HZ BOOKS

经 典 原 版 书 库

PEARSON
Prentice Hall

Java程序设计导论

（英文版 · 第5版）

Introduction to

JAVA™ PROGRAMMING

CORE VERSION

FIFTH EDITION

Y. DANIEL LIANG

（美） Y. Daniel Liang 著

机械工业出版社
China Machine Press

经 典 原 版 书 库

Java程序设计导论

（英文版 · 第5版）

Introduction to Java Programming, Core Version

(Fifth Edition)

（美） Y. Daniel Liang 著

机 械 工 业 出 版 社
China Machine Press

本书版权登记号：图字：01-2005-3624

图书在版编目（CIP）数据

Java程序设计导论（英文版·第5版）/（美）梁（Liang, Y. D.）著. －北京：机械工业出版社，2005.9

（经典原版书库）

书名原文：Introduction to Java Programming, Core Version, Fifth Edition

ISBN 7-111-15896-2

Ⅰ.J…　Ⅱ.梁…　Ⅲ.JAVA语言－程序设计－英文　Ⅳ.TP312

中国版本图书馆CIP数据核字（2004）第141025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：迟振春

北京京北制版厂印刷·新华书店北京发行所发行

2005年9月第1版第1次印刷

787mm × 1092mm 1/16 · 41.5印张

印数：0 001-3 000册

定价：65.00元

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，还揭橥了学术的源变，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、研究及度藏。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，有的还专程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们服务，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，我们的联系方法如下:

电子邮件：hzjsj@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

PREFACE

In the past seven years, five editions of *Introduction to Java Programming* have been published. Each new edition substantially improved the previous edition in clarity, content, presentation, examples, and exercises, thanks to comments and suggestions by instructors and students. The Fifth Edition is a gigantic leap forward. I invite you to take a close look and be the judge. I am constantly improving the book. Please continue to send me your comments and suggestions to help further improve it.

Custom Versions

The book is published in a comprehensive version of twenty-nine chapters and can also be printed in custom versions with substantial savings for students. The first sixteen chapters form the custom core. You can customize the book by adding new chapters to the custom core. The following diagram summarizes the materials in the comprehensive version.

Introduction to Java Programming, 5E, Comprehensive Version

Part I Fundamentals of Programming
- Chapter 1 Introduction to Computers, Programs, and Java
- Chapter 2 Primitive Data Types and Operations
- Chapter 3 Control Statements
- Chapter 4 Methods
- Chapter 5 Arrays

Part II Object-Oriented Programming
- Chapter 6 Objects and Classes
- Chapter 7 Strings
- Chapter 8 Inheritance and Polymorphism
- Chapter 9 Abstract Classes and Interfaces
- Chapter 10 Object-Oriented Modeling

Part III GUI Programming
- Chapter 11 Getting Started with GUI Programming
- Chapter 12 Event-Driven Programming
- Chapter 13 Creating User Interfaces
- Chapter 14 Applets, Images, and Audio

Part IV Exception Handling and IO
- Chapter 15 Exceptions and Assertions
- Chapter 16 Simple Input and Output

Custom Core

Part V Data Structures and Collections Framework
- Chapter 17 Object-Oriented Data Structures
- Chapter 18 Java Collections Framework

Part VI Threads and Internationalization
- Chapter 19 Multithreading
- Chapter 20 Internationalization

Part VII Advanced GUI Programming
- Chapter 21 JavaBeans, Bean Events, and MVC
- Chapter 22 Containers, Layout Managers, and Borders
- Chapter 23 Menus, Toolbars, Dialogs, and Internal Frames
- Chapter 24 Advanced Swing Components

Part VIII Web Programming
- Chapter 25 Java Database Programming
- Chapter 26 Servlets
- Chapter 27 JavaServer Pages

Part IX Distributed Computing
- Chapter 28 Networking
- Chapter 29 Remote Method Invocation

Appendixes

Please contact your Prentice Hall sales representative or your Pearson Custom Editor to order custom versions.

