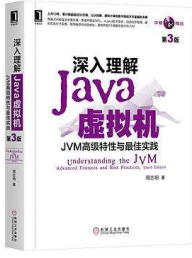


书名、作者、ISBN

购书单 电子图书 豆瓣书店 2021年度榜单 2021书影音报告 购物车

深入理解Java虚拟机（第3版）



作者: [周志明](#)

出版社: [机械工业出版社](#)

副标题: JVM高级特性与最佳实践

出版年: 2019-12

页数: 540

定价: 129.00元

装帧: 平装

丛书: [华章原创精品](#)

ISBN: 9787111641247

豆瓣评分

9.5

641人评价

5星 79.6%

4星 16.8%

3星 3.3%

2星 0.3%

1星 0.0%

想读 在读 读过 评价: ☆☆☆☆☆

写笔记 写书评 加入购书单 分享到

推荐

内容简介 · · · · · ·

内容介绍

这是一部从工作原理和工程实践两个维度深入剖析JVM的著作，是计算机领域公认的经典，繁体版在台湾也颇受欢迎。

自2011年上市以来，前两个版本累计印刷36次，销量超过30万册，两家主要网络书店的评论近90000条，内容上近乎零差评，是原创计算机图书领域不可逾越的丰碑。

第3版在第2版的基础上做了重大修订，内容更丰富、实战性更强：根据新版JDK对内容进行了全方位的修订和升级，围绕新技术和生产实践新增逾10万字，包含近50%的全新内容，并对第2版中含糊、瑕疵和错误内容进行了修正。

全书一共13章，分为五大部分：

第一部分（第1章）走近Java

系统介绍了Java的技术体系、发展历程、虚拟机家族，以及动手编译JDK，了解这部分内容能对学习JVM提供良好的指引。

第二部分（第2~5章）自动内存管理

详细讲解了Java的内存区域与内存溢出、垃圾收集器与内存分配策略、虚拟机性能监控与故障排除等与自动内存管理相关的内容，以及10余个经典的性能优化案例和优化方法；

第三部分（第6~9章）虚拟机执行子系统

深入分析了虚拟机执行子系统，包括类文件结构、虚拟机类加载机制、虚拟机字节码执行引擎，以及多个类加载及其执行子系统的实战案例；

第四部分（第10~11章）程序编译与代码优化

详细讲解了程序的前、后端编译与优化，包括前端的易用性优化措施，如泛型、主动装箱拆箱、条件编译等的内容的深入分析；以及后端的性能优化措施，如虚拟机的热点探测方法、HotSpot的即时编译器、提前编译器，以及各种常见的编译期优化技术；

第五部分（第12~13章）高效并发

主要讲解了Java实现高并发的原理，包括Java的内存模型、线程与协程，以及线程安全和锁优化。

全书以实战为导向，通过大量与实际生产环境相结合的案例分析和展示了解决各种Java技术难题的方案和技巧。

在线试读：
 豆瓣阅读
 得到

作者简介 · · · · · ·

资深Java技术、机器学习和企业级开发技术专家，现任远光软件研究院院长，人工智能博士在读，著有知名畅销书《深入理解Java虚拟机：JVM高级特性与最佳实践》。

https://book.douban.com/subject/34907497/

1/9

开源技术的积极倡导者和推动者，对计算机科学和相关的多个领域都有深刻的见解，尤其是人工智能、Java开发和敏捷开发等领域。曾受邀在InfoQ和IBM DeveloperWorks等网站撰写技术专栏。

著作颇丰，著有《深入理解Java虚拟机》《深入理解OSGi》，翻译了《Java虚拟机规范》等著作。其中《深入理解Java虚拟机》第1版出版于2011年，已经出至第2版，不仅销量好，而且口碑更好，累计印刷超过30次，仍长销不衰，是中文计算机图书领域公认的、难得一见的佳作。

