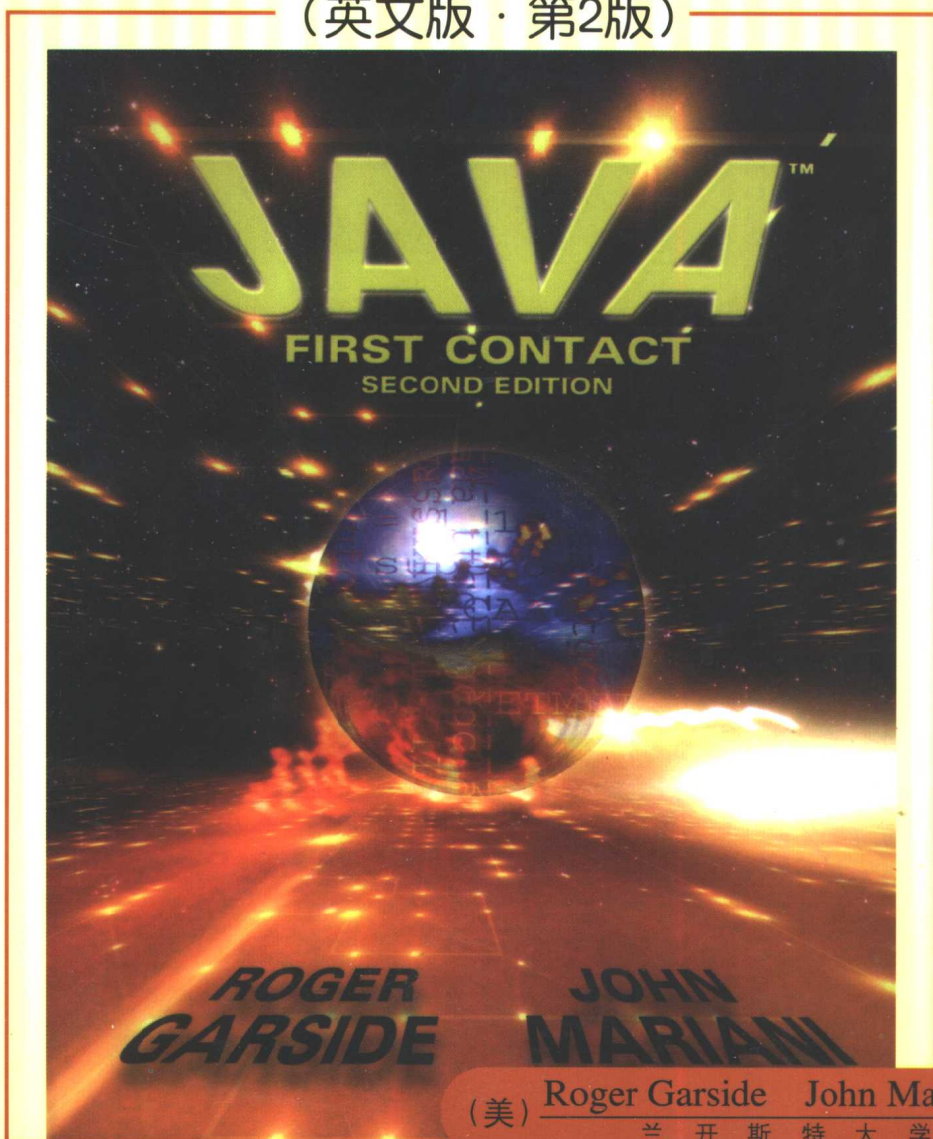


Java 教程

(英文版·第2版)



(美) Roger Garside John Mariani 著
兰 开 斯 特 大 学

经典原版书库

Java 教程

(英文版·第2版)

Java First Contact
(Second Edition)

江苏工业学院图书馆
藏书章

(美) Roger Garside John Mariani 著
兰开斯特大学



机械工业出版社
China Machine Press

Roger Garside and John Mariani: Java First Contact, Second Edition (ISBN: 0-534-37816-1).

Original edition copyright © 2003 Brooks/Cole.

First published by Brooks/Cole, an imprint of Thomson Learning, United States of America.

All rights reserved.

