原版风暴 · 开发大师系列

"Java之父" James Gosling 作品

PEARSON Addison Wesley

The Java Programming Language Third Edition

Java 编程语言

（第三版·影印版）

[美] Ken Arnold James Gosling David Holmes 著

Pearson Education

中国电力出版社

www.infopower.com.cn

原版风暴·开发大师系列

"Java之父" James Gosling 作品

The Java Programming Language Third Edition

Java 编程语言

（第三版·影印版）

[美]Ken Arnold James Gosling David Holmes 著

中国电力出版社

The Java Programming Language, Third Edition (ISBN 0-201-70433-1)

北京市版权局著作合同登记号：图字：01-2003-1011

图书在版编目（CIP）数据

Java 编程语言（第三版）/（美）阿诺德（Arnold,K.），（美）高林（Gosling,J.），（美）福尔摩斯（Holmes,D.）著. —影印本. —北京：中国电力出版社，2003
（原版风暴·开发大师系列）
ISBN 7-5083-1499-9
Ⅰ.J... Ⅱ.①阿...②高...③福... Ⅲ.JAVA 语言—程序设计—英文 Ⅳ.TP312
中国版本图书馆 CIP 数据核字（2003）第 023076 号

责任编辑：姚贵胜
丛 书 名：原版风暴·开发大师系列
书　　名：Java编程语言（第三版·影印版）
编　　著：（美）Ken Arnold, James Gosling, David Holmes
出版发行：中国电力出版社
地址：北京市三里河路6号 邮政编码：100044
电话：（010）88515918 传真：（010）88423191
印　　刷：北京地矿印刷厂
发 行 者：新华书店总店北京发行所
开　　本：787×1092 1/16　　**印　　张：**43.75
书　　号：ISBN 7-5083-1499-9
版　　次：2003年5月北京第一版
印　　次：2003年5月第一次印刷
定　　价：65.00 元

Preface

Beautiful buildings are more than scientific. They are true organisms, spiritually conceived; works of art, using the best technology by inspiration rather than the idiosyncrasies of mere taste or any averaging by the committee mind.
—Frank Lloyd Wright

THE Java™ programming language has been warmly received by the world community of software developers and Internet content providers. Users of the Internet and World Wide Web benefit from access to secure, platform-independent applications that can come from anywhere on the Internet. Software developers who create applications in the Java programming language benefit by developing code only once, with no need to "port" their applications to every software and hardware platform.

For many, the language was known first as a tool to create applets for the World Wide Web. An *applet* is a mini-application that runs inside a Web page. An applet can perform tasks and interact with users on their browser pages without using resources from the Web server after being downloaded. Some applets may, of course, talk with the server to do their job, but that's their business.

The Java programming language is indeed valuable for distributed network environments like the Web. However, it goes well beyond this domain to provide a powerful general-purpose programming language suitable for building a variety of applications that either do not depend on network features, or want them for different reasons. The ability to execute downloaded code on remote hosts in a secure manner is a critical requirement for many organizations.

Other groups use it as a general-purpose programming language for projects in which machine independence is less important. Ease of programming and safety features help you quickly produce working code. Some common programming errors never occur because of features like garbage collection and type-safe references. Support for multithreading caters to modern network-based and graphical user interface–based applications that must attend to multiple tasks simulta-

neously, and the mechanisms of exception handling ease the task of dealing with error conditions. While the built-in tools are powerful, it is a simple language in which programmers can quickly become proficient.

The Java programming language is designed for maximum portability with as few implementation dependencies as possible. An `int`, for example, is a 32-bit signed two's-complement integer in all implementations, irrespective of the CPU architecture on which the program executes. Defining everything possible about the language and its runtime environment enables users to run compiled code anywhere and share code with anyone who has a Java runtime environment.

About This Book

This book teaches the Java programming language to people who are familiar with basic programming concepts. It explains the language without being arduously formal or complete. This book is not an introduction to object-oriented programming, although some issues are covered to establish a common terminology. Other books in this series, and much online documentation, focus on applets, graphical interfaces, databases, components, and other specific kinds of programming tasks. For other references, see "Further Reading" on page 563.

This third edition includes the changes introduced in the Java 2 Platform, such as the new `strictfp` keyword, collection classes, and reference objects, as implemented in the Java 2 SDK, Standard Edition Version 1.3 (sometimes colloquially referred to as JDK 1.3 or simply 1.3). You will also find brief coverage of the other main packages. If you have already read the second edition, you will find that much of the information in this edition has been restructured to improve the presentation of language features—such as nested classes and interfaces—and class API's. This edition will give you a lot of new information, but since most of the language is unchanged, and almost all main package types are still usable, you will want to pay most attention to the newer areas.

