HZ BOOKS
华章教育

经　典　原　版　书　库

PRENTICE
HALL

Java语言程序设计

基础篇

（英文版 · 第6版）

INTRODUCTION TO

JAVA™

PROGRAMMING

SIXTH EDITION

（美） Y. Daniel Liang 著

机械工业出版社
China Machine Press

经 典 原 版 书 库

Java语言程序设计

基础篇

（英文版·第6版）

Introduction to Java Programming

(Sixth 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：01-2006-6511

图书在版编目（CIP）数据

Java语言程序设计　基础篇（英文版 · 第6版）/（美）梁（Liang, Y. D.）著. —北京：机械工业出版社，2008.2

（经典原版书库）

书名原文：Introduction to Java Programming, Sixth Edition

ISBN 978-7-111-23367-1

Ⅰ. J…　Ⅱ. 梁…　Ⅲ. JAVA语言－程序设计－教材－英文　Ⅳ. TP312

中国版本图书馆CIP数据核字（2008）第009292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
2008年2月第 1 版第 1 次印刷
170mm × 242mm · 42印张
标准书号：ISBN 978-7-111-23367-1
定价：66.00元

凡购本书，如有倒页、脱页、缺页，由本社发行部调换
本社购书热线：(010) 68326294

PREFACE

Welcome to *Introduction to Java Programming, Sixth Edition.* This edition is a substantial improvement on the previous edition in respect to clarity, content, presentation, code listings, and exercises, thanks to comments and suggestions by instructors and students. Overall, it is a great leap forward. We invite you to take a close look and be the judge.

Versions

The book is available in two versions:

- The fundamentals first version (Chapters 1–19).[1]
- The comprehensive version (Chapters 1–36).[2]

The following diagram summarizes the contents of the comprehensive version:

Introduction to Java Programming, Sixth Edition, Comprehensive Version

Part 1 Fundamentals of Programming
Chapter 1 Introduction to Computers, Programs, and Java
Chapter 2 Primitive Data Types and Operations
Chapter 3 Selection Statements
Chapter 4 Loops
Chapter 5 Methods
Chapter 6 Arrays

Part 2 Object-Oriented Programming
Chapter 7 Objects and Classes
Chapter 8 Strings and Text I/O
Chapter 9 Inheritance and Polymorphism
Chapter 10 Abstract Classes and Interfaces
Chapter 11 Object-Oriented Design

Part 3 GUI Programming
Chapter 12 Getting Started with GUI Programming
Chapter 13 Graphics
Chapter 14 Event-Driven Programming
Chapter 15 Creating User Interfaces
Chapter 16 Applets and Multimedia

Part 4 Exception Handling, I/O, and Recursion
Chapter 17 Exceptions and Assertions
Chapter 18 Binary I/O
Chapter 19 Recursion

Part 5 Data Structures
Chapter 20 Lists, Stacks, Queues, Trees, and Heaps
Chapter 21 Generics
Chapter 22 Java Collections Framework
Chapter 23 Algorithm Efficiency and Sorting

Part 6 Concurrency, Networking, and Internationalization
Chapter 24 Multithreading
Chapter 25 Networking
Chapter 26 Internationalization

Part 7 Advanced GUI Programming
Chapter 27 JavaBeans and Bean Events
Chapter 28 Containers, Layout Managers, and Borders
Chapter 29 Menus, Toolbars, Dialogs, and Internal Frames
Chapter 30 MVC and Swing Models
Chapter 31 JTable and JTree

Part 8 Web Programming
Chapter 32 Java Database Programming
Chapter 33 Advanced Java Database Programming
Chapter 34 Servlets
Chapter 35 JavaServer Pages
Chapter 36 Remote Method Invocations

fundamentals first

Fundamentals First introduces the fundamentals of programming, problem-solving, object-oriented programming, and GUI programming. This version is suitable for an introductory course on problem-solving and object-oriented programming.

comprehensive version

The *Comprehensive Version* contains all the chapters in the fundamentals first version. Additionally, it covers data structures, networking, internationalization, advanced GUI programming, and Web programming.

