Broadview
www.broadview.com.cn

Java技术大系

一样的JDK，不一样的深入解析！

深入浅出 JDK 6.0

涂传滨 著

光盘包含：
本书所有程序代码

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

深入浅出JDK 6.0

涂传滨 著

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

内 容 简 介

本书从 Java 语言的特点、语法、开发环境入手，深入讲解如何利用 JDK 提供的 API 进行应用开发，如输入输出、线程、图形界面、数据库存取、XML 等，并对 Java 语言的高级开发技巧，如分布式计算、本地调用、Annotation、反射、动态代理、JMX 等进行详细的阐述。本书基于最新的 JDK 6.0，对该版本新引入的特性：JDBC 4.0、内嵌 Apache Derby 数据库、动态语言支持、本地化桌面支持、增强的 XML 解析引擎等，均开辟专题予以介绍。并对 Java 相关技术和常用软件工具包进行详细讲解，包括：Java 客户端开发技术 SWT 和 JavaFX，数据库存取框架 Hibernate 和 iBatis，Groovy，以及 Java 与 Ruby、Python、PHP 等动态语言的结合，日志和日程工具包等。

本书既可供 Java 语言的初学者作为入门书籍使用，也可供具有一定开发经验的人员作为进阶材料阅读。

图书在版编目（CIP）数据

深入浅出 JDK6.0 / 涂传滨著. —北京：电子工业出版社，2008.1
（Java 技术大系）
ISBN 978-7-121-05503-4

I. 深… II. 涂… III. JAVA 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字（2007）第 185049 号

责任编辑：李　冰
印　　刷：北京智力达印刷有限公司
装　　订：三河市金马印装有限公司
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
开　　本：787×1092　1/16　印张：44　字数：989 千字
印　　次：2008 年 1 月第 1 次印刷
印　　数：5000 册　定价：79.80 元（含光盘 1 张）

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线：（010）88258888。

前　言

当博文视点的毕宁老师邀请笔者写一本关于 JDK 6.0 方面的技术书籍时，没有多加思索便答应了下来，当时认为凭借对 Java 语言的熟悉程度，要完成这样一本 Java 基础书籍是不难的。不料当真正进入到写作状态时才发现，离开了终日倚重的集成开发环境和企业服务容器的支持，要把 Java 变量存储、垃圾回收机制、线程、输入输出等基础概念阐述清楚，远比笔者之前参与的 JBuilder 和 WebSphere 题材难得多！

以 Java 语言为开发工具的程序员群体组成了国内最大规模的技术社区。但是笔者发现，在日益增多的 Java 程序员中，流行着一种错误的观点：学习 Java 语言重在掌握一些常用的开源框架——这和当年的“掌握了 EJB 就是 Java 高手”的观点是多么的相似！于是，出现了一种奇怪的现象：从事了多年 Java 应用开发的程序员，却不知道“堆”和“栈”的区别、不理解对象序列化的真实目的、不懂得如何编写线程安全的程序代码…… 他们眼中仿佛只有实用 API 和一些只知其一、不知其二的开源框架。这类程序员虽然能够完成日常的开发任务，但是却不能开发出具备高可用性的应用系统，当应用系统出现故障时的诊断和恢复能力也不足。

本书特色

市面上介绍 Java 开发的技术书籍琳琅满目。与其他书籍相比，本书有以下特色。

❑ 与 Java 教材比较

各类培训机构使用的 Java 教材重在介绍 Java 语法和基础 API。本书仅花费了一章的篇幅来介绍 Java 语法，因为今日的面向对象语言在语法方面实际上已经趋同，而且语法从来不是学习 Java 语言的难点。本书重在阐述 Java 程序的运行原理和高级开发技巧。

❑ 与介绍开源框架和工具包的书籍比较

开源框架和工具包在 Java 应用开发领域正发挥着越来越关键的作用，虽然视它们为 Java 开发的全部是错误的，但是忽视它们的地位同样是不对的。实际上，各类开源框架和工具包的涌现与 Java 阵营的策略方向是吻合的——面对大量语法越来越简单、入门门槛越来越低的新型语言的挑战，Java 没有大幅度改变自身语言结构，而是把突围的重点放在增强工具的易用性方面：近年来，Java 集成开发环境发生了突飞猛进的变化、由全世界聪明的程序员贡献的开源框架和工具包迅速增多。开源框架和工具包不仅大幅度提升了 Java 应用开发的产能，彻底改善了 Java 企业开发烦琐、冗长的过程，而且正在为 JDK 注入新鲜血液：新加入 JDK 的并发工具包（java.util.concurrent 包）、XML 与 Java 对象的映射引擎 JAXB 等都源自开源工具包。

