

- Carlo A. Mazzone -

PROGRAMMARECON

C e C++

Le chiavi della programmazione



Interfacce grafiche e programmazione guidata da eventi >>

I costrutti fondamentali della programmazione >>

Strutture dati complesse: dalle struct alle classi >>

La programmazione orientata agli oggetti >>

***pro DigitalLifeStyle**

EDIZIONI
FAG
MILANO

*Questo lavoro è dedicato a tutti coloro per i quali la tenacia, la passione e la curiosità
per le cose del mondo rappresentano capisaldi della propria esistenza.*
Carlo A. Mazzone

PROGRAMMARECON

C e C++

Le chiavi della programmazione

Carlo A. Mazzone

EDIZIONI
FAG
MILANO

C e C++ | Le chiavi della programmazione

L'autore: Carlo A. Mazzone

Collana:

PROGRAMMARECON

Publisher: Fabrizio Comolli

Editor: Marco Aleotti

Progetto grafico e impaginazione: Roberta Venturieri

Coordinamento editoriale, pre stampa e stampa: escom - Milano

Immagine di copertina: Gianluca Rotondo

ISBN: 978-88-6604-361-4

Copyright © 2013 **Edizioni FAG Milano**

Via G. Garibaldi 5 - 20090 Assago (MI) - www.fag.it

Finito di stampare in Italia presso Press Grafica - Gravellona Toce - VB nel mese di novembre 2013

Nessuna parte del presente libro può essere riprodotta, memorizzata in un sistema che ne permetta l'elaborazione, né trasmessa in qualsivoglia forma e con qualsivoglia mezzo elettronico o meccanico, né può essere fotocopiata, riprodotta o registrata altrimenti, senza previo consenso scritto dell'editore, tranne nel caso di brevi citazioni contenute in articoli di critica o recensioni.

La presente pubblicazione contiene le opinioni dell'autore e ha lo scopo di fornire informazioni precise e accurate. L'elaborazione dei testi, anche se curata con scrupolosa attenzione, non può comportare specifiche responsabilità in capo all'autore e/o all'editore per eventuali errori o inesattezze.

Nomi e marchi citati nel testo sono generalmente depositati o registrati dalle rispettive aziende. L'autore detiene i diritti per tutte le fotografie, i testi e le illustrazioni che compongono questo libro, salvo quando diversamente indicato.

Sommario

INTRODUZIONE	11
--------------------	----

Parte 1 - Il linguaggio C e i fondamenti della programmazione

1. PREMESSE E MOTIVAZIONI PER UNIRE LA TEORIA E LA PRATICA DELLA PROGRAMMAZIONE.....	21
La battaglia ormai persa dell'analogico	22
In principio era il bit	23
Quanti ne rappresento?	26
Nibble e byte	26
Nuove basi numeriche	26
Rappresentazione dei dati alfanumerici	28
Hardware e software	29
Linguaggi naturali e artificiali	37
I programmi traduttori	37
Interpreti e compilatori	40
A spasso con una "macchina virtuale"	41
Il C e le sue origini	42
Il C come primo linguaggio	44
Ambienti di sviluppo	44
Gli ambienti di sviluppo per il C	45
Creare programmi con interfaccia grafica	50
2. SALVE, MONDO	51
"Salve, mondo" con Bloodshed Dev-C++	54
Salutiamo il mondo da Code::Blocks	62
3. LE VARIABILI E IL MONDO ESTERNO	69
Andiamo a capo.....	72
Input e output di base	73
Tipologie di variabili.....	75
Anche l'occhio vuole la sua parte.....	76
I nomi delle variabili.....	77
Le costanti	77
Le enumerazioni	79

