

Prefazione



Questo è un testo introduttivo di informatica che focalizza l'attenzione sui principi della programmazione e sull'ingegneria del software, seguendo alcune linee guida.

- **Insegnare gli oggetti con gradualità.**
Nel Capitolo 2 gli studenti imparano a usare oggetti e classi della libreria standard, mentre nel Capitolo 3 apprendono le strategie per la realizzazione di classi *a partire da specifiche assegnate*, usando poi semplici oggetti per approfondire la conoscenza di diramazioni, cicli e array. La vera e propria progettazione orientata agli oggetti inizia nel Capitolo 8. Un approccio così graduale consente agli studenti di utilizzare oggetti anche mentre studiano la parte algoritmica di base, senza ricorrere all'insegnamento di cattive abitudini che dovrebbero essere abbandonate in seguito.
- **Insistere su pratiche ingegneristiche adeguate.**
Concentrando l'attenzione sullo sviluppo orientato al collaudo, si incoraggiano gli studenti a collaudare i propri programmi in modo sistematico; analogamente, la presentazione degli errori più frequenti e di molti consigli utili per migliorare la qualità del software agevola lo sviluppo di buone abitudini di programmazione.
- **Assistere gli studenti con guide mirate e esempi completi.**
Spesso i programmatori principianti chiedono: “Come posso cominciare? E ora cosa devo fare?” Un'attività complessa come la programmazione non può, ovviamente, ridursi a un ricettario di istruzioni, però guide mirate e dettagliate possono essere estremamente utili per acquisire confidenza con la materia, costituendo uno schema da tenere sotto mano durante la soluzione dei problemi assegnati. Il libro contiene

un'ampia rassegna di tali guide, denominate “Consigli pratici” e focalizzate su problemi comuni, seguite dalla presentazione di ulteriori “Esempi completi”.

- **Concentrarsi sugli elementi essenziali, pur rimanendo tecnicamente corretti.**

Una trattazione enciclopedica non è di alcun aiuto per un programmatore principiante, ma non lo è nemmeno un approccio opposto, che riduca il materiale a un elenco di punti semplificati, che danno soltanto l'illusione di aver compreso. In questo libro, gli aspetti essenziali di ciascun argomento sono presentati con trattazioni di dimensioni digeribili, aggiungendo note che consentono di approfondire le buone pratiche di programmazione o le caratteristiche del linguaggio, una volta che il lettore sia pronto per tali informazioni aggiuntive.

- **Usare il linguaggio Java standard.**

Il libro insegna il linguaggio Java standard, non un ambiente speciale, dedicato all'apprendimento. Il livello di approfondimento, nella presentazione del linguaggio, della libreria e degli strumenti, è tale da consentire la soluzione di problemi di programmazione reali.

- **Presentare la programmazione grafica in un percorso opzionale.**

Le forme grafiche sono splendidi esempi di oggetti e molti studenti amano scrivere programmi grafici o, comunque, con interfaccia utente grafica. Se lo si desidera, questi argomenti possono essere integrati nella trattazione, utilizzando il materiale che si trova alla fine dei Capitoli 2, 3, 9 e 10.

Le novità di questa edizione

Giunto alla sesta edizione (e quinta edizione italiana), il libro è stato nuovamente aggiornato e rivisto con cura, migliorando alcune caratteristiche e introducendone di nuove.

Maggiore aiuto ai programmatori principianti

- I “Consigli pratici” sono stati aggiornati ed espansi, aggiungendone quattro. Quindi, nuovi “Esempi completi” guidano gli studenti passo dopo passo alla soluzione di problemi complessi e interessanti.
- È stata migliorata la trattazione della progettazione di algoritmi, con l'uso di pseudocodice fin dal Capitolo 1.
- Sono stati rivisti tutti i capitoli, in modo da focalizzare ciascun paragrafo su uno specifico obiettivo di apprendimento, secondo la filosofia dei *learning object*, che si riflette anche in una nuova organizzazione del riassunto, presente al termine di ogni capitolo per aiutare gli studenti a valutare i propri progressi.

Discussione di esempi

- I diagrammi sintattici sono stati rivisitati graficamente, introducendo esempi tipici che attirino l'attenzione degli studenti verso gli elementi chiave della sintassi. Ulteriori annotazioni evidenziano casi speciali, errori comuni e buone pratiche relative a quella particolare sintassi.

