学习Java（影印版）

第四版
上册

Learning Java™

O'REILLY®
東南大學出版社

Patrick Niemeyer
& Daniel Leuck 著

（第四版／上册）

学习Java（影印版）

Learning Java

Patrick Niemeyer, Daniel Leuck 著

O'REILLY®

Beijing · Cambridge · Farnham · Köln · Sebastopol · Tokyo

O'Reilly Media, Inc.授权东南大学出版社出版

南京 东南大学出版社

图书在版编目（CIP）数据

学习 Java：第 4 版：英文/(美)尼麦耶（Niemeyer, P.），(美)洛伊克（Leuck, D.)著．—影印本．—南京：东南大学出版社，2014.1

书名原文：Learning Java, 4E

ISBN 978-7-5641-4596-5

I. ① 学… II. ①尼… ②洛… III. ① JAVA 语言－程序设计－英文 IV. ① TP312

中国版本图书馆 CIP 数据核字（2013）第 246089 号

江苏省版权局著作权合同登记

图字：10-2013-129 号

学习 Java 第四版（影印版）

出版发行：东南大学出版社
地　　址：南京四牌楼 2 号　　邮编：210096
出 版 人：江建中
网　　址：http://www.seupress.com
电子邮件：press@seupress.com
印　　刷：扬中市印刷有限公司
开　　本：787 毫米 × 980 毫米　16 开本
印　　张：64.5
字　　数：1263 千字
版　　次：2014 年 1 月第 1 版
印　　次：2014 年 1 月第 1 次印刷
书　　号：ISBN 978-7-5641-4596-5
定　　价：138.00 元（上下册）

本社图书若有印装质量问题，请直接与营销部联系。电话（传真）：025-83791830

Preface

This book is about the Java programming language and environment. Whether you are a software developer or just someone who uses the Internet in your daily life, you've undoubtedly heard about Java. Its introduction was one of the most exciting developments in the history of the Web and Java applications have powered much of the growth of business on the Internet in the past 15 years. Java is, arguably, the most popular programming language in the world, used by millions of developers on almost every kind of computer imaginable. In the past decade, Java has surpassed languages such as C++ and Visual Basic in terms of developer demand and has become the de facto language for certain kinds of development—especially for web-based services. Most universities are now using Java in their introductory courses alongside the other important modern languages. Perhaps you are using this text in one of your classes right now!

This book gives you a thorough grounding in Java fundamentals and APIs. *Learning Java*, Fourth Edition, attempts to live up to its name by mapping out the Java language and its class libraries, programming techniques, and idioms. We'll dig deep into interesting areas and at least scratch the surface of the rest. Other titles from O'Reilly pick up where we leave off and provide more comprehensive information on specific areas and applications of Java.

Whenever possible, we provide compelling, realistic, and fun examples and avoid merely cataloging features. The examples are simple, but hint at what can be done. We won't be developing the next great "killer app" in these pages, but we hope to give you a starting point for many hours of experimentation and inspired tinkering that will lead you to develop one yourself.

Who Should Read This Book

This book is for computer professionals, students, technical people, and Finnish hackers. It's for everyone who has a need for hands-on experience with the Java language with an eye toward building real applications. This book could also be considered a

crash course in object-oriented programming, networking, GUIs, and XML. As you learn about Java, you'll also learn a powerful and practical approach to software development, beginning with a deep understanding of the fundamentals of Java and its APIs.

Superficially, Java looks like C or C++, so you'll have a tiny head start in using this book if you have some experience with one of these languages. If you do not, don't worry. Don't make too much of the syntactic similarities between Java and C or C++. In many respects, Java acts like more dynamic languages such as Smalltalk and Lisp. Knowledge of another object-oriented programming language should certainly help, although you may have to change some ideas and unlearn a few habits. Java is considerably simpler than languages such as C++ and Smalltalk. If you learn well from concise examples and personal experimentation, we think you'll like this book.

The last part of this book branches out to discuss Java in the context of web applications, web services, and XML processing, so you should be familiar with the basic ideas behind web browsers, servers, and documents.

New Developments

This edition of *Learning Java* is actually the sixth edition—updated and retitled—of our original, popular *Exploring Java*. With each edition, we've taken great care not only to add new material covering additional features, but to thoroughly revise and update the existing content to synthesize the coverage and add years of real-world perspective and experience to these pages.

One noticeable change in recent editions is that we've deemphasized the use of applets, reflecting their diminished role in recent years in creating interactive web pages. In contrast, we've greatly expanded our coverage of Java web applications, web services, and XML, which are now mature technologies.

We cover all of the important features of the latest release of Java, officially called Java Standard Edition (SE) 7, JDK 1.7. Sun (Java's keeper before Oracle) has changed the naming scheme many times over the years. Sun coined the term *Java 2* to cover the major new features introduced in Java version 1.2 and dropped the term *JDK* in favor of *SDK*. With the sixth release, Sun skipped from Java version 1.4 to Java 5.0, but reprieved the term JDK and kept its numbering convention there. After that, we had Java 6 and now we reach Java 7.