当然，本书不是专门介绍开源框架和工具包的书籍，而是对它们采取为我所用、拓宽视野，进而达到让读者加深理解的目的。例如，在阐述输入输出的过程中涉及的 HttpUnit、J-FTP 等；在阐述数据库存取的过程中涉及的数据库存取框架 Hibernate 和 iBatis；嵌入式

数据库 BerkeleyDB 和 StelsEngine；在阐述动态代理的过程中涉及的 Hibernate 拦截器和 Spring AOP；在阐述 Java 动态编程的过程中涉及的 Javassist 和 Spring 方法注入；在阐述 XML 解析、编辑、转换、校验、映射的过程中涉及的多种工具包。此外，本书还独立开辟了一章篇幅集中介绍常用工具包，包括 Quartz、XFire、Velocity 等。

❑ 与介绍 JavaEE 的书籍比较

JavaEE 与 JavaSE 的界线正在日渐模糊，比如 Web Services 开发在传统上属于 JavaEE 的范畴，但是 JavaSE 已经提供了通过 Annotation 快速开发 Web Services 的能力。在处理与 JavaEE 的关系方面，本书采取不拘一格的态度：如果有助于读者理解知识点的话，则适度地引入一些 JavaEE 的知识。例如，在阐述基于 HTTP 协议的开发过程中，为了帮助理解 Web 站点如何保存用户状态，本书对 Web 容器工作原理进行了简单介绍。

本书的读者对象

本书涉猎广泛，适合以下读者群体：

❑ Java 语言的初学者

从全日制院校毕业的计算机相关专业的学生，以及在各类培训机构接受过 Java 语言培训的学员，因为缺少实际的项目经验，急需寻找有别于课堂上使用的 Java 基础教材的技术书籍。本书作者拥有多年的一线开发和设计经验，深谙日常练习题和高可用的复杂系统之间的天壤之别。初学者们可以从提炼自实际项目的大量开发专题中汲取丰富的营养。

❑ 谋求进阶的 Java 程序员

如上所述，大量的程序员对诸如 Java 内存堆栈、对象序列化、线程安全等概念存在模糊认识，通过阅读本书，将能在短期内迅速澄清。更为难能可贵的是，本书所介绍的 Java 本地调用、Java 并行开发、Java 动态编程、JMX 资源管理容器、嵌入式数据库等章节均属于 Java 技术书籍中的“罕见”内容。例如，围绕 IoC 和 AOP 出现了大量的介绍 Spring 框架的技术书籍，但是唯独本书指导读者们如何自己动手实现一个动态代理框架。

❑ 需要与异质语言互操作的 Java 程序员，以及需要与 Java 语言互操作的其他程序员

笔者日常使用的语言不限一种，深知异质语言、异构平台互操作需求的普遍性。通过本书，读者们不仅能找到 Java 语言与 Ruby、PHP、Python、Groovy 等动态语言互操作的途径，而且还能在 Linux 平台上体验 Java 调用原生的 C 程序，在 Windows 平台上体验 Java 调用 Delphi 程序，以及通过 JCOM 调用 COM 组件的无限乐趣。

❑ 需要补充新知、了解动态、开拓视野的软件开发从业人员

为了迎接 Web 2.0 时代的到来，JDK 在编程的动态性和桌面 GUI 方面适时予以增强。另一方面，围绕着 Java 基础 API，近年来涌现出了大量的开源框架和工具包。这些新知识对于 Java 程序员来说，既是对产能提升的机遇，也是对能力提升的挑战。所幸的是，本书的读者将能收获大量的 JDK 之外的知识。

