

《Java多线程编程实战指南（设计模式）》

图书基本信息

书名：《Java多线程编程实战指南（设计模式篇）》

13位ISBN编号：9787121270064

出版时间：2015-10

作者：黄文海

页数：284

版权说明：本站所提供下载的PDF图书仅提供预览和简介以及在线试读，请支持正版图书。

更多资源请访问：www.tushu000.com

《Java多线程编程实战指南（设计模式）》

内容概要

随着CPU多核时代的到来，多线程编程在充分利用计算资源、提高软件服务质量方面扮演了越来越重要的角色。而

解决多线程编程中频繁出现的普遍问题可以借鉴设计模式所提供的现成解决方案。然而，多线程编程相关的设计模式书籍多采用C++作为描述语言，且书中所举的例子多与应用开发人员的实际工作相去甚远。《Java多线程编程实战指南（设计模式篇）》采用Java（JDK1.6）语言和UML为描述语言，并结合作者多年工作经历的相关实战案例，介绍了多线程环境下常用设计模式的来龙去脉：各个设计模式是什么样的及其典型的实际应用场景、实际应用时需要注意的事项以及各个模式的可复用代码实现

。《Java多线程编程实战指南（设计模式篇）》适合有一定Java多线程编程基础、经验的读者。

《Java多线程编程实战指南（设计模式）》

作者简介

黄文海，2004年开始从事软件开发工作，近几年从事软件项目管理工作。在其工作过程中积累了丰富的技术指导经验和企业内部培训经验。曾在InfoQ中文站和IBM developerWorks上发表过十几篇技术、项目管理文章。

媒体评论

书籍目录

1章 Java多线程编程实战基础

- 1
- 1.1 无处不在的线程
1
- 1.2 线程的创建与运行
2
- 1.3 线程的状态与上下文切换
5
- 1.4 线程的监视
7
- 1.5 原子性、内存可见性和重排序——重新认识synchronized和volatile
10
- 1.6 线程的优势和风险
11
- 1.7 多线程编程常用术语
13

第2章 设计模式简介

- 17
- 2.1 设计模式及其作用
17
- 2.2 多线程设计模式简介
20
- 2.3 设计模式的描述
21

第3章 Immutable Object（不可变对象）模式

- 23
- 3.1 Immutable Object模式简介
23
- 3.2 Immutable Object模式的架构
25
- 3.3 Immutable Object模式实战案例
27
- 3.4 Immutable Object模式的评价与实现考量
31
- 3.5 Immutable Object模式的可复用实现代码
32
- 3.6 Java标准库实例
32
- 3.7 相关模式
34
- 3.7.1 Thread Specific Storage模式（第10章）
34
- 3.7.2 Serial Thread Confinement模式（第11章）
34
- 3.8 参考资源
34

第4章 Guarded Suspension（保护性暂挂）模式

35	
4.1	Guarded Suspension模式简介
35	
4.2	Guarded Suspension模式的架构
35	
4.3	Guarded Suspension模式实战案例解析
39	
4.4	Guarded Suspension模式的评价与实现考量
45	
4.4.1	内存可见性和锁泄漏（Lock Leak）
46	
4.4.2	线程过早被唤醒
46	
4.4.3	嵌套监视器锁死
47	
4.5	Guarded Suspension模式的 reusable 实现代码
50	
4.6	Java标准库实例
50	
4.7	相关模式
51	
4.7.1	Promise模式（第6章）
51	
4.7.2	Producer-Consumer模式（第7章）
51	
4.8	参考资源
51	
第5章	Two-phase Termination（两阶段终止）模式
52	
5.1	Two-phase Termination模式简介
52	
5.2	Two-phase Termination模式的架构
53	
5.3	Two-phase Termination模式实战案例
56	
5.4	Two-phase Termination模式的评价与实现考量
63	
5.4.1	线程停止标志
63	
5.4.2	生产者-消费者问题中的线程停止
64	
5.4.3	隐藏而非暴露可停止的线程
65	
5.5	Two-phase Termination模式的 reusable 实现代码
65	
5.6	Java标准库实例
66	
5.7	相关模式
66	

5.7.1 Producer-Consumer模式（第7章）	66
5.7.2 Master-Slave模式（第12章）	66
5.8 参考资源	66
第6章 Promise（承诺）模式	67
6.1 Promise模式简介	67
6.2 Promise模式的架构	68
6.3 Promise模式实战案例解析	70
6.4 Promise模式的评价与实现考量	74
6.4.1 异步方法的异常处理	75
6.4.2 轮询（Polling）	75
6.4.3 异步任务的执行	75
6.5 Promise模式的可复用实现代码	77
6.6 Java标准库实例	77
6.7 相关模式	78
6.7.1 Guarded Suspension模式（第4章）	78
6.7.2 Active Object模式（第8章）	78
6.7.3 Master-Slave模式（第12章）	78
6.7.4 Factory Method模式	78
6.8 参考资源	79
第7章 Producer-Consumer（生产者/消费者）模式	80
7.1 Producer-Consumer模式简介	80
7.2 Producer-Consumer模式的架构	80
7.3 Producer-Consumer模式实战案例解析	83
7.4 Producer-Consumer模式的评价与实现考量	87
7.4.1 通道积压	