- Sono state introdotte nuove tabelle di esempi per presentare in modo chiaro e compatto una varietà di casi tipici e speciali. Ogni esempio è accompagnato da una breve nota esplicativa sul suo utilizzo e sui valori che ne risultano.
- L'introduzione degli oggetti è stata resa ancor più graduale, con esempi aggiuntivi e approfondimenti fin dai primi capitoli.

Aggiornamento a Java 7

- Le caratteristiche introdotte nella versione 7 di Java sono trattate in alcuni “Argomenti avanzati”, in modo che gli studenti le possano approfondire. In questa edizione, usiamo ancora Java 5 e 6 per la trattazione principale.

Più opportunità per esercitarsi

- Il vasto insieme di esercizi, graduati per tipologia e difficoltà, è stato ulteriormente espanso e migliorato. Gli esercizi relativi alla grafica e al collaudo (*testing*) sono poi ulteriormente evidenziati mediante, rispettivamente, la lettera G o T. Al termine di ogni capitolo sono presenti esercizi di ripasso, mentre i più completi esercizi e progetti di programmazione si trovano nel sito web dedicato al libro.

Una panoramica del libro

Il libro può essere suddiviso in tre parti in modo molto naturale, come si può vedere nella Figura 1, dove vengono evidenziate graficamente anche le propedeuticità fra i capitoli, che consentono la medesima flessibilità che era caratteristica dell'edizione precedente.

Parte A: I fondamenti (Capitoli da 1 a 7)

Il Capitolo 1 contiene una breve introduzione relativa all'informatica e alla programmazione Java, il Capitolo 2 mostra come manipolare oggetti di classi predefinite e il Capitolo 3 insegna a costruire semplici classi a partire dalle loro specifiche.

I capitoli che vanno dal 4 al 7 trattano i tipi di dati fondamentali, le strutture sintattiche per il controllo di flusso (diramazioni e cicli) e gli array.

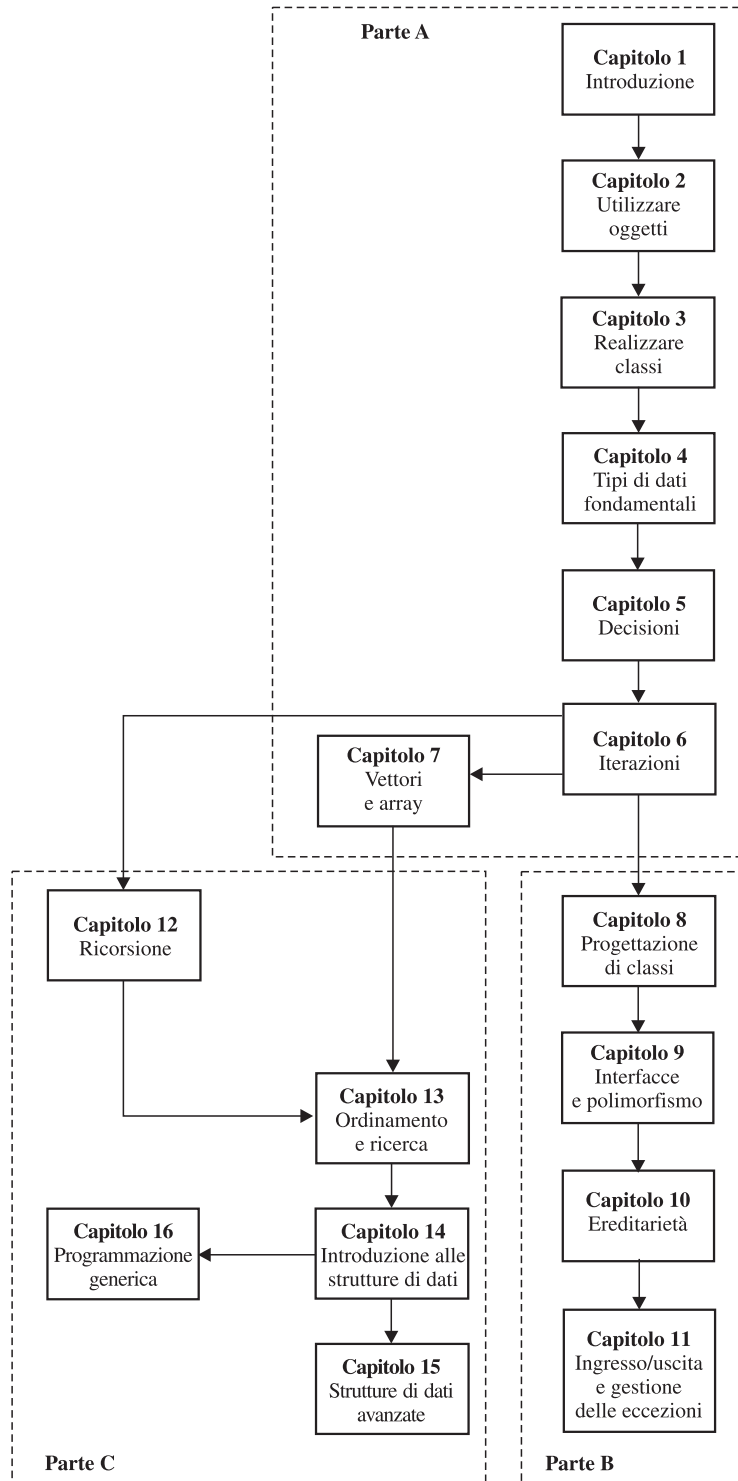
Parte B: Progettazione orientata agli oggetti (Capitoli da 8 a 11)