4.	OPERIAMO SUI DATI	81
	Aiuto, quanti problemi!.....	81
	Algoritmo, la parola magica	82
	I diagrammi di flusso	83
	Operatori aritmetici	87
	Printf, la stampa e le specifiche di formattazione.....	89
	Il casting delle variabili.....	91
	Operatori di incremento e decremento	92
	Operatori relazionali.....	94
	Operatori logici	95
5.	IL DEBUGGING DI UN'APPLICAZIONE	99
	Fare debug con Dev-C++.....	100
	Il debug con Code::Blocks	108
6.	ISTRUZIONI CONDIZIONALI: OPERIAMO DELLE SCELTE	111
	Il costrutto if - else.....	111
	Il costrutto else if	114
	Strutture nidificate.....	115
	Il costrutto switch	117
7.	CICLI, OVVERO NOI LA MENTE E LA MACCHINA IL BRACCIO.....	119
	Programmazione strutturata	120
	Il ciclo for	120
	Il ciclo while	125
	Il ciclo do-while.....	128
	Facciamo un break	130
8.	COMINCIAMO A FARE SUL SERIO: GLI ARRAY	133
	Premesse sulla complessità dei dati.....	133
	Strutture dati composte	134
	Gli array	135
	Inizializziamo un array	138
	Array di caratteri, usiamo le stringhe.....	140
	Caratteri "numerici"	141
	Input di caratteri, la funzione getchar	143
	Una pausa alternativa	143
	Do ut des, ovvero getchar e putchar	144
	Il pappagallo rimbambito	144
	Input e output di stringhe: gets e puts.....	145
	Il pappagallo 2, la vendetta.....	146
	Alcune funzioni tipiche per le stringhe.....	147

9.	ARRAY BIDIMENSIONALI.....	151
	Array a due dimensioni.....	151
	La scansione di un array 2D.....	153
	La generazione di numeri casuali.....	156
10.	ORDINIAMO LE COSE, GLI ALGORITMI NOTEVOLI	159
	Il selection sort.....	159
	Bubble sort.....	162
11.	SOLUZIONI SEMPLICI A PROBLEMI COMPLESSI, LE FUNZIONI... ..	165
	Approccio Top-Down alla programmazione	165
	Procedure e funzioni.....	166
	Dichiarazione e definizione di una funzione.....	167
	Nessun input, nessun output	168
	Input si, output no.....	169
	Input e output nella stessa funzione	171
	La visibilità delle variabili.....	173
12.	LA RICORSIONE E LE FUNZIONI RICORSIVE.....	179
	I numeri fattoriali.....	179
	Ancora ricorsione, Fibonacci.....	182
13.	I PUNTATORI	185
	I puntatori.....	185
	Puntatori e array.....	187
	Puntatori e funzioni.....	190
	Array, funzioni e passaggio per indirizzo	192
	Comunichiamo con il programma, il passaggio di parametri al main.....	193
14.	LE STRUTTURE, MOLTI DATI E UN'UNICA VARIABILE.....	195
	Campi, record e strutture.....	195
	Una struttura dentro l'altra	198
	Array di strutture.....	201
	Le strutture e le funzioni.....	203
	Nuovi tipi con typedef.....	206
	I campi di bit.....	208
	Le unioni	209
15.	I FILE E GLI ARCHIVI DI DATI SECONDO IL C.....	213
	Archivi e file.....	213
	Apertura e chiusura di un file.....	214
	Modalità di apertura di un file.....	217
	Scriviamo e leggiamo file di testo: fprintf e fscanf.....	217
	Gestire gli errori.....	220

Un carattere per volta: fgetc e fputc.....	222
Scrivere e leggere file binari: fwrite e fread	223
Gestire gli archivi di dati	226

Parte 2 - Oltre le frontiere del C

16. LA PROGRAMMAZIONE ORIENTATA AGLI OGGETTI E LO SVILUPPO DI SOFTWARE COMPLESSO.....	229
I linguaggi orientati agli oggetti	231
Evoluzione dei linguaggi di programmazione.....	232
Vita, morte e “miracoli” di un software	232
Non solo codice	233
Il versioning	236
Dall’alfa al rilascio: versioni alfa, beta, RC.....	238
17. DAL C AL C++	241
“Salve mondo” con il C++	242
Diamo spazio ai nomi.....	245
Visibilità delle variabili in C++	250
I riferimenti	251
18. INPUT E OUTPUT DI DATI.....	255
Console input	255
Cin e le stringhe: un rapporto difficile.....	257
Qualche dettaglio sull’output	260
19. OGGETTI E CLASSI	265
Dalle struct alle classi	265
Incapsulamento e occultamento dell’informazione	267
Parliamo agli oggetti: messaggi e metodi	269
Oggetti e istanze	271
Come disegnare le classi.....	272
20. L’ARTE DEL CREARE E... DEL DISTRUGGERE.....	275
Il costruttore	275
Il distruttore	277
La compilazione separata.....	278
21. IL POLIMORFISMO DEL C++	285
Monomorfismo e polimorfismo	286
Una firma per la classe	289
Codice e ambiguità	290
Quando il numero conta: array di oggetti	291

