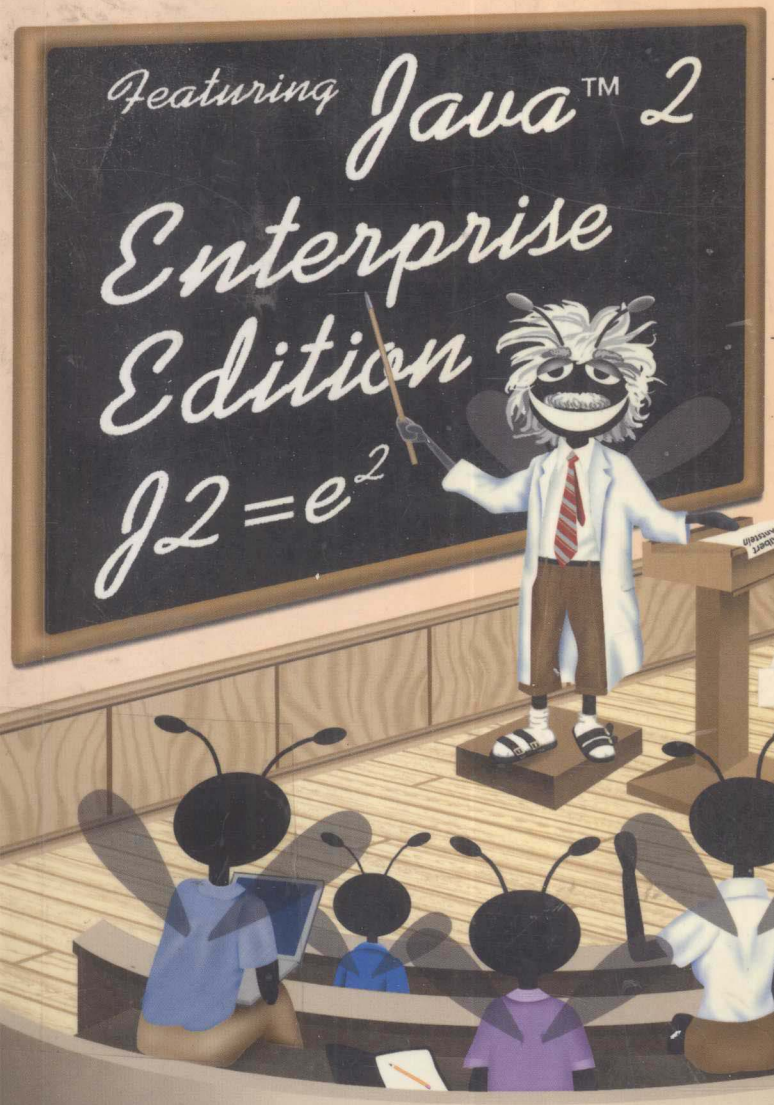
国外计算机科学教材系列

随书附带光盘中含有多种实用软件

高级 Java 2大学教程

英文版



Advanced Java 2 Platform How to Program

J2EE 企业级系统开发宝典

Harvey M. Deitel
[美] Paul J. Deitel 著
Sean E. Santry

DEITEL™
PEARSON Prentice Hall
PHEI
電子工業出版社
Publishing House of Electronics Industry
http://www.phei.com.cn

国外计算机科学教材系列

高级 Java 2 大学教程

（英文版）

Advanced Java 2 Platform How to Program

Harvey M. Deitel
[美] Paul J. Deitel 著
Sean E. Santry

電子工業出版社
Publishing House of Electronics Industry
北京 · BEIJING

内容简介

本书是一本高级Java 2编程方面的优秀教材，全面介绍了Java 2平台的多种常用及前沿技术。本书从高级GUI编程入手，讲解了Java 2D、Java 3D图形设计以及JavaBean组件模型；讨论了分布式编程，其中包括RMI、Jini、Jiro、JMX、CORBA以及JavaSpace的概念；介绍了有关网络服务的内容，并通过实例讲解了servlet和JSP的应用，然后列举了与其他Web服务相关的技术，例如WML、SOAP等。本书还讨论了构造企业级Java应用的关键技术，其中包括安全、JDBC、EJB等，并给出一个利用Java技术实现的网上书店。本书所附的光盘上含有书中用到的一些软件，并提供了全部的程序代码。全书内容丰富、结构严谨、条理清晰，写作方法别具一格，并且给出了大量的实例和练习，是一本难得的高级Java 2编程教材。

本书是高等院校进行中高级Java编程语言教学的教材，是软件设计人员进行企业级Java应用开发的宝贵参考资料，也适合所有想深入学习Java的读者使用。

版权贸易合同登记号　图字：01-2003-7707

图书在版编目（CIP）数据

高级Java 2大学教程 = Advanced Java 2 Platform How to Program/（美）戴特尔（Deitel, H. M.）等著.
-北京：电子工业出版社，2004.1
（国外计算机科学教材系列）
ISBN 7-5053-9562-9

