

Java Threads

第三版
涵盖J2SE 5.0

JavaTM 线程



O'REILLY®
東南大學出版社

Scott Oaks & Henry Wong 著
O'Reilly Taiwan公司 编译

TP312
2056

Java™ 线程

第三版

Scott Oaks & Henry Wong 著

O'Reilly Taiwan 公司 编译

O'REILLY®

Beijing • Cambridge • Farnham • Köln • Paris • Sebastopol • Taipei • Tokyo

O'Reilly Media, Inc. 授权东南大学出版社出版

东南大学出版社

图书在版编目 (CIP) 数据

Java™ 线程: 第3版 / (美) 奥克斯 (Oaks, S.),
(美) 王 (Wong, H.) 著; O'Reilly Taiwan 公司编译
—南京: 东南大学出版社, 2006.3

书名原文: Java™ Threads, Third Edition

ISBN 7-5641-0239-X

I. J... II. ①奥... ②王... ③ O... III. Java 语言—程序设计
IV. TP312

中国版本图书馆 CIP 数据核字 (2005) 第 057381 号

江苏省版权局著作权合同登记

图字: 10-2005-088 号

©2004 by O'Reilly Media, Inc.

Simplified Chinese Edition, jointly published by O'Reilly Media, Inc. and Southeast University Press, 2006. Authorized translation of the English edition, 2004 O'Reilly Media, Inc., the owner of all rights to publish and sell the same.

All rights reserved including the rights of reproduction in whole or in part in any form.

英文原版由 O'Reilly Media, Inc. 出版 2004。

简体中文版由东南大学出版社出版 2006。英文原版的翻译得到 O'Reilly Media, Inc. 的授权。此简体中文版的出版和销售得到出版权和销售权的所有者——O'Reilly Media, Inc. 的许可。

版权所有, 未得书面许可, 本书的任何部分和全部不得以任何形式重制。

书 名 / Java™ 线程 (第三版)

书 号 / ISBN 7-5641-0239-X

责任编辑 / 张烨

封面设计 / Emma Colby, 张健

出版发行 / 东南大学出版社 (press.seu.edu.cn)

地 址 / 南京四牌楼 2 号 (邮政编码 210096)

印 刷 / 扬中市印刷有限公司

开 本 / 787 毫米 × 980 毫米 16 开本 22.25 印张

版 次 / 2006 年 3 月第 1 版 2006 年 3 月第 1 次印刷

印 数 / 0001-3000 册

定 价 / 39.00 元 (册)

O'Reilly Media, Inc.介绍

为了满足读者对网络和软件技术知识的迫切需求，世界著名计算机图书出版机构 O'Reilly Media, Inc.授权东南大学出版社，翻译出版一批该公司久负盛名的英文经典技术专著。

O'Reilly Media, Inc.是世界上在 UNIX、X、Internet 和其他开放系统图书领域具有领导地位的出版公司，同时是联机出版的先锋。

从最畅销的《The Whole Internet User's Guide & Catalog》（被纽约公共图书馆评为二十世纪最重要的 50 本书之一）到 GNN（最早的 Internet 门户和商业网站），再到 WebSite（第一个桌面 PC 的 Web 服务器软件），O'Reilly Media, Inc.一直处于 Internet 发展的最前沿。

许多书店的反馈表明，O'Reilly Media, Inc.是最稳定的计算机图书出版商——每一本书都一版再版。与大多数计算机图书出版商相比，O'Reilly Media, Inc.具有深厚的计算机专业背景，这使得 O'Reilly Media, Inc.形成了一个非常不同于其他出版商的出版方针。O'Reilly Media, Inc.所有的编辑人员以前都是程序员，或者是顶尖级的技术专家。O'Reilly Media, Inc.还有许多固定的作者群体——他们本身是相关领域的技术专家、咨询专家，而现在编写著作，O'Reilly Media, Inc.依靠他们及时地推出图书。因为 O'Reilly Media, Inc.紧密地与计算机业界联系着，所以 O'Reilly Media, Inc.知道市场上真正需要什么图书。