Puntare sulla classe	292
Da punto a punto: puntare su sé stessi	293
Overloading anche sugli operatori.....	296
22. EREDITARIETÀ	301
L'ereditarietà del C++	302
Proteggere i membri della propria classe.....	306
Diversi ma non troppo: la sovrapposizione	308
Quando il virtuale è realmente utile	309
Dal virtuale all'astratto.....	312
L'amicizia è una funzione importante	313
23. LA PROGRAMMAZIONE GENERIC E LA LIBRERIA STANDARD DEL C++	317
Riutilizzo del codice e applicazioni legacy	317
Modelli, template e classi generiche.....	318
Specializzare un template	324
La programmazione generica.....	325
STL: Standard Template Library.....	326
24. FLUSSI E FILE NEL C++	327
Le classi per la gestione dei file	327
Apertura e chiusura di un file.....	329
Scriviamo file di testo.....	330
Modalità di apertura.....	331
Leggiamo il contenuto di un file.....	332
File binari e archivi di dati.....	334
Un archivio reale: gestiamo una collezione di fumetti	335
25. LA GESTIONE DEGLI ERRORI E DELLE ECCEZIONI	345
Try e catch.....	345
Gestori multipli	348
Eccezioni standard	349
26. ALLOCAZIONE DINAMICA DELLA MEMORIA	351
Stack e heap: come gestire la memoria.....	353
New e delete: operiamo dinamicamente sulla memoria	354
Il dinamismo di un array	355
I memory leaks.....	356
Verificare la disponibilità di spazio di memoria.....	357
Allochiamo dinamicamente oggetti	358

27. LE INTERFACCE GRAFICHE E LA PROGRAMMAZIONE	
GUIDATA DA EVENTI.....	361
La programmazione guidata da eventi.....	361
Che fatica aprire una finestra.....	363
WinMain, il punto di ingresso dell'applicazione	367
La struttura di una finestra.....	368
Un loop per i messaggi	369
Rispondiamo ai messaggi.....	369
Messaggi complessi.....	370
Le risorse indispensabili per un'applicazione.....	372
INDICE ANALITICO	379

Introduzione

La programmazione è una forma d'arte, così come possono esserlo la pittura, il disegno o la scultura. Realizzare un programma rappresenta, infatti, un processo di vera e propria creazione di qualcosa che prende vita grazie al talento di un essere umano. Nel contesto informatico, il creatore in questione è identificato con differenti nomi: programmatore, sviluppatore o anche developer. Si tratta in ogni caso di uno "strano essere" dotato di capacità tecniche che gli consentono di istruire la macchina a realizzare determinate attività e funzionalità.

Questo testo vuole fornire gli strumenti di base per intraprendere il lungo viaggio che può portare chi è dotato di passione per le macchine a diventare un vero programmatore e quindi a dialogare con esse. Per farlo individua quelle che potrebbero essere definite "le chiavi della programmazione" nell'uso di due linguaggi che rappresentano elementi imprescindibili per un qualsiasi programmatore: C e C++.

La prima parte del testo è dedicata appunto al C e consente, anche a chi si avvicina per la prima volta alla programmazione, di realizzare programmi di media difficoltà. In questa prima parte vengono infatti forniti non solo i rudimenti della sintassi del linguaggio ma anche, passo dopo passo, le indicazioni sui costrutti fondamentali della programmazione e i concetti di base relativi alle strutture dati, sia di tipo semplice sia complesso.