本书内容

章　名	核心内容
第 1 章 Java 基础	本章的目的在于帮助读者建立对 Java 语言的基本认识。首先从 Java 语言的特点谈起，然后分析 Java 语言的运行环境和，最后阐述 Java 语言诸多特点中最本质、最重要的特点——对象性
第 2 章 Java 进阶	本章内容可以分为 3 部分：Java 内存管理机制（包括垃圾收集、对象引用类型、变量存储和复制的原理）、Java 语法进阶（包括异常、集合、泛型、修饰等）、实用工具（包括代码注释和 Javadoc 工具、类包管理工具 jar 命令、定制 JVM 启动参数、Java 程序调试工具、Java 进程监控平台、JVM 内存分析工具）
第 3 章 第一个 Java 程序	本章首先下载、安装、设置 Java 开发工具——JDK 6.0，接下来动手在新建立的环境中编写我们的第一个 Java 应用程序
第 4 章 文件系统	本章开篇先鸟瞰式地介绍 Java 输入输出的体系结构，然后进入利用输入输出 API 操作文件系统的正题，分为顺序存取和随机存取两种操作方式，最后利用一些实用例程帮助读者们进一步熟悉 Java 文件及目录对象的常用方法和属性
第 5 章 网络通信基础	Java 作为一种高级语言，从 OSI 参考模型之传输层开始向程序员提供编程接口，本章围绕这些核心编程接口向读者们介绍如何开发基础的网络通信程序
第 6 章 输入输出综合	本章首先回顾 Java 输入输出 API，包括字节流、字符流和对象流。接下来对对象序列化、HTTP 开发、大文件传输、字符集问题展开专题讨论。最后是一些输入输出相关的经典范例
第 7 章 Java 线程	本章首先介绍线程理论，然后介绍多线程编程的基础技能：创建和启动线程、管理线程的状态、线程组，接着介绍如何利用管道在线程之间实现传输信息。最后利用本章介绍的多线程技术开发一个类似网络蚂蚁的多线程下载程序
第 8 章 并发情况下的多线程编程	本章内容包括：分析线程安全问题的现象和本质，并寻求各种现象的解决方案；介绍现代网络服务器经常采用的线程池工作模式，并带领读者们开发基本的线程池；并发工具包在开发线程池方面带来的便利
第 9 章 Java 安全	本章一方面阐述安全技术的基石，包括：消息摘要、对称和不对称加密、数字签名和数字证书；另一方面指出应用系统安全的目的是信息存储安全、信息传输安全、访问控制（访问控制又可以分为身份认证和权限管理）。向读者们揭示如何在 Java 安全框架下利用基础安全技术实现应用系统的安全需求
第 10 章 图形界面开发	本章首先介绍利用 java. awt. Graphics 进行基础绘图，然后系统阐述 AWT 和 Swing 开发的基础知识，包括控件、布局管理器和事件模型。接下来在 JBuilder 中利用可视化环境实际开发 Java 图形界面应用程序。最后介绍 JDK 6.0 的桌面 API 增强
第 11 章 Applet、JavaWebStart、SWT 与 JavaFX	本章内容分为 Applet、JavaWebStart、SWT 和 JavaFX 共 4 节，其中的重点在于 Applet 和 SWT，而 JavaWebStart 是一项借助于浏览器的应用程序发布途径、JavaFX 目前尚未进入成熟阶段
第 12 章 数据库存取	本章首先介绍 JDBC 开发的基础知识，包括 JDBC 驱动程序、增删改查数据等基础开发、元数据、JDBC 异常。然后分成数据集二次处理、RowId、动态游标、预编译执行计划、批处理、操作大二进制数据、事务共 7 个专题详细地介绍 JDBC 开发的技巧。最后向读者们展示了一种有别于数据库的持久化途径——Preferences API，以及在 Java 命名与目录服务（JNDI）的名字空间中绑定数据源对象

续表

章　名	核心内容
第 13 章 开源数据库产品	本章内容包括两部分：开源的数据库存取框架（包括 Hibernate 和 iBatis）和嵌入式数据库（包括文件数据库 BerkeleyDB 和内存数据库 StelsEngine）
第 14 章 分布式计算	本章先鸟瞰式地介绍可供选择的分布式组件技术的概貌。然后分别以专门篇幅介绍 RMI 和 CORBA。其中 RMI 是 Java 领域私有的分布式组件技术，而 CORBA 的难度更大，所以在 CORBA 部分花费的篇幅更多
第 15 章 Java 本地调用	本章首先介绍 JNI 原理，然后分别介绍两种 JNI 的应用场景：Java 语言调用 C 程序和 Java 语言调用 Delphi 程序
第 16 章 反射机制及其应用	本章先介绍 Java 反射机制，读者们将在熟悉和理解反射机制的基础上，继续深入学习建立在反射机制之上的高级应用，包括 Annotation、动态代理和 java. beans 包的使用
第 17 章 Java 动态编程	本章内容分为 5 个方面： 1．在程序中利用 Compiler API 编译源文件 2．基于 Instrumentation，获得整体转换类字节码的能力 3．基于 Javassist，深入到方法级别转换类字节码 4．Spring 在替换类方法方面的能力 5．使用 ClassEditor 达到在缺少源文件支持的情况下，修改二进制类文件的目的
第 18 章 与动态语言的结合	本章介绍 Java 语言与 JavaScript、Ruby、PHP、Python、Groovy 语言的互操作。最后介绍 JDK 6. 0 内嵌的 HTTP Server
第 19 章 JMX 资源管理容器	本章首先介绍 JMX 的基本概念和技术架构，然后重点讲解 MBean 的 4 种类型，尤其是标准 MBean 和动态 MBean，最后介绍利用 JDMK 开发包为 JMX 生成管理界面，以及 JMX 的通知服务
第 20 章 XML 开发	本章介绍的是对 XML 势必涉及的基本操作：解析、编辑、转换、校验、映射（数据绑定）。XML 的应用是如此的广泛，各种语言和平台都对它提供了完备的支持，这也就导致了一种现象：往往可以找到多种方案来实现一种操作。本章本着开拓视野的原则，引入了较多的 XML 开发类库
第 21 章 常用工具包	本章将要介绍的工具包分为 4 类：AOP 工具（包括日志和日程工具）、特殊格式文件存取工具（包括 ZIP 和 Excel 格式）、Java 在 Windows 平台上的优化（包括将 Java 程序封装成 Windows NT Service、与 COM 组件互操作）、开发辅助类工具（包括生成 Web Services、模板引擎、性能监视工具）

