Broadview
www.broadview.com.cn

Java 并发编程之美

翟陆续 薛宾田 著

中国工信出版集团
電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

Java 并发编程之美

翟陆续 薛宾田 著

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

内容简介

并发编程相比 Java 中其他知识点的学习门槛较高，从而导致很多人望而却步。但无论是职场面试，还是高并发 / 高流量系统的实现，却都离不开并发编程，于是能够真正掌握并发编程的人成为了市场迫切需求的人才。

本书通过图文结合、通俗易懂的讲解方式帮助大家完成多线程并发编程从入门到实践的飞跃！全书分为三部分，第一部分为 Java 并发编程基础篇，主要讲解 Java 并发编程的基础知识、线程有关的知识和并发编程中的其他相关概念，这些知识在高级篇都会有所使用，掌握了本篇的内容，就为学习高级篇奠定了基础；第二部分为 Java 并发编程高级篇，讲解了 Java 并发包中核心组件的实现原理，让读者知其然，也知其所以然，熟练掌握本篇内容，对我们在日常开发高并发、高流量的系统时会大有裨益；第三部分为 Java 并发编程实践篇，主要讲解并发组件的使用方法，以及在使用过程中容易遇到的问题和解决方法。

本书适合 Java 初级、中高级研发工程师，对 Java 并发编程感兴趣，以及希望探究 JUC 包源码原理的人员阅读。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有，侵权必究。

图书在版编目（CIP）数据

Java 并发编程之美 / 翟陆续，薛宾田著 . —北京：电子工业出版社，2018.11
ISBN 978-7-121-34947-8

Ⅰ . ① J… Ⅱ . ①翟… ②薛… Ⅲ . ① JAVA 语言 – 程序设计 Ⅳ . ① TP312

中国版本图书馆 CIP 数据核字 (2018) 第 199378 号

策划编辑：刘　皎
责任编辑：牛　勇
印　　刷：三河市华成印务有限公司
装　　订：三河市华成印务有限公司
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：787×980　1/16　印张：22.5　字数：432 千字
版　　次：2018 年 11 月第 1 版
印　　次：2018 年 11 月第 1 次印刷
定　　价：89.00 元

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）88254888，88258888。

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。

本书咨询联系方式：010-51260888-819，faq@phei.com.cn。

业界好评

Java 的并发编程太重要，又太迷人，所以自 Goetz 的 Java Concurrency in Practice 在 2006 年出版，2012 年重译以来，国内的众多作者又陆陆续续出版了若干本相关主题的书籍。那我手上的这本，是又一本 Java 并发编程 (Yet Another ...) 吗？为了找一个大家再次购买的理由，我快速翻完了全书。

显而易见，作者是一位喜欢用代码说话的同学，第一部分基础知识中的每个知识点都伴随一段简短的示例及证明的代码，代码不撒谎。

作者对代码的爱，也带到了第二部分。书中针对 Java 并发库中的主要组件，进行了代码级的原理讲解，而且紧贴时代脉搏，涵盖了 JDK 8 的内容。如果你能耐下心来，跟随作者进行一番代码级的探究，所产生的印象比阅读文章、死记结论，无疑要深刻得多。

到了最后的实践部分，依然没有模式、架构之类的宏大叙事，而是作者自己一个个的实践例子。

所以，如果要简单概括，这就是一本有好奇心的 Coder，写给另一位有好奇心的 Coder 的 Java 并发编程书。

——肖桦（江南白衣），唯品会资深架构师，公众号“春天的旁边”

JDK 1.5 之前，我们必须自己编写代码实现一些并发编程的逻辑；之后到了 JDK 1.5，Doug Lea 解救了广大 Java 用户，在 JDK 里特意设计并实现了一套 JUC 的框架，给大家提供了非常好的并发编程体验。本书作者在阿里经历过大量并发的场景，积攒了不少并发编程的经验，并毫无保留地写入本书。通过书中对 JUC 源码的解读，读者可以揭开 JUC 的神秘面纱。这是一本值得仔细品读的好书。

——你假笨 / 寒泉子，PerfMa CEO，公众号“你假笨”

Java 并发编程所涉及的知识点比较多，多线程编程所考虑的场景相对比较复杂，包括线程间的资源共享、竞争、死锁等问题。并发编程相比 Java 中其他知识点，学习起来门槛相对较高，学习难度较大，从而导致很多人望而却步。加多的《Java 并发编程之美》这本书刚好填补了这个空缺，作者在并发编程领域深耕多年。本书用浅显易懂的文字为大家系统地介绍了 Java 并发编程的相关内容，推荐大家关注学习。