La seconda parte del libro inizia, anche metaforicamente, nel punto in cui si ferma il linguaggio C in termini di potenza e di strumenti dati al programmatore.

Viene presentato il linguaggio C++ come logica e naturale conseguenza dei limiti del C, nella direzione di rendere possibile la creazione di programmi sempre più complessi e più facilmente manutenibili. In questa ottica si espongono i concetti che sono alla base della programmazione orientata agli oggetti e si illustra il loro uso nel contesto del C++.

A tutti voi che mi seguirete nelle pagine successive: buon viaggio!

Struttura e contenuti del libro

Il testo è organizzato in due principali sezioni. La prima parte è introduttiva alla programmazione in linguaggio C, presentando i concetti di base dello sviluppo software: le strutture dati e gli algoritmi che operano su di esse. Vengono così forniti i concetti relativi a variabili e costanti e ai principali costrutti di programmazione (sequenza, selezione, ripetizione) con abbondanza di esempi stimolanti anche per il lettore già avvezzo a tali argomenti. La prima sezione si chiude con la presentazione della struttura dati `struct` del C quale esempio di modalità complessa di gestione di dati ponendo le basi per le successive discussioni relative alla creazione di nuovi tipi di dati e operatori.

Il Capitolo 1 vuole fornire una base sostanziale rispetto ai concetti generali dell'informatica con speciale attenzione relativa a tutto ciò che riguarda lo sviluppo software. Viene innanzitutto presentata la dicotomia analogico-digitale e quindi, rispetto al contesto digitale, si fornisce una panoramica essenziale, ma corposa, relativa alla logica binaria e alle tecniche di rappresentazione dei dati (ASCII e Unicode).

Vengono introdotti i concetti e le descrizioni di base riguardanti i principali dispositivi hardware e una panoramica del mondo del software sia in merito a una semplice classificazione tipologica sia rispetto a ciò che attiene alla commerciabilità del software stesso.

Si introduce dunque il lettore al mondo dei linguaggi con indicazioni relative alla traduzione del codice: interpretazione, compilazioni e linguaggi bytecode.

Il primo capitolo termina quindi, motivando la scelta del linguaggio C come elemento cardine per tutte le discussioni di tipo operativo che proseguiranno nel resto del libro, con una carrellata in merito ai principali ambienti di sviluppo utilizzabili per produrre codice con tale linguaggio.

Il Capitolo 2 introduce alla compilazione, passo dopo passo, del primo semplice programma C proponendo l'ambiente Bloodshed Dev-C++ come strumento principale da utilizzarsi per implementare tutti gli spunti di esempio e di esercizio proposti nella prima parte del testo. In ogni caso si presenta, per coerenza di contenuti e di discussione, anche l'ambiente Code::Blocks fornendo anche per questo secondo sistema le istruzioni di base per compilare un primo semplice programma C. Seppure il consiglio sia quello di utilizzare tale secondo ambiente di sviluppo per i contenuti della seconda parte del testo, relativi al linguaggio C++, nulla osta a sceglierlo come principale strumento sin dall'inizio. Tale scelta può essere obbligata nel caso si utilizzi una macchina del mondo Mac in quanto Dev-C++ è specifico per l'ambiente Windows. Il consiglio di utilizzare come primo ambiente proprio il Dev-C++ è motivato dalla maggiore semplicità d'uso dell'ambiente stesso, che consente di meglio concentrarsi sugli aspetti del

linguaggio. In ogni caso il lettore più esperto potrà scegliere lo strumento di sviluppo che preferisce sin dall'inizio.

Nel Capitolo 3 vengono introdotti i fondamentali concetti di variabile, costante ed enumerazione presentandone le varie tipologie unitamente ai consigli per una scelta corretta per i loro nomi e modalità d'uso. Per farlo si introduce da subito il lettore all'input e output di base del linguaggio C. Iniziano già in tale capitolo una serie di suggerimenti di carattere generale che riguardano le buone pratiche di programmazione che, nel contesto specifico, sono relative all'usabilità dell'interfaccia utente.