本书看点

在软件开发技术发展的历史上，一种处于领导地位的主流技术，一旦显露出一点点的进展缓慢，就有可能出现一种新概念或是技术试图取而代之。Java 语言在步入中年之后，也正面临着前所未有的挑战。最表层的原因是，经过了十余年的快速发展，在 Java 的成熟度日渐提高的同时，基于 Java 技术的失败项目也在逐渐增多。其中的原因有立项时技术选型的失误，即把 Java 用于其所不擅长的领域，也不排除 Java 软件开发工程师在设计和实现能力方面的不足。

Java 语言所面临的挑战更多地来自于其他技术阵营，这突出地体现为两点：动态语言的兴起和客户端地位的提升。

❑ 动态语言的兴起

人们已经习惯于将应用开发划分为 Java 和. NET 两大阵营，却往往容易忽视动态语言作为另一股力量一直在顽强成长，这股力量在两大强权的夹缝中求生存，拥有不容忽视的拥护力量和成功案例。今日的应用系统普遍信奉“以善变为荣、以僵化为耻”的信条，强调应用系统的功能和结构快速响应用户需求的变化。系统架构师们苦苦纠缠于面向构件、面向服务等思想以期待实现快速业务生成之时，动态语言则另辟蹊径，从编程语言入手，提供给我们一套快速上手、快速成型、快速响应变化的神兵利器。

Java 语言应对挑战的策略，一是在语言中主动加入动态编程的特征，如 CompilerAPI、Instrumentation，二是热烈拥抱动态语言，我们欣喜地发现，Java 语言与 JavaScript、Ruby、PHP、Python 等动态语言几乎都存在结合产品，甚至诞生了一种完全建立在 Java 语言基础之上的动态语言——Groovy。关于动态编程和 Java 与动态语言的结合，在本书中均可以找到相应的章节。

❑ 客户端地位的提升

随着 Web 2. 0 概念的一夜窜红，包括浏览器和桌面两大主题的客户端应用开发重新站在了各大软件厂商争夺的焦点位置。不仅互联网的新型应用模式犹如雨后春笋，而且也急速地改变着用户的使用体验，最终用户已经不再被动地接受基于纯文档传递手段的瘦客户端界面。

而 Java 语言对图形界面开发的支持一直是饱受诟病的软肋。桌面领域的开发有其自身与众不同的规律：最终用户对于美观和快速的追求，远胜于对于程序可移植性的认同。本地化桌面增强、SWT 和 JavaFX 是 Java 阵营响应挑战的三大举措，读者们可以在本书中找到相关内容。

很喜欢这句格言：山不辞土，故能成其高；海不辞水，故能成其深。在书中笔者曾提出一个观点：人类永远不会认为自己的计算机太快了。同理，程序员永远不能认为自己的知识太多了，因为对核心技术保持深度理解、对技术动态保持敏锐感知是软件开发从业人员的基本素质。在漫漫征途上，笔者愿意和大家共相勉励。限于水平，本书难免存在错误和不当之处，欢迎反馈您的宝贵意见至：t-c-b@163. com。