目录 ·····

前言

致谢

第一部分 走近Java

第1章 走近Java 2

1.1 概述 2

1.2 Java技术体系 3

1.3 Java发展史 4

1.4 Java虚拟机家族 12

1.4.1 虚拟机始祖：Sun Classic/Exact VM 12

1.4.2 武林盟主：HotSpot VM 13

1.4.3 小家碧玉：Mobile/Embedded VM 14

1.4.4 天下第二：BEA JRockit/IBM J9 VM 15

1.4.5 软硬合璧：BEA Liquid VM/Azul VM 16

1.4.6 挑战者：Apache Harmony/Google Android Dalvik VM 17

1.4.7 没有成功，但并非失败：Microsoft JVM及其他 18

1.4.8 百家争鸣 19

1.5 展望Java技术的未来 21

1.5.1 无语言倾向 21

1.5.2 新一代即时编译器 23

1.5.3 向Native迈进 24

1.5.4 灵活的胖子 26

1.5.5 语言语法持续增强 27

1.6 实战：自己编译JDK 29

1.6.1 获取源码 29

1.6.2 系统需求 31

1.6.3 构建编译环境 33

1.6.4 进行编译 34

1.6.5 在IDE工具中进行源码调试 36

1.7 本章小结 39

第二部分 自动内存管理

第2章 Java内存区域与内存溢出异常 42

2.1 概述 42

2.2 运行时数据区域 42

2.2.1 程序计数器 43

2.2.2 Java虚拟机栈 43

2.2.3 本地方法栈 44

2.2.4 Java堆 44

2.2.5 方法区 46

2.2.6 运行时常量池 47

2.2.7 直接内存 47

2.3 HotSpot虚拟机对象探秘 48

2.3.1 对象的创建 48

2.3.2 对象的内存布局 51

2.3.3 对象的访问定位 52

2.4 实战：OutOfMemoryError异常 53

2.4.1 Java堆溢出 54

2.4.2 虚拟机栈和本地方法栈溢出 56

2.4.3 方法区和运行时常量池溢出 61

2.4.4 本机直接内存溢出 65

2.5 本章小结 66

第3章 垃圾收集器与内存分配策略 67

3.1 概述 67

3.2 对象已死？ 68

3.2.1 引用计数算法 68

3.2.2 可达性分析算法 70

3.2.3 再谈引用 71

3.2.4 生存还是死亡？ 72

3.2.5 回收方法区 74

3.3 垃圾收集算法 75

3.3.1 分代收集理论 75

3.3.2 标记-清除算法 77

3.3.3 标记-复制算法 78

- 3.3.4 标记-整理算法 79
- 3.4 HotSpot的算法细节实现 81
 - 3.4.1 根节点枚举 81
 - 3.4.2 安全点 82
 - 3.4.3 安全区域 83
 - 3.4.4 记忆集与卡表 84
 - 3.4.5 写屏障 85
 - 3.4.6 并发的可达性分析 87
- 3.5 经典垃圾收集器 89
 - 3.5.1 Serial收集器 90
 - 3.5.2 ParNew收集器 92
 - 3.5.3 Parallel Scavenge收集器 93
 - 3.5.4 Serial Old收集器 94
 - 3.5.5 Parallel Old收集器 95
 - 3.5.6 CMS收集器 96
 - 3.5.7 Garbage First收集器 98
- 3.6 低延迟垃圾收集器 104
 - 3.6.1 Shenandoah收集器 105
 - 3.6.2 ZGC收集器 112
- 3.7 选择合适的垃圾收集器 121
 - 3.7.1 Epsilon收集器 121
 - 3.7.2 收集器的权衡 121
 - 3.7.3 虚拟机及垃圾收集器日志 122
 - 3.7.4 垃圾收集器参数总结 127
- 3.8 实战：内存分配与回收策略 129
 - 3.8.1 对象优先在Eden分配 130
 - 3.8.2 大对象直接进入老年代 131
 - 3.8.3 长期存活的对象将进入老年代 132
 - 3.8.4 动态对象年龄判定 134
 - 3.8.5 空间分配担保 135
- 3.9 本章小结 137
- 第4章 虚拟机性能监控、故障处理工具 138
 - 4.1 概述 138
 - 4.2 基础故障处理工具 138
 - 4.2.1 jps：虚拟机进程状况工具 141
 - 4.2.2 jstat：虚拟机统计信息监视工具 142
 - 4.2.3 jinfo：Java配置信息工具 143
 - 4.2.4 jmap：Java内存映像工具 144
 - 4.2.5 jhat：虚拟机堆转储快照分析工具 145
 - 4.2.6 jstack：Java堆栈跟踪工具 146
 - 4.2.7 基础工具总结 148
 - 4.3 可视化故障处理工具 151
 - 4.3.1 JHSDb：基于服务性代理的调试工具 152
 - 4.3.2 JConsole：Java监视与管理控制台 157
 - 4.3.3 VisualVM：多合-故障处理工具 164
 - 4.3.4 Java Mission Control：可持续在线的监控工具 171
 - 4.4 HotSpot虚拟机插件及工具 175
 - 4.5 本章小结 180
- 第5章 调优案例分析与实战 181
 - 5.1 概述 181
 - 5.2 案例分析 181
 - 5.2.1 大内存硬件上的程序部署策略 182
 - 5.2.2 集群间同步导致的内存溢出 184
 - 5.2.3 堆外内存导致的溢出错误 185
 - 5.2.4 外部命令导致系统缓慢 187
 - 5.2.5 服务器虚拟机进程崩溃 187
 - 5.2.6 不恰当数据结构导致内存占用过大 188
 - 5.2.7 由Windows虚拟内存导致的长时间停顿 189
 - 5.2.8 由安全点导致长时间停顿 190
 - 5.3 实战：Eclipse运行速度调优 192
 - 5.3.1 调优前的程序运行状态 193
 - 5.3.2 升级JDK版本的性能变化及兼容问题 196
 - 5.3.3 编译时间和类加载时间的优化 200
 - 5.3.4 调整内存设置控制垃圾收集频率 203
 - 5.3.5 选择收集器降低延迟 206
 - 5.4 本章小结 209
- 第三部分 虚拟机执行子系统
- 第6章 类文件结构 212
 - 6.1 概述 212
 - 6.2 无关性的基石 212