Reprinted for the People's Republic of China by Thomson Asia Pte Ltd and China Machine Press under the authorization of Thomson Learning. No part of this book may be reproduced in any form without the express written permission of Thomson Learning Asia and China Machine Press.

本书英文影印版由汤姆森学习出版社与机械工业出版社合作出版。未经出版者书面许可，不得以任何方式复制或抄袭本书内容。

版权所有，侵权必究。

本书版权登记号：图字：01-2003-5133

图书在版编目 (CIP) 数据

Java教程 (英文版·第2版) / (美)加赛德 (Garside, R.), (美)马瑞尼 (Mariani, J.) 著. -北京: 机械工业出版社, 2003.8

(经典原版书库)

书名原文: Java First Contact, Second Edition

ISBN 7-111-12606-8

I. J… II. ①加… ②马… III. Java语言 - 程序设计 - 教材 - 英文 IV. TP312

中国版本图书馆CIP数据核心字 (2003) 第062070号

机械工业出版社 (北京市西城区百万庄大街22号 邮政编码 100037)

责任编辑: 杨海玲

北京昌平奔腾印刷厂印刷·新华书店北京发行所发行

2003年8月第1版第1次印刷

787mm × 1092mm 1/16 · 40.5印张

印数: 0 001-3 000册

定价: 55.00元

凡购本书, 如有倒页、脱页、缺页, 由本社发行部调换

出版者的话

文艺复兴以降，源远流长的科学精神和逐步形成的学术规范，使西方国家在自然科学的各个领域取得了垄断性的优势；也正是这样的传统，使美国在信息技术发展的六十多年间名家辈出、独领风骚。在商业化的进程中，美国的产业界与教育界越来越紧密地结合，计算机学科中的许多泰山北斗同时身处科研和教学的最前线，由此而产生的经典科学著作，不仅擘划了研究的范畴，还揭集了学术的源变，既遵循学术规范，又自有学者个性，其价值并不会因年月的流逝而减退。

近年，在全球信息化大潮的推动下，我国的计算机产业发展迅猛，对专业人才的需求日益迫切。这对计算机教育界和出版界都既是机遇，也是挑战；而专业教材的建设在教育战略上显得举足轻重。在我国信息技术发展时间较短、从业人员较少的现状下，美国等发达国家在其计算机科学发展的几十年间积淀的经典教材仍有许多值得借鉴之处。因此，引进一批国外优秀计算机教材将对我国计算机教育事业的发展起积极的推动作用，也是与世界接轨、建设真正的世界一流大学的必由之路。

机械工业出版社华章图文信息有限公司较早意识到“出版要为教育服务”。自1998年开始，华章公司就将工作重点放在了遴选、移译国外优秀教材上。经过几年的不懈努力，我们与Prentice Hall, Addison-Wesley, McGraw-Hill, Morgan Kaufmann等世界著名出版公司建立了良好的合作关系，从它们现有的数百种教材中甄选出Tanenbaum, Stroustrup, Kernighan, Jim Gray等大师名家的一批经典作品，以“计算机科学丛书”为总称出版，供读者学习、研究及收藏。大理石纹理的封面，也正体现了这套丛书的品位和格调。

“计算机科学丛书”的出版工作得到了国内外学者的鼎力襄助，国内的专家不仅提供了中肯的选题指导，还不辞劳苦地担任了翻译和审校的工作；而原书的作者也相当关注其作品在中国的传播，有的还专诚为其书的中译本作序。迄今，“计算机科学丛书”已经出版了近百个品种，这些书籍在读者中树立了良好的口碑，并被许多高校采用为正式教材和参考书籍，为进一步推广与发展打下了坚实的基础。

随着学科建设的初步完善和教材改革的逐渐深化，教育界对国外计算机教材的需求和应用都步入一个新的阶段。为此，华章公司将加大引进教材的力度，在“华章教育”的总规划之下出版三个系列的计算机教材：除“计算机科学丛书”之外，对影印版的教材，则单独开辟出“经典原版书库”；同时，引进全美通行的教学辅导书“Schaum's Outlines”系列组成“全美经典学习指导系列”。为了保证这三套丛书的权威性，同时也为了更好地为学校和老师服务，华章公司聘请了中国科学院、北京大学、清华大学、国防科技大学、复旦大学、上海交通大学、南京大学、浙江大学、中国科技大学、哈尔滨工业大学、西安交通大学、中国人民大学、北京航空航天大学、北京邮电大学、中山大学、解放军理工大学、郑州大学、湖北工学院、中国

