HZ BOOKS
华章IT

资深Java专家10年经验总结，全程案例式讲解，首本全面介绍Java多线程编程技术的专著

结合大量实例，全面讲解Java多线程编程中的并发访问、线程间通信、锁等最难突破的核心技术与应用实践

Java
核心技术
系列

Java
多线程编程核心技术

Java Multi-thread Programming

高洪岩 著

机械工业出版社
China Machine Press

Java
核心技术
系列

Java
多线程编程核心技术

Java Multi-thread Programming

高洪岩 著

机械工业出版社
China Machine Press

图书在版编目（CIP）数据

Java 多线程编程核心技术 / 高洪岩著 . —北京：机械工业出版社，2015.6
（Java 核心技术系列）

ISBN 978-7-111-50206-7

I. J… II. 高… III. JAVA 语言 – 程序设计 IV. TP312

中国版本图书馆 CIP 数据核字（2015）第 098874 号

Java 多线程编程核心技术

出版发行：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：高婧雅 责任校对：董纪丽
印 刷：三河市宏图印务有限公司 版 次：2015 年 6 月第 1 版第 1 次印刷
开 本：186mm × 240mm 1/16 印 张：19.75
书 号：ISBN 978-7-111-50206-7 定 价：69.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换
客服热线：（010）88379426 88361066 投稿热线：（010）88379604
购书热线：（010）68326294 88379649 68995259 读者信箱：hzit@hzbook.com

华章IT
HZBOOKS | Information Technology

Preface 前　言

为什么要写这本书

早在几年前笔者就曾想过整理一份与 Java 多线程有关的稿件，因为市面上所有的 Java 书籍都是以一章或两章的篇幅介绍多线程技术，并没有完整地覆盖该技术的知识点，但可惜，苦于当时的时间及精力有限，一直没有达成所愿。

也许是注定的安排，我目前所在的单位是集技术与教育为一体的软件类企业。我在工作中发现很多学员在学习完 JavaSE/JavaEE 之后想对更深入的技术进行探索，比如在对大数据、分布式、高并发类的专题进行攻克时，立即遇到针对 java.lang 包中 Thread 类的学习，但 Thread 类的学习并不像 JDBC 那样简单，学习多线程会遇到太多的问题、弯路以及我们所谓的“坑”，为了带领学员在技术层面上进行更高的追求，我将多线程的技术点以教案的方式进行整理，在课堂上与同学们一起学习、交流，同学们反响也非常热烈。此至，若干年前的心愿终于了却，学员们也很期待这本书能出版发行，因为这样他们就有了真正的纸质参考资料，其他爱好 Java 多线程的朋友们也在期盼本书的出版。本书能促进他们相互交流与学习，这就是我最大的心愿。

本书秉承大道至简的主导思想，只介绍 Java 多线程开发中最值得关注的内容，希望能抛砖引玉，以个人的一些想法和见解，为读者拓展出更深入、更全面的思路。

本书特色

在本书写作的过程中，我尽量减少“啰嗦”的文字语言，全部用案例来讲解技术点的实现，使读者看到代码及运行结果后就可以知道此项目要解决的是什么问题，类似于网络中的博客风格，可让读者用最短的时间学完相关知识点，明白这些知识点是如何应用的，以及在

使用时要避免什么。本书就像“瑞士军刀”一样，精短小，但却非常锋利，可帮读者快速学习知识并解决问题。

读者对象

本书适合所有 Java 程序员阅读，尤其适合以下读者：

- Java 多线程开发者
- Java 并发开发者
- 系统架构师
- 大数据开发者
- 其他对多线程技术感兴趣的人员

如何阅读本书

在整理本书时，我一直本着实用、易懂的原则，最终整理出 7 章：

第 1 章讲解了 Java 多线程的基础，包括 Thread 类的核心 API 的使用。

第 2 章讲解了在多线程中对并发访问的控制，主要就是 synchronized 的使用，由于此关键字在使用上非常灵活，所以书中用了很多案例来介绍此关键字的使用，为读者学习同步相关内容打好坚实的基础。

第 3 章介绍线程并不是孤独的，它们之间要通信，要交互。本章主要介绍 wait()、notifyAll() 和 notify() 方法的使用，使线程间能互相通信，合作完成任务。本章还介绍了 ThreadLocal 类的使用。学习完本章，读者就能在 Thread 多线程中进行数据的传递了。

第 4 章讲解了 synchronized 关键字，它使用起来比较麻烦，所以在 Java 5 中提供了 Lock 对象，以求能更好地实现并发访问时的同步处理，包括读写锁等相关技术点。