Ⅰ.高… Ⅱ.戴… Ⅲ.JAVA语言-程序设计-高等学校-教材-英文 Ⅳ.TP312

中国版本图书馆CIP数据核字（2003）第124310号

责任编辑：冯小贝
印　　刷：北京兴华印刷厂
出版发行：电子工业出版社
　　北京市海淀区万寿路173信箱　邮编：100036
经　　销：各地新华书店
开　　本：787 × 980　1/16　　印张：92　字数：2061千字
印　　次：2004年1月第1次印刷
定　　价：128.00元（附光盘1张）

出 版 说 明

21世纪初的5至10年是我国国民经济和社会发展的重要时期，也是信息产业快速发展的关键时期。在我国加入WTO后的今天，培养一支适应国际化竞争的一流IT人才队伍是我国高等教育的重要任务之一。信息科学和技术方面人才的优劣与多寡，是我国面对国际竞争时成败的关键因素。

当前，正值我国高等教育特别是信息科学领域的教育调整、变革的重大时期，为使我国教育体制与国际化接轨，有条件的高等院校正在为某些信息学科和技术课程使用国外优秀教材和优秀原版教材，以使我国在计算机教学上尽快赶上国际先进水平。

电子工业出版社秉承多年来引进国外优秀图书的经验，翻译出版了“国外计算机科学教材系列”丛书，这套教材覆盖学科范围广、领域宽、层次多，既有本科专业课程教材，也有研究生课程教材，以适应不同院系、不同专业、不同层次的师生对教材的需求，广大师生可自由选择和自由组合使用。这些教材涉及的学科方向包括网络与通信、操作系统、计算机组织与结构、算法与数据结构、数据库与信息处理、编程语言、图形图像与多媒体、软件工程等。同时，我们也适当引进了一些优秀英文原版教材，本着翻译版本和英文原版并重的原则，对重点图书既提供英文原版又提供相应的翻译版本。

在图书选题上，我们大都选择国外著名出版公司出版的高校教材，如Pearson Education培生教育出版集团、麦格劳－希尔教育出版集团、麻省理工学院出版社、剑桥大学出版社等。撰写教材的许多作者都是蜚声世界的教授、学者，如道格拉斯·科默（Douglas E. Comer）、威廉·斯托林斯（William Stallings）、哈维·戴特尔（Harvey M. Deitel）、尤利斯·布莱克（Uyless Black）等。

为确保教材的选题质量和翻译质量，我们约请了清华大学、北京大学、北京航空航天大学、复旦大学、上海交通大学、南京大学、浙江大学、哈尔滨工业大学、华中科技大学、西安交通大学、国防科学技术大学、解放军理工大学等著名高校的教授和骨干教师参与了本系列教材的选题、翻译和审校工作。他们中既有讲授同类教材的骨干教师、博士，也有积累了几十年教学经验的老教授和博士生导师。

在该系列教材的选题、翻译和编辑加工过程中，为提高教材质量，我们做了大量细致的工作，包括对所选教材进行全面论证；选择编辑时力求达到专业对口；对排版、印制质量进行严格把关。对于英文教材中出现的错误，我们通过与作者联络和网上下载勘误表等方式，逐一进行了修订。

此外，我们还将与国外著名出版公司合作，提供一些教材的教学支持资料，希望能为授课老师提供帮助。今后，我们将继续加强与各高校教师的密切联系，为广大师生引进更多的国外优秀教材和参考书，为我国计算机科学教学体系与国际教学体系的接轨做出努力。

电子工业出版社

教材出版委员会

Preface

Live in fragments no longer. Only connect.
Edward Morgan Forster

Welcome to *Advanced Java 2 Platform How to Program* and the exciting world of advanced-programming concepts with the three major Java platforms—*Java™ 2 Enterprise Edition (J2EE)*, *Java 2 Standard Edition (J2SE)* and *Java 2 Micro Edition (J2ME)*. Little did we know when we attended the November 1995 Internet/World Wide Web conference in Boston what that session would yield—four editions of *Java How To Program* (the world's best-selling Java textbook), and now this book about Java software-development technologies for upper-level college courses and professional developers.

Before Java appeared, we were convinced that C++ would replace C as the dominant application-development language and systems-programming language for the next decade. However, the combination of the World Wide Web and Java now increases the prominence of the Internet in information-systems planning and implementation. Organizations want to integrate the Internet "seamlessly" into their information systems. Java is more appropriate than C++ for this purpose—as evidenced by Sun Microsystems' announcement in 2001 that over 96% of enterprise application servers support J2EE.