目录

前言	1
第一章 Thread 导论	9
Java 术语	9
关于范例	12
为何要用 Thread?	14
总结	17
第二章 Thread 的创建与管理	18
什么是 Thread?	18
创建 Thread	22
Thread 的生命周期	30
两种停止 Thread 的方式	34
Runnable Interface	38
Thread 与对象	42
总结	43

第三章 数据同步	45
Synchronized 关键字	45
Volatile 关键字	48
更多 Race Condition 的讨论	50
明确的 (explicit) Locking	56
Lock Scope	59
选择 Locking 机制	61
Nested Lock	63
死 锁	65
Lock 公平 (Fairness)	71
总 结	72
 第四章 Thread Notification	74
等待与通知	74
条件变量	82
总 结	85
 第五章 极简同步技巧	86
能避免同步吗?	86
Atomic 变量	91
Thread 局部变量	110
总 结	112
 第六章 高级同步议题	114
同步术语	114
J2SE 5.0 中加入的同步 Class	116
防止死锁	122
死锁检测	128
Lock 饥饿	141
总 结	145

第七章 Thread 与 Swing	146
Swing Threading 的限制	146
事件派发 Thread 的处理	147
使用 invokeLater()与 invokeAndWait()	148
长时间运行的事件回调	150
总 结	153
 第八章 Thread 与 Collection Class	155
Collection Class 的概述	155
同步与 Collection Class	159
生产者 / 消费者模式	166
使用 Collection Class	168
总 结	170
 第九章 Thread 调度	171
Thread 调度的概述	172
以 Thread 优先级来调度	179
常见 Threading 的实现	180
总 结	186
 第十章 Thread Pool	188
为何要用 Thread Pool?	188
Executor	191
使用 Thread Pool	192
Queue 与大小	194
创建 Thread	198
Callable Task 与 Future 结果	199
单一 Thread 化的访问	201
总 结	202

第十一章 Task 的调度	204
Task 调度的概述	204
java.util.Timer Class	206
javax.swing.Timer Class	211
ScheduledThreadPoolExecutor Class	214
总结	221
 第十二章 Thread 与 I/O	 223
传统的 I/O 服务器	224
新的 I/O 服务器	233
被中断的 I/O	242
总结	245
 第十三章 各种 Thread 议题	 247
Thread Group	247
Thread 与 Java 安全性	249
Daemon Thread	251
Thread 与 Class 的加载	252
Thread 与异常处理	254
Thread、Stack、内存的使用	257
总结	261
 第十四章 Thread 性能	 262
性能的概述	262
同步的 Collection	264
Atomic 变量与有竞争的同步	265
Thread 的创建与 Thread Pool	267
总结	268

第十五章 多处理器计算机的并行化循环	270
对单一 Threaded 程序的并行化	271
多处理器扩大 (scaling)	296
总 结	307
 附录 被撤换的 Threading 功能	309
 索 引	327

前言

当 Sun Microsystems 于 1995 年发表 alpha 版本的 Java 时，全世界的开发者都注意到了。Java 有许多功能吸引着这些开发者，而不只是 Sun 用来营销这种程序语言所用的花言巧语。Java 是坚固、安全、结构中立、便携、面向对象、简单且 multithreaded。对许多开发者来说，最后两个词似乎是矛盾的：一个 multithreaded 的程序语言怎么可能会是简单的？

事实证明，至少 Java 的 threading 系统跟其他的 threading 系统相比较要简单得多。如此的简化让 Java 的 threading 系统很容易学习，甚至对 thread 不熟悉的开发者也能够比较轻松地着手基本的 thread 程序设计。

在早期的 Java 版本中，简化也是有利有弊，在其他 threading 系统上可以看到的某些高级功能在 Java 中找不到。Java 2 Standard Edition Version 5.0 (J2SE 5.0) 就不一样了，它提供了非常多的与 thread 相关的新 class，能够让撰写 multithreaded 程序的工作变得简单了许多。

但运用 thread 的程序设计还是一项复杂的工作。本书要对读者展示如何使用 Java 的 threading 功能进行基本的 threaded 程序设计工作，以及如何将其扩展至更复杂的程序上以执行更高级的工作。