6.3	Class类文件的结构	214
6.3.1	魔数与Class文件的版本	215
6.3.2	常量池	218
6.3.3	访问标志	224
6.3.4	类索引、父类索引与接口索引集合	225
6.3.5	字段表集合	226
6.3.6	方法表集合	229
6.3.7	属性表集合	230
6.4	字节码指令简介	251
6.4.1	字节码与数据类型	251
6.4.2	加载和存储指令	253
6.4.3	运算指令	254
6.4.4	类型转换指令	255
6.4.5	对象创建与访问指令	256
6.4.6	操作数栈管理指令	256
6.4.7	控制转移指令	257
6.4.8	方法调用和返回指令	257
6.4.9	异常处理指令	258
6.4.10	同步指令	258
6.5	公有设计，私有实现	259
6.6	Class文件结构的发展	260
6.7	本章小结	261
第7章	虚拟机类加载机制	262
7.1	概述	262
7.2	类加载的时机	263
7.3	类加载的过程	267
7.3.1	加载	267
7.3.2	验证	268
7.3.3	准备	271
7.3.4	解析	272
7.3.5	初始化	277
7.4	类加载器	279
7.4.1	类与类加载器	280
7.4.2	双亲委派模型	281
7.4.3	破坏双亲委派模型	285
7.5	Java模块化系统	287
7.5.1	模块的兼容性	288
7.5.2	模块化下的类加载器	290
7.6	本章小结	292
第8章	虚拟机字节码执行引擎	293
8.1	概述	293
8.2	运行时栈帧结构	294
8.2.1	局部变量表	294
8.2.2	操作数栈	299
8.2.3	动态连接	300
8.2.4	方法返回地址	300
8.2.5	附加信息	301
8.3	方法调用	301
8.3.1	解析	301
8.3.2	分派	303
8.4	动态类型语言支持	315
8.4.1	动态类型语言	316
8.4.2	Java与动态类型	317
8.4.3	java.lang.invoke包	318
8.4.4	invokedynamic指令	321
8.4.5	实战：掌控方法分派规则	324
8.5	基于栈的字节码解释执行引擎	326
8.5.1	解释执行	327
8.5.2	基于栈的指令集与基于寄存器的指令集	328
8.5.3	基于栈的解释器执行过程	329
8.6	本章小结	334
第9章	类加载及执行子系统的案例与实战	335
9.1	概述	335
9.2	案例分析	335
9.2.1	Tomcat：正统的类加载器架构	335
9.2.2	OSGi：灵活的类加载器架构	338
9.2.3	字节码生成技术与动态代理的实现	341
9.2.4	Backport工具：Java的时光机器	345
9.3	实战：自己动手实现远程执行功能	348
9.3.1	目标	348