Advanced Java 2 Platform How to Program is the first book in our *Advanced How to Program* series. We discuss Java technologies that may be unfamiliar and challenging to the average Java programmer. We structured each chapter discussion to provide the reader with an introduction to leading-edge and complex Java technologies, rather than provide a detailed analysis of every nuance of each topic. In fact, each topic we present could be a 600–800 page book in itself.

We use a different approach with the examples in this book than that of programming examples in our previous books. We provide fewer programs, but these programs are more substantial and illustrate sophisticated coding practices. We integrate many technologies to create a book for developers that enables you to "go beyond" and experiment with the most

up-to-date technologies and most widely employed design concepts. What better way to learn than to work with actual technologies and code?

When determining the appropriate topics for this book, we read dozens of journals, reviewed the Sun Microsystems Web site and participated in numerous trade shows. We audited our material against the latest technologies presented at the JavaOne conference—the leading Java-developer conference sponsored by Sun Microsystems—and at other popular Java conferences. We also reviewed books on specialized Java topics. After this extensive research, we created an outline for this book and sent it for professional review by Java experts. We found so many topics we wanted to include that we wound up with over 1800 pages of material (several hundred of those pages appear as PDF documents on the CD that accompanies this book). We apologize if this is inconvenient, but the material and the number of topics are voluminous. We will most likely split the next edition into two volumes.

This book benefitted from an unusually large pool of excellent reviewers and the detailed documentation that Sun makes available on their Web site (**www.sun.com**). We were excited to have a number of reviewers from Sun and many other distinguished industry reviewers. We wanted experienced developers to review our code and discussions, so we could offer "expert advice" from people who actually work with the technologies in industry.

We are pleased to include a discussion of application servers in Chapter 21. The three most popular application server software products are BEA's *WebLogic,* IBM's *WebSphere* and Sun/Netscape's *iPlanet.* Originally, we had planned to include all three on the book's accompanying CD, but we have included only WebLogic and WebSphere. iPlanet was about to publish a new version as this book went to publication. By mutual agreement between iPlanet and Deitel & Associates, Inc., we decided not to include this software, but iPlanet provides a link to a site specific to this book—**www.iplanet.com/ias_deitel**—where readers can download the latest iPlanet software. We also include a discussion of how to deploy our case study on the iPlanet server. You can find this discussion on our Web site—**www.deitel.com**.

We moved four chapters from *Java How to Program, Third Edition*—RMI, Servlets, JavaBeans and JDBC—to *Advanced Java 2 Platform How to Program.* Prentice Hall has published a paperback supplement (ISBN: 0-13-074367-4) containing these four chapters for readers who have purchased *Java How to Program, Fourth Edition.*

The world of Java is growing so rapidly that *Advanced Java 2 Platform How to Program* and its companion text, *Java How to Program, Fourth Edition,* total 3400 pages! The books are so large that we had to put several chapters from each on the accompanying CDs. This creates tremendous challenges and opportunities for us as authors, for our publisher—Prentice Hall, for instructors, for students and for professionals. We hope you enjoy the results of these challenges as much as we have enjoyed the process of tackling them.

Features of *Advanced Java 2 Platform How to Program*

This book contains many features including:

- ***Full-Color Presentation.*** This book is in full color to enable readers to see sample outputs as they would appear on a color monitor. Also, we now syntax color all the Java code, as do many of today's Java integrated development environments and code editors. Our syntax-coloring conventions are as follows:

```
comments appear in green
keywords appear in dark blue
constants and literal values appear in light blue
JSP delimiters appear in red
all other code appears in black
```

- ***"Code Washing."*** This is our own term for the process we use to format the programs in the book with a carefully commented, open layout. The code is in full color and grouped into small, well-documented pieces. This greatly improves code readability—an especially important goal for us given that this book contains almost 40,000 lines of code.

- ***Advanced Graphical User Interface (GUI) Design.*** Starting with Chapter 2, we use advanced Java Swing features to create real-world Java components, including a Web-browser application with a multiple-document interface. In Chapter 3, we introduce the Model-View-Controller (MVC) architecture and its implementation in the Swing API. In Chapters 4 and 5, we create 2D graphics and 3D worlds. The Java 2D Drawing Application with Design Patterns Case Study in Chapter 5 presents a complex drawing program with which the user can create shapes in various colors and gradients. We are also pleased to add Java 3D coverage. One of the book's adopters said these chapters were ideal for a course in advanced GUI programming. (We wanted to include multimedia programming with the Java Media Framework, but instead we decided to include this material in the companion book, *Java How to Program, Fourth Edition.*)

- ***Enterprise Java and Our Enterprise Java Case Study.*** Developers use Java for building "heavy-duty" enterprise applications. Chapters 7–11, 14–16 and 21 explore the necessary components for implementing enterprise solutions—including security, database manipulation, servlets, JavaServer Pages, distributed transactions, message-oriented middleware and application servers. In Chapter 7, Security, we discuss secure communications and secure programming. Chapters 17–20 showcase an Enterprise Java Case Study that integrates many technologies, such as Enterprise JavaBeans, servlets, RMI-IIOP, XML, XSLT, XHTML, (and for wireless application development) WML and cHTML—into an online-bookstore application. The Deitel Bookstore demonstrates how to use the MVC architecture introduced in Chapter 3 to build enterprise applications. This bookstore uses technologies to provide support for almost any type of client, including cell phones, mobile devices and Web browsers. In this world of networks and wireless networks, business information must be delivered securely and reliably to the intended recipients.