第 5 章讲解了 Timer 定时器类，其内部实现就是使用的多线程技术。定时器的计划任务执行是很重要的技术点，包括在 Android 开发时都会有深入的使用，所以会为读者详细讲解。

第 6 章讲解的单例模式虽然很简单，但如果遇到多线程将会变得非常麻烦，如何在多线程中解决这么棘手的问题呢？本章将全面介绍解决方案。

第 7 章，在整理稿件的过程中肯定会出现一些技术知识点的空缺，前面被遗漏的技术案例将在本章进行补充，以帮助读者形成完整的多线程的知识体系。编写本章的目的就是尽量使本书不存在技术空白点。

勘误和支持

由于我的水平有限，编写时间仓促，书中难免会出现一些错误或者不准确的地方，恳请读者批评指正，让我与大家一起，在技术之路上互勉共进。我的邮箱是 279377921@qq.com，期待能够得到你们的真挚反馈。本书的源代码可以在华章网站（www.hzbook.com）下载。

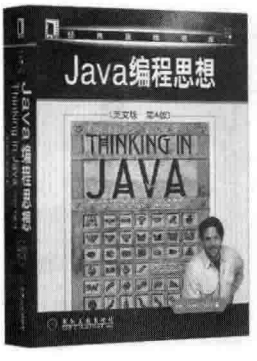
致谢

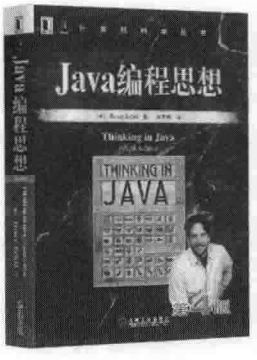
感谢所在单位领导的支持与厚爱，使我在技术道路上更有信心。

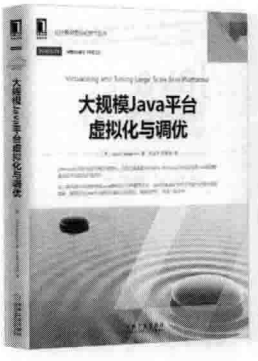
感谢机械工业出版社华章公司的高婧雅和杨福川，因为有了你们的鼓励、帮助和引导，我才能顺利完成本书。

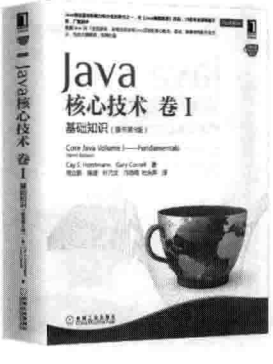
高洪岩

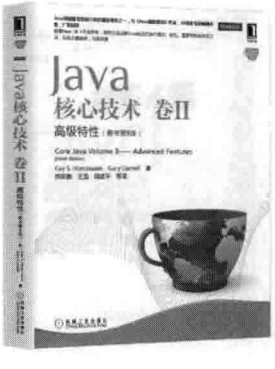
推荐阅读


Java编程思想
THINKING IN JAVA

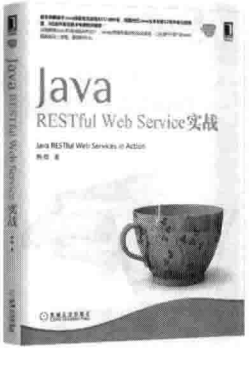

Java编程思想
Thinking in Java
THINKING IN JAVA

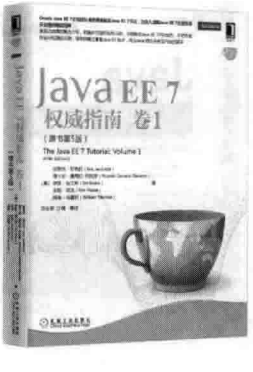

大规模Java平台
虚拟化与调优

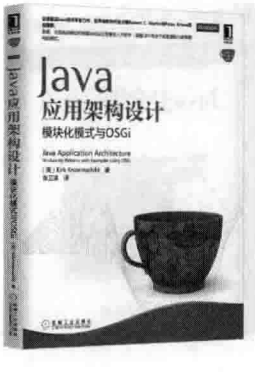

Java
核心技术 卷I
基础知识


Java
核心技术 卷II
高级特性

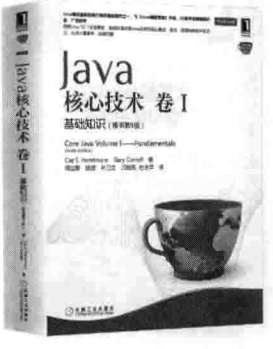

Java
虚拟机规范
(Java SE 7版)