[1] 本部分对应影印版的基础篇。——编辑注

[2] 原书完全版的影印版分为基础篇和进阶篇。本次出版的是基础篇（第1～19章）。——编辑注

Teaching Strategies

imperative and OOP

Both imperative and OOP are important programming paradigms with distinct advantages for certain applications. Some programs should be developed using the imperative approach and others are better developed using the object-oriented approach. Today's students need to know both paradigms and use them effectively. This book introduces both imperative and OOP paradigms. Students will learn when and how to apply these two paradigms effectively.

fundamentals first

There are several strategies in teaching Java. This book 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, and advanced features.

fundamental programming techniques

My own experience, confirmed by the experience of many colleagues, demonstrates that learning basic logic and *fundamental programming techniques* like loops and step-wise refinement is essential for new programmers to succeed. 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.

using OOP effectively

The fundamentals-first approach reinforces 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.

object-early failed?

object-right

At every SIGCSE (Computer Science Education) conference prior to 2005, the object-early approach was trumpeted and the voice for the fundamentals-first approach was muted. This changed when some former proponents of object-early began to air their frustrations and declared that object-early was a failure. This book is fundamentals-first and *object-right*. OOP is introduced right after fundamental programming techniques are covered. Many instructors of this book, from research universities to community colleges, have embraced the approach and have succeeded.

problem solving

Programming isn't just syntax, classes, or objects. It is really *problem solving*. Loops, methods, and arrays are fundamental techniques for problem solving. From fundamental programming techniques to object-oriented programming, there are many layers of abstraction. Classes are simply a layer of abstraction. 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 layers of abstraction in solving problems and to see problems in small and in large. The examples and exercises throughout the book center on problem solving and foster the concept of developing reusable methods and classes 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

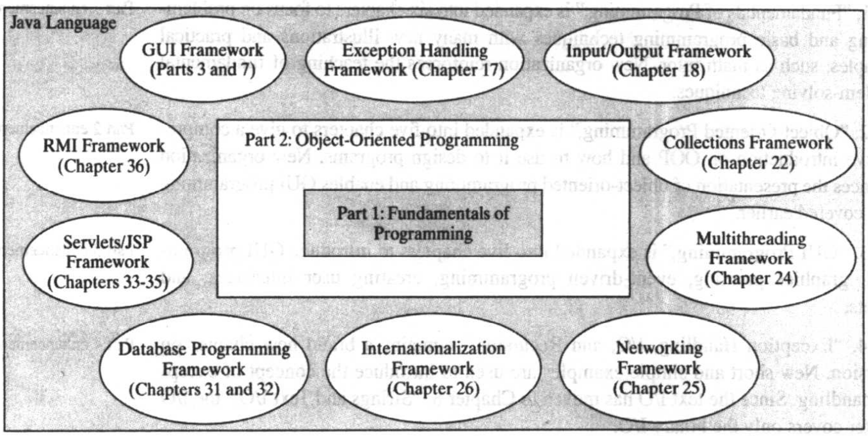
For first-time programmers, learning Java is like learning any high-level programming language. The fundamental point in learning programming is to develop the critical skills of formulating *programmatic solutions* for real problems and translating them into programs using selection statements, loops, 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 the 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 3 and 7)
Exception Handling Framework (Chapter 17)
Input/Output Framework (Chapter 18)
RMI Framework (Chapter 36)
Part 2: Object-Oriented Programming
Collections Framework (Chapter 22)
Part 1: Fundamentals of Programming
Servlets/JSP Framework (Chapters 33-35)
Multithreading Framework (Chapter 24)
Database Programming Framework (Chapters 31 and 32)
Internationalization Framework (Chapter 26)
Networking Framework (Chapter 25)

What's New in This Edition?