- ***Distributed Systems.*** Enterprise applications are usually so complex that they run more efficiently when program components are distributed among different machines in organizations' networks. This book introduces several technologies for building distributed systems—Remote Method Invocation (RMI), Jini, JavaSpaces, Java Management Extensions (JMX), Jiro and Common Object Request Broker Architecture (CORBA). CORBA, controlled by the Object Management Group (OMG), is a mature distributed computing technology for integrating distributed components written in many disparate languages. Java was originally intended for networks of programmable devices—Jini assumes that technology role

now. JMX and Jiro are technologies specifically for network management (LANs, WANs, intranets, the Internet, extranets, etc.).

- ***Java 2 Micro Edition (J2ME) and Wireless Applications.*** It is estimated that by 2003, more people worldwide will access the Internet through wireless devices than through desktop computers. The Java platform for wireless devices with limited capabilities such as cell phones and personal digital assistants is Java 2 Micro Edition (J2ME). Chapter 12, Wireless Java-Based Applications Development and J2ME, contains a case study that sends content from a centralized data store to several wireless clients, including a J2ME client.
- ***Web Services.*** Web services are applications that expose public interfaces usable by other applications over the Web. The area of Web services builds on existing protocols, such as HTTP, and communicate with XML-based messages. Directory services enable clients to perform lookups to discover available Web services. The Simple Object Access Protocol (SOAP) uses XML to provide communication in many Web services. Many of the technologies in this book can be used to build Web services.
- ***Employing Design Patterns.*** The book's largest case studies—such as the Java 2D drawing program in Chapter 5, the three-tier servlet and JavaServer Pages case study in Chapter 11, the three-tier wireless application in Chapter 12 and the Deitel Bookstore Enterprise Case Study in Chapters 17–20—each contain thousands of lines of code. Larger systems, such as automated teller machines or air-traffic control systems, can contain hundreds of thousands, or even millions, of lines of code. Effective design is crucial to the proper construction of such complex systems. Over the past decade, the software engineering industry has made significant progress in the field of *design patterns*—proven architectures for constructing flexible and maintainable object-oriented software.[1] Using design patterns can substantially reduce the complexity of the design process. We used many design patterns when building the software in this book. Chapter 1 introduces design patterns, discusses why they are useful and lists those design patterns we use throughout this book
- ***XML.*** XML (Extensible Markup Language) use is exploding in the software-development industry and we use it pervasively throughout the text. As a platform-independent syntax for creating markup languages, XML's data portability integrates well with Java's portable applications and services. If you do not know XML, Appendices A–D of this book provide an introduction to XML. Appendices A and B introduce XML basics and DTDs, which define standard XML document structures. Appendix C introduces the Document Object Model (DOM) API for manipulating XML documents. Appendix D covers XSLT (Extensible Stylesheet Language Transformations—an XML vocabulary for transforming XML documents into other text-based documents.
- ***Peer-to-Peer Applications.*** Peer-to-peer (P2P) applications—such as instant messaging and document-sharing programs—have become extremely popular. Chap-

1. Gamma, Erich, Richard Helm, Ralph Johnson, and John Vlissides. *Design Patterns; Elements of Reusable Object-Oriented Software*. (Massachusetts: Addison-Wesley, 1995).

ter 28, Peer-to-Peer Applications and JXTA, introduces this architecture, in which each node performs both client and server duties. JXTA (short for the term "Juxtapose"), defines protocols for implementing peer-to-peer applications. This chapter includes two P2P application case studies—one written with Jini and RMI and the other written in multicast sockets and RMI. Both implement a P2P instant messaging application. We wanted a capstone example for Jini and decided this chapter should have it. The first case study is somewhat centralized—and therefore not a "true" P2P application (some developers think that Jini has too much overhead for a peer-to-peer application). We developed the second to demonstrate a lighter-weight, decentralized implementation.