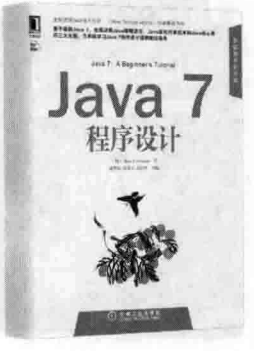

Java
RESTful Web Service实战

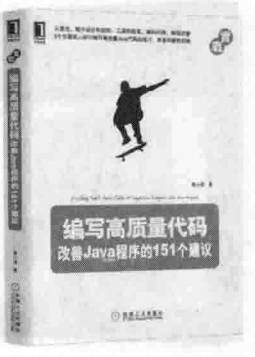

Java EE 7
权威指南 卷1

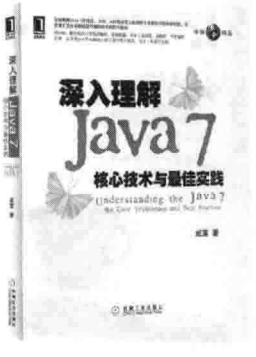

Java
应用架构设计
模块化模式与OSGi

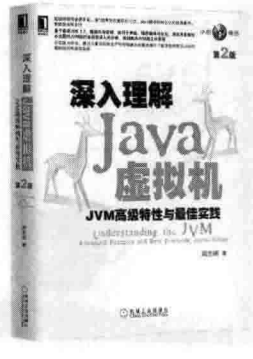
推荐阅读

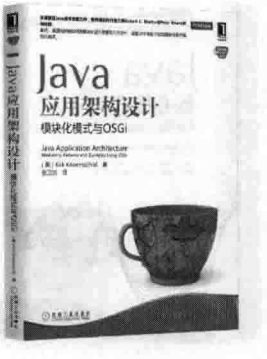

Java
核心技术 卷I
基础知识

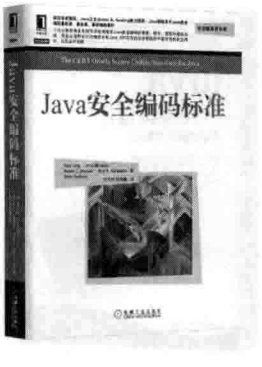

Java 7
程序设计

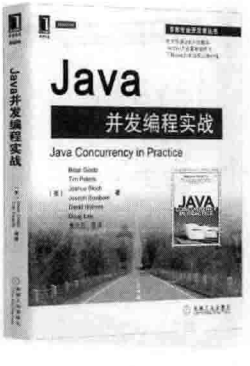

编写高质量代码
改善Java程序的151个建议

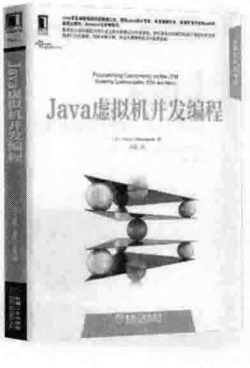

深入理解
Java 7
核心技术与最佳实践


深入理解
Java
虚拟机
JVM高级特性与最佳实践


Java
应用架构设计
模块化模式与OSGi


Java安全编码标准


Java
并发编程实战
Java Concurrency in Practice


Java虚拟机并发编程

目　录 *Contents*

第 1 章 Chapter 1

Java 多线程技能

作为本书的第 1 章，一定要引导读者快速进入 Java 多线程的学习，所以本章中主要介绍 Thread 类中的核心方法。Thread 类的核心方法较多，读者应该着重掌握如下关键技术点：

