fosse possibile in C++98 o in C++11. Inoltre, i programmi che ne risultano vengono controllati meglio dal compilatore e girano più velocemente.

Questo libro offre una panoramica di C++ come definito da C++20, l'attuale standard ISO C++, e implementato dai principali fornitori di C++. Inoltre, menziona un paio di componenti di libreria attualmente in uso, ma non pianificati per l'inclusione nello standard fino a C++23.

Come altri linguaggi moderni, C++ è esteso, ed esiste un gran numero di librerie necessarie per un suo utilizzo efficace. Questo piccolo libro intende offrire al programmatore esperto un'idea di ciò che costituisce il moderno C++, trattando la maggior parte delle caratteristiche del linguaggio e i principali componenti della libreria standard.

Questo libro può essere letto in uno o due giorni ma, ovviamente, per scrivere bene in C++ serve molto di più di quanto si possa imparare in quel lasso di tempo. Tuttavia, qui l'obiettivo non è fornire una preparazione approfondita, ma piuttosto offrire una panoramica, proporre esempi efficaci e aiutare il programmatore a cominciare.

Si parte dal presupposto che abbiate già programmato in precedenza. Se così non fosse, dovrete leggere un libro di testo, come *Programming: Principles and Practice Using C++ (Second edition)* [Stroustrup, 2014], prima di proseguire. Anche se avete già qualche esperienza di programmazione, il linguaggio che avete utilizzato o le applicazioni che avete scritto potrebbero essere molto differenti dallo stile C++ presentato in questo testo.

Pensate a un giro turistico in una città, come Copenaghen o New York. Nel giro di poche ore vedrete rapidamente le principali attrazioni, vi verranno fornite alcune informazioni contestuali e dati suggerimenti su cosa fare dopo. Dopo un tour di questo tipo *non* conoscerete la città, *non* capirete tutto quello che avete visto e sentito, e alcune storie vi sembreranno strane o addirittura non plausibili. *Non* saprete districarvi tra le regole formali e informali che determinano la vita della città. Per conoscere davvero una città bisogna viverci, spesso per anni. Tuttavia, con un po' di fortuna, riuscirete a ottenere una visione d'insieme, a farvi un'idea di cosa c'è di speciale nella città e di cosa potrebbe interessarvi. Dopo il tour, può iniziare la vera esplorazione.

Questo tour presenta le caratteristiche principali del linguaggio C++ a supporto degli stili di programmazione, come la programmazione a oggetti e quella generica, senza

voler essere un manuale di riferimento dettagliato, che analizza tutte le singole caratteristiche del linguaggio. Nella migliore tradizione dei libri di testo, cerco di spiegare una caratteristica prima di utilizzarla, ma non è sempre possibile e non tutti leggono il testo rigorosamente in sequenza. Parto dal presupposto che i miei lettori abbiano una certa preparazione tecnica. Pertanto, il lettore è incoraggiato a usare i riferimenti incrociati e l'indice analitico.

In modo simile, questo tour presenta le librerie standard in termini di esempi, più che in modo esauriente. Il lettore è incoraggiato a cercare materiale aggiuntivo e di supporto se necessario. C'è molto di più nell'ecosistema C++ oltre alle semplici funzionalità offerte dallo standard ISO (per esempio librerie, sistemi di compilazione, strumenti di analisi e ambienti di sviluppo). C'è un'enorme quantità di materiale (di varia qualità) disponibile sul Web. La maggior parte dei lettori troverà utili tutorial e video introduttivi di conferenze come CppCon e Meeting C++. Per i dettagli tecnici del linguaggio e della libreria offerti dallo standard ISO C++, consiglio [Cppreference]. Per esempio, quando menziono una funzione o una classe della libreria standard, la sua definizione può essere ricercata facilmente e, esaminando la sua documentazione, è possibile trovare molte informazioni correlate.

Questo tour presenta C++ come un'unità integrata anziché come una serie di strati. Di conseguenza, raramente identifico le caratteristiche del linguaggio come presenti in C, C++98 o standard ISO successivi. È possibile trovare tali informazioni nel Capitolo 19, "Storia e compatibilità". Mi concentro sugli aspetti fondamentali e cerco di essere breve, ma non ho resistito completamente alla tentazione di dare particolare rilievo a caratteristiche nuove, come i moduli (§3.2.2), i concetti (§8.2) e le coroutine (§18.6). Avere un occhio di riguardo verso gli sviluppi recenti sembra anche soddisfare la curiosità di molti lettori che già conoscono qualche versione precedente di C++.

Uno standard o un manuale di riferimento sul linguaggio di programmazione indica semplicemente cosa si può fare, ma i programmatori sono spesso più interessati a imparare come usare bene il linguaggio. Questo aspetto è affrontato in parte nella selezione degli argomenti trattati, in parte nel testo, e in particolare nei paragrafi dei consigli. Potete trovare ulteriori consigli su ciò che costituisce un buon C++ moderno nelle C++ Core Guidelines [Stroustrup, 2015]. Le "linee guida di base" possono essere una buona fonte per un ulteriore approfondimento delle idee presentate in questo libro. Potreste notare una notevole somiglianza nella formulazione dei consigli e persino nella numerazione dei consigli tra le Core Guidelines e questo libro. Uno dei motivi è che la prima edizione del libro è stata una delle principali fonti delle linee guida di base iniziali.

Ringraziamenti

Grazie a tutti coloro che hanno contribuito a completare e correggere le precedenti edizioni di questo libro, in particolare agli studenti del mio corso "Design Using C++" alla Columbia University. Grazie a Morgan Stanley per avermi dato il tempo di scrivere questa terza edizione. Grazie a Chuck Allison, Guy Davidson, Stephen Dewhurst, Kate Gregory, Danny Kalev, Gor Nishanov e J.C. van Winkel per aver revisionato il libro e suggerito molti miglioramenti.

Questo libro è stato realizzato dall'autore con il software `troff` con macro provenienti da Brian Kernighan.