Teaching Strategies

There are three popular strategies in teaching Java. The first, known as *GUI-first*, is to mix Java applets and GUI programming with object-oriented programming concepts. The second, known as *object-first,* is to introduce object-oriented programming (OOP) from the start. The third strategy, known as *fundamentals-first,* is a step-by-step approach, first laying a sound foundation on programming concepts, control statements, methods, and arrays, then introducing object-oriented programming, and then moving on to graphical user interface (GUI), applets, and finally to exception handling, simple I/O, and other advanced subjects.

GUI-first

The GUI-first strategy, starting with GUI and applets, seems attractive, but requires substantial knowledge of object-oriented programming and a good understanding of the Java event-handling model; thus, students may never fully understand what they are doing.

object-first

The object-first strategy is based on the notion that objects should be introduced first because Java is an object-oriented programming language. This notion, however, overlooks the importance of the fundamental techniques required for writing programs in any programming language. Furthermore, this approach inevitably mixes static and instance variables and methods before students can fully understand classes and objects and use them to develop useful programs. Students are overwhelmed by having to master object-oriented programming and basic rules of programming simultaneously in the early stage of learning Java. This is a common source of frustration for first-year students learning object-oriented programming.

fundamentals-first

From my own experience, confirmed by the experiences of many colleagues, I have found that learning basic logic and fundamental programming techniques like loops is a struggle for most first-year students. *Students who cannot write code in procedural programming are not able to learn object-oriented programming.* A good introduction on primitive data types, control statements, methods, and arrays prepares students to learn object-oriented programming. Therefore, this text adopts the fundamentals-first strategy, proceeding at a steady pace through all the necessary and important basic concepts, then moving to object-oriented programming, and then to the use of the object-oriented approach to build interesting GUI applications and applets with exception handling, simple I/O, and advanced features. The fundamentals-first approach can reinforce object-oriented programming by first presenting the procedural solutions and demonstrating how they can be improved using the object-oriented approach. Students can learn when and how to apply OOP effectively.

problem solving

This book is not simply about how to program, for it teaches, as well, how to solve problems using programs. Applying the concept of abstraction in the design and implementation of software projects is the key to developing software. The overriding objective of the book, therefore, is to teach students to use many levels of abstraction in solving problems and to see problems in small and in large. *The examples and exercises throughout the book foster the concept of developing reusable components and using them to create practical projects.*

Learning Strategies

practice

A programming course is quite different from other courses. In a programming course, you learn from examples, from practice, and from mistakes. You need to devote a lot of time to writing programs, testing them, and fixing errors.

programmatic solution

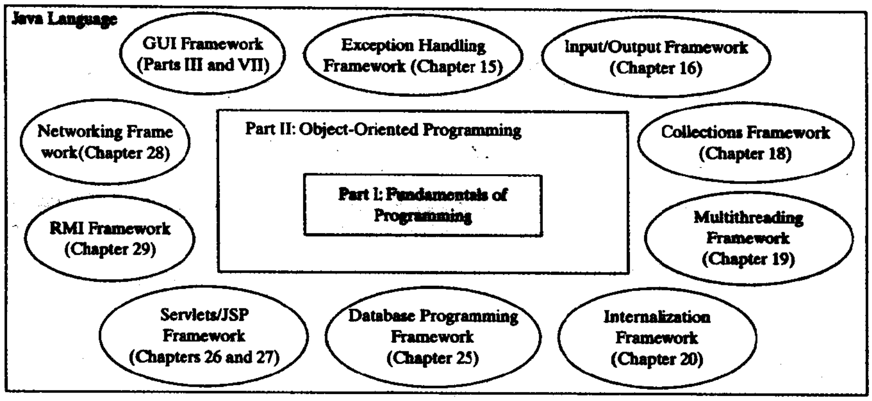
For first-time programmers, learning Java is like learning any high-level programming language. The fundamental point in learning programming is to develop the critical skills of formulating programmatic solutions for real problems and translating them into programs using selection statements, loops, and methods.

object-oriented programming

Once you acquire the basic skills of writing programs using loops, methods, and arrays, you can begin to learn object-oriented programming. You will learn how to develop object-oriented software using class encapsulation and class inheritance.