87	
7.4.2	工作窃取算法
88	
7.4.3	线程的停止
92	
7.4.4	高性能高可靠性的Producer-Consumer模式实现
92	
7.5	Producer-Consumer模式的可复用实现代码
92	
7.6	Java标准库实例
93	
7.7	相关模式
93	
7.7.1	Guarded Suspension模式（第4章）
93	
7.7.2	Thread Pool模式（第9章）
93	
7.8	参考资源
93	
第8章	Active Object（主动对象）模式
94	
8.1	Active Object模式简介
94	
8.2	Active Object模式的架构
95	
8.3	Active Object模式实战案例解析
98	
8.4	Active Object模式的评价与实现考量
105	
8.4.1	错误隔离
107	
8.4.2	缓冲区监控
108	
8.4.3	缓冲区饱和和处理策略
108	
8.4.4	Scheduler空闲工作者线程清理
109	
8.5	Active Object模式的可复用实现代码
109	
8.6	Java标准库实例
111	
8.7	相关模式
112	
8.7.1	Promise模式（第6章）
112	
8.7.2	Producer-Consumer模式（第7章）
112	
8.8	参考资源
112	

第9章 Thread Pool（线程池）模式	113
9.1 Thread Pool模式简介	113
9.2 Thread Pool模式的架构	114
9.3 Thread Pool模式实战案例解析	116
9.4 Thread Pool模式的评价与实现考量	117
9.4.1 工作队列的选择	118
9.4.2 线程池大小调校	119
9.4.3 线程池监控	121
9.4.4 线程泄漏	122
9.4.5 可靠性与线程池饱和处理策略	122
9.4.6 死锁	125
9.4.7 线程池空闲线程清理	126
9.5 Thread Pool模式的可复用实现代码	127
9.6 Java标准库实例	127
9.7 相关模式	127
9.7.1 Two-phase Termination模式（第5章）	127
9.7.2 Promise模式（第6章）	127
9.7.3 Producer-Consumer模式（第7章）	127
9.8 参考资源	128
第10章 Thread Specific Storage（线程特有存储）模式	129
10.1 Thread Specific Storage模式简介	129
10.2 Thread Specific Storage模式的架构	131
10.3 Thread Specific Storage模式实战案例解析	133
10.4 Thread Specific Storage模式的评价与实现考量	135
10.4.1 线程池环境下使用Thread Specific Storage模式	

138	
10.4.2	内存泄漏与伪内存泄漏
139	
10.5	Thread Specific Storage模式的 reusable 实现代码
145	
10.6	Java标准库实例
146	
10.7	相关模式
146	
10.7.1	Immutable Object模式（第3章）
146	
10.7.2	Proxy（代理）模式
146	
10.7.3	Singleton（单例）模式
146	
10.8	参考资源
147	
第11章	Serial Thread Confinement（串行线程封闭）模式
148	
11.1	Serial Thread Confinement模式简介
148	
11.2	Serial Thread Confinement模式的架构
148	
11.3	Serial Thread Confinement模式实战案例解析
151	
11.4	Serial Thread Confinement模式的评价与实现考量
155	
11.4.1	任务的处理结果
155	
11.5	Serial Thread Confinement模式的 reusable 实现代码
156	
11.6	Java标准库实例
160	
11.7	相关模式
160	
11.7.1	Immutable Object模式（第3章）
160	
11.7.2	Promise模式（第6章）
160	
11.7.3	Producer-Consumer模式（第7章）
160	
11.7.4	Thread Specific Storage（线程特有存储）模式（第10章）
161	
11.8	参考资源
161	
第12章	Master-Slave（主仆）模式
162	
12.1	Master-Slave模式简介
162	