- ***Appendix H, Career Opportunities.*** This appendix introduces career services on the Internet. We explore online career services from both the employer's and employee's perspectives. We suggest Web sites at which you can submit applications, search for jobs and review applicants (if you are interested in hiring someone). We also review services that build recruiting pages directly into e-businesses. One of our reviewers told us that he had just gone through a job search largely using the Internet and this chapter would have expanded his search dramatically.
- ***Appendix I, Unicode.*** This appendix overviews the *Unicode Standard.* As computer systems evolved worldwide, computer vendors developed numeric representations of character sets and special symbols for the local languages spoken in different countries. In some cases, different representations were developed for the same languages. Such disparate character sets made communication between computer systems difficult. Java supports the Unicode Standard (maintained by a non-profit organization called the *Unicode Consortium*), which defines a single character set with unique numeric values for characters and special symbols in most spoken languages. This appendix discusses the Unicode Standard, overviews the Unicode Consortium Web site (**`unicode.org`**) and shows a Java example that displays "Welcome" in many different languages.
- ***Bibliography and Resources.*** Chapters in this book contain bibliographies when appropriate and URLs that offer additional information about the technologies. We did this so those readers who would like to study a topic further could begin with the resources we found helpful when developing this book.

Some Notes to Instructors

A World of Object Orientation

When we wrote the first edition of *Java How to Program*, universities were still emphasizing procedural programming in languages like Pascal and C. The leading-edge courses were using object-oriented C++, but these courses were generally mixing a substantial amount of procedural programming with object-oriented programming—something that C++ lets you do, but Java does not. By the third edition of *Java How to Program*, many universities were switching from C++ to Java in their introductory curricula, and instructors were emphasizing a pure object-oriented programming approach. In parallel with this activity, the software engineering community was standardizing its approach to modeling ob-

ject-oriented systems with the UML, and the design-patterns movement was taking shape. This book takes a 100% object-oriented approach and emphasizes Java design patterns and adherence to Java idiom.

The prerequisite for this book is *Java How to Program, Fourth Edition* (or equivalent Java knowledge), which provides a solid foundation in Java programming. *Java How to Program, Fourth Edition* includes the following chapters and appendices, for a more detailed Table of Contents, visit **www.deitel.com**: Introduction to Computers, the Internet and the Web; Introduction to Java Applications; Introduction to Java Applets; Control Structures: Part 1; Control Structures: Part 2; Methods; Arrays; Object-Based Programming; Object-Oriented Programming; Strings and Characters; Graphics and Java 2D; Graphical User Interface Components: Part 1; Graphical User Interface Components: Part 2; Exception Handling; Multithreading; Files and Streams; Networking; Multimedia: Images, Animation, Audio and Video; Data Structures; Java Utilities Package and Bit Manipulation; Collections; Java Media Framework and Java Sound; Java Demos; Java Resources; Operator Precedence Chart; ASCII Character Set; Number Systems; Creating HTML Documentation with **javadoc**; Elevator Events and Listener Interfaces; Elevator Model; Elevator View; Career Opportunities; Unicode; Bibliography.

Students Like Java

Students are highly motivated by the fact that they are learning a leading-edge language (Java) and a leading-edge programming paradigm (object-oriented programming) for building entire systems. Java immediately gives them an advantage when they head into a world in which the Internet and the World Wide Web have a massive prominence and corporations need enterprise systems programmers. Students quickly discover that they can do great things with Java, so they are willing to put in the extra effort. Java helps programmers unleash their creativity. We see this in the Java and advanced Java courses Deitel & Associates, Inc. teaches.

Focus of the Book

Our goal was clear—produce an advanced Java textbook for higher-level university courses in computer programming for students with intermediate-level Java programming experience, and offer the depth and the rigorous treatment of theory and practice demanded by professionals. To meet these goals, we produced a book that challenges Java programmers. We present clear examples of advanced topics and often overlooked topics. We adhere to Java idiom and follow sophisticated coding style and practices (i.e., not just the code formatting, but the idiomatic use of Java API's, constructs and technologies). This book presents substantial Java applications that readers can use to start working with these technologies immediately.

***Evolution of* Advanced Java 2 Platform How to Program**

Advanced Java 2 Platform How to Program was finished fresh on the heels of *Java How to Program, Fourth Edition*. Hundreds of thousands of university students and professionals worldwide have learned Java from our texts. Upon publication in September 2001, *Advanced Java 2 Platform How to Program* will be used in universities, corporations and government organizations worldwide. Deitel & Associates, Inc. taught Java courses internationally to thousands of students as we were writing the various editions of *Java How to*

Program and *Advanced Java 2 Platform How to Program.* We carefully monitored the effectiveness of material and tuned the books accordingly.

Conceptualization of Java
We believe in Java. Its conceptualization by Sun Microsystems, the creator of Java, was brilliant. Sun based the new language on C and C++, two of the world's most widely used implementation languages. This immediately gave Java a huge pool of highly skilled programmers who were implementing most of the world's new operating systems, communications systems, database systems, personal-computer applications and systems software. Sun removed the more complex and error-prone C/C++ features (such as explicit pointers, operator overloading and multiple inheritance, among others). They kept the language concise by removing special-purpose features used by only small segments of the programming community. They made the language truly portable for implementing Internet-based and Web-based applications, and they included features developers need such as strings, graphics, GUI components, exception handling, multithreading, multimedia (audio, images, animation and video), prepackaged data structures, file processing, database processing, Internet and Web-based client/server networking, distributed computing and enterprise computing. Then they made the language available *at no charge* to millions of potential programmers worldwide.