This release of Java reflects a mature language with relatively few syntactic changes but significant updates to APIs and libraries. We've tried to capture these new features and update every example in this book to reflect not only the current Java practice, but style as well.

New in This Edition (Java 6 and 7)

This edition of the book has been significantly reworked to be as complete and up-to-date as possible. It incorporates changes from both the Java 6 and Java 7 releases that occurred since the last edition of this book. New topics in this edition include:

- New language features, including type inference in generics and improved exception handling and automatic resource management syntax
- New concurrency utilities including the Fork-Join framework
- The new NIO Files API, which allows new types of filesystem access to be implemented in Java
- New versions of the Java Servlets (3.0) and web services APIs, including use of the new annotations-based deployment and built-in web service container
- New version of JAXB (2.2) Java XML Binding, including use of the new annotations for binding Java to XML
- Improved Swing desktop integration and enhancements to key Swing components such as JTable
- Updated examples and analysis throughout the book

Using This Book

This book is organized roughly as follows:

- Chapters 1 and 2 provide a basic introduction to Java concepts and a tutorial to give you a jump start on Java programming.
- Chapter 3 discusses fundamental tools for developing with Java (the compiler, the interpreter, and the JAR file package).
- Chapters 4 through 7 describe the Java language itself, beginning with the basic syntax and then covering classes and objects, exceptions, arrays, enumerations, annotations, and much more.
- Chapter 8 covers generics and parameterized types in Java.
- Chapter 9 covers the language's built-in thread facilities and the Java Concurrency package, which should be of particular interest to advanced programmers.
- Chapter 10 covers text processing, formatting, scanning, string utilities, and the powerful regular expressions API.
- Chapter 11 covers much of the core API including utilities and collections.
- Chapter 12 covers Java I/O, streams, files, and the NIO package.

- Chapters 13 and 14 cover Java networking, including sockets and NIO, URLs, and RMI.
- Chapter 15 covers web applications using servlets, servlet filters, and WAR files, as well as web services.
- Chapters 16 through 21 cover GUI development with the Abstract Window Toolkit (AWT) and Swing, which provide graphical user interface (GUI) and image support.
- Chapter 22 covers the JavaBeans component architecture and introduces the NetBeans IDE.
- Chapter 23 covers applets.
- Chapter 24 covers the Java APIs for working with XML and XSLT, including XML Schema, validation, XPath, and XInclude, as well as XML binding with JAXB.
- Appendix A covers using the Eclipse IDE with the examples in this book.
- Appendix B describes BeanShell, a lightweight scripting language for Java developed by the authors of this book.

If you're like us, you don't read books from front to back. If you're really like us, you usually don't read the Preface at all. However, on the off chance that you will see this in time, here are a few suggestions:

- If you are an experienced programmer who has to learn Java in the next five minutes, you are probably looking for the examples. You might want to start by glancing at the tutorial in Chapter 2. If that doesn't float your boat, you should at least look at the information in Chapter 3, which explains how to use the compiler and interpreter, or Appendix A, which shows how to run the examples in the Eclipse IDE. This should get you started.
- Chapters 12 through 15 are essential if you are interested in writing advanced networked or web-based applications and services. This is one of the more interesting and important parts of Java.
- Chapters 16 through 22 discuss Java's graphics features and component architecture. You should read this if you are interested in writing graphical Java applications or applets.
- Chapter 24 covers the Java APIs for working with XML, including SAX, DOM, DTDs, XML Schema, and using XSL to render output for the Web. XML technology is becoming key to cross-platform development.

Online Resources

There are many online sources for information about Java. Oracle's official website for Java topics is *http://java.sun.com*; look here for the software, updates, and Java releases. This is where you'll find the JDK, which includes the compiler, the interpreter, and other tools.

You should also visit O'Reilly's Java site at *http://oreilly.com/java*. There you'll find information about other O'Reilly Java books, and a pointer to the home page for *Learning Java*, *http://oreil.ly/Java_4E*, where you'll find the source code examples for this book.

Conventions Used in This Book

The font conventions used in this book are quite simple.

Italic is used for:

- Unix pathnames, filenames, and program names
- Internet addresses, such as domain names and URLs
- New terms where they are defined
- Program names, compilers, interpreters, utilities, and commands
- Threads

`Constant width` is used for:

- Anything that might appear in a Java program, including method names, variable names, and class names
- Tags that might appear in an HTML or XML document
- Keywords, objects, and environment variables

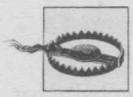
`Constant width bold` is used for:

- Text that is typed by the user on the command line

`Constant width italic` is used for:

- Replaceable items in code

This icon designates a note, which is an important aside to the nearby text.



This icon designates a warning relating to the nearby text.

In the main body of text, we always use a pair of empty parentheses after a method name to distinguish methods from variables and other creatures.