家信息安全测评认证中心等国内重点大学和科研机构在计算机的各个领域的著名学者组成“专家指导委员会”，为我们提供选题意见和出版监督。

这三套丛书是响应教育部提出的使用外版教材的号召，为国内高校的计算机及相关专业的教学度身订造的。其中许多教材均已为M. I. T., Stanford, U.C. Berkeley, C. M. U. 等世界名牌大学所采用。不仅涵盖了程序设计、数据结构、操作系统、计算机体系结构、数据库、编译原理、软件工程、图形学、通信与网络、离散数学等国内大学计算机专业普遍开设的核心课程，而且各具特色——有的出自语言设计者之手、有的历经三十年而不衰、有的已被全世界的几百所高校采用。在这些圆熟通博的名师大作的指引之下，读者必将在计算机科学的宫殿中由登堂而入室。

权威的作者、经典的教材、一流的译者、严格的审校、精细的编辑，这些因素使我们的图书有了质量的保证，但我们的目标是尽善尽美，而反馈的意见正是我们达到这一终极目标的重要帮助。教材的出版只是我们的后续服务的起点。华章公司欢迎老师和读者对我们的工作提出建议或给予指正，我们的联系方式如下：

电子邮件：hzedu@hzbook.com

联系电话：(010) 68995264

联系地址：北京市西城区百万庄南街1号

邮政编码：100037

专家指导委员会

(按姓氏笔画顺序)

尤晋元
石教英
张立昂
邵维忠
周克定
郑国梁
高传善
裘宗燕

王 珊
吕 建
李伟琴
陆丽娜
周傲英
施伯乐
梅 宏
戴 葵

冯博琴
孙玉芳
李师贤
陆鑫达
孟小峰
钟玉琢
程 旭

史忠植
吴世忠
李建中
陈向群
岳丽华
唐世渭
程时端

史美林
吴时霖
杨冬青
周伯生
范 明
袁崇义
谢希仁

Dedication

To Tessa, la migliore fabbra, and to Dan and Beth—R.G.G.

To Louise, and her own personal inheritance hierarchy: Moira, Elizabeth, Louis, Vera, and Thomas—J.A.M.

Preface

This is a book about programming in the Java™ programming language. It is intended for people who have no previous programming experience, so it is unlike many other Java programming books, which appear to have been written by experts for experts. It was written from scratch and not based on previous C or C++ textbooks by the same authors, which also makes it different from many other books. This book was developed for a specific purpose and is not an evolution from an earlier book. The book is thus well suited for a first-year undergraduate programming course. This edition has been thoroughly revised and updated for the latest versions of Java.

The authors have between them taught the first-year programming course at Lancaster University for more than 14 years, first in Pascal, then in Ada, and since 1996–1997 in Java, one of the first British universities to use the Java language for first-year programming. This book is based on our experience of teaching the language to classes of some 150 students. One author has experience in object-oriented software development and database management systems from Objective-C to C++ and ONTOS to POET. The other has written a number of textbooks on natural language processing and computer architecture.

Java is an interesting development in the field of programming languages. We briefly explore some of the history of Java in our first chapter, but for now, suffice it to say that it is very much a language for the new millennium. Part of the “buzz” around Java is because it is the language of the World Wide Web (itself a major aspect of computing now and in the future), but a significant feature of the language is that it is *object-oriented*. This means that we can think of systems as composed of a number of *objects* that provide and request services from each other to accomplish an overall task. The object model can be applied from the earliest stages of system requirements and design and carried through to coding, testing, and maintenance.

Rather than avoid or delay object orientation, we have decided to make objects our departure point. We present our study of Java in terms of objects from the very start rather than begin as a conventional programming text and graft on objects at a later point.