2.5 Million Java Developers
Java was promoted in 1995 as a means of adding "dynamic content" to Web pages. Instead of Web pages with only text and static graphics, Web pages could now "come alive" with audios, videos, animations, interactivity—and soon, 3D imaging. But we saw much more in Java than this. Java's features are precisely what businesses and organizations need to meet today's information-processing requirements. So we immediately viewed Java as having the potential to become one of the world's key general-purpose programming languages. In fact, Java has revolutionized software development with multimedia-intensive, platform-independent, object-oriented code for conventional, Internet-, Intranet- and Extranet-based applications and applets. Java now has 2.5 million developers worldwide—a stunning accomplishment when considering that it has been available publicly for only six years. No other programming language has ever acquired such a large developer base so quickly.

Teaching Approach

Advanced Java 2 Platform How to Program, First Edition contains a rich collection of examples, exercises and projects drawn from many fields to provide readers with a chance to solve interesting real-world problems. The book concentrates on the principles of good software engineering and stresses program clarity, especially important when creating substantial programs like those covered in this book. We avoid arcane terminology and syntax specifications in favor of teaching by example. Our code examples have been tested on popular Java platforms. We are educators who teach edge-of-the-practice topics in industry classrooms worldwide. The text emphasizes good pedagogy.

Learning Java via the live-code™ Approach
The book is loaded with live-code™ examples. This is how we teach and write about programming, and is the focus of each of our multimedia Cyber Classrooms and Web-based

training courses. We present each new concept in the context of a complete, working Java program, immediately followed by screen captures that show the program's output. We call this style of teaching and writing our ***live-code™ approach***. *We use the language to teach the language.* Reading these programs (almost 40,000 lines of code) is much like entering and running them on a computer.

Java Programming from Chapter Two
Advanced Java 2 Platform How to Program, "jumps right in" with substantial programs right from Chapter 2. This is the beginning of an aggressive pace that challenges readers with graphical, multithreaded, database-intensive, network-based programming. Throughout the book, readers learn by implementing impressive projects.

World Wide Web Access
All the code for *Advanced Java 2 Platform How to Program* is on the CD that accompanies this book. The code also is available at the following Web sites:

```
www.deitel.com
www.prenhall.com/deitel
```

Objectives
Each chapter begins with *Objectives* that inform the reader what to expect and provides an opportunity, after reading the chapter, to determine if the reader has met these objectives. It is a confidence builder and a source of positive reinforcement.

Quotations
The learning objectives are followed by quotations. Some are humorous, some are philosophical and some offer interesting insights. Our readers enjoy relating the quotations to the chapter material. The quotations are worth a "second look" after you read each chapter.

Outline
The chapter outline helps the reader approach the material in top-down fashion. This, too, helps students anticipate what is to come and set a comfortable and effective learning pace.

Almost 40,000 Lines of Code in 126 Example Programs (with Program Outputs)
We present Java features in the context of complete, working Java programs. The programs in this book are substantial, with hundreds to thousands of lines of code (e.g., 10,000 lines of code for the bookstore case study example). Students should use the program code from the CD that accompanies the book and run each program while studying that program in the text.

841 Illustrations/Figures
Many of the figures are code examples, but this book still offers many charts, line drawings and program outputs. For example, Chapter 4 and 5, Graphics Programming with Java 2D and Java 3D, provides stunning graphics, and the architectural overview of the Enterprise Java case study in Chapter 17 is impressive.

235 Programming Tips

We have included programming tips to help students focus on important aspects of program development. We highlight numerous tips in the form of *Good Programming Practices*, *Common Programming Errors*, *Testing and Debugging Tips*, *Performance Tips*, *Portability Tips*, *Software Engineering Observations* and *Look-and-Feel Observations*. These tips and practices represent the best we have gleaned from decades of programming and teaching experience. One of our students—a mathematics major—told us that she feels this approach is like the highlighting of axioms, theorems and corollaries in mathematics books; it provides a basis on which to build good software.

Good Programming Practices

We highlight Good Programming Practices *techniques for writing programs that are clearer, more understandable, more debuggable and more maintainable.*

Common Programming Errors

Focusing on these Common Programming Errors *helps readers avoid making the same errors.*

Testing and Debugging Tips

When we first designed this "tip type," we thought we would use it strictly to tell people how to test and debug Java programs. In fact, many of the tips describe aspects of Java that reduce the likelihood of "bugs" and thus simplify the testing and debugging process.

