

Come qualunque linguaggio umano, Java mette a disposizione un modo per esprimere concetti. Se ha successo, questo veicolo di espressione sarà significativamente più agevole e più flessibile delle alternative, a mano a mano che i problemi si fanno più grandi e più complessi.

Non è possibile vedere Java come una semplice raccolta di funzionalità: alcune di queste non hanno senso considerate in modo isolato. Potete utilizzare la somma delle parti soltanto se pensate al *progetto*, non semplicemente alla codifica. E per capire Java in questo modo, dovete capire i suoi problemi e quelli della programmazione in generale. Questo libro parla dei problemi della programmazione, del perché sono problemi e dell'approccio seguito da Java per risolverli. Quindi, l'insieme di caratteristiche funzionali che spiego in ciascun capitolo si basa sul modo in cui un tipo particolare di problema viene risolto con il linguaggio. In questa maniera, spero di potervi aiutare, un po' per volta, fino ad arrivare al punto nel quale la mentalità di Java diventa la vostra lingua nativa.

Lungo tutto il libro riterrò che voi vogliate costruirvi un modello mentale che vi permetta di capire in profondità il linguaggio; se vi trovate davanti un rompicapo, riuscirete a darlo in pasto al vostro modello e a dedurne la soluzione.

## Prerequisiti

Questo libro dà per scontato che abbiate una certa familiarità con la programmazione: vi rendete conto che un programma è una raccolta di enunciati, vi sono chiare le idee di subroutine/funzione/macro, gli enunciati di controllo tipo "if" e i costrutti per creare cicli tipo "while", ecc. Tuttavia, potreste avere imparato queste cose in molti contesti, per esempio programmando con un macro linguaggio o lavorando con uno strumento tipo il Perl. Nella misura in cui avrete programmato fino a sentirvi a vostro agio con le idee base della programmazione, sarete in grado di lavorare su questo libro. Naturalmente, il libro sarà più *facile* per i programmatori in C e ancora di più per i programmatori in C++, ma non chiamatevi fuori se non siete esperti di questi linguaggi (però tenetevi pronti a lavorare con impegno). Presenterò i concetti di programmazione orientata agli oggetti (object-oriented programming: OOP) e i meccanismi di controllo fondamentali di Java, per cui avrete modo di sperimentarli e fin dai primi esercizi vedrete gli enunciati base per il controllo dei flussi.

Sebbene si faranno spesso riferimenti a caratteristiche dei linguaggi C e C++, questi non vanno intesi come commenti per addetti ai lavori, ma invece come un aiuto rivolto a tutti

i programmatori per collocare Java nella prospettiva di quei linguaggi, dai quali, dopo tutto, Java è stato derivato. Tenterò di rendere semplici questi riferimenti e di spiegare tutto ciò che, secondo me, potrebbe non essere familiare a un programmatore che non ha pratica del C++.

## Obiettivi

Come il mio libro precedente, *Thinking in C++*, questo si è venuto strutturando intorno al processo di insegnamento del linguaggio. In particolare, sono motivato a creare qualcosa che mi dia un aiuto per insegnare il linguaggio nei miei seminari. Quando penso a un capitolo del libro, lo considero dal punto di vista di come dovrebbe essere una buona lezione durante un seminario. Il mio obiettivo è quello di ottenere porzioni sufficientemente compatte da poter essere insegnate nell'arco di un ragionevole periodo di tempo, seguite da esercizi che si possano svolgere nel contesto di un corso.

I miei obiettivi per questo libro sono:

1. presentare il materiale un semplice passo per volta, in modo che ciascun concetto sia facilmente assimilabile prima di procedere oltre;
2. utilizzare esempi che siano quanto più è possibile semplici e brevi. Questo talvolta mi impedisce di affrontare problemi del “mondo reale”, ma ho scoperto che i principianti di solito sono più contenti quando possono capire ogni singolo particolare di un esempio, piuttosto che essere impressionati dalla portata del problema che risolve. Inoltre, esiste un rigido limite alla quantità di codice che può essere assorbita nel contesto di una lezione in aula. Qualcuno sicuramente mi criticherà perché mi servo di “esempi giocattolo”, però è una critica che sono disposto ad accettare, se è il prezzo da pagare per produrre qualcosa di pedagogicamente utile;
3. disporre accuratamente in sequenza le caratteristiche e le funzionalità, in modo che non vi troviate davanti qualcosa che non vi è stato prima spiegato. Naturalmente, questo non sempre è possibile; in quelle situazioni, fornisco una breve descrizione introduttiva;
4. darvi quello che secondo me è importante che voi capiate del linguaggio, invece di tutto quello che so. Sono convinto che esista una gerarchia dell'importanza delle informazioni, e che vi siano alcuni fatti che il 95 per cento dei programmatori non avrà mai bisogno di conoscere e che creano soltanto confusione nella testa delle persone, facendo sembrare il linguaggio ancora più complesso. Per prendere un esempio dal C, se imparate a memoria la tabella della precedenza degli operatori (cosa che non ho mai fatto), potete scrivere del codice molto intelligente. Ma se per scriverlo ci dovete pensare, confonderete il lettore/gestore di quel codice. Quindi lasciate perdere la precedenza e servitevi di parentesi quando le cose non sono chiare;
5. tenere ciascuna sezione dei capitoli ben focalizzata in modo che la durata della lezione – e il tempo fra i periodi dedicati agli esercizi – sia breve. Non soltanto questo tiene più sveglia la mente di chi partecipa al seminario, e li coinvolge di più durante le esercitazioni pratiche, ma dà al lettore anche un maggior senso di appagamento;
6. fornirvi una solida base che vi consenta di capire le questioni quanto basta per poter procedere con corsi e libri più difficili.

## Documentazione online

Il linguaggio Java e le librerie della Sun Microsystems (tutte scaricabili gratuitamente) sono distribuiti insieme con una documentazione in forma elettronica, leggibile mediante

un browser Web e praticamente ogni implementazione di Java realizzata da terzi ha questo sistema di documentazione o uno equivalente. Quasi tutti i libri su Java hanno duplicato questa documentazione. Per cui o l'avete già o potete scaricarla e, salvo quando è necessario, questo libro non ripeterà quella documentazione perché di solito farete prima a trovare le descrizioni delle classi col vostro browser Web che a cercarle in un libro (e la documentazione online sarà probabilmente più aggiornata). Questo libro vi fornirà ulteriori descrizioni delle classi soltanto quando sarà necessario per completare la documentazione in modo da farvi capire un esempio particolare.

## Capitoli

Questo libro è stato concepito con una sola cosa in mente: il modo in cui le persone apprendono il linguaggio Java. I feedback ricevuti nel corso dei miei seminari mi hanno aiutato a capire le parti difficili che hanno bisogno di essere illuminate. Nelle aree dove mi sono lasciato prendere la mano dall'ambizione e ho inserito troppe caratteristiche tutte insieme, mi sono reso conto – nel presentare il materiale – che se uno mette dentro un sacco di caratteristiche poi le deve spiegare tutte e questo finisce facilmente per aggravare la confusione nella testa di chi studia. Di conseguenza, ho fatto tutto il possibile per presentare le caratteristiche sempre in piccole quantità per volta.

L'obiettivo, quindi, di ciascun capitolo è quello di insegnare una singola caratteristica, o un piccolo gruppo di funzionalità associate, senza far riferimento a ulteriori caratteristiche. In questo modo vi sarà facile assimilare ciascuna parte nel contesto di quello che sapete al momento, prima di procedere oltre.

## Esercizi

Ho scoperto che semplici esercizi sono eccezionalmente utili per completare l'apprendimento durante un seminario, per cui ne troverete un po' alla fine di ciascun capitolo.

Gran parte degli esercizi è concepita in modo che siano sufficientemente facili da poter essere svolti in un periodo di tempo ragionevole in una situazione didattica durante la quale l'istruttore osserva, verificando che tutti gli studenti assorbano il materiale. Alcuni esercizi sono più avanzati, per evitare che gli studenti esperti si annoino. La maggior parte è concepita per essere risolta in poco tempo e per consentire un test e una messa a punto di quanto si è appreso. Alcuni sono impegnativi, ma nessuno presenta sfide impegnative. Le soluzioni degli esercizi sono disponibili nel documento elettronico *The Thinking in Java Annotated Solution Guide*, reperibile nel sito [www.bruceeckel.com](http://www.bruceeckel.com).