We believe that Java is an elegant language with an extensive class library that supports the programming of graphical interfaces and the use of computer networks. We therefore believe it is a good language with which to introduce the concepts of computing, and it is rewarding because the programmer is able to write significant pieces of code quickly.

The book consists of five parts:

- In Part 1, we concentrate on using objects. Here objects are supplied either from the standard Java library of classes or from example classes of our own creation. The programmer writes applications that call upon the services of these preexisting objects and interprets the results.
 - In Part 2, we change emphasis and look at what is required to write (and use) objects of our own.
-

- In Part 3, we explore a major concept in object models, that of *inheritance*, and how we can apply it in Java. This part also covers two areas of Java that rely heavily on inheritance: the use of exceptions for dealing with error conditions in programs and the Abstract Windowing Toolkit (AWT), which supports the creation of graphical user interfaces.
- In Part 4, we examine more advanced features of Java, such as the input/output facilities, and how applets—applications that can be executed over the World Wide Web—can be written. We also examine how common data structures can be realized in Java, introducing ideas about the use of pointers and linked data structures.
- In Part 5, we look at object-oriented design and show how we can identify objects from the requirements of a system and move forward through design and into coding. This section is supported by examining the development of a system from start to finish.

We cover all the major parts of the basic language except multithreading, which we introduce only briefly in Chapter 21. In a book of this size, it is possible to cover only a small part of the extensive facilities provided by Java. Thus, we introduce only a few of the main classes of the AWT, and we consider access to networks only in terms of the communication between an applet and its originating computer. Some of the sections are marked as optional on a first reading. We see Java as having a future as a good general-purpose programming language rather than as only a vehicle for Web programming. We therefore spend most of the book discussing normal application programs and discuss how applets can be written to be executed over the Web only in Chapter 20.

The book contains many examples and exercises to illustrate the principles discussed. All example programs are also available through a Web site (see Appendix A).

A number of environments are available for developing Java programs. This book describes Version 1.3 of Java, and the development environment it describes is the basic Software Development Kit (SDK) from Sun Microsystems. The examples have all been tested under the Windows NT and UNIX operating systems. Appendix A describes how to obtain a copy of the SDK, as well as the special classes used in this book.

We believe that Java is an exciting development in programming language design. We hope you will enjoy this introduction to the world of Java programming.

*R.G.G. & J.A.M.
Lancaster*

Acknowledgments

We would like to thank Professors Doug Shepherd and Ian Sommerville of the Computing Department at Lancaster University for supporting our efforts with this book. We introduced Java as our first-year teaching language in October 1996, one of the first UK universities to do so. We would also like to thank Richard Cardoe for his assistance in developing course material and Ian Warren for his help with the material in Part 5. We would especially like to thank our first-year students for serving as guinea pigs for various versions of the material and exercises in this book. Finally, we would like to thank our long-suffering wives for putting up with us during the long process of producing this book.

*R.G.G.
J.A.M.
Lancaster*

Contents

Part 1 Using Objects

Programming and the Java Language	2
1.1 Programs and Programming	2
1.2 Algorithms	3
1.3 High-Level Languages and Programs	5
1.4 A Simple Computer	8
1.5 Machine Code	9
1.6 Files and the Filing System	14
1.7 The World Wide Web	15
1.8 The Java Programming Language	17
1.9 Getting Started with Java	20
1.10 A First Java Program	21
1.11 The <code>print</code> and <code>println</code> Methods	24
1.12 How This Book Is Arranged	25
1.13 Key Points in Chapter 1	26
1.14 Exercises	27
2 Object Orientation	29
2.1 Objects and Classes	29
2.2 Software Objects	32
2.3 More about Single Objects	36
2.4 An Object-Oriented Program	39
2.5 Types	40
2.6 Classes and Instances Revisited	43