The Java programming language shares many features common to most programming languages in use today. The language should look familiar to C and C++ programmers because it was designed with C and C++ constructs where the languages are similar. That said, this book is neither a comparative analysis nor a "bridge" tutorial—no knowledge of C or C++ is assumed. C++ programmers, especially, may be as hindered by what they must unlearn as they are helped by their knowledge.

Chapter 1—*A Quick Tour*—gives a quick overview of the language. Programmers who are unfamiliar with object-oriented programming notions should read the quick tour, while programmers who are already familiar with object-oriented programming paradigms will find the quick tour a useful introduction to the object-oriented features of the language.

Chapters 2, 3, 4, and 5 cover the object-oriented core features of the language, namely, class declarations that define components of a program, and objects manufactured according to class definitions. Chapter 2—*Classes and Objects*—describes the basis of the language: classes. Chapter 3—*Extending Classes*—describes how an existing class can be *extended*, or *subclassed*, to create a new class with additional data and behavior. Chapter 4—*Interfaces*—describes how to declare interface types which are abstract descriptions of behavior that provide maximum flexibility for class designers and implementors. Chapter 5—*Nested Classes and Interfaces*—describes how classes and interfaces can be declared inside other classes and interfaces, and the benefits that provides.

Chapters 6 and 7 cover standard constructs common to most languages. Chapter 6—*Tokens, Operators, and Expressions*—describes the tokens of the language from which statements are constructed, how the tokens and operators are used to build expressions, and how expressions are evaluated. Chapter 7—*Control Flow*—describes how control statements direct the order of statement execution.

Chapter 8—*Exceptions*—describes the language's powerful error-handling capabilities. Chapter 9—*Strings*—describes the built-in language and runtime support for `String` objects.

Chapter 10—*Threads*—explains the language's view of multithreading. Many applications, such as graphical interface–based software, must attend to multiple tasks simultaneously. These tasks must cooperate to behave correctly, and threads meet the needs of cooperative multitasking.

Chapter 11—*Programming with Types*—describes the type-related classes: individual objects that describe each class and interface, and classes that wrap primitive data types such as integers and floating-point values into their own object types.

Chapter 12—*Garbage Collection and Memory*—talks about garbage collection, finalization, and lower-strength reference objects.

Chapter 13—*Packages*—describes how you can group collections of classes and interfaces into separate packages.

Chapter 14—*Documentation Comments*—shows how to write reference documentation in comments.

Chapters 15 through 19 cover the main packages. Chapter 15—*The I/O Package*—describes the input/output system, which is based on *streams*. Chapter 16—*Collections*—covers the *collection* or *container classes* such as sets and lists. Chapter 17—*Miscellaneous Utilities*—covers the rest of the *utility classes* such as bit sets and random number generation. Chapter 18—*System Programming*—leads you through the *system classes* that provide access to features of the underlying platform. Chapter 19—*Internationalization and Localization*—covers some of the tools used to create programs that can run in many linguistic and cultural environments.

Chapter 20—*Standard Packages*—briefly explores the packages that are part of the standard platform, giving overviews of those packages not covered in more detail in this book.

Appendix A—*Runtime Exceptions*—lists all the runtime exceptions and errors that the runtime system itself can throw.

Appendix B—*Useful Tables*—has tables of information that you may find useful for quick reference.

Finally, *Further Reading* lists works that may be interesting for further reading on complete details, object orientation, programming with threads, software design, and other topics.

Examples and Documentation

All the code examples in the text have been compiled and run on the latest version of the language available at the time the book was written, which was the Java 2 SDK, Standard Edition, Version 1.3. Only supported features are covered—deprecated types, methods, and fields are ignored except where unavoidable. We have also covered issues beyond writing programs that simply compile. Part of learning a language is to learn to use it well. For this reason, we have tried to show principles of good programming style and design.

In a few places we refer to online documentation. Development environments provide a way to automatically generate documentation (usually HTML documents) from a compiled class using the documentation comments. This documentation is normally viewed using a Web browser.

Acknowledgments (First Edition)

No technical book-writing endeavor is an island unto itself, and ours was more like a continent. Many people contributed technical help, excellent reviews, useful information, and book-writing advice.

Contributing editor Henry McGilton of Trilithon Software played the role of "chief editorial firefighter" to help make this book possible. Series editor Lisa Friendly contributed dogged perseverance and support.

A veritable multitude of reviewers took time out of their otherwise busy lives to read, edit, advise, revise, and delete material, all in the name of making this a better book. Kevin Coyle performed one of the most detailed editorial reviews at all levels. Karen Bennet, Mike Burati, Patricia Giencke, Steve Gilliard, Bill Joy, Rosanna Lee, Jon Madison, Brian O'Neill, Sue Palmer, Stephen Perelgut, R. Anders Schneiderman, Susan Sim, Bob Sproull, Guy Steele, Arthur van Hoff, Jim Waldo, Greg Wilson, and Ann Wollrath provided in-depth review. Geoff Arnold, Tom Cargill, Chris Darke, Pat Finnegan, Mick Jordan, Doug Lea, Randall Murray,

Roger Riggs, Jimmy Torres, Arthur van Hoff, and Frank Yellin contributed useful comments and technical information at critical junctures.