## Codice sorgente

Tutto il codice sorgente presentato in questo libro è disponibile sotto forma di freeware coperto da copyright, distribuito in un unico package, presso il booksite Apogeo, [www.apogeonline.com/libri/00998/allegati/](http://www.apogeonline.com/libri/00998/allegati/) e presso il sito Web [www.bruceeckel.com](http://www.bruceeckel.com).

Per garantirvi di ottenere la versione più aggiornata, questo è il sito ufficiale per la distribuzione del codice e della versione elettronica del libro. Potete trovare versioni rispecchiate del libro elettronico e del codice su altri siti (alcuni di questi sono indicati in [www.bruceeckel.com](http://www.bruceeckel.com)), ma farete bene a verificare sempre sul sito ufficiale per accertarvi che la versione rispecchiata sia in effetti quella più recente. Potete distribuire il codice in aula e in altri contesti didattici.

Il copyright serve soprattutto per assicurare che la fonte del codice venga citata in modo corretto e per impedire che il codice venga pubblicato a stampa senza autorizzazione.

(Nella misura in cui la fonte viene citata, l'utilizzo degli esempi tratti dal libro nella maggior parte dei media non è in generale un problema.)

## Standard di codifica

Nel testo di questo libro tutti gli identificatori (nomi di funzioni, variabili e classi) sono in **grassetto**. Anche la maggior parte delle parole chiave è in grassetto, salvo quelle utilizzate talmente spesso, come “class” per le quali l'uso del grassetto sarebbe fastidioso.

Utilizzo uno stile di codifica particolare per gli esempi di questo libro. È uno stile che segue quello che la stessa Sun utilizza praticamente in tutto il codice che troverete nel suo sito ([java.sun.com/docs/codeconv/index.html](http://java.sun.com/docs/codeconv/index.html)) e che risulta essere supportato dalla maggior parte degli ambienti di sviluppo per Java. Se avete letto qualche altro mio lavoro, noterete anche che lo stile di codifica della Sun coincide con il mio: questo mi fa piacere, anche se non ho nulla a che fare con esso. L'argomento dello stile di formattazione si presta a essere dibattuto accanitamente per ore, per cui mi limiterò a dire che non intendo imporre uno stile corretto tramite i miei esempi; ho i miei motivi per attenermi allo stile che mi sono dato. Siccome Java è un linguaggio di programmazione in formato libero, potete continuare a utilizzare qualunque stile vi torni più comodo.

I programmi di questo libro sono file che sono stati inseriti nel testo direttamente da file compilati. Per questa ragione, i file stampati nel libro dovrebbero funzionare tutti senza errori di compilazione. Gli errori che dovrebbero provocare segnalazioni di errore da parte del compilatore sono stati mascherati con i caratteri di commento `///` in modo che sia possibile scoprirli subito e collaudarli utilizzando mezzi automatici. Gli errori che verranno scoperti e segnalati all'autore compariranno prima nel codice sorgente distribuito e successivamente negli aggiornamenti del libro (che saranno anch'essi disponibili nel sito dell'autore).

## Versioni di Java

In generale mi baso sull'implementazione Sun di Java come criterio di riferimento per determinare se un certo comportamento è corretto.

Nel corso del tempo, la Sun ha rilasciato tre versioni principali di Java: la 1.0, la 1.1 e la 2 (che viene chiamata versione 2 anche se il rilascio del JDK da parte della Sun continua a utilizzare il criterio di numerazione 1.2, 1.3, 1.4 eccetera). La versione 2 sembra abbia portato infine Java alla sua maturità, in particolare per quanto riguarda gli strumenti per l'interfaccia utente. Questo libro si basa su ed è stato collaudato con Java 2, anche se talvolta faccio qualche concessione alle prime caratteristiche di Java 2 in modo che il codice si possa compilare sotto Linux (tramite il JDK Linux che era disponibile all'epoca in cui scrivevo questo libro).