9.3.2	思路	349
9.3.3	实现	350
9.3.4	验证	355
9.4	本章小结	356
第四部分 程序编译与代码优化		
第10章	前端编译与优化	358
10.1	概述	358
10.2	Javac编译器	359
10.2.1	Javac的源码与调试	359
10.2.2	解析与填充符号表	362
10.2.3	注解处理器	363
10.2.4	语义分析与字节码生成	364
10.3	Java语法糖的味道	367
10.3.1	泛型	367
10.3.2	自动装箱、拆箱与遍历循环	375
10.3.3	条件编译	377
10.4	实战：插入式注解处理器	378
10.4.1	实战目标	379
10.4.2	代码实现	379
10.4.3	运行与测试	385
10.4.4	其他应用案例	386
10.5	本章小结	386
第11章	后端编译与优化	388
11.1	概述	388
11.2	即时编译器	389
11.2.1	解释器与编译器	389
11.2.2	编译对象与触发条件	392
11.2.3	编译过程	397
11.2.4	实战：查看及分析即时编译结果	398
11.3	提前编译器	404
11.3.1	提前编译的优劣得失	405
11.3.2	实战：Javac的提前编译	408
11.4	编译器优化技术	411
11.4.1	优化技术概览	411
11.4.2	方法内联	415
11.4.3	逃逸分析	417
11.4.4	公共子表达式消除	420
11.4.5	数组边界检查消除	421
11.5	实战：深入理解Graal编译器	423
11.5.1	历史背景	423
11.5.2	构建编译调试环境	424
11.5.3	JVMCI编译器接口	426
11.5.4	代码中间表示	429
11.5.5	代码优化与生成	432
11.6	本章小结	436
第五部分 高效并发		
第12章	Java内存模型与线程	438
12.1	概述	438
12.2	硬件的效率与一致性	439
12.3	Java内存模型	440
12.3.1	主内存与工作内存	441
12.3.2	内存间交互操作	442
12.3.3	对于volatile型变量的特殊规则	444
12.3.4	针对long和double型变量的特殊规则	450
12.3.5	原子性、可见性与有序性	450
12.3.6	先行发生原则	452
12.4	Java与线程	455
12.4.1	线程的实现	455
12.4.2	Java线程调度	458
12.4.3	状态转换	460
12.5	Java与协程	461
12.5.1	内核线程的局限	461
12.5.2	协程的复苏	462
12.5.3	Java的解决方案	464
12.6	本章小结	465
第13章	线程安全与锁优化	466
13.1	概述	466
13.2	线程安全	466
13.2.1	Java语言中的线程安全	467
13.2.2	线程安全的实现方法	471

13.3 锁优化 479

13.3.1 自旋锁与自适应自旋 479

13.3.2 锁消除 480

13.3.3 锁粗化 481

13.3.4 轻量级锁 481

13.3.5 偏向锁 483

13.4 本章小结 485

附录A 在Windows系统下编译OpenJDK 6 486

附录B 展望Java技术的未来（2013年版） 493

附录C 虚拟机字节码指令表 499


附录D 对象查询语言（OQL）简介 506

附录E JDK历史版本轨迹 512

..... (收起)

原文摘录 (全部)

GC日志开头的"[GC"和"[Full GC"说明了这次垃圾收集的停顿类型，而不是 用来区分新生代GC还是老年代GC的。如果有"Full"，说明这次GC是发生了Stop-The-World 的 (查看原文)




RednaxelaFX

4 回复 15赞 2014-02-11 03:39:44

—— 引自第89页

由于使用了准确式内存管理，Exact VM可以抛弃掉以前Classic VM基于handler的对象查找方式（原因是GC后对象有可能会被移动位置，如果地址为123456的对象移动到654321，在没有明确信息表明内存中哪些数据是reference的前提下，那虚拟机是不敢把内存中所有为123456的值改成654321的，所以要使用句柄来保持reference值的稳定），这样每次定位对象都少了一次间接查找的开销，提升执行性能。(查看原文)