Performance Tips

We have included 13 Performance Tips *that highlight opportunities for improving program performance—making programs run faster or minimizing the amount of memory that they occupy.*

Portability Tips

One of Java's "claims to fame" is "universal" portability, so some programmers assume that if they implement an application in Java, the application will automatically be "perfectly" portable across all Java platforms. Unfortunately, this is not always the case. We include Portability Tips *to help readers write portable code and to provide insights on how Java achieves its high degree of portability.*

Software Engineering Observations

The object-oriented programming paradigm requires a complete rethinking about the way we build software systems. Java is an effective language for performing good software engineering. The Software Engineering Observations *highlight architectural and design issues that affect the construction of software systems, especially large-scale systems.*

Look-and-Feel Observations

We provide Look-and-Feel Observations *to highlight graphical user interface conventions. These observations help readers design their own graphical user interfaces in conformance with industry norms.*

Summary (949 Summary bullets)

Each chapter ends with additional pedagogical devices. We present a thorough, bullet-list-style summary of the chapter. On average, there are 26 summary bullets per chapter. This helps the readers review and reinforce key concepts.

Terminology (1904 Terms)
We include in a *Terminology* section an alphabetized list of the important terms defined in the chapter—again, further reinforcement. On average, there are 51 terms per chapter.

394 Self-Review Exercises and Answers (Count Includes Separate Parts)
Self-review exercises and answers are included for self-study. These reinforce the knowledge the reader gained from the chapter.

189 Exercises (Count Includes Separate Parts)
Each chapter concludes with a set of exercises. The exercises cover many areas. This enables instructors to tailor their courses to the unique needs of their audiences and to vary course assignments each semester. Instructors can use these exercises to form homework assignments, quizzes and examinations. The solutions for most of the exercises are included on the *Instructor's Manual* CD that is *available only to instructors* through their Prentice-Hall representatives. [***NOTE*: Please do not write to us requesting the instructor's manual. Distribution of this publication is strictly limited to college professors teaching from the book. Instructors may obtain the Instructor's manual only from their Prentice Hall representatives. We regret that we cannot provide the solutions to professionals.**] Solutions to approximately half of the exercises are included on the *Advanced Java 2 Platform Multimedia Cyber Classroom* CD, which also is part of *The Complete Advanced Java 2 Platform Training Course*. For ordering instructions, please see the last few pages of this book or visit **`www.deitel.com`**.

Approximately 3,080 Index Entries (with approximately 4648 Page References)
This book includes an extensive index. This helps the reader find any term or concept by keyword. The index is useful to developers who use the book as a reference. The terms in the Terminology sections generally appear in the index (along with many more index items from each chapter).

"Double Indexing" of Java live-code™ Examples and Exercises
Advanced Java 2 Platform How to Program has 126 live-code™ examples and 189 exercises (including parts). Many exercises are challenging problems or projects that require substantial effort. We have "double indexed" the live-code™ examples. For every Java source-code program in the book, we took the file name with the **`.java`** extension, such as **`WebBrowser.java`** and indexed it both alphabetically (in this case under "W") and as a subindex item under "Examples." This makes it easier to find examples using particular features.

Software Included with *Advanced Java 2 Platform How to Program*

There are a number of for-sale Java products available. However, you do not need them to get started with Java. We wrote *Advanced Java 2 Platform How to Program* using the *Java 2 Software Development Kit (J2SDK) Standard Edition Version 1.3.1 for Windows and Linux (Intel x86)* and other software programs that we include on the CD that accompanies this book. For your convenience, Sun's J2SDK also can be downloaded from the Sun Microsystems Java Web site **`java.sun.com/j2se`**. We include some of the most popular

server software so you can set up and run live systems. This software includes *BEA WebLogic Server™, Version 6.0 (Windows/Linux) with Service Pack 2, 30-Day Trial, Enterprise Edition, 6.0, Testdrive*; *IBM® WebSphere® Application Server, Advanced Single Server Edition, Version 4.0 for Windows NT® and Windows® 2000 Evaluation Copy,* and *Apache Tomcat 3.2.3 from the Apache Software Foundation.* We also include Informix Software's *Cloudscape 3.6.4* database software. With Sun's cooperation, we also were able to include on the CD a powerful Java integrated development environment (IDE)—Sun Microsystem's *Forte for Java Community Edition. Forte* is a professional IDE written in Java that includes a graphical user interface designer, code editor, compiler, visual debugger and more. J2SDK 1.3.1 must be installed before installing *Forte*. If you have any questions about using this software, please read the introductory *Forte* documentation on the CD. We will provide additional information on our Web site **www.deitel.com**.

The CD also contains the book's examples and a Web page with links to the Deitel & Associates, Inc. Web site (**www.deitel.com**), the Prentice Hall Web site (**www.prenhall.com/deitel**) and the many Web sites listed at the end of each chapter. If you have access to the Internet, this Web page can be loaded into your Web browser to give you quick access to all the resources. Finally, because we wrote much more than we originally intended, a number of chapters and appendices have been off-loaded to the CD.