This edition substantially enhances *Introduction to Java Programming, Fifth Edition*. The major changes are as follows:

syntax coloring

- All the Java code in the book now uses syntax coloring for keywords, literals, and comments. This makes the code more readable and understandable.

full Java 5 update

- The Java 5 features are fully integrated in this edition. (Note: the fifth edition treats Java 5 features in separate optional sections.)

complete revision

- The book is completely revised in every detail to enhance clarity, content, presentation, and exercises. All code samples and listings have been shortened and revised to be more compelling and to keep students even more engaged.

GUI options in Part 2

- Optional GUI sections are provided at the end of Chapters 7–10 in Part 2, "Object-Oriented Programming," while a complete introduction of GUI programming is presented in Part 3, "GUI Programming," GUI components are excellent examples to demonstrate OOP. The optional GUI sections can be used as additional examples for OOP.

early interesting programs

- Students can now write short, interesting, graphical game programs starting from Chapter 2.

`JOptionPane` or `Scanner`

- Chapter 2, "Primitive Data Types and Operations," introduces the use of `JOptionPane` and `Scanner` to receive input. Students can use either `JOpationPane` or `Scanner` for obtaining input.

new recursion chapter

- The discussion of recursion is expanded to form Chapter 19, "Recursion," a brand-new chapter. It can be covered after Chapter 5, "Methods."

early text I/O

- High-level text I/O using the `Scanner` and `PrintWriter` classes is introduced in Chapter 8 along with strings. Binary I/O is covered in Chapter 18 "Binary I/O."

early `ArrayList`

- `ArrayList` is introduced early in Chapter 9, "Inheritance and Polymorphism."

UML exercises

- New exercises are provided in Part 2, "Object-Oriented Programming," for achieving three objectives: (1) design and draw UML for classes; (2) implement classes from the UML; (3) use classes to develop applications.

extensive supplements

- Extensive supplements (e.g., installing and configuring JDK, IDE tutorials, design patterns, rapid GUI development, database design, SQL) are provided for instructors to customize a course.

Part 1 enhancement

- Part 1, "Fundamentals of Programming," is expanded into six chapters to focus on problem-solving and basic programming techniques with many new illustrations and practical examples, such as math tutor. New organization reinforces the teaching of fundamental problem-solving techniques.

Part 2 enhancement

- Part 2, "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 enhances the presentation of object-oriented programming and enables GUI programming to be covered earlier.

Part 3 enhancement

- Part 3, "GUI Programming," is expanded into five chapters to introduce GUI programming, graphics painting, event-driven programming, creating user interfaces, and applets.

Part 4 enhancements

- Part 4, "Exception Handling, I/O, and Recursion," contains a brand new chapter on recursion. New short and simple examples are used to introduce the concept of exception handling. Since the text I/O has moved to Chapter 8, "Strings and Text I/O," the I/O chapter covers only the binary I/O.

Part 5 enhancement

- Part 5, "Data Structures," is expanded to cover data structure design and implementation (array list, linked list, stack, queue, heap, priority queue, and binary tree), generics, Java Collections Framework, and algorithm efficiencies and sorting.

Part 6 enhancement

- Part 6, "Concurrency, Networking, and Internationalization," is updated to cover Java 5 thread pooling, locks, and semaphores.

Part 7 enhancement

- Part 7, "Advanced GUI Programming," is expanded into five chapters with short, simple new examples to teach complex subjects. For example, a new example is used to demonstrate how to develop source components. The MVC architecture is introduced along with the Swing models.

Part 8 enhancement

- Part 8, "Web Programming," is expanded into five chapters. Advanced database programming is in a separate chapter and may be skipped. Several new examples are presented to introduce the JSP.

Pedagogical Features

teaching by example
learning by doing

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:

- **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.
- **Code Listings** are used to teach programming concepts. Syntax coloring makes the code easier to follow.
- **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.
- **Optional Sections** cover nonessential but valuable features. Instructors may choose to include or skip an optional section, or cover it later. The section headers of optional sections are marked by ❖.
- **Review Questions** are grouped by sections to help students track their progress and evaluate their learning.
- **Margin Notes** are featured throughout the book to emphasize key terms and important concepts.

