李建平◎著

Java 多线程与大数据处理实战

作者用**通俗易懂的语言**+**丰富详尽的实例**+**40多幅手绘图表**，向读者展示了一幅**Java编程之美**与**大数据处理之巧**交织辉映的美好画卷。

北京大学出版社
PEKING UNIVERSITY PRESS

Java 多线程与大数据处理实战

李建平◎著

北京大学出版社
PEKING UNIVERSITY PRESS

内 容 提 要

本书对Java的多线程及主流大数据中间件对数据的处理进行了较为详细的讲解。本书主要讲了Java的线程创建方法和线程的生命周期，方便我们管理多线程的线程组和线程池，设置线程的优先级，设置守护线程，学习多线程的并发、同步和异步操作，了解Java的多线程并发处理工具（如信号量、多线程计数器）等内容。同时，本书还引入了Spring Boot、Spring Batch、Quartz、Kafka等大数据中间件。这为学习Java多线程和大数据处理的读者提供了良好的参考。多线程和大数据的处理是许多开发岗位面试中最容易被问到的知识点，一些一线开发的重要岗位面试会将多线程作为压轴问题或重要的考察点。所以，学好多线程的知识点，无论是对于日后的开发工作，还是正要前往一线开发岗位的面试准备，都是非常有用的。

本书既适合高等院校的计算机类专业的学生学习，也适合从事软件开发相关行业的初级和中级开发人员。

图书在版编目(CIP)数据

Java多线程与大数据处理实战 / 李建平著. — 北京:北京大学出版社，2020.4
ISBN 978-7-301-31283-4

Ⅰ.①J… Ⅱ.①李… Ⅲ.①JAVA语言－程序设计 Ⅳ.①TP312.8

中国版本图书馆CIP数据核字(2020)第040196号

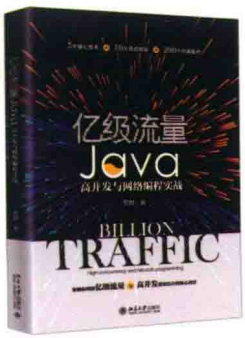
书　　名 Java 多线程与大数据处理实战
JAVA DUOXIANCHENG YU DASHUJU CHULI SHIZHAN

著作责任者 李建平　著
责任编辑 吴晓月　王继伟
标准书号 ISBN 978-7-301-31283-4
出版发行 北京大学出版社
地　　址 北京市海淀区成府路205 号　100871
网　　址 http://www.pup.cn　　新浪微博：@ 北京大学出版社
电子信箱 pup7@ pup.cn
电　　话 邮购部 010-62752015　发行部 010-62750672　编辑部 010-62570390
印 刷 者 北京市科星印刷有限责任公司
经 销 者 新华书店
787毫米×1092毫米　16开本　19.75印张　448千字
2020年4月第1版　2020年4月第1次印刷
印　　数 1-4000册
定　　价 79.00元

作者简介

李建平，从小学开始接触编程，并多次在参赛中获奖。大学选择自己喜欢的软件工程专业进行了更为系统的学习。本科毕业之后，先后在中兴通讯、汇丰银行（HSBC）、广电运通研究院等数据处理部门担任过一线的中级、高级软件研发工程师等职务。能熟练使用Java、Python等处理大数据，并拥有10年以上的开发经验。

好书荐读



北京大学出版社
PEKING UNIVERSITY PRESS
第七事业部

智慧创造价值
品质铸就卓越

“博雅读书社”
微信公众号

读者邮箱：2751801073@qq.com
投稿邮箱：pup7@pup.cn
出版咨询：010-62570390

封面设计：李尘工作室

前　言

多线程技术在大型互联网系统及大数据处理中有着广泛的应用，它能够更合理地利用系统的硬件资源，提供并发执行任务的能力，使系统处理数据的效率大大提高。因此，掌握好了多线程技术，手中就多了一把利器，以前一些系统应用中遇到的问题也都可以迎刃而解了。同时，理解透彻常用的多线程的数据处理方法，也为我们开发处理能力更为卓越的系统提供了方向和可行性。

Java 对多线程提供了全方位的支持，其具备多种线程的创建方法，能按照自己的需要创建合适的线程。Java 提供了线程组和线程池，方便管理多线程。对于多线程的调度，Java 也提供了多种方法，可以满足众多场景下的多线程处理。同时，还可以设置线程的优先级、守护线程等。对于多线程的并发，可以进行同步和异步操作。Java 的多线程并发处理包中也提供了如信号量、多线程计数器等众多并发处理辅助工具。所以，相比起其他一些较为偏门的编程语言，以 Java 的多线程作为入口，能够帮助开发人员更为全面地理解多线程的相关知识。

多线程对于入门级别的开发人员来说，难度会略高于其他的编程基础知识点。但它并非“奇技淫巧”，相反，当需要开发性能更高、处理数据量更大的系统时，多线程的重要性就越发突显。

本书特色

（1）内容实用，图文并茂，提供众多生活中的案例，并将浅明的例子融入开发中。书中每讲解一个多线程的知识点，作者都尽力去寻找身边或生活中合适的、易懂的例子，将问题融入代码中进行解决。同时，对于一些较难用言语表达清楚的知识点，作者都尝试加入示意图，作为补充说明。

（2）多线程与大数据处理相结合，同时引入了一些主流的大数据中间件，结合多线程的知识点，进行大数据处理系统的开发。本书使用了许多主流的开源框架，如 Spring Boot、Spring Batch、Netty 等，以及大数据消息中间件 Kafka、大数据任务调度框架 Quartz 等。