Java API

Once you understand the concept of object-oriented programming, learning Java becomes a matter of learning the Java API. The Java API establishes a framework for programmers to develop applications using Java. You have to use these classes and interfaces in the API and follow their conventions and rules to create applications. The best way to learn the Java API is to imitate examples and do exercises. The following diagram highlights the API covered in the book.


Java Language
GUI Framework (Parts III and VII)
Exception Handling Framework (Chapter 15)
Input/Output Framework (Chapter 16)
Networking Frame work(Chapter 28)
Part II: Object-Oriented Programming
Part I: Fundamentals of Programming
Collections Framework (Chapter 18)
RMI Framework (Chapter 29)
Multithreading Framework (Chapter 19)
Servlets/JSP Framework (Chapters 26 and 27)
Database Programming Framework (Chapter 25)
Internalization Framework (Chapter 20)

Pedagogical Features

The philosophy of the Liang Java Series is *teaching by example and learning by doing*. Basic features are explained by example so that you can learn by doing. The book uses the following elements to get the most from the material:

teaching by example
learning by doing

- **Objectives** list what students should have learned from the chapter. This will help them to determine whether they have met the objectives after completing the chapter.
- **Introduction** opens the discussion with a brief overview of what to expect from the chapter.
- **Examples,** carefully chosen and presented in an easy-to-follow style, teach programming concepts. Each example has a problem statement, solution steps, complete source code, sample run, and review.
- **Chapter Summary** reviews the important subjects that students should understand and remember. It helps them to reinforce the key concepts they have learned in the chapter.
- **Review Questions** are grouped by sections to help students track their progress and evaluate their learning.
- **Programming Exercises** are grouped by sections to provide students with opportunities to apply the skills on their own. The level of difficulty is rated easy (no asterisk), moderate (*), hard (**), or challenging (***). The trick of learning programming is practice, practice, and practice. To that end, the book provides a large number of exercises.
- **Interactive Self-Test** lets students test their knowledge interactively online. The Self-Test is accessible from the Companion Website. It provides more than nine hundred multiple-choice questions organized by sections in each chapter.
- **Notes, Tips,** and **Cautions** are inserted throughout the text to offer valuable advice and insight on important aspects of program development.

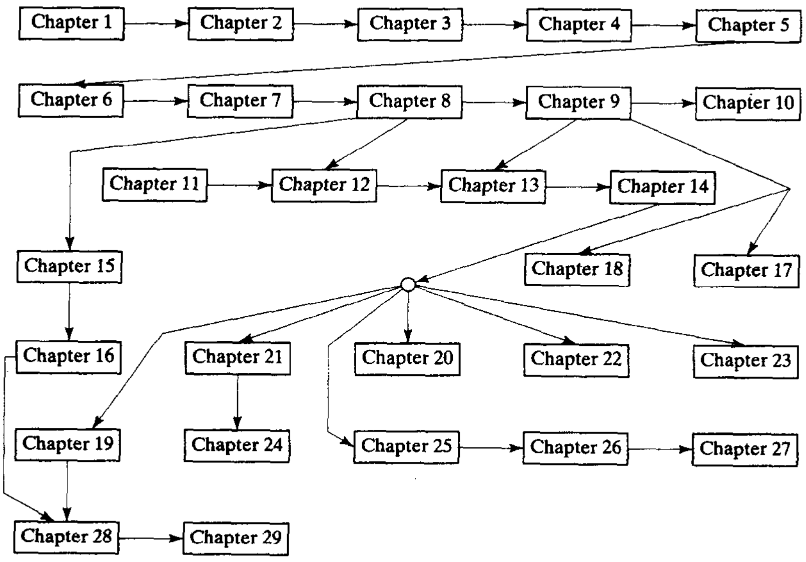
NOTE
Provides additional information on the subject and reinforces important concepts.

TIP
Teaches good programming style and practice.

CAUTION
Helps students steer away from the pitfalls of programming errors.