Il Capitolo 4 presenta il concetto cardine di algoritmo e fornisce spunti di esercizio in merito a semplici problemi risolvibili con l'ausilio dei diagrammi di flusso. La necessità di realizzare delle operazioni sui dati è lo spunto per introdurre i principali operatori aritmetici e relazionali e la presentazione delle problematiche relative alla trasformazione di un dato da un certo tipo a un altro.

Il capitolo si chiude con l'introduzione degli operatori logici e quindi con una panoramica sull'algebra di Boole.

Il Capitolo 5 presenta i meccanismi relativi al debugging (verifica degli errori) delle applicazioni software sia in relazione all'ambiente Dev-C++ sia per l'ambiente Code::Blocks. Vengono così proposti i concetti di punto di interruzione (breakpoint) e sono poste sotto osservazione specifiche variabili utilizzando esempi concreti.

Nel Capitolo 6 vengono fornite indicazioni dettagliate, corredate da esempi d'uso, relative alle istruzioni condizionali per consentire la realizzazione di programmi con un livello di complessità maggiore rispetto agli esempi iniziali.

Il Capitolo 7 completa la presentazione degli strumenti fondamentali di programmazione introducendo il costrutto iterazione. Vengono messi a confronto i cicli for e while cercando di evidenziare al meglio il loro campo di applicazione. Si discute quindi di programmazione di tipo strutturato come superamento delle problematiche legate alla cosiddetta spaghetti programming.

Nei Capitoli 8 e 9 vengono introdotte le strutture dati di tipo complesso. Vengono così presentati gli array monodimensionali con una dettagliata panoramica sulle problematiche connesse al trattamento delle stringhe. Successivamente è il turno degli array bidimensionali, subito dopo i quali si coglie l'occasione per illustrare la possibilità di generare numeri casuali. In tal modo si rendono disponibili gli strumenti che consen-

tono la realizzazione di programmi autonomi rispetto all'input del programmatore o dell'utente.

Il Capitolo 10 presenta il concetto di algoritmo notevole come esempio di risoluzione di problemi mediante metodologie consolidate nella letteratura e dalla pratica informatica. Vengono esaminati in dettaglio gli algoritmi di ordinamento selection sort e bubble sort con accenni a concetti di complessità computazionale.

Il Capitolo 11 prende in esame l'approccio alla programmazione software di tipo Top-Down introducendo concetti ed esempi reali relativi a procedure e funzioni. È il contesto giusto in cui possono essere definiti, quindi, diversi dettagli relativi ai concetti di visibilità e ciclo di vita delle variabili presentando inoltre le classi di memorizzazione esterne e statiche.

Nel Capitolo 12, ormai noto il concetto di funzione, se ne presenta una tipologia molto particolare: la funzione ricorsiva. Vengono così proposti concetti ed esempi relativi al calcolo del fattoriale e alle sequenze di Fibonacci.

Il Capitolo 13 presenta i puntatori, argomento per sua natura ostico. Data la particolarità del tema si cerca di motivarne il più possibile l'utilizzo analizzando prima i collegamenti tra puntatori e array, e facendo successivamente vedere quale sia il collegamento tra puntatori e funzioni, esaminando le chiamate per valore e per indirizzo delle funzioni stesse.

Il Capitolo 14 descrive la struttura dati struct del C motivandone l'uso come contenitore fondamentale per raggruppare dati di tipo complesso tra di essi omogenei. Tale capitolo rappresenta un punto di snodo: si iniziano a delineare i limiti dello sviluppo del software con il linguaggio C e l'approccio di tipo procedurale. Vengono così enfatizzati gli aspetti relativi al trattamento dei dati introducendo la parola chiave typedef per la definizioni di nuovi tipi e le strutture dati union e campi di bit.

Con il Capitolo 15 si chiude la prima sezione del libro focalizzata sugli elementi del C, facendo vedere come vengano gestiti i file utilizzando l'approccio proprio di tale linguaggio di programmazione.