提示：本书所涉及的源代码已上传到百度网盘，供读者下载。请读者关注封底“博雅读书社”微信公众号，找到“资源下载”栏目，根据提示获取。

本书内容及体系结构

第 1 章　Java 多线程基础

本章从 Java 单线程的简介开始，慢慢地引出多线程的内容；同时通过简单的示例，帮助读者了解线程的多种建立方式。通过这样的对比，大家可以轻松地掌握多种编写简单线程的方法。

第 2 章　线程的生命周期

本章将从线程的多种状态及生命周期来更深入地讲解线程，这样可在实际开发多线程中更好地理解和分析一些问题，即知其然并知其所以然。

第 3 章　多线程的调度方式

本章将讲解多线程的另一个重要内容：线程之间的调度。由操作系统的调度原理入手，讲解 Java 多线程之间的调度；同时，通过实例讲解睡眠、唤醒、让步、插队等不同情况下的线程调度方式。

第 4 章　多线程的线程组与线程池

本章介绍的线程组和线程池是 Java 多线程中的两个应用，特别是线程池，其在许多商业和企业级的系统中都会使用。所以，了解线程组，学习和运用好线程池对开发真正的企业级系统非常有帮助。

第 5 章　多线程的异常处理

本章介绍的异常处理是每一个企业级项目中都必须存在的重要且必要的环节。好的系统当然需要良好的异常处理机制来保证系统的健壮性。

第 6 章　多线程定时任务 TimerTask

本章将介绍 Java 自带的多线程定时任务 TimerTask 工具，其在创建任务的过程中，实际上也会创建一个新的线程。

第 7 章　多线程并发处理

从本章开始，讲解一些线程及多线程情况下的线程的高级特性，其中多线程的并发处理是重点和难点。能够把握好多线程的并发处理，基本上就掌握了运用多线程的能力。

第 8 章　批处理 Spring Batch 与多线程

本章将讲解大数据批处理框架 Spring Batch。同时，Spring Batch 能在 Step 中使用多线程，实现在大数据情况下批处理过程的再次加速。

第 9 章　大数据任务调度框架 Quartz 与多线程

本章介绍的 Quartz 是 Java 大数据任务调度框架，其已经在许多大型的商用软件中发挥了重要的作用，能更好地替代 TimerTask 定时任务工具。

第 10 章　大数据中间件 Kafka 与多线程

本章将讲解大数据中间件 Kafka。在分布式的子系统、微服务之间的通信中，经常会使用到消息队列（Message Queue，MQ）。其中，Kafka 作为大数据 MQ，其在大数据处理上有着突出的表现。

第 11 章　多线程实战训练

本章将通过几个简单的小项目，抛砖引玉，让大家多思考，看看是否也可以在自己的业务开发中或更大的项目中引入类似的多线程处理功能。

目　录

第1章

Java 多线程基础

本章将从 Java 单线程的简介开始，慢慢地引出多线程的内容；同时，通过简单的示例，帮助读者了解线程的多种建立方式。通过这样的对比，大家可以轻松地掌握多种编写简单线程的方法。

本章内容主要涉及以下知识点。

- 程序、进程、线程三者之间的关系。
- 多线程的优势。
- 守护线程和用户线程。
- 在 Java 中，用户创建线程的三种方法。
- 搭建好环境，通过 IDEA 进行线程开发和调试。

1.1 初识线程

首先介绍线程的概念。我们可以把线程看作有活力、有生命力的，它是让看似安静的一段段代码活动起来的一种形式。在学习本章前，读者如果已经进行过 Java 简易入门基础的学习，那么理解线程会是一件非常容易的事情。理解了线程的概念后，会由单线程向多线程进军。

1.1.1 线程是什么?

如今，智能手机与我们的生活密不可分。智能手机之所以这样吸引我们，与其能提供丰富多彩的应用程序有密切的关系。在使用这些应用程序，如查阅资讯、单击图标、拉取列表、播放视频和音乐等时，会给人们以视觉和听觉上的享受。同时，智能手机能及时地对我们的操作进行反馈，非常友好。这里的每一次反馈，都可能是有一个线程在专心致志地为我们服务。所以，看似陌生的线程实际上已经默默服务人们多时。

每一个刚接触程序设计的初级人员，在学习了某种编程语言后都会开始尝试编写一些基本的短小的代码段。在 Java 中，这些短小的代码段一般会被放入一个 class，然后保存到一个扩展名为 .java 的文件中；之后通过命令行或集成开发环境工具的编译，生成 .class 文件并让这个 .class 文件运行起来，得到我们想要的结果。

例如，有一个简单的模仿游戏打开宝箱得到礼品的程序代码，参考如下：

```
1  public class OpenBox {
2      public static void main(String[] args) {
3          // 设置宝箱中可能包含的水果
4          List<String> fruits = new ArrayList<String>();
5          fruits.add("green apple");
6          fruits.add("red apple");
7          fruits.add("banana");
8          fruits.add("cherry");
9          fruits.add("watermelon");
10         // 获取随机的下标，用于生成随机的水果，范围为 0 至最大水果链表的下标
11         Random randomUtil = new Random();
12         int randomInt = randomUtil.nextInt(fruits.size());
13         System.out.println(" 打开宝箱，得到了 " +
14           fruits.get(randomInt) + "!");
15     }
16 }
```

我们将其以文件形式保存到系统中，如图 1.1 所示。