Il Capitolo 8 si occupa della progettazione di classi in modo più sistematico e introduce un sottoinsieme semplificato della notazione UML.

La discussione sul polimorfismo e l'ereditarietà è suddivisa in due capitoli: il Capitolo 9 tratta le interfacce e il polimorfismo, mentre il Capitolo 10 si occupa dell'ereditarietà vera e propria. L'introduzione delle interfacce prima dell'ereditarietà è utile per un motivo fondamentale: gli studenti vedono immediatamente il polimorfismo prima di dover affrontare dettagli tecnici complessi, come la costruzione della superclasse.

La gestione delle eccezioni (la cui gerarchia costituisce un ottimo esempio di ereditarietà) e gli strumenti basilari per la gestione dei file di testo sono argomento del Capitolo 11.

Figura 1
Dipendenze fra i capitoli



Parte C: Algoritmi e strutture di dati (Capitoli da 12 a 16)

I Capitoli 12, 13 e 14 contengono un'introduzione agli algoritmi e alle strutture di dati, con la trattazione di ricorsione, ordinamento e ricerca, liste concatenate, pile e code. Come si può vedere nella Figura 1, tali argomenti possono essere trattati, in alternativa, dopo il Capitolo 7.

La ricorsione viene presentata da un punto di vista orientato agli oggetti: un oggetto che risolve un problema ricorsivamente costruisce un altro oggetto della stessa classe che risolve un problema più semplice. Fare in modo che questo nuovo oggetto svolga il compito più semplice è molto più plausibile per gli studenti rispetto a una funzione che invoca se stessa.

Il Capitolo 13 si occupa degli algoritmi di ordinamento fondamentali e introduce l'analisi delle prestazioni mediante la notazione O-grande. Il Capitolo 14 tratta liste concatenate, pile e code, sia come tipi di dati astratti sia come realizzazioni concrete presenti nella libreria standard di Java. Il Capitolo 15 presenta alcune strutture di dati avanzate: tabelle hash, alberi di ricerca binari e heap.

Infine, il Capitolo 16, disponibile sul sito web dedicato al libro, presenta le caratteristiche del linguaggio Java che consentono la programmazione generica: si tratta di un capitolo adatto a studenti che vogliono realizzare proprie classi generiche o propri metodi generici.

Appendici

L'Appendice A contiene le risposte alle domande di auto-valutazione presenti in tutti i capitoli, mentre le appendici seguenti contengono materiale di riferimento sugli operatori e sulle parole riservate del linguaggio Java, per proseguire con la rappresentazione binaria dei numeri e le operazioni sui bit. Altre appendici, disponibili nel sito web dedicato al libro, riportano un utile compendio sintattico del linguaggio Java e linee guida per la scrittura del codice sorgente.

Una panoramica degli ausili didattici

Questa nuova edizione è basata sugli elementi pedagogici già utilizzati nell'edizione precedente, con l'aggiunta di ulteriori ausili per il lettore, rendendo il testo adeguato sia ai principianti sia a chi voglia apprendere Java come secondo linguaggio di programmazione.

L'inizio di ciascun capitolo offre la consueta panoramica sugli obiettivi del capitolo stesso e le motivazioni introduttive.

Note a margine evidenziano i punti in cui vengono introdotti nuovi concetti: costituiscono uno schema delle idee chiave e sono poi riassunte alla fine di ogni capitolo, suddivise per obiettivi di apprendimento, e possono essere utilizzate per un rapido ripasso.

I riquadri sintattici forniscono una rapida panoramica dei costrutti del linguaggio: già presenti nelle precedenti edizioni, sono stati arricchiti da annotazioni che evidenziano gli errori più frequenti e le migliori consuetudini.

