

《Java程序员修炼之道》

图书基本信息

书名：《Java程序员修炼之道》

13位ISBN编号：9787115321954

10位ISBN编号：7115321957

出版时间：2013-7

出版社：人民邮电出版社

作者：[英] Benjamin J. Evans,[荷兰] Martijn Verburg

页数：395

译者：吴海星

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.tushu111.com

《Java程序员修炼之道》

内容概要

本书分为四部分，第一部分全面介绍Java 7的新特性，第二部分探讨Java 关键编程知识和技术，第三部分讨论JVM 上的新语言和多语言编程，第四部分将平台和多语言编程知识付诸实践。从介绍Java 7 的新特性入手，本书涵盖了Java 开发中最重要的技术，比如依赖注入、测试驱动的开发和持续集成，探索了JVM 上的非Java 语言，并详细讲解了多语言项目，特别是涉及Groovy、Scala 和Clojure 语言的项目。此外，书中含有大量代码示例，帮助读者从实践中理解Java 语言和平台。本书适合Java 开发人员以及对Java7 和JVM 新语言感兴趣的各领域人士阅读。

《Java程序员修炼之道》

作者简介

作者简介：

Benjamin J. Evans

是伦敦Java用户组发起人、Java社区过程执行委员会成员。他拥有多年Java开发经验，现在是一家面向金融业的Java技术公司的CEO。

Martijn Verburg

是jClarity的CTO、伦敦Java用户组领导人。作为一名技术专家和众多初创企业的OSS导师，他拥有十多年的经验。Martijn经常应邀出席Java界的大型会议（JavaOne、Devoxx、OSCON、FOSDEM等）并发表演讲，人送雅号“开发魔头”，赞颂他敢于向行业现状挑战的精神。

译者简介：

吴海星

具有10多年的Java软件开发经验，熟悉Java语言规范、基于Java的Web软件开发以及性能调优，曾获SCJP及SCWCD证书。

书籍目录

目 录

第一部分 用Java 7做开发

第1章 初识Java 7 2

- 1.1 语言与平台 2
- 1.2 Coin项目：浓缩的都是精华 4
- 1.3 Coin项目中的修改 7
 - 1.3.1 switch语句中的String 7
 - 1.3.2 更强的数值文本表示法 8
 - 1.3.3 改善后的异常处理 9
 - 1.3.4 try-with-resources (TWR) 11
 - 1.3.5 钻石语法 13
 - 1.3.6 简化变参方法调用 14
- 1.4 小结 15

第2章 新I/O 17

- 2.1 Java I/O简史 18
 - 2.1.1 Java 1.0到1.3 19
 - 2.1.2 在Java 1.4中引入的NIO 19
 - 2.1.3 下一代I/O-NIO.2 20
- 2.2 文件I/O的基石：Path 20
 - 2.2.1 创建一个Path 23
 - 2.2.2 从Path中获取信息 23
 - 2.2.3 移除冗余项 24
 - 2.2.4 转换Path 25
 - 2.2.5 NIO.2 Path和Java已有的File类 25
- 2.3 处理目录和目录树 26
 - 2.3.1 在目录中查找文件 26
 - 2.3.2 遍历目录树 27
- 2.4 NIO.2的文件系统I/O 28
 - 2.4.1 创建和删除文件 29
 - 2.4.2 文件的复制和移动 30
 - 2.4.3 文件的属性 31
 - 2.4.4 快速读写数据 34
 - 2.4.5 文件修改通知 35
 - 2.4.6 SeekableByteChannel 37
- 2.5 异步 I/O操作 37
 - 2.5.1 将来式 38
 - 2.5.2 回调式 40
- 2.6 Socket和Channel的整合 41
 - 2.6.1 NetworkChannel 42
 - 2.6.2 MulticastChannel 42
- 2.7 小结 43