Flexible Chapter Orderings

The book provides flexible chapter orderings to enable GUI, IO, or Collections to be covered earlier. Many of the chapters after Chapter 14 can be covered immediately after Chapter 14. The following diagram shows the chapter dependencies.



NOTE
Some of the optional examples and exercises in a later chapter may be dependent on earlier chapters. In such cases the examples and exercises can be omitted. For example, Chapter 25 has an example that uses `JTable` from Chapter 24. If you have not covered `JTable`, this type of examples and excercises can be skipped.

What's New in This Edition?

This edition improves upon *Introduction to Java Programming, Fourth Edition*. The major changes are as follows:

- The book is completely revised in every detail to improve clarity, content, presentation, examples, and exercises.
- The book provides many new illustrations and uses short examples to demonstrate concepts and techniques. Longer examples are presented in case studies with overall discussions and thorough line-by-line explanations.

- Part I, "Fundamentals of Programming," focuses on problem-solving and basic programming techniques with many new illustrations and practical examples. This part uses `JOptionPane` input dialog to receive input, but console input using the `MyInput` class and the JDK 1.5 `Scanner` class are also introduced to provide alternative ways for input.
- Part II, "Object-Oriented Programming," is expanded into five chapters to give a comprehensive introduction on OOP and how to use it to design programs. New organization improves the presentation of object-oriented programming and enables GUI programming to be covered earlier.
- Part III, "GUI Programming," is expanded into four chapters to introduce GUI programming, event-driven programming, creating user interfaces, and applets. Advanced GUI features are now covered in Part VII, "Advanced GUI Programming."
- Chapter 16, "Simple Input and Output," is completely overhauled. It first introduces the `File` class, then text I/O, binary I/O, object I/O, and random access files. Short examples are used to demonstrate concepts and techniques. Three cases studies on using various I/O classes are presented in this chapter.
- The comprehensive version covers the Java collections framework, threads, JavaBeans, advanced GUI components, JDBC, Servlets, JSP, networking, and RMI.
- Purely mathematical examples, such as computing deviations and matrix multiplications, have been replaced by practical examples, such as computing loan payments, taxes, and printing payroll statements.
- The number of exercises is almost doubled to cover a variety of problems with simple or complex solutions. The level of difficulty is rated easy (no asterisk), moderate (*), hard (**), or challenging (***).
- The book is updated to JDK 1.5.

How are the New Features in JDK 1.5 Treated?

There are already more features in Java than an introductory course can cover. This edition does not aim to cover all the new features in JDK 1.5. Nevertheless, some of the useful features of JDK 1.5 are appropriately introduced to beginners. Specifically,

- Formatted output (`System.out.printf`) is covered in Chapter 2.
- The enhanced `for` loop is covered in Chapters 5 and 18.
- The `java.util.Scanner` class is covered in Chapter 2 and Supplement T for console input, and in Chapter 7 to complement the `StringTokenizer` class.
- Boxing and unboxing of primitives is covered in Chapter 9.
- Static import is covered in Chapter 11.
- Generic types are covered in Chapter 18 and Supplement Q.

To facilitate the use of this book in courses based on JDK 1.4 and to enable instructors to choose JDK 1.5 topics freely, all the sections on JDK 1.5 are marked *JDK 1.5 Features* and can be skipped.

NOTE
Sun MicroSystems recently renamed JDK 1.5 to JDK 5.0. Since most programmers are familiar with JDK 1.5, Sun uses JDK 5.0 and JDK 1.5 interchangeably. So does this book.

JDK 1.5 = JDK 5.0

Java Development Tools

You can use a text editor, such as the Windows Notepad or WordPad, to create Java programs, and compile and run the programs from the command window. You can also use a Java development tool, such as TextPad, JBuilder, NetBeans, or Eclipse. These tools support an integrated development environment (IDE) for rapidly developing Java programs. Editing, compiling, building, and executing programs are integrated in one graphical user interface. Using these tools effectively will greatly increase your programming productivity. TextPad is a primitive IDE tool. JBuilder, NetBeans, and Eclipse are more sophisticated. It may take a while to become familiar with a tool, but the time you invest will pay off in the long run. Tutorials on TextPad, JBuilder, NetBeans, and Eclipse are in the supplements on the Companion Website.