RednaxelaFX

4 回复 6赞 2013-07-08 17:33:47

—— 引自第10页

> 全部原文摘录

丛书信息

华章原创精品 (共24册), 这套丛书还有 《编译系统透视》,《c语言解惑》,《深入理解C++11》,《系统分析师UML实务手册》,《深入理解Java虚拟机》 等。

喜欢读"深入理解Java虚拟机（第3版）"的人也喜欢的电子书

支持 Web、iPhone、iPad、Android 阅读器



Java并发编程实战
25.00元



MySQL技术内幕：
InnoDB存储引擎
(第2版)
14.99元



架构整洁之道
27.99元



实战Java虚拟机
(第2版)
53.40元



Redis设计与实现
30.00元

喜欢读"深入理解Java虚拟机（第3版）"的人也喜欢



Effective Java中文
版（第3版）



数据密集型应用系
统设计



图解Java多线程设
计模式



微服务架构设计模
式



Kafka权威指南



高性能MySQL(第3版)



操作系统导论



图解设计模式



虚拟机设计与实现



算法（第4版）

短评 ····· (全部 181 条)

我来说两句

热门 / 最新 / 好友

adam2020-05-05 13:44:501 有用

讲gc，字节码，编译优化，并发等底层知识。

withpope2020-01-14 18:26:006 有用

针对最新版本jdk更新了一部分内容，但还是不太够，再版可能需要“重构”了

于小鱼2022-01-28 11:07:210 有用

第三版终于基于新的JDK 13了，依然经典，不过部分基于老JDK的内容还是没有删除或改写

mistwave2020-02-22 14:08:261 有用

不愧经典

TianbingJ2020-01-20 23:34:1113 有用

先打个4分怕周志明老师骄傲，读完再来评价。读完改成5星，两个礼拜读完，比第二版增加了不少内容。

> 更多短评 181 条

深入理解Java虚拟机（第3版）的书评 ····· (全部 76 条)

我要写书评

热门 / 最新 / 好友 / 只看本版本的评论

lcyFenix2011-07-26 23:08:53机械工业出版社2011版

大家好，我是本书作者

update 第三版最新的勘误地址为：https://github.com/fenixsoft/jvm_book 广告：下一本书的预告《[凤凰架构：构建可靠的大型分布式系统]》：[\[https://lcyfenix.cn\]](https://lcyfenix.cn) 这是一部以“如何构筑一套可靠的分布式大型软件系统”为叙事主线的开源文档，是一幅帮助开发人员整理现代软件... (展开)

△ 95 ▽ 2 19回应

pps10862011-07-08 10:30:13机械工业出版社2011版

对于国人能写出这样的书感到欣慰

在JavaEye上关注过这本书作者，拜读过他几篇关于虚拟机的文章。这本书一出版的第一时间我就买了，花了差不多1周时间囫圇吞枣地初读了一遍。10年前那本翻译外国的《深入Java虚拟机》我也读过，客观地说，对虚拟机细节的描写，这本书与10年前那本还是有差距，但是这本书的确... (展开)

△ 40 ▽ 1 19回应

风纪扣v2013-11-10 10:33:34机械工业出版社2011版

这是本好书，但还是不够

这本书我读到第8章，之后就是在读不下去了。读到后面是一种痛苦的体验，太多的东西是不全面的，大量的专有名词是没有解释的，读到最后很多东西仅仅是一个侧面，所以我觉得，这本书不适合初学者看，很适合有一定理论基础而缺乏实践的人去读。有时候... (展开)

△ 26 ▽ 4 5回应

 CodeMing

2019-12-23 17:42:29

JVM从入门到放弃？不要这么轻易就放弃啊

每一个JVM编程语言开发工程师，不管Java还是Groovy，Scala，Clojure，乃至新兴的Kotlin语言，在好奇心和求知欲驱使下，不断深度学习本领域的新东西时一定会接触到JVM。可能从基友，公众号，网络或书籍上了解到，然后走上了从入门到放弃之路。但是出来工作以后，JVM这一关，终... (展开)

△ 4

▽ 1 回应

 Chuan

2021-03-31 11:14:35

Java虚拟机一本通

正在二刷这本书，写的真是太精彩了，就Java虚拟机方面的书，特意逛了一下美亚，我发现：即使是英语世界--计算机科学的大本营，也找不到一本能和周老师这本书相媲美的书，建议周老师把它译成英文版，造福全世界的程序员。从我本人的经历来说，应该是Java领域的老兵了。我大学毕... (展开)