- **Programming Exercises** are grouped by sections to provide students with opportunities to apply on their own the new skills they have learned. The level of difficulty is rated as easy (no asterisk), moderate (*), hard (**), or challenging (***). The trick of learning programming is practice, practice, and practice. To that end, the book provides a great many exercises.
- **Interactive Self-Test** lets students test their knowledge interactively online. The Self-Test is accessible from the Companion Website. It provides more than one thousand multiple-choice questions organized by sections in each chapter. The Instructor Resource Website contains the quiz generator with additional multiple-choice questions.
- **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, exception handling, generics, and the Java Collections Framework to be covered earlier. Three common alternative orderings are shown as follows:

GUI Early

Chapter 9	Inheritance and Polymorphism
Chapter 12	GUI Basics
Chapter 13	Graphics
Chapter 10	Abstract Classes and Interfaces
Chapter 14	Event-Driven Programming
Chapter 15	Creating User Interfaces
Chapter 16	Applets and Multimedia

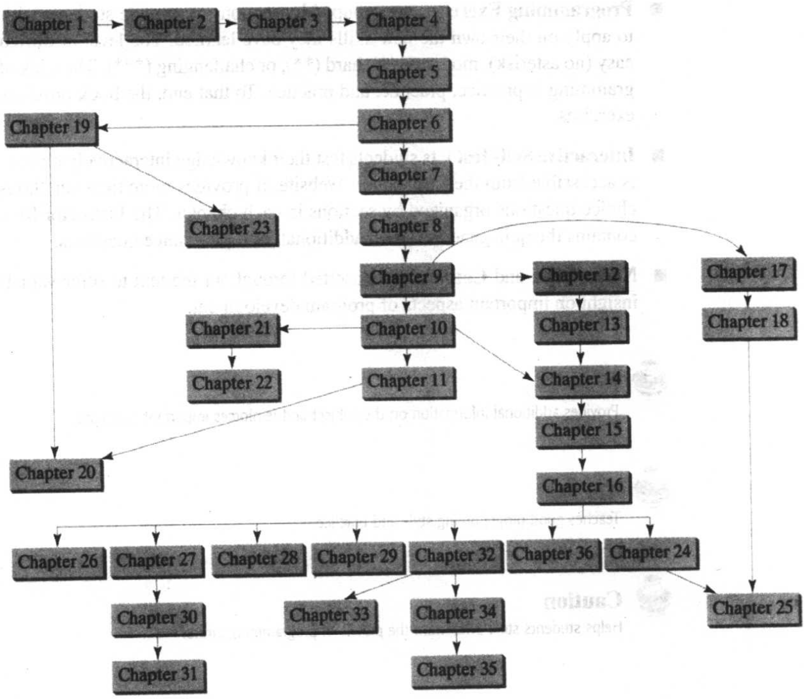
Exception Early

Chapter 9	Inheritance and Polymorphism
Chapter 17	Exceptions and Assertions

Data Structures Early

Chapter 9	Inheritance and Polymorphism
Chapter 10	Abstract Classes and Interfaces
Chapter 20	Lists, Stacks, Queues, Trees, and Heaps
Chapter 21	Generics
Chapter 22	Java Collections Framework

Many of the chapters after Chapter 16 can be covered in any order. The following diagram shows the chapter dependencies.



Note

Some of the optional examples and exercises in later chapters may be dependent on earlier chapters. In such cases the examples and exercises can be omitted. For example, Chapter 17 has an example that uses GUI components from Chapter 15. If you have not covered Chapter 15, these examples and exercises can be skipped.

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, executing, and debugging 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, but they are easy to use if you follow the tutorials. Tutorials on TextPad, JBuilder, NetBeans, and Eclipse will be found in the supplements on the Companion Website.

Companion Website

The Companion Website at www.prenhall.com/liang features a host of resources for students and instructors, including the following:

- Answers to review questions
- Solutions to programming exercises
- Source code for the code listings in the book

- Links to software and Web resources
- The **Online Quiz Generator** enables instructors to generate quizzes as well as assign students to take quizzes online with the results emailed to the instructor. This can be done in a closed lab or at home. Instructors may assign a set of 10, 15, or 20 questions and specify that the questions to be randomly generated. Go to www.prenhall.com/liang to access the Online Quiz.