——纯洁的微笑，第三方支付公司技术总监，公众号“纯洁的微笑”

Java 并发编程无处不在，Java 多线程、并发处理是深入学习 Java 必须要掌握的技术。本书涵盖了 Java 并发包中的核心类、API 以及框架等内容，并辅以详尽的案例讲解，帮助读者快速学习、迅速掌握。如果你希望成长为一名优秀的 Java 程序员，有必要读一读本书。

——许令波，《深入分析 Java Web 技术内幕》作者

第一作者加多是一位非常勤奋的技术人员，经常发布各种技术文章，有时候甚至能做到每天一篇，在并发编程网已经累计发布了近百篇文章。本书是他多年的积累，厚积薄发，从并发编程的基础知识一直到实战娓娓道来，希望读者喜欢。

——方腾飞，并发编程网创始人

试读结束：需要全本请在线购买： www.ertongbook.com

前　言

本书特色

不像其他并发类书籍那样晦涩难懂，本书的特色之一是通俗易懂，对 Java 有一定基础的开发人员都可以看懂。本书在基础篇专门讲解并发编程基础，笔者根据在项目实践中对这些知识的理解，总结了并发编程中常用的基础知识以及常用的概念，并通过图文结合的方式降低理解的难度，使用少量的代码讲解就可以让读者轻松掌握并发编程的基础知识，让读者逐步建立起自信。在高级篇主要讲解 JUC 并发包下并发组件的实现原理，首先介绍 JUC 里面最简单的原子类，让读者学会使用在基础篇里介绍的最简单的 CAS 操作，再逐步加大难度让读者慢慢适应：比如一开始打算把并发 List 放到锁后面讲解，因为并发 List 里面使用了锁，但是锁的理解难度比 List 大太多，所以最终还是坚持从易入难的原则，先讲解 List，再讲解锁。在实践篇中首先讲解并发组件在开源框架或者项目中的运用，让读者不仅可以知道并发组件的原理，而且可以了解怎么使用这些组件。最后总结笔者在项目中或者其他同事在项目中经常遇到的并发编程问题，并对其进行分析，给出解决方案。

如果你只想使用并发包，那么可以阅读本书，因为本书在讲解代码时基本都是用的实例；如果你想研究源码却一筹莫展——不知道如何下手或者感觉吃力，也可以阅读本书，因为本书对核心代码进行了讲解；如果你想了解并发编程中的常见问题，增加对并发的认识，也可以阅读本书，因为本书对这类问题进行了总结。

如何阅读本书

本书分为基础篇、高级篇和实践篇，其中基础篇讲解线程的知识和并发编程中的基本概念以及基础知识，高级篇则介绍并发包下常用的并发组件的原理，实践篇讲解并发组件的具体使用方法和在并发编程中会遇到的一些并发问题及解决方法。

阅读开源框架源码的一点心得

为什么要看源码

我们在做项目的时候一般会遇到下面的问题：

（1）不知道如何去设计。比如刚入职场时，来一个需求需做概要设计，不知如何下手，不得不去看当前系统类似需求是如何设计的，然后仿照去设计。

（2）设计的时候，考虑问题不周全。相比职场新手，这类人对一个需求依靠自己的经验已经能够拿出一个概要设计，但是设计中经常会遗漏一些异常细节，比如使用多线程有界队列执行任务，遇到机器宕机了，如果队列里面的任务不存盘的话，那么机器下次启动的时候这些任务就丢失了。

对于这些问题，说到底主要还是因为经验不够，而经验主要从项目实践中积累，所以招聘单位一般都会限定工作时间大于 3 年，因为这些人的项目经验相对较丰富，在项目中遇到的场景相对较多。工作经验的积累来自于年限与实践，然而看源码可以扩展我们的思路，这是变相增加我们经验的不错方法。虽然不能在短时间内通过时间积累经验，但是可以通过学习开源框架、开源项目来获取经验。

另外，进职场后一般都要先熟悉现有系统，如果有文档还好，没文档的话就得自己去翻代码研究。如果之前对阅读源码有经验，那么在研究新系统的代码逻辑时就不会那么费劲了。

还有一点就是，当你使用框架或者工具做开发时，如果你对它的实现有所了解，就能最大化地减少出故障的可能。比如并发队列 ArrayBlockingQueue 里面关于元素入队有个 offer 方法 和 put 方法，虽然某个时间点你知道使用 offer 方法时，当队列满了就会丢弃要入队的元素，之后 offer 方法会返回 false，而不会阻塞当前线程；而使用 put 方法时，当队列满了，则会挂起当前线程，直到队列有空闲，元素入队成功后才返回。但是人是善忘的，一段时间不使用，就会忘记它们的区别，当你再去使用时，需进入 offer 和 put 方法的内部，看它们的源码实现。进入 offer 方法一看，哦，原来队列满后直接返回了 false；进入 put 方法一看，哦，原来队列满后，直接使用条件变量的 await 方法挂起了当前线程。知道了它们的区别，你就可以根据自己的需求来选择了。