Il Capitolo 16 apre dunque la seconda sezione del testo presentando sin da subito le prime discussioni critiche sull'approccio che utilizza il paradigma a oggetti nella programmazione software. Preparando il lettore per lo sviluppo di applicativi di sem-

pre maggiore complessità si definiscono le problematiche proprie della corretta progettazione del software come momento fondamentale e preliminare alla scrittura del codice. Si introducono quindi i concetti di ciclo di vita nello sviluppo software e di conseguenza vengono presentati gli ambiti propri dell'ingegneria del software. Il capitolo si chiude con una panoramica sulle modalità di versioning.

I Capitoli 17 e 18 introducono nel dettaglio la sintassi e l'approccio alla programmazione del linguaggio C++. In tale ambito si suggerisce l'utilizzo dell'ambiente di sviluppo Code::Blocks fornendone le specifiche d'uso di massima. Vengono da subito introdotti i concetti di spazio dei nomi delle variabili e si vede come queste possano variare la loro visibilità nei differenti blocchi di codice. Il Capitolo 18 è specifico e dettagliato rispetto alle funzionalità di input e output proprie del C++.

Il Capitolo 19 presenta finalmente i concetti di classe e oggetto. È un momento fondamentale nel quale si mostra come, a partire da una struct del C, si possa costruire un oggetto di programmazione inserendo al suo interno la possibilità di realizzare specifiche azioni con l'utilizzo di apposite funzioni. Si discute quindi di incapsulamento e occultamento dell'informazione e di come si possa di conseguenza comunicare con gli oggetti.

Il Capitolo 20 fornisce la sintassi per la gestione del ciclo di vita degli oggetti relativamente alla loro creazione e successiva distruzione. Il capitolo presenta inoltre le modalità per la cosiddetta compilazione separata ovvero la possibilità di organizzare il proprio codice in differenti file, operazione indispensabile al crescere delle dimensioni dei propri programmi.

I Capitoli 21 e 22 completano la descrizione delle proprietà fondanti della programmazione orientata agli oggetti descrivendo polimorfismo ed ereditarietà. Infatti, il Capitolo 21 continua l'analisi delle caratteristiche degli oggetti fornendo dettagli sul polimorfismo, contrapponendolo al comportamento monomorfo delle funzioni C, relativamente alla tecnica dell'overloading. Il Capitolo 22, d'altra parte, introduce il concetto di ereditarietà e quindi la possibilità di riutilizzare al meglio il codice già scritto per ampliarlo adattandolo a nuove necessità. È in questo contesto che si presentano le classi virtuali e astratte.

Il Capitolo 23 presenta la programmazione generica e quindi l'uso dei template. Si illustra così come sia possibile realizzare un approccio alternativo al polimorfismo classico, usando funzioni e classi parametriche, nell'ottica di produrre codice sorgente il più efficiente e generico possibile in merito alle aspettative del programmatore.

Nel Capitolo 24 si riprende la discussione già affrontata nel Capitolo 15 in relazione alla gestione dei file e degli archivi. In questo contesto si affronta però l'approccio usato dal linguaggio C++ mostrando una diversa metodologia per un comune problema e dando al contempo la possibilità di esplorare dettagli ulteriori della programmazione C++ e della libreria standard. Ancora una volta vengono fornite indicazioni dettagliate per la realizzazione di una piccola ma reale applicazione di gestione di un archivio di dati.

Il Capitolo 25 affronta le problematiche connesse alla gestione degli errori e propone gli strumenti messi a disposizione dal linguaggio C++ per prevenire e gestire al meglio le condizioni anomale nelle quali possono venire a trovarsi le nostre applicazioni. Viene così proposto il concetto relativo alle eccezioni e il costrutto try - catch per la loro corretta gestione.

Il Capitolo 26 torna sulle questioni attinenti alla gestione della memoria focalizzando l'attenzione sulle corrette procedure da utilizzarsi per superare i vincoli imposti dalle dimensioni fisse richieste nella dichiarazione di variabili statiche automatiche. Si discute quindi di allocazione dinamica della memoria confrontando stack e heap e dando adeguati suggerimenti per evitare situazioni potenzialmente disastrose come i memory leaks.