12.2 Master-Slave模式的架构	162
12.3 Master-Slave模式实战案例解析	164
12.4 Master-Slave模式的评价与实现考量	171
12.4.1 子任务的处理结果的收集	172
12.4.2 Slave参与者实例的负载均衡与工作窃取	173
12.4.3 可靠性与异常处理	173
12.4.4 Slave线程的停止	174
12.5 Master-Slave模式的 reusable 实现代码	174
12.6 Java标准库实例	186
12.7 相关模式	186
12.7.1 Two-phase Termination模式（第5章）	186
12.7.2 Promise模式（第6章）	186
12.7.3 Strategy（策略）模式	186
12.7.4 Template（模板）模式	186
12.7.5 Factory Method（工厂方法）模式	186
12.8 参考资源	187
第13章 Pipeline（流水线）模式	188
13.1 Pipeline模式简介	188
13.2 Pipeline模式的架构	189
13.3 Pipeline模式实战案例解析	194
13.4 Pipeline模式的评价与实现考量	208
13.4.1 Pipeline的深度	209
13.4.2 基于线程池的Pipe	209
13.4.3 错误处理	212
13.4.4 可配置的Pipeline	

212	
13.5 Pipeline模式的可复用实现代码	
212	
13.6 Java标准库实例	
222	
13.7 相关模式	
222	
13.7.1 Serial Thread Confinement模式（第11章）	
222	
13.7.2 Master-Slave模式（第12章）	
222	
13.7.3 Composite模式	
223	
13.8 参考资源	
223	
第14章 Half-sync/Half-async（半同步/半异步）模式	
224	
14.1 Half-sync/Half-async模式简介	
224	
14.2 Half-sync/Half-async模式的架构	
224	
14.3 Half-sync/Half-async模式实战案例解析	
226	
14.4 Half-sync/Half-async模式的评价与实现考量	
234	
14.4.1 队列积压	
235	
14.4.2 避免同步层处理过慢	
235	
14.5 Half-sync/Half-async模式的可复用实现代码	
236	
14.6 Java标准库实例	
240	
14.7 相关模式	
240	
14.7.1 Two-phase Termination模式（第5章）	
240	
14.7.2 Producer-Consumer模式（第7章）	
241	
14.7.3 Active Object模式（第8章）	
241	
14.7.4 Thread Pool模式（第9章）	
241	
14.8 参考资源	
241	
第15章 模式语言	
242	
15.1 模式与模式间的联系	
242	

15.2 mmutable Object（不可变对象）模式	244
15.3 Guarded Suspension（保护性暂挂）模式	244
15.4 Two-phase Termination（两阶段终止）模式	245
15.5 Promise（承诺）模式	246
15.6 Producer-Consumer（生产者/消费者）模式	247
15.7 Active Object（主动对象）模式	248
15.8 Thread Pool（线程池）模式	249
15.9 Thread Specific Storage（线程特有存储）模式	250
15.10 Serial Thread Confinement（串行线程封闭）模式	251
15.11 Master-Slave（主仆）模式	252
15.12 Pipeline（流水线）模式	253
15.13 Half-sync/Half-async（半同步/半异步）模式	254
附录 本书常用UML图指南	255
A.1 UML简介	255
A.2 类图（Class Diagram）	256
A.1.1 类的属性、方法和立体型（Stereotype）	256
A.1.2 类与类之间的关系	258
A.3 序列图（Sequence Diagram）	261
参考文献	263

精彩短评

1、特别好的书，过瘾

2、我是本书的作者。感谢@York Cai的反馈！该问题系排版问题，已经在本书第2次印刷中更正。由此给读者带来阅读上的不便在此表示歉意！

另外，本书的所有实战案例代码都是可以实际运行的，以便读者进行实验。

本书的源码可以从<https://github.com/Viscent>上下载，那里面的代码是我实际编写、测试过的。书上的毕竟都是贴上去并经过编辑、排版的。

另外，清单5-2那个类[io.github.viscent.mtpattern.ch14.hsha.example.AlarmSendingThread](https://github.com/viscent/mtpattern.ch14.hsha.example.AlarmSendingThread)贴自第14章的源码。

也欢迎读者就本书阅读过程中遇到的疑问在此或者知乎上提出。

我的知乎主页<https://www.zhihu.com/people/da-qian-56-87>。

3、感觉写的不上心，两阶段终止任务那章清单5-2代码不对，中间少了一些代码导致前后都不通，还有少了AlarmAgent的代码 @作者 书上的代码编译器能通过么

精彩书评

1、正在读。总体感受是作者行文流畅，文字表达条理清晰，同时辅以实战代码，把多线程编程技术中的关键点以及体会剖析的明明白白。近几年，随着中国软件技术的不断进步和成熟，涌现出不少对热点技术的原创书籍。这类书籍着重描写对技术的剖析、总结与归纳以及在实战中所获得的宝贵体会和经验。但是不得不说，能够把道理讲明白将彻底的不多，能够让人获得较大收获的不多。而该书则是中国软件原创书籍中少有的佳作，即把道理讲明白讲透彻，让人有较大收获。

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：www.tushu000.com