看源码最大的好处是可以开阔思维，提升架构设计能力。有些东西仅靠书本和自己思考是很难学到的，必须通过看源码，看别人如何设计，然后思考为何这样设计才能领悟到。能力的提高不在于你写了多少代码，做了多少项目，而在于给你一个业务场景时，你是否能拿出几种靠谱的解决方案，并且说出各自的优缺点。而如何才能拿出来，一来靠经验，二来靠归纳总结，而看源码可以快速增加你的经验。

如何看源码

那么如何阅读源码呢？在你看某一个框架的源码前，先去 Google 查找这个开源框架的官方介绍，通过资料了解该框架有几个模块，各个模块是做什么的，之间有什么联系，每个模块都有哪些核心类，在阅读源码时可以着重看这些类。

然后对哪个模块感兴趣就去写个小 demo，先了解一下这个模块的具体作用，然后再 debug 进入看具体实现。在 debug 的过程中，第一遍是走马观花，简略看一下调用逻辑，都用了哪些类；第二遍需有重点地 debug，看看这些类担任了架构图里的哪些功能，使用了哪些设计模式。如果第二遍有感觉了，便大致知道了整体代码的功能实现，但是对整体代码结构还不是很清晰，毕竟代码里面多个类来回调用，很容易遗忘当前断点的来处；那么你可以进行第三遍 debug，这时候你最好把主要类的调用时序图以及类图结构画出来，等画好后，再对着时序图分析调用流程，就可以清楚地知道类之间的调用关系，而通过类图可以知道类的功能以及它们相互之间的依赖关系。

另外，开源框架里面每个功能类或者方法一般都有注释，这些注释是一手资料，比如 JUC 包里的一些并发组件的注释，就已经说明了它们的设计原理和使用场景。

在阅读源码时，最好画出时序图和类图，因为人总是善忘的。如果隔一段时间你再去看之前看过的源码，虽然有些印象，但当你想去看某个模块的逻辑时，又需根据 demo 再从头 debug 了。而如果有了这俩图，就可以从这俩图里面直接找，并且看一眼时序图就知道整个模块的脉络了。

此外，查框架使用说明最好去官网查（这些信息是源头，是没有经过别人翻译的），虽然是英文，但是看久了就好了，毕竟还有 Google 翻译呐！

当然研究代码时不一定非要 debug 三遍，其实这里说的是三种掌握程度，如果你 debug 一遍就能掌握，那自然更好啦。

目　录

第一部分　Java 并发编程基础篇

试读结束：需要全本请在线购买： www.ertongbook.com

第三部分　Java 并发编程实践篇

第一部分

Java并发编程基础篇

本篇主要介绍并发编程的基础知识，包含两章内容，分别为并发编程线程基础以及并发编程的其他概念与原理解析。

第1章

并发编程线程基础

1.1 什么是线程

在讨论什么是线程前有必要先说下什么是进程，因为线程是进程中的一个实体，线程本身是不会独立存在的。进程是代码在数据集合上的一次运行活动，是系统进行资源分配和调度的基本单位，线程则是进程的一个执行路径，一个进程中至少有一个线程，进程中的多个线程共享进程的资源。

操作系统在分配资源时是把资源分配给进程的，但是 CPU 资源比较特殊，它是被分配到线程的，因为真正要占用 CPU 运行的是线程，所以也说线程是 CPU 分配的基本单位。

在 Java 中，当我们启动 main 函数时其实就启动了一个 JVM 的进程，而 main 函数所在的线程就是这个进程中的一个线程，也称主线程。

进程和线程的关系如图 1-1 所示。

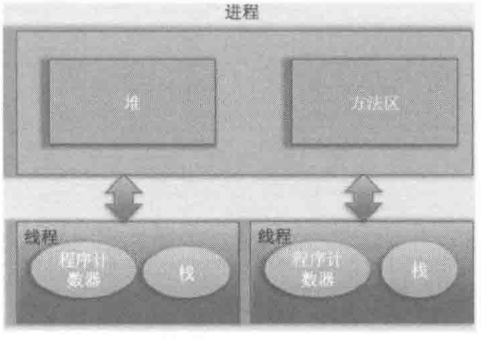

图 1-1

试读结束：需要全本请在线购买： www.ertongbook.com