Supplements

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

Instructor Resource Center

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

- Over 2000 Microsoft PowerPoint lecture slides. They bring the illustrations from the text to life in the classroom. Featuring full-color, syntax-highlighted source code and the ability to run programs live without leaving the slides and trace program execution using animation.
- Sample exams. In general, each exam has four parts:
  1. Multiple-choice questions or short-answer questions (most of these are different from the questions in the self-test on the Companion Website)
  2. Correct programming errors
  3. Trace programs
  4. Write programs
- Solutions to all the exercises (available to instructors only). Students will have access to the solutions of even-numbered exercises.
- UML Diagram Solutions to all the UML exercises (for instructors only). Students are provided solutions to even-numbered exercises.
- TestGen, a test bank of over 2000 programming questions.
- Online quiz. (Students can take the online quiz for each chapter and a quiz report will be sent to the instructor.)
- Ten sample exams, each with multiple choice or short-answer questions. They test how students correct programming errors, test programs, and write their own programs.
- Online Homework and Assessment. GOAL (Gradiance Online Accelerated Learning) is a sophisticated online homework tool that can be purchased and used alongside the textbook. Multiple choice problems based on the concept of a "root question" and challenging programming labs are designed to save instructors time grading programming assignments and to deliver meaningful feedback to students immediately. All code is tested against a real data set. Contact your local Pearson Prentice Hall sales representative to learn more about GOAL.

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

Acknowledgments

I would like to thank Ray Greenlaw 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.

This book was greatly enhanced thanks to outstanding reviews. The reviewers for the previous editions were:

Yang Ang	University of Wollongong (Australia)
David Champion	DeVry Institute
James Chegwidden	Tarrant County College
Harold Grossman	Clemson University
Ron Hofman	Red River College (Canada)
Hong Lin	DeVry Institute
Dan Lipsa	Armstrong Atlantic State University
Vladan Jovanovic	Georgia Southern University
Larry King	University of Texas at Dallas
Nana Kofi	Langara College (Canada)
Roger Kraft	Purdue University at Calumet
Debbie Masada	Sun Microsystems
Blayne Mayfield	Oklahoma State University
Michel Mitri	James Madison University
Kenrick Mock	University of Alaska Anchorage
Jun Ni	University of Iowa
Gavin Osborne	University of Sakatchewan
Kevin Parker	Idaho State University
Mary Ann Pumphrey	De Anza Junior College
Ronald F. Taylor	Wright State University
Carolyn Schauble	Colorado State University
David Scuse	University of Manitoba
Ashraf Shirani	San Jose State University
Daniel Spiegel	Kutztown University
Lixin Tao	Pace University
Russ Tront	Simon Fraser University
Deborah Trytten	University of Oklahoma
Kent Vidrine	George Washington University
Bahram Zartoshty	California State University at Northridge

The reviewers for this edition are:

Kevin Bierre	Rochester Institute of Technology
James Chegwidden	Tarrant County College
Charles Dierbach	Towson University
Deena Engel	New York University
Vladan Jovanovic	Georgia Southern University
Frank Malinowski	Aelera Corporation
John McGrath	J.P. McGrath Consulting
Shyamal Mitra	University of Texas at Austin
Benjamin Nystuen	University of Colorado at Colorado
Roger Priebe	University of Texas at Austin

It is a great pleasure, honor, and privilege to work with Prentice Hall. I would like to thank Marcia Horton, Tracy Dunkelberger, Robin O'Brien, Christianna Lee, Jennifer Cappello, Vince O'Brien, Camille Trentacoste, Craig Little, Xiaohong Zhu, and their colleagues for organizing, producing, and promoting this project, and Robert Milch for copy editing.

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/intro6e

CONTENTS

[⊖] 自本部分开始，后面部分是进阶篇内容。——编辑注