本书包含一系列富有启发意义的精彩例程，如多线程下载、文件传输、文件切割机等，希望读者们能从中受益。在随书光盘中，按照章节组织例程源代码和搭建运行环境所需的软件。

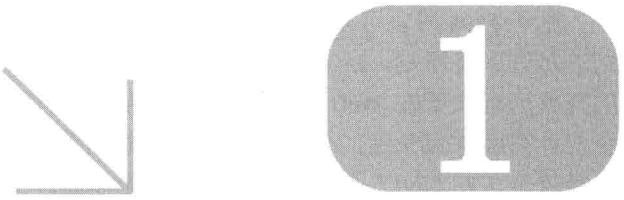
祝大家阅读愉快！

涂传滨

2007 年 10 月 13 日于厦门

目 录

第1章 Java基础



在经历了以大型主机为代表的集中计算模式和以PC为代表的单机计算模式之后，互联网的出现使得计算模式进入了网络计算时代。

网络计算模式的一个特点是计算机是异构的，即计算机的类型和操作系统是不一样的。例如，Sun工作站的硬件是Sparc体系、操作系统是Solaris，而PC的硬件是Intel体系、操作系统是Windows或者Linux。而相应的编程语言基本上只适用于某种特定平台。

网络计算模式的另一个特点是代码可以通过网络在各种计算机上移植。这就迫切需要一种跨平台的编程语言，用它编写的程序能够在网络中的各种计算机上正常运行，Java就是在这种需求下应运而生的。

正是因为Java语言符合了互联网时代的发展趋势，才使它获得了巨大的成功。

本章主要内容包括：

- 认识Java
- Java语法
- Java语言的对象性

本章的目的在于帮助读者建立对 Java 语言的基本认识。首先从 Java 语言的特点谈起，然后分析 Java 语言的运行环境。和一切编程语言一样，掌握和精通 Java 需要从熟悉其语法开始，最后阐述 Java 语言诸多特点中最本质、最重要的特点——对象性。

1.1 认识 Java

Java 是由 Sun 公司开发的一种编程语言。使用它可在各种不同机器、不同操作系统的网络环境中开发软件。Java 已经成为网络应用的主要开发语言。它彻底改变了应用软件的开发模式，带来了自 PC 以来的又一次技术革命。

基于 Java 语言可以在服务器、客户机和嵌入式设备上开发应用程序。其中需要特别提到的是在服务器端，Java 语言已经成为事实上的开发标准。甚至可以说，Java 在服务器端的成就远远大于在 Web 浏览器和桌面等用户终端领域。其中的主要原因在于 Java 语言对企业计算所重点关注的事务、安全、扩展性等方面的良好支持，同时得益于主要的竞争对手微软公司在 UNIX 和 Linux 平台上的无作为。

1.1.1 Java 语言的特点

Java 语言拥有以下显著的特点。

- 面向对象：现实世界中的任何实体都可以看做是对象，对象之间通过消息相互作用。另一方面，现实世界中任何实体都可归属于某类事物，任何对象都是某一类事物的实例。如果说传统的过程式编程语言是以过程为中心、以算法为驱动的话，面向对象的编程语言则是以对象为中心、以消息为驱动。所有面向对象编程语言都支持 3 个概念：封装、多态和继承。
- 可移植性：这是最初 Java 被关注的首要原因。所谓可移植性，就是在 A 系统上开发的程序经过一次编译后可以移植到 B 系统上运行，只要简单的复制即可，而不需要再次编译。
- 安全性：在任何一台机器上运行 Java 程序，必须拥有相应的权限。Java 安全模型是沙箱模型，自 Java 语言诞生之初发展到今天，目前的 Java 语言采用基于策略（Policy）的沙箱模型，可以根据使用者的需求来定义策略，形成灵活的安全解决方案。Java 从设计之初便考虑到了安全性，因此 Java 的安全性是在语言层次实现的。Java 的安全性由下列 3 个方面保证：语言特性（包括数组的边界检查、类型转换、取消指针型变量）、资源访问控制（包括本地文件系统访问、连接访问网络等）、代码数字签名（通过数字签名来确认代码源以及代码是否完整）。
- 并发性：Java 语言支持多线程技术，也就是允许多个线程并行执行，基于线程技术模拟多任务并发执行。多线程是 Java 语言的一个重要特征，Java 把线程融合进入语言本身的结构中——这是一个非常明智的战略决策，因为如果一方面要求应用程序启用多线程机制，一方面又强调应用程序的可移植性的话，对于开发人员来说是一个非常棘手的难题。基于 C/C++等传统语言虽然可以开发出多线程应用程序，但是每种操作系统在线程机制的运用和线程的处理方面存在相当的差异。