谁该读这本书？

这本书是写给想要学习在 Java 程序中使用 thread 的各种经验程度的程序设计师，这包括了之前就已经使用过 Java 撰写 threaded 程序的开发者。J2SE 5.0 具有非常多的与 thread 相关的新 class 与功能。因此，即使读者已经写过 Java 的 threaded 程序，本书还是能够提供对 Java 新功能的指引以撰写出更有效的程序。

本书的前几章从基本的内容开始讨论Java的threaded程序设计议题，并未假设开发者已经有很丰富的threaded程序设计经验。随着章节的进展，在信息的呈现与对开发者经验的要求上，内容将更为高级。对于刚接触threaded程序设计的开发者来说，这样的安排应该可以对他们就此议题提供循序渐进的学习过程。

这本书相当适合着眼于第二波Java程序（完全释放Java的threading系统威力的复杂程序）的开发者。我们假设本书的读者已经熟悉Java的语法与特点。在某些部分我们会展示出需要对Java其他功能具有一定程度了解的复杂程序，例如说AWT、Swing、NIO等。然而，我们所展示的基本原则是应该对任何具有基本Java知识的人都是可以理解的。我们也发现讨论其他API的书都会尝试对如何在多个thread下完整地运用这些Java的功能有简短的说明（反之也是如此，我们并不会试着要解释与thread无关的Java API）。

虽然本书所展现的内容并没有设定需要对以往thread的知识有所了解，但的确假设读者已经具有其他领域的Java API知识且能够撰写简单的Java程序。

本书使用的版本

在Internet的时代写本书是很艰苦的事情——我们所依赖的基础不停地在变动。但我们还是得在基础上圈起地基，而我们所根据就是Sun Microsystems的Java 2 Standard Edition (J2SE) Version 5.0。此软件之前又被称为J2SE Version 1.5。

这个版本之后的Java版本很可能会在threading系统上面具有一些本书没有讨论到的变化。我们同样也会在正文的行进中指出J2SE 5.0与之前版本的Java间相异之处，因此使用较早版本的Java开发者同样也能够使用本书。

大部分在J2SE 5.0中的threading新功能对早期版本的Java来说（包括本书旧版所开发的class）可通过第三方厂商（根据不同的API）来取得。因此，即使不是使用J2SE 5.0，你还是可以从本书所涵盖的主题中得到丰富的收获。

本版新增部分

这个版本涵盖了关于J2SE 5.0的信息。J2SE 5.0中最重要的改变是引入了Java Specification Request (JSR) 166，此规范通常又称为“concurrency utilities（并发性功能）”。JSR-166对已经存在的API以及一整组新提供的API定义了一群与thread相关的加强功能。

这些新的 API 包括了：

Atomic variable

一组提供 thread 安全性操作而不具 synchronization 的 class。

Explicit lock

可程序化已获得的与释放的 synchronization lock。

Condition variable

当特定条件出现时可作为被锁定的 notification 主题的变量。

Queue

具 thread 感知的 collection class。

Synchronization primitive

执行复杂类型 synchronization 的新 class。

Thread pool

可管理 thread 的 pool 以执行特定 task 的 class。

Thread scheduler

可在特定的时间点执行 task 的 class。

我们已经在此版的文字中结合了 J2SE 5.0 的新功能。这些新功能可以分为下列三类：

现有功能的新实现

该语言一直都有执行 data synchronization 与 thread notification 的能力。然而，这些功能的实现是有限制的。举例来说，你可以同步化程序代码段或者整个 method，但跨 method 与 class 间的同步化需要额外的程序设计。在 J2SE 5.0 中，explicit lock 与 condition variable 能够让你在使用这些功能时更具灵活性。

这些新的实现并没有为开发者引进新的概念。无论是使用 J2SE 5.0 的 explicit lock 还是更基本的 synchronized 关键字，想要撰写 thread 安全性程序的开发者必须要确保自己的数据有正确的同步化。因此，两者皆在涉及 data synchronization 时被拿出来一起讨论。提供 thread notification 的 condition variable 也是如此，并会跟着 Java 的 wait() 与 notify() 两个 method 一起讨论，还有 queue 也会与 Java 其他的 collection class 一并讨论。