2.7	Key Points in Chapter 2	46
2.8	Exercises	47
3	Declaring Objects and Calling Methods	48
3.1	Program Structure	48
3.2	The class <code>Person</code>	49
3.3	A Program to Manipulate a Person	51
3.4	The <code>import</code> Statement	52
3.5	Declaring Objects	53
3.6	Using Methods to Set the Attributes of Objects	57
3.7	Using Methods to Extract Object Attributes	60
3.8	Using Constants in Java	63
3.9	Using Objects and Methods	65
3.10	A Variety of Methods	66
3.11	Constructors Revisited	68
3.12	Input to a Program (Optional)	72
3.13	Key Points in Chapter 3	73
3.14	Exercises	74
4	Selecting Among Alternatives	77
4.1	Reading Values from the Keyboard	77
4.2	More on Integer Variables	80
4.3	Type Checking	84
4.4	Making Decisions	85
4.5	Selection Statements in Java	89
	4.5.1 Statements	90
	4.5.2 Relational Operators and Boolean Expressions	92
4.6	The Boolean Type	96
4.7	The <code>switch</code> Statement	101
4.8	Testing a New Class	107
4.9	Key Points in Chapter 4	110
4.10	Exercises	111
5	Repetition	113
5.1	Repetition as a Basic Control Structure	113
5.2	Looping a Predetermined Number of Times: The <code>for</code> Statement	115

5.3	Looping an Indeterminate Number of Times: The <code>while</code> Statement	121
5.4	Stopping in the Middle of an Iteration	124
5.5	<code>for</code> Loops and <code>while</code> Loops	126
5.6	Nested Loops	127
5.7	Boolean Expressions for Loops	126
5.8	Testing at the End of the Loop	134
5.9	Other Java Loop Features	136
5.10	Key Points in Chapter 5	138
5.11	Exercises	139
6	Basic Java Data Types	142
6.1	Objects and Basic Data Types	142
6.1.1	Declaration	142
6.1.2	Setting a Value	144
6.1.3	Obtaining a Value	146
6.1.4	Arguments to Methods	147
6.2	The <code>int</code> Data Type	147
6.3	Other Whole Number Data Types in Java (Optional)	151
6.4	Floating-Point Data Types	153
6.5	The <code>boolean</code> Data Type	155
6.6	The <code>char</code> Data Type	156
6.7	The <code>String</code> Class	157
6.8	Methods for the <code>String</code> Class	161
6.9	Wrapper Classes (Optional)	162
6.10	Key Points in Chapter 6	165
6.11	Exercises	166

Part 2 Writing Objects

7	A Simple Class	169
7.1	What We Are Trying to Achieve	169
7.2	Providing the <code>Person</code> Class	172
7.3	Methods for the <code>Person</code> Class	175
7.4	Actual and Formal Arguments	178
7.5	Modes of Argument Passing	182
7.6	Return Values	185

7.7	Lexical Conventions within a Class	185
7.8	Key Points in Chapter 7	186
7.9	Exercises	187
8	More on the Simple Class	189
8.1	Constructor Methods	189
8.2	Overloading	193
8.3	Class Constants	195
8.4	Class Variables	196
8.5	Private Methods	198
8.6	Class or Static Methods	201
8.7	Revisiting the Main Class	202
8.8	Packages and Directories	206
8.8.1	The <code>import</code> Statement	209
8.9	Scope and Visibility	210
8.9.1	Intraclass Visibility	210
8.9.2	Use of <code>this</code>	214
8.9.3	Interclass Visibility	214
8.10	Key Points in Chapter 8	215
8.11	Exercises	216
9	Arrays	218
9.1	Collections of Elements	218
9.2	Arrays of Objects	221
9.3	Searching an Array	223
9.4	Binary Search	226
9.5	Sorting an Array	227
9.6	Arrays as Arguments	234
9.7	Multidimensional Arrays	237
9.8	Nonrectangular Arrays (Optional)	241
9.9	Key Points in Chapter 9	242
9.10	Exercises	244
10	Objects within Objects	249
10.1	What We Are Trying to Achieve	249
10.2	Writing the <code>OurDate</code> Class	250

10.3	Using the <code>OurDate</code> Class	251
10.4	Objects as Arguments	253
10.5	Multiple References to the Same Object	255
10.6	Objects as Arguments and Return Values: Call by Reference	260
10.6.1	Changing the Contents of the Formal and Actual Arguments	263
10.7	Hiding References to Other Objects	266
10.8	Key Points in Chapter 10	267
10.9	Exercises	268
11	Putting Objects to Work	270
11.1	A Task Organizer Program	270
11.2	A Priority Queue Class	271
11.3	Implementing a Priority Queue with an Array	275
11.4	Alternative Implementations of <code>PriorityQueue</code> (Optional)	281
11.5	Testing the <code>PriorityQueue</code> Class	282
11.6	Using the <code>PriorityQueue</code> Class	287
11.7	Outstanding Issues	289
11.8	Key Points in Chapter 11	291
11.9	Exercises	291