Alka Deshpande, Sharon Flank, Nassim Fotouhi, Betsy Halstead, Kee Hinckley, Dr. K. Kalyanasundaram, Patrick Martin, Paul Romagna, Susan Snyder, and Nicole Yankelovich collaborated to make possible the five words of non-ISO-Latin-1 text on pages 140 and 406. Jim Arnold provided research help on the proper spelling, usage, and etymology of "smoog" and "moorge." Ed Mooney helped with the document preparation. Herb and Joy Kaiser were our Croatian language consultants. Cookie Callahan, Robert E. Pierce, and Rita Tavilla provided the support necessary to keep this project going at many moments when it would otherwise have stalled with a sputtering whimper.

Thanks to Kim Polese for supplying us the capsule summary of why the Java programming language is important to computer users as well as programmers.

Support and advice were provided at critical moments by Susan Jones, Bob Sproull, Jim Waldo, and Ann Wollrath. And we thank our families, who, besides their loving support, would at times drag us out to play when we should have been working, for which we are deeply grateful.

And thanks to the folks at Peet's Coffee and Tea, who kept us buzzed on the best Java on the planet.

ACKNOWLEDGMENTS (SECOND EDITION)

The cast of characters for this second edition is much like the first.

Series Editor Lisa Friendly continued to be doggedly supportive and attentive. The set of reviewers was smaller, overlapping, and certainly as helpful and thorough. Overall reviews by Steve Byrne, Tom Cargill, Mary Dageforde, Tim Lindholm, and Rob Murray were critical to clarity. Brian Beck, Peter Jones, Doug Lea, Bryan O'Sullivan, Sue Palmer, Rosanna Lee, Lori Park, Mark Reinhold, Roger Riggs, Ann Wollrath, and Ken Zadek contributed focused reviews of important parts. Guy Steele's support was ongoing and warm. Rosemary Simpson's extensive and intensive efforts to make a useful index are deeply appreciated. Carla Carlson and Helen Leary gave logistic support that kept all the wheels on the tracks instead of in the ditch. Gerry Wiener provided the Tibetan word on page 406, and we also had help on this from Craig Preston and Takao Miyatani. All who submitted errata and suggestions from the first edition were helpful.

For some inexplicable reason we left the friendly folks of Addison-Wesley off the original acknowledgments—luckily, most of them were present again for this edition. A merged list for both editions includes Kate Duffy, Rosa Gonzales, Mike Hendrickson, Marina Lang, Shannon Patti, Marty Rabinowitz, Sarah Weaver, and Pamela Yee. Others did much that we are blissfully unaware of, but for which we are nonetheless abidingly grateful.

The revision was additionally aided by Josh Bloch, Joe Fialli, Jimmy Torres, Benjamin Renaud, Mark Reinhold, Jen Volpe, and Ann Wollrath.

And Peet's Coffee and Tea continued its supporting role as purveyor to the caffeine-consuming connoisseur.

ACKNOWLEDGMENTS (THIRD EDITION)

The third edition required yet more reviews and work, and the helper list is equally critical. Lisa Friendly continued her attempts to keep the project in line; someday we will cooperate better. The set of reviewers included new faces and old friends, all helpful: Joshua Bloch, Joseph Bowbeer, Gilad Bracha, Keith Edwards, Joshua Engel, Rich Gillam, Peter Haggar, Cay Horstmann, Alexander Kuzmin, Doug Lea, Keith Lea, Tim Lindholm, David Mendenhall, Andrew M. Morgan, Ray Ortigas, Brian Preston, Mark Schuldenfrei, Peter Sparago, Guy Steele, Antoine Trux, and our Russian compatriots Leonid Arbouzov, Valery Shakurov, Viatcheslav Rybalov, Eugene Latkin, Dmitri Khukhro, Konstantin Anisimov, Alexei Kaigorodov, Oleg Oleinik, and Maxim Sokolnikov. Several people let us bend their ears to figure out how to approach things better: Peter Jones, Robert W. Scheifler, Susan Snyder, Guy Steele, Jimmy Torres, and Ann Wollrath. Helen Leary made the logistics work smoothly, as always.

Material support is always provided by the Addison-Wesley team: Julie DiNicola, Mike Hendrickson, and Tracy Russ.

And since the last edition, Peet's Coffee and Tea has opened up on the East Coast, so the eastern part of this writing team can enjoy it regularly. The world continues to improve apace.

Any errors or shortcomings that remain in this book—despite the combined efforts of these myriads—are completely the responsibility of the authors.

Results! Why, man, I have gotten a lot of results.
I know several thousand things that won't work.
—Thomas Edison

Contents