重要的 thread 功能

在某些时间点上，几乎所有撰写 threaded 程序的开发者都需要使用 pool 或 scheduler 之类的基本 thread 功能，其中有许多也同样需要使用高级的 synchronization primitive。对此事实的认知正是驱动 JSR-166 的动力之一 —— 在之前的 Java 版本

中当然能够自行开发 thread 的 pool 与 scheduler, 但考虑到 threading 在 Java 平台上的重要性, 加入这些基本的功能能够大幅提高程序设计师的生产力。

极简的 *synchronization* 功能

Java 的新的 atomic class 提供了一种方法让开发者能够在必要时避开 synchronization 撰写应用程序。这会让程序有更高的并发性。

如果你读过本书之前的版本, 应该已经熟悉前两类的功能。在旧版中, 我们开发过自定义的 *synchronization class* 与 *thread pool* 等。在那些版本中, 我们详细地解释了自定义的实现是如何运作的, 并在数个范例中加以运用。在这个版本中, 我们只专注于如何有效地运用这些 class。

在本版中第三类的信息是全新编写的。执行极简的 *synchronization class* 需要从新的 virtual machine 本身中取得支持, 并且也不能独立于这些新变化之外来开发。

本书内容编排

以下是本书的内容概要, 其中有十五章与一个附录:

第一章 Thread 导论

这一章对 thread 的议题作基本的介绍: 它为何有用以及我们对它的讨论方式。

第二章 Thread 的创建与管理

这一章在解释 thread 运作的基本原理的同时展示如何创建 thread 与 runnable 对象。

第三章 数据同步

这一章在基础水平上讨论 thread 间安全地共享数据 —— 于任意时间点协调某个 thread 能够访问数据。接下来的四章中会有 thread 间共享数据的议题。

第四章 Thread Notification

这一章讨论 thread 间在数据变化时用来互相通信的基本技巧。这可以让 thread 间响应数据的变化而不是轮替查询这些变化。

第五章 极简同步技巧

这一章讨论使用最少的同步化来设计出数据安全性的 class 与程序设计方式。

第六章 高级同步议题

在这一章中, 我们会完成对数据共享与同步化的讨论, 也包括 deadlock、starvation 与其他 locking class 的讨论。

第七章 Thread 与 Swing

Swing class 并不具有 thread 的安全性。这一章讨论 multithreaded 程序如何完全发挥 Swing 的优点。

第八章 Thread 与 Collection Class

Java 的 collection class 是写给各种不同的状况的。有些是具有 thread 安全性的而某些则不是, J2SE 5.0 特地为 thread 功能引进新的 collection class。在这一章中我们会将它们列出。

第九章 Thread 调度

调度是单一 CPU 怎样选择 thread 来运行的进程。Thread 调度比较是操作系统 (OS) 而不是 Java 程序的事情, 这一章会讨论此领域中 virtual machine 与 OS 之间的关系。

第十章 Thread Pool

这一章讨论 thread pool —— 一群用来执行非特定 task 的 thread 的集合。我们使用 J2SE 5.0 的 thread pool 实现来讨论使用 thread pool 的基本原则。

第十一章 Task 的调度

task scheduler 在未来的某一时间点一次或多次地执行某一 task。这一组 class 包括 timer (Java 从 JDK 1.3 开始就有 timer 的 class) 与 J2SE 5.0 中可见的普通的 task scheduler。

第十二章 Thread 与 I/O

处理 I/O 是开发者为何在 Java 中使用 thread 的主要原因之一。在这一章中, 我们使用 Java 全部的 threading 功能来展示如何在 multithreaded 程序中有效地处理 I/O。

第十三章 各种 Thread 议题

在这一章中, 我们通过察看 thread security、thread group、thread stack 与其他议题来完成对 Java 中与 thread 有关功能的讨论。