Part 3 Advanced Objects

12	Introduction to Inheritance	293
12.1	Motivation	293
12.1.1	Data Modeling	293
12.1.2	Programming	298
12.2	What's the Difference?	299
12.3	Overriding Inherited Methods	300
12.4	Access Rights and Subclasses	304
12.5	Airplane Reservations: An Example	305
12.6	Key Points in Chapter 12	310
12.7	Exercises	311
13	Class and Method Polymorphism	313
13.1	Person and Student: An Example	313

13.2	Constructor Methods and Inheritance	314
13.2.1	Constructor Chaining	315
13.3	Multiple Levels of Inheritance: The Inheritance Hierarchy	316
13.4	The Class Object	317
13.5	Polymorphism	319
13.6	Polymorphism and Heterogeneous Collections	322
13.6.1	Dynamic Method Binding (Late Binding)	325
13.7	Calling Overridden Methods	325
13.8	Methods in Derived Classes	327
13.9	Key Points in Chapter 13	329
13.10	Exercises	330
14	Abstract Classes and Interfaces	333
14.1	Abstract Classes	333
14.2	Polymorphism	338
14.3	Interfaces	340
14.4	Key Points in Chapter 14	346
14.5	Exercises	347
15	Throwing and Catching Exceptions	351
15.1	Motivation: Robust Programs	351
15.2	Defining a New Exception	353
15.3	Throwing an Exception	354
15.4	Catching an Exception	355
15.4.1	The finally Clause (Optional)	361
15.5	Key Points in Chapter 15	363
15.6	Exercises	365
16	Graphics and the Abstract Windowing Toolkit	366
16.1	Graphical User Interfaces	366
16.2	A Simple Program with a Graphical Interface	368
16.3	Writing the Chapter16n0 Class	372
16.3.1	The Constructor for the Chapter16n0 Class	372
16.3.2	Other Layout Managers	375
16.3.3	The main Method for the Chapter16n0 Class	376
16.3.4	The actionPerformed Method of the Chapter16n0 Class	376
16.3.5	The windowClosing Method of the Chapter16n0 Class	377

16.4	Writing the Canvas0 Class	378
16.5	Writing Text on the Canvas	382
16.6	Animating the Simple Graphics Program	383
16.7	Input of Character Strings in a Graphical Interface	386
16.7.1	Setting Up the Picture	387
16.7.2	Getting a String from a <code>TextField</code>	389
16.7.3	Drawing the Thermometer	391
16.8	Menus, Files, and Images (Optional)	392
16.8.1	Setting Up Menus	393
16.8.2	Selecting a File	397
16.8.3	Displaying an Image	398
16.8.4	Tracking the Mouse	399
16.9	Key Points in Chapter 16	400
16.10	Exercises	401

Part 4 Advanced Java

17	Linked Data Structures	406
17.1	Linear and Linked Data Structures	406
17.2	Implementing a Priority Queue Using a Linked Data Structure	409
17.3	Methods for the <code>PriorityQueue</code> Class	413
17.3.1	The <code>length</code> Method	413
17.3.2	The <code>first</code> Method	415
17.3.3	The <code>remove</code> Method	416
17.4	The <code>insert</code> Method	417
17.5	Deletion from a Linked Data Structure (Optional)	423
17.6	Doubly Linked Lists (Optional)	425
17.7	Using Linked Data Structures	428
17.8	Key Points in Chapter 17	429
17.9	Exercises	430
18	Recursion and Binary Trees	434
18.1	Recursion	434
18.2	Solving the Towers of Hanoi Problem	437
18.2.1	A Recursive Solution to the Towers of Hanoi Problem	439
18.2.2	An Iterative Solution to the Towers of Hanoi Problem	441