In the Java source listings, we follow the coding conventions most frequently used in the Java community. Class names begin with capital letters; variable and method names begin with lowercase. All the letters in the names of constants are capitalized. We don't use underscores to separate words in a long name; following common practice, we capitalize individual words (after the first) and run the words together. For example: `thisIsAVariable`, `thisIsAMethod()`, `ThisIsAClass`, and `THISISACONSTANT`. Also, note that we differentiate between static and nonstatic methods when we refer to them. Unlike some books, we never write `Foo.bar()` to mean the `bar()` method of `Foo` unless `bar()` is a static method (paralleling the Java syntax in that case).

Using Code Examples

This book is here to help you get your job done. In general, if this book includes code examples, you may use the code in your programs and documentation. You do not need to contact us for permission unless you're reproducing a significant portion of the code. For example, writing a program that uses several chunks of code from this book does not require permission. Selling or distributing a CD-ROM of examples from O'Reilly books does require permission. Answering a question by citing this book and quoting example code does not require permission. Incorporating a significant amount of example code from this book into your product's documentation does require permission.

We appreciate, but do not require, attribution. An attribution usually includes the title, author, publisher, and ISBN. For example: "*Learning Java*, Fourth Edition, by Patrick Niemeyer and Daniel Leuck. Copyright 2013 Patrick Niemeyer and Daniel Leuck, 978-1-449-31924-3."

If you feel your use of code examples falls outside fair use or the permission given above, feel free to contact us at *permissions@oreilly.com*.

Safari® Books Online

Safari Books Online (*www.safaribooksonline.com*) is an on-demand digital library that delivers expert content in both book and video form from the world's leading authors in technology and business.

Technology professionals, software developers, web designers, and business and creative professionals use Safari Books Online as their primary resource for research, problem solving, learning, and certification training.

Safari Books Online offers a range of product mixes and pricing programs for organizations, government agencies, and individuals. Subscribers have access to thousands of books, training videos, and prepublication manuscripts in one fully searchable database from publishers like O'Reilly Media, Prentice Hall Professional, Addison-Wesley Professional, Microsoft Press, Sams, Que, Peachpit Press, Focal Press, Cisco Press, John Wiley & Sons, Syngress, Morgan Kaufmann, IBM Redbooks, Packt, Adobe Press, FT Press, Apress, Manning, New Riders, McGraw-Hill, Jones & Bartlett, Course Technology, and dozens more. For more information about Safari Books Online, please visit us online.

How to Contact Us

Please address comments and questions concerning this book to the publisher:

O'Reilly Media, Inc.
1005 Gravenstein Highway North
Sebastopol, CA 95472
800-998-9938 (in the United States or Canada)
707-829-0515 (international or local)
707-829-0104 (fax)

We have a web page for this book, where we list errata, examples, and any additional information. You can access this page at *http://oreil.ly/Java_4E*.

To comment or ask technical questions about this book, send email to *bookquestions@oreilly.com*.

For more information about our books, courses, conferences, and news, see our website at *http://www.oreilly.com*.

Find us on Facebook: *http://facebook.com/oreilly*

Follow us on Twitter: *http://twitter.com/oreillymedia*

Watch us on YouTube: *http://www.youtube.com/oreillymedia*

Acknowledgments

Many people have contributed to putting this book together, both in its *Exploring Java* incarnation and in its current form as *Learning Java*. Foremost, we would like to thank Tim O'Reilly for giving us the opportunity to write this book. Thanks to Mike Loukides, the series editor, whose patience and experience helped us get started on this

journey. Thanks to Paula Ferguson and John Posner, who contributed their organizational and editing abilities at various times. And a special thanks to Deb Cameron, the tireless editor of this book, without whom the previous two editions might never have been finished and certainly wouldn't have resembled English. We could not have asked for a more skillful or responsive team of people with whom to work.

Speaking of borrowings, the original version of the glossary came from David Flanagan's book *Java in a Nutshell* (O'Reilly). We also borrowed several class hierarchy diagrams from David's book. These diagrams were based on similar diagrams by Charles L. Perkins.

Thanks also to Marc Wallace and Steven Burkett for reading the original work in progress and for the support of our friends at Washington University: Bryan O'Connor and Brian Gottlieb. Thanks also to Josh Peck, coauthor of the original book, *Exploring Java*. Thanks to all those who reviewed or answered questions: David Flanagan for generics; Henry Wong for the concurrency utilities; Jim Elliott, Marc Loy, and Brian Cole for Swing; Jack Shirazi for NIO; Tim Boudreau for NetBeans; Martin Aeschlimann, Jim Farley, and John Norman for Eclipse; Ed Howland for XML; and Ian Darwin for regular expressions. (Check out Ian's *Java Cookbook* [O'Reilly] for more examples.) Thanks also to Ray O'Leary, Mario Aquino, and Mark Volkmann for their reviews. And finally, thanks to my beautiful wife, Ellen Song, for putting up with me through all this work.

Table of Contents