Il Capitolo 27 chiude il viaggio attraverso la programmazione e nel C/C++ fornendo spunti relativamente alla realizzazione di applicazioni con interfaccia grafica. Alle tante possibili declinazioni relative alla creazione di applicazioni di tipo grafico si preferisce l'approccio basato sulle funzioni native di Windows. Tale scelta è sufficientemente generica da fornire la giusta panoramica per comprendere i fondamenti dello sviluppo di applicazioni grafiche utilizzando la metodologia nota come programmazione guidata da eventi.

Materiale di supporto ed esempi

Il testo mantiene in tutta la sua organizzazione un approccio pratico e a tale scopo propone numerosi esempi di codice reale. Tale materiale è disponibile in formato digitale all'indirizzo:

www.tesseract.it/lechiavi

Allo stesso indirizzo è possibile riferirsi per approfondimenti e novità relativi al libro e agli argomenti in esso trattati.

Ringraziamenti

È abitudine consolidata quella di inserire, all'inizio di ogni testo, dei ringraziamenti da parte dell'autore nei confronti di una serie di persone che hanno, in un modo o nell'altro, contribuito alla realizzazione del testo medesimo. Trovo tale consuetudine doverosa.

Ognuno di noi cresce, vive e si realizza attraverso le proprie azioni in conseguenza degli stimoli e del sostegno che riceve dalle persone che lo circondano. Cosicché ogni nostra realizzazione è in un certo modo un prodotto collettivo. Non sfugge a tale contesto il presente volume.

Questo lavoro non avrebbe infatti visto la luce senza il sostegno che ho ricevuto negli anni in primo luogo dalla mia famiglia. Essa è stata da sempre fucina di stimoli e propulsione per una crescita basata sulla critica costruttiva e l'interesse sempre meravigliato verso tutte le cose del mondo. Ringrazio così in particolare i miei genitori e i miei fratelli. Nello specifico un forte tributo alla mia dolce mamma Rosa per l'affetto incondizionato, per le gite in libreria quando ero bambino e, tra le tantissime altre cose, un grazie particolare per avermi comprato il mio primo vero computer, il mitico Sinclair QL.

Uno speciale abbraccio è poi dovuto a mio fratello Ermanno per il continuo appoggio e sostegno in tutto ciò che faccio.

A mia moglie va il riconoscimento per la sopportazione relativa al fatto di aver sposato un uomo con DNA digitale e la mia eterna gratitudine per la sua capacità di riuscire a completarmi. Ai miei figli, unitamente al mio non misurabile affetto, vanno le scuse per quelle volte in cui, per il troppo lavoro, li ho costretti alla mia assenza.

Questo libro deve molto ai miei amici che hanno fatto da cavie nella lettura e nello studio delle bozze. Tra i tanti voglio ringraziare gli ingegneri Francesco Ambrosio, Stefano Ciaramella, Amedeo Lepore, Giuseppe Pica e in particolare l'ingegnere Luigi Della Gala. Un ringraziamento speciale va poi al mio fraterno amico Roberto Frattin, informatico brillante del quale ho sempre apprezzato la capacità di schematizzare in forma grafica il mondo della matematica.

È spontaneo e sincero il riconoscimento che tributo agli ingegneri Fabrizio D'Aloia e Dario Pica che mi hanno onorato della loro fiducia relativamente alle mie competenze informatiche in questi anni di continue evoluzioni tecnologiche.

Un ringraziamento specifico in merito al presente volume è poi per tutti i miei innumerevoli alunni che hanno fatto sì che potessi migliorarmi nelle capacità comunicative e non perdessi mai la voglia di continuare a studiare e stare dalla parte di chi impara. Tra i tantissimi ai quali va la mia riconoscenza, voglio ricordarne alcuni in particolare: Antonio Esposito, Andrea Micco e Pietro Goglia.

Tuttavia, questo libro non avrebbe visto la luce in questa forma senza l'appoggio e la condivisione di intenti di Fabrizio Comolli e Marco Aleotti; spero di poter lavorare ancora con voi.

E infine, a tutti coloro i quali, anche qui non menzionati, che in un modo o nell'altro hanno incrociato positivamente la mia esistenza: un grazie dal profondo del cuore.

Carlo A. Mazzone