第十四章 Thread 性能

与 thread 有关功能的性能 (特别是 synchronization 构造) 是撰写 multithreaded 程序的关键。在这一章中, 我们测试各种低水平的程序设计特征并探索关于 thread 性能的一些真相与神话。

第十五章 多处理器计算机的并行化循环

在这一章中, 我们展示运用多处理器计算机的能力于并行的循环中进行 CPU 密集的运算。

附录 被撤换的 Threading 功能

J2SE 5.0 引进许多与 thread 有关的 class，这些 class 中有许多与本书旧版中所开发的 class 类似。我们在附录中列出这些 class 以便帮助还不能升级到 J2SE 5.0 的开发者的。

排版约定

下面是本书使用的排版惯例：

斜体字 (*Italic*)

表示 URL 或者文件名称，也用于介绍新术语。有时我们用问答的形式来解释 thread 的功能，其中由读者提出的问题用斜体字表示。

等宽字 (Constant Width)

表示正文中的程序代码范例、method、变量、参数与关键字。

等宽黑体字 (Constant Width Bold)

表示使用者输入的数据，例如在命令行输入的命令。

程序范例

所有在本书中展示的范例都是完整的、可以运行的应用程序。然而，有许多的程序仅简短列出是因为考虑到空间与可读性的缘故。完整的范例可从在线资源中下载，网址是：<http://www.oreilly.com/catalog/jthreads3>。

这本书是用来帮助你完成工作的。一般来说，你可以在自己的程序与说明文件中使用本书的程序代码。除非复制大部分的程序代码，否则你无需取得我们的使用许可。举例来说，使用本书数个程序代码的片段来撰写程序无需事先取得许可。销售或发布奥莱理丛书随附的 CD-ROM 则必须先取得许可。引用本书与列举本书范例程序代码来回答问题无需许可。在你的产品说明文件中采用本书一定数量的范例程序代码需要取得许可。

我们感谢但不强制要求注明引文出处。引文出处通常包括了书名、作者、出版商与 ISBN。例如：Java Threads, Third Edition, by Scott Oaks and Henry Wong. Copyright 2004 O'Reilly Media, 0-596-00782-5。

建议与评论

本书的内容都经过测试，尽管我们做了最大的努力，但错误和疏忽仍然是在所难免的。如果你发现有什么错误，或者是对将来的版本有什么建议，请通过下面的地址告诉我们：

美国：

O'Reilly Media, Inc.
1005 Gravenstein Highway North
Sebastopol, CA 95472

中国：

100080 北京市海淀区知春路 49 号希格玛公寓 B 座 809 室
奥莱理软件（北京）有限公司

O'Reilly 的每一本书都有专属网页，你可以在此找到关于本书的相关信息，包括范例程序的下载、勘误表与相关资源的链接。

<http://www.oreilly.com/catalog/jthreads3>

询问技术问题或对本书的评论，请发电子邮件到：

info@mail.oreilly.com.cn
bookquestions@oreilly.com

最后，您可以在以下站点找到我们：

<http://www.oreilly.com>
<http://www.oreilly.com.cn>

致谢

读过前言的读者都知道，编撰出一本书绝对不只是独占封面的作者一个人的功劳。我们深深地感激下列这些人给予的帮助与鼓励：Michael Loukides，在我们认为这是个重要的主题时相信我们并于创意构想期间给我们以指引；David Flanagan，对初稿给予无价的指教；Deb Cameron，有条不紊地编辑凌乱文字；Hong Zhang，对Windows的threading议题给予我们帮助；以及Reynold Jabbour、Wendy Talmont、Steve Wilson与Tim Cramer过去六年来在工作上给予的协助。

最重要的，我们必须感谢各自的家人。James，他给予 Scott 必要的支持与鼓励才能够使他完成这本书（以及处理他不断分心的状态）；还有 Nini，知道在灵感来时留给 Henry 自己一些时间，并在其余的时间给予鼓励。感谢你们所做的一切！

最后，我们必须感谢许多对本书旧版本给予无价意见的读者。我们已经尽力地试着回答他们所提出的每一个疑问。请继续来信！