Ogni paragrafo si conclude con esercizi di auto-valutazione, che lo studente può usare per verificare di aver compreso i nuovi argomenti. Le relative risposte si trovano nell'Appendice A.

Molte (nuove) tabelle agevolano il compito dei principianti, fornendo esempi concreti, per un veloce ripasso delle caratteristiche salienti del linguaggio e dei relativi errori frequenti. Un aiuto analogo viene da figure che illustrano, in fasi successive, il flusso di esecuzione di parti salienti di un programma, affiancandosi al codice sorgente di molti casi concreti.

Esistono, poi, sette diverse tipologie di note inserite nel testo, evidenziate in modo particolare per non interrompere il flusso del materiale principale. Alcune sono piuttosto brevi, altre occupano più di una pagina: ad ogni argomento è stato assegnato lo spazio necessario per una piena e convincente spiegazione, piuttosto che tentare di farle rientrare in “consigli” di un unico paragrafo.

- I **Consigli pratici**, ispirati dalle guide “HOW TO” di Linux, hanno lo scopo di rispondere a domande frequenti degli studenti, del tipo “Cosa devo fare ora?”, fornendo istruzioni sull’esecuzione dei compiti più comuni passo dopo passo, concentrandosi in particolar modo sulla pianificazione dell’attività di programmazione e sul relativo collaudo.
- Gli **Esempi completi**, novità di questa edizione, applicano i Consigli pratici a un esempio diverso, illustrando come quei consigli siano effettivamente assai utili per pianificare, realizzare e collaudare la soluzione di un altro problema di programmazione.
- Gli **Errori comuni** descrivono i tipi di errori compiuti spesso dagli studenti, con una spiegazione del motivo dell’errore e di come rimediarvi.
- I **Consigli per la qualità** illustrano buone abitudini di programmazione. Dato che molti di essi richiedono uno sforzo iniziale per l’apprendimento, queste note danno attente motivazioni per i consigli stessi e spiegano come lo sforzo verrà ripagato in seguito.
- I **Consigli per la produttività** insegnano agli studenti come usare in modo più efficace gli strumenti e il tempo che hanno a disposizione, spingendoli a essere più produttivi.
- Gli **Argomenti avanzati** trattano materiale non essenziale o più complesso, oltre alle novità relative a Java 7.
- Le **Note di cronaca** forniscono informazioni storiche e sociali sull’informatica.

Sito web dedicato al libro

All’indirizzo <http://www.apogeeonline.com/libri/9788850329564/scheda> si trova il sito web dedicato a questo libro, dove è possibile scaricare il codice sorgente degli esempi utilizzati nel testo, insieme ad altre risorse per studenti e docenti, tra le quali: esercizi e progetti di programmazione relativi a ciascun capitolo del libro, un capitolo dedicato alla programmazione generica e appendici che riportano un utile compendio sintattico del linguaggio Java e linee guida per la scrittura del codice sorgente.

Ringraziamenti

I miei ringraziamenti vanno a Beth Golub, Lauren Sapira, Andre Legaspi, Don Fowley, Mike Berlin, Janet Foxman, Lisa Gee e Bud Peters di John Wiley & Sons, oltre a Vickie Piercey del gruppo Publishing Services, per l'aiuto che hanno dato a questo progetto. Un ringraziamento speciale e profondo va a Cindy Johnson per l'incredibile cura con cui ha lavorato e per i suoi preziosi consigli.

Sono grato a Suzanne Dietrich, Rick Giles, Kathy Liszka, Stephanie Smullen, Julius Dichter, Patricia McDermott-Wells e David Woolbright per la realizzazione del materiale di supporto.

Devo, poi, ringraziare le numerosissime persone che hanno rivisto il manoscritto di questa edizione, dando utili consigli e portando alla mia attenzione un numero imbarazzante di errori e omissioni. Tra tutti, citerò:

Ian Barland, *Radford University*
Rick Birney, *Arizona State University*
Paul Bladek, *Edmonds Community College*
Robert P. Burton, *Brigham Young University*
Teresa Cole, *Boise State University*
Geoffrey Decker, *Northern Illinois University*
Eman El-Sheikh, *University of West Florida*
David Freer, *Miami Dade College*
Ahmad Ghafarian, *North Georgia College & State University*
Norman Jacobson, *University of California, Irvine*
Mugdha Khaladkar, *New Jersey Institute of Technology*
Hong Lin, *University of Houston, Downtown*
Jeanna Matthews, *Clarkson University*
Sandeep R. Mitra, *State University of New York, Brockport*
Parviz Partow-Navid, *California State University, Los Angeles*
Jim Perry, *Ulster County Community College*
Kai Qian, *Southern Polytechnic State University*
Cyndi Rader, *Colorado School of Mines*
Chaman Lal Sabharwal, *Missouri University of Science and Technology*
John Santore, *Bridgewater State College*
Stephanie Smullen, *University of Tennessee, Chattanooga*
Monica Sweat, *Georgia Institute of Technology*
Shannon Tauro, *University of California, Irvine*
Russell Tessier, *University of Massachusetts, Amherst*
Jonathan L. Tolstedt, *North Dakota State University*
David Vineyard, *Kettering University*
Lea Wittie, *Bucknell University*

Ogni nuova edizione si basa sui suggerimenti e sulle esperienze dei revisori e degli utenti delle edizioni precedenti. Sono, quindi, grato a tutti quanti per il loro prezioso contributo:

Tim Andersen, *Boise State University*
 Ivan Bajic, *San Diego State University*
 Ted Bangay, *Sheridan Institute of Technology*
 George Basham, *Franklin University*
 Sambit Bhattacharya, *Fayetteville State University*
 Joseph Bowbeer, *Vizrea Corporation*
 Timothy A. Budd, *Oregon State University*
 Frank Butt, *IBM*
 Jerry Cain, *Stanford University*
 Adam Cannon, *Columbia University*
 Nancy Chase, *Gonzaga University*
 Archana Chidanandan, *Rose-Hulman Institute of Technology*
 Vincent Cicirello, *The Richard Stockton College of New Jersey*
 Deborah Coleman, *Rochester Institute of Technology*
 Valentino Crespi, *California State University, Los Angeles*
 Jim Cross, *Auburn University*
 Russell Deaton, *University of Arkansas*
 H. E. Dunsmore, *Purdue University*
 Robert Duvall, *Duke University*
 Henry A. Etlinger, *Rochester Institute of Technology*
 John Fendrich, *Bradley University*
 John Fulton, *Franklin University*
 David Geary, *Sabware, Inc.*
 Margaret Geroch, *Wheeling Jesuit University*
 Rick Giles, *Acadia University*
 Stacey Grasso, *College of San Mateo*
 Jianchao Han, *California State University, Dominguez Hills*
 Lisa Hansen, *Western New England College*
 Elliotte Harold
 Eileen Head, *Binghamton University*
 Cecily Heiner, *University of Utah*
 Brian Howard, *Depauw University*
 Lubomir Ivanov, *Iona College*
 Curt Jones, *Bloomsburg University*
 Aaron Keen, *California Polytechnic State University, San Luis Obispo*
 Elliot Koffman, *Temple University*

Kathy Liszka, *University of Akron*
 Hunter Lloyd, *Montana State University*
 Youmin Lu, *Bloomsburg University*
 John S. Mallozzi, *Iona College*
 John Martin, *North Dakota State University*
 Scott McElfresh, *Carnegie Mellon University*
 Joan McGrory, *Christian Brothers University*
 Carolyn Miller, *North Carolina State University*
 Teng Moh, *San Jose State University*
 John Moore, *The Citadel*
 Faye Navabi, *Arizona State University*
 Kevin O’Gorman, *California Polytechnic State University, San Luis Obispo*
 Michael Olan, *Richard Stockton College*
 Kevin Parker, *Idaho State University*
 Cornel Pokorny, *California Polytechnic State University, San Luis Obispo*
 Roger Priebe, *University of Texas, Austin*
 C. Robert Putnam, *California State University, Northridge*
 Neil Rankin, *Worcester Polytechnic Institute*
 Brad Rippe, *Fullerton College*
 Pedro I. Rivera Vega, *University of Puerto Rico, Mayaguez*
 Daniel Rogers, *SUNY Brockport*
 Carolyn Schauble, *Colorado State University*
 Christian Shin, *SUNY Geneseo*
 Jeffrey Six, *University of Delaware*
 Don Slater, *Carnegie Mellon University*
 Ken Slonneger, *University of Iowa*
 Peter Stanchev, *Kettering University*
 Ron Taylor, *Wright State University*
 Joseph Vybihal, *McGill University*
 Xiaoming Wei, *Iona College*
 Todd Whittaker, *Franklin University*
 Robert Willhoft, *Roberts Wesleyan College*
 David Womack, *University of Texas at San Antonio*
 Catherine Wyman, *DeVry University*
 Arthur Yanushka, *Christian Brothers University*
 Salih Yurttas, *Texas A & M University*