Companion Website

The Companion Website accessible from `www.prenhall.com/liang` contains the following resources:

- Interactive Self-Test
- Supplements
- Answers to review questions
- Solutions to even-numbered programming exercises
- Source code for the examples in the book
- Download links for JDK 1.5, JBuilder, NetBeans, Eclipse, TextPad, JCreator LE, JEdit, JGrasp, BlueJ, WinZip, MySQL, and Apache Tomcat.

Instructor Resource Website

The Instructor Resource Website accessible from `www.prenhall.com/liang` contains the following resources:

- Microsoft PowerPoint slides with interactive buttons to view full-color, syntax-highlighted source code and to run programs without leaving the slides.
- Sample exams. In general, each exam has four parts:
  1. Multiple-choice questions or short-answer questions (most of these are different from the ones in the Self-Test on the Companion Website)
  2. Correct programming errors
  3. Trace programs
  4. Write programs
- Solutions to all the exercises. Students will have access to the solutions of even-numbered exercises from the Companion Website.
- Quiz generator developed using Java.

Some readers have requested the materials in the Instructor Resource Website. Please understand that these are for instructors only. Such requests will not be answered.

Supplements

The text covers the core subjects. The supplements extend the text to introduce additional topics that might be of interest to readers. The following supplements are available from the Companion Website.

A. Installing and Configuring JDK 1.5
B. Compiling and Running Java from the Command Window
C. Compiling and Running Java from TextPad
D. Java Coding Style Guidelines
E. HTML Tutorial
F. Glossary
G. SQL statements for creating and initializing tables for Chapters 25, 26, 27, and 29
H. JBuilder Tutorial
I. NetBeans Tutorial
J. Eclipse Tutorial
K. Tutorial for MySQL
L. Tutorial for Oracle
M. Tutorial for Microsoft Access
N. Tutorial for Tomcat
O. Creating Shortcuts for Java Applications on Windows
P. Supplemental Case Study for Chapter 10: Design a `GenericMatrix` Class
Q. Creating Generic Types (JDK 1.5)
R. Enumerated Types (JDK 1.5)
S. Semaphores (JDK 1.5)
T. Obtaining Input from the Console Using the `Scanner` Class (JDK 1.5)

Acknowledgments

I would like to thank Ray Greenlaw, Chuck Shipley, and my colleagues at Armstrong Atlantic State University for enabling me to teach what I write and for supporting me in writing what I teach. Teaching is the source of inspiration for continuing to improve the book. I am grateful to the instructors and students who have offered comments, suggestions, bug reports, and praise. Their enthusiastic support has contributed to the success of my Java series.

This book was greatly improved thanks to outstanding reviews by James Chegwidden of Tarrant County College, Dan Lipsa of Armstrong Atlantic State University, Vladan Jovanovic of Georgia Southern University, Kenrick Mock of the University of Alaska—Anchorage, Ronald F. Taylor of Wright State University, and Lixin Tao of Pace University. Professors Mutsumi Nakamura and Lian Yu of Arizona State University are so kind to provide us with the errata and suggestions, which came just in time before the book is finished.

It is a great pleasure and privilege to work with the legendary computer science team at Prentice Hall. I would like to thank Alan Apt, Toni Holm, Patrick Lindner, Sarah Parker, Camille Trentacoste, John Keegan, Xiaohong Zhu, Pamela Hersperger, Barrie Reinhold, Toni Callum, Barbara Taylor-Laino, Meredith Maresca, and their colleagues for organizing, managing, and promoting this project, and to Robert Milch for copy editing.

During the production of this book, I was in Asia. I thank Pearson Education in Taiwan, Malaysia, and Singapore for their hospitality and valuable assistance.

As always, I am indebted to my wife, Samantha, for her love, support, and encouragement.

Y. Daniel Liang
liang@armstrong.edu
www.cs.armstrong.edu/liang/intro5e.html

Brief Contents

CONTENTS