△ 3

▽ 0 回应

 摘浅的相思


2019-12-23 15:17:50

值得反复拜读的好书

每年读一遍《深入理解Java虚拟机（第2版）》，每年就会有一些新的感悟。技术总在更新，但思想原理不会过时，书中的思想部分吸收理解了对以后的工作学习有很大帮助。此书作为jvm基础入门的书很可以，相对全面易懂。本书选取了JDK7的JVM非常地恰到好处, 因为在其之后到目前虽然... (展开)

△ 2

▽ 0 回应

 看山


2019-12-12 11:06:54

深入理解Java虚拟机：JVM高级特性与最佳实践 读后感

写这篇读后感的缘由是这本书的第三版即将面世了，先拜谢周教授，相信很多人得益于周教授的这本书。14年入手的第二版，当时准备跳槽，需要理论知识武装一下。于是囫圇吞枣的扫了第一遍，就两个字，不懂，那种感觉就仿佛身在雾中，一切都看不通透，感觉浑身不舒服，想来是自己水... (展开)

△ 2

▽ 0 回应

 一缕炊烟

2019-12-12 09:26:39

Java虚拟机，有这本就够了

之前看过第二版，对于想了解jvm的小伙伴建议入手这本书。类加载、内存分析、垃圾收集机制等部分都是进入jvm大门的必备知识，如果想更深入了解，还有字节码层面的介绍，还结合了一些案例进行讲解。而第三版是在jdk12的基础上进行介绍的，想了解当前jvm的最新进展的小伙伴，本... (展开)

△ 2

▽ 0 回应

 Derek

2012-03-01 22:28:45

机械工业出版社2011版

java进阶必看

玩java不懂JVM那岂能行，此书乃java进阶必看的书籍，对java内存模型，JVM工作原理，基本JVM调优和内存溢出排查有很大的帮助。以前对于JVM运行机制不是很了解，遇到很多问题都无从下手，即便解决了也不懂其原理，读此书上述问题就游刃有余。 (展开)

△ 3

▽ 1 0 回应

> 更多书评 76篇

读书笔记 ·····

按有用程度 按页码先后 最新笔记

我来写笔记

 安全点和安全区域的具象化理解

元颯

在OopMap的协助下，HotSpot可以快速准确地完成GC Roots枚举，但一个很现实的问题随之而来：可能导致引用关系变化，或者说导致OopMap内容变化的指令非常多，如果为每一条指令都生成对应的OopMap，

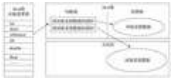
那将会需要大量的额外存储空间，这样垃圾收集伴随而来的空间成本就会变得无法忍受的高昂。实际上 HotSpot也的确没有为每条指令都生成OopMap，前面已经提到，只是在“特定的位置”记录了这些信息，这些位置被称为安全点（Safepoint） ...

2020-06-03 07:27:32



垃圾回收对引用对象地址值定位的影响

元颯



以前我一直有个问题，就是Java是面向对象语言，栈里面保存的是地址值。新生代的垃圾回收算法是复制算法。而复制算法意味着移动存活对象。那么，如果这个对象存活在栈中，而且垃圾回收的时候被复制了。虚拟机如何能在垃圾回收之后准确定位到这个对象的位置而不产生内存泄露。这个就涉及到Java虚拟机的对象访问方式，在这本书中提到以下说法：对象访问方式也是由虚拟机实现而定的，主流访问方式主要有使用句柄和直接指针两种...

2020-06-03 07:06:38

论坛 ·····

求此书pdf	来自藏心	2 回应	2022-01-15 17:09:58
三色标记算法为什么黑色对象不能直接引用白色对象呢	来自蓝猫和魂之挽歌	1 回应	2021-06-30 17:09:33
311页 showmethemoney	来自郑能亮	1 回应	2020-06-06 10:28:12
改变了国内技术数籍不如外国的印象	来自夏夜春秋秋意寒		2020-03-05 16:12:50
请问有必要买第3版吗？公司只到8	来自大刀	2 回应	2019-12-26 00:06:40