- 线程的启动
- 如何使线程暂停
- 如何使线程停止
- 线程的优先级
- 线程安全相关的问题

上面的 5 点也是本章学习的重点与思路，掌握这些内容是学习 Java 多线程的必经之路。

1.1 进程和多线程的概念及线程的优点

本节主要介绍在 Java 语言中使用多线程技术。但讲到多线程这个技术时不得不提及“进程”这个概念，“百度百科”里对“进程”的解释如图 1-1 所示。

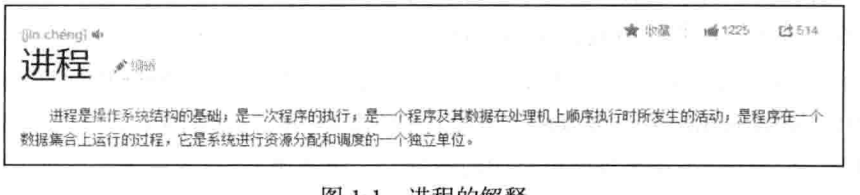


图 1-1 进程的解释

初看这段文字会觉得十分的抽象，难以理解，但如果你看到图 1-2 所示的内容，那么你对进程还不能理解吗？

图 1-2 Windows 7 系统中的进程列表

难道可以将一个正在操作系统中运行的 exe 程序理解成一个“进程”吗？没错！

通过查看“Windows 任务管理器”中的列表，完全可以将运行在内存中的 exe 文件理解成进程，进程是受操作系统管理的基本运行单元。

那什么是线程呢？线程可以理解成是在进程中独立运行的子任务。比如，QQ.exe 运行时就有很多的子任务在同时运行。再如，好友视频线程、下载文件线程、传输数据线程、发送表情线程等，这些不同的任务或者说功能都可以同时运行，其中每一项任务完全可以理解成是“线程”在工作，传文件、听音乐、发送图片表情等功能都有对应的线程在后台默默地运行。

这样做有什么优点呢？更具体来讲，使用多线程有什么优点呢？其实如果读者有使用“多任务操作系统”的经验，比如 Windows 系列，那么它的方便性大家应该都有体会：使用多任务操作系统 Windows 后，可以最大限度地利用 CPU 的空闲时间来处理其他的任务，比如一边让操作系统处理正在由打印机打印的数据，一边使用 Word 编辑文档。而 CPU 在这些任务之间不停地切换，由于切换的速度非常快，给使用者的感受就是这些任务似乎在同时运

行。所以使用多线程技术后，可以在同一时间内运行更多不同种类的任务。

为了更加有效地理解多线程的优势，看一下如图 1-3 所示的单任务的模型图，理解一下单任务的缺点。

在图 1-3 中，任务 1 和任务 2 是两个完全独立、互不相关的任务，任务 1 是在等待远程服务器返回数据，以便进行后期的处理，这时 CPU 一直处于等待状态，一直在“空运行”。如果任务 2 是在 10 秒之后被运行，虽然执行任务 2 用的时间非常短，仅仅是 1 秒，但也必须在任务 1 运行结束后才可以运行任务 2。本程序是运行在单任务环境中，所以任务 2 有非常长的等待时间，系统运行效率大幅降低。单任务的特点就是排队执行，也就是同步，就像在 cmd 中输入一条命令后，必须等待这条命令执行完才可以执行下一条命令一样。这就是单任务环境的缺点，即 CPU 利用率大幅降低。

而多任务的环境如图 1-4 所示。

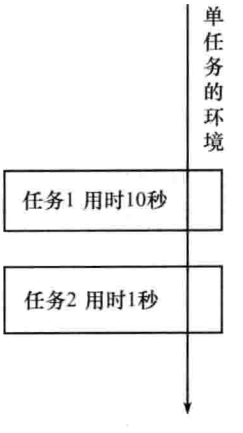


图 1-3　单任务运行环境

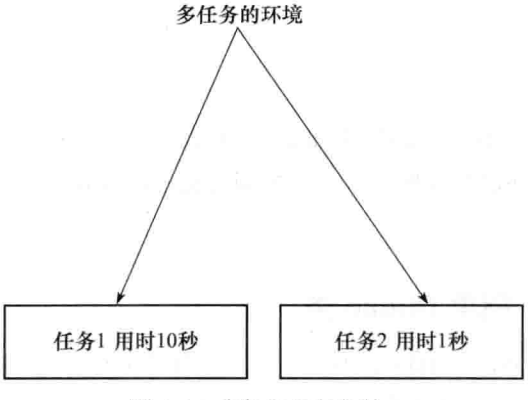


图 1-4　多任务运行环境

在图 1-4 中可以发现，CPU 完全可以在任务 1 和任务 2 之间来回切换，使任务 2 不必等到 10 秒再运行，系统的运行效率大大得到提升。这就是要使用多线程技术、要学习多线程的原因。这是多线程技术的优点，使用多线程也就是在使用异步。

注意　多线程是异步的，所以千万不要把 Eclipse 里代码的顺序当成线程执行的顺序，线程被调用的时机是随机的。

1.2　使用多线程

想学习一个技术就要“接近”它，所以在本节，首先用一个示例来接触一下线程。