Ancillary Package for *Advanced Java 2 Platform How to Program*

Advanced Java 2 Platform How to Program has extensive ancillary materials for instructors teaching from the book. The Instructor's Manual CD contains solutions to the vast majority of the end-of-chapter exercises. We also provide PowerPoint® slides containing all the code and figures in the text. You are free to customize these slides to meet your own classroom needs. Prentice Hall provides a *Companion Web Site* (**www.prenhall.com/deitel**) that includes resources for instructors and students. For instructors, the Web site has a *Syllabus Manager* for course planning, links to the PowerPoint slides and reference materials from the appendices of the book (such as the character sets and Web resources). For students, the Web site provides chapter objectives, true/false exercises with instant feedback, chapter highlights and reference materials. **[*NOTE*: Please do not write to us requesting the instructor's manual. Distribution of this publication is strictly limited to college professors teaching from the book. Instructors may obtain the solutions manual only from their regular Prentice Hall representatives. We regret that we cannot provide the solutions to professionals.]**

Advanced Java 2 Platform Multimedia Cyber Classroom (CD and Web-Based Training Versions) and *The Complete Advanced Java 2 Platform Training Course*

We have prepared an interactive, CD-based, software version of *Advanced Java 2 Platform How to Program,* called the *Advanced Java 2 Platform Multimedia Cyber Classroom.* It is loaded with features for learning and reference. The *Cyber Classroom* is wrapped with the textbook at a discount in *The Complete Advanced Java 2 Platform Training Course.* If you already have the book and would like to purchase the *Advanced Java 2 Platform Multimedia Cyber Classroom* (ISBN: 0-13-091276-x) separately, please visit **www.infor-**

mit.com/cyberclassrooms. All Deitel *Cyber Classrooms* are generally available in CD and Web-based training formats.

The CD has an introduction with the authors overviewing the *Cyber Classroom*'s features. Many of the live-code™ examples in the textbook truly "come alive" in the *Cyber Classroom*. If you are viewing a program and want to execute it, you click the lightning bolt icon and the program will run. You will immediately see—and hear for the audio-based multimedia programs—the program's outputs. If you want to modify a program and see and hear the effects of your changes, simply click the floppy-disk icon that causes the source code to be "lifted off" the CD and "dropped into" one of your own directories so you can edit the text, recompile the program and try out your new version. Click the audio icon and one of the authors will talk about the program and "walk you through" the code.

The *Cyber Classroom* also provides navigational aids including extensive hyperlinking. The *Cyber Classroom* is browser based, so it remembers recent sections you have visited and allows you to move forward or backward among these sections. The thousands of index entries are hyperlinked to their text occurrences. You can search for a term using the "find" feature and the *Cyber Classroom* locates its occurrences throughout the text. The Table of Contents entries are "hot"—so clicking a chapter name takes you to that chapter.

Students tell us that they particularly like the fact that solutions to about half the exercises in the book are included with the *Cyber Classroom*. Studying and running these extra programs is a great way for students to enhance their learning experience.

Students and professional users of our *Cyber Classrooms* tell us they like the interactivity and that the *Cyber Classroom* is an effective reference because of the extensive hyperlinking and other navigational features. We received an email from a person who said that he lives "in the boonies" and cannot take a live course at a university, so the *Cyber Classroom* was the solution to his educational needs.

Professors tell us that their students enjoy using the *Cyber Classroom*, spend more time on the course and master more of the material than in textbook-only courses. We have published (and will be publishing) many other *Cyber Classroom* and *Complete Training Course* products. For a complete list of the available and forthcoming *Cyber Classrooms* and *Complete Training Courses*, see the *Deitel™ Series* page at the beginning of this book or the product listing and ordering information at the end of this book. You can also visit **www.deitel.com** or **www.prenhall.com/deitel** for more information.

Acknowledgments

One of the great pleasures of writing a textbook is acknowledging the efforts of the many people whose names may not appear on the cover, but whose hard work, cooperation, friendship and understanding were crucial to the production of the book.

Several people at Deitel & Associates, Inc. devoted long hours to this project. We would like to acknowledge the efforts of our full-time Deitel & Associates, Inc. colleagues Jonathan Gadzik, Tem Nieto, Su Zhang, Kyle Lomeli, Matthew Kowalewski, Rashmi Jayaprakash, Kate Steinbuhler, Abbey Deitel and Betsy DuWaldt.

- Jonathan Gadzik, a graduate of the Columbia University School of Engineering and Applied Science (BS in Computer Science) co-authored Chapter 1, Introduction, and Chapter 12, Java-Based Wireless Applications Development and J2ME, and contributed to Chapter 4 and the design patterns material throughout the book. He also reviewed Chapter 28, Peer-to-Peer Applications.