第二部分 关键技术

第3章 依赖注入 46

- 3.1 知识注入：理解IoC和DI 46
 - 3.1.1 控制反转 47
 - 3.1.2 依赖注入 48

3.1.3	转成DI	49
3.2	Java中标准化的DI	53
3.2.1	@Inject注解	54
3.2.2	@Qualifier注解	55
3.2.3	@Named注解	57
3.2.4	@Scope注解	57
3.2.5	@Singleton注解	57
3.2.6	接口Provider<T>	58
3.3	Java中的DI参考实现：Guice 3	59
3.3.1	Guice新手指南	59
3.3.2	水手绳结：Guice的各种绑定	62
3.3.3	在Guice中限定注入对象的生命周期	64
3.4	小结	66
第4章	现代并发	67
4.1	并发理论简介	68
4.1.1	解释Java线程模型	68
4.1.2	设计理念	69
4.1.3	这些原则如何以及为何会相互冲突	70
4.1.4	系统开销之源	71
4.1.5	一个事务处理的例子	71
4.2	块结构并发（Java 5之前）	72
4.2.1	同步与锁	73
4.2.2	线程的状态模型	74
4.2.3	完全同步对象	74
4.2.4	死锁	76
4.2.5	为什么是synchronized	77
4.2.6	关键字volatile	78
4.2.7	不可变性	79
4.3	现代并发应用程序的构件	80
4.3.1	原子类：java.util.concurrent.atomic	81
4.3.2	线程锁：java.util.concurrent.locks	81
4.3.3	CountDownLatch	85
4.3.4	ConcurrentHashMap	86
4.3.5	CopyOnWriteArrayList	87
4.3.6	Queue	90
4.4	控制执行	95
4.4.1	任务建模	96
4.4.2	ScheduledThread-ThreadPoolExecutor	97
4.5	分支/合并框架	98
4.5.1	一个简单的分支/合并例子	99
4.5.2	ForkJoinTask与工作窃取	101
4.5.3	并行问题	102
4.6	Java内存模型	103
4.7	小结	104
第5章	类文件与字节码	106
5.1	类加载和类对象	107
5.1.1	加载和连接概览	107
5.1.2	验证	108
5.1.3	Class对象	108

5.1.4	类加载器	109
5.1.5	示例：依赖注入中的类加载器	110
5.2	使用方法句柄	111
5.2.1	MethodHandle	112
5.2.2	MethodType	112
5.2.3	查找方法句柄	113
5.2.4	示例：反射、代理与方法句柄	114
5.2.5	为什么选择MethodHandle	116
5.3	检查类文件	117
5.3.1	介绍javap	117
5.3.2	方法签名的内部形式	118
5.3.3	常量池	119
5.4	字节码	121
5.4.1	示例：反编译类	121
5.4.2	运行时环境	123
5.4.3	操作码介绍	124
5.4.4	加载和储存操作码	125
5.4.5	数学运算操作码	125
5.4.6	执行控制操作码	126
5.4.7	调用操作码	126
5.4.8	平台操作操作码	127
5.4.9	操作码的快捷形式	127
5.4.10	示例：字符串拼接	127
5.5	invokedynamic	129
5.5.1	invokedynamic如何工作	129
5.5.2	示例：反编译invokedynamic调用	130
5.6	小结	132
第6章	理解性能调优	133
6.1	性能术语	134
6.1.1	等待时间	135
6.1.2	吞吐量	135
6.1.3	利用率	135
6.1.4	效率	135
6.1.5	容量	136
6.1.6	扩展性	136
6.1.7	退化	136
6.2	务实的性能分析法	136
6.2.1	知道你在测量什么	137
6.2.2	知道怎么测量	137
6.2.3	知道性能目标是什么	138
6.2.4	知道什么时候停止优化	139
6.2.5	知道高性能的成本	139
6.2.6	知道过早优化的危险	140
6.3	哪里出错了？我们担心的原因	140
6.3.1	过去和未来的性能趋势：摩尔定律	141
6.3.2	理解内存延迟层级	142
6.3.3	为什么Java性能调优存在困难	143
6.4	一个来自于硬件的时间问题	144
6.4.1	硬件时钟	144

6.4.2	麻烦的nanoTime()	144
6.4.3	时间在性能调优中的作用	146
6.4.4	案例研究：理解缓存未命中	147
6.5	垃圾收集	149
6.5.1	基本算法	149
6.5.2	标记和清除	150
6.5.3	jmap	152
6.5.4	与GC相关的JVM参数	155
6.5.5	读懂GC日志	156
6.5.6	用VisualVM查看内存使用情况	157
6.5.7	逸出分析	159
6.5.8	并发标记清除	160
6.5.9	新的收集器：G1	161
6.6	HotSpot的JIT编译	162
6.6.1	介绍HotSpot	163
6.6.2	内联方法	164
6.6.3	动态编译和独占调用	165
6.6.4	读懂编译日志	166
6.7	小结	167
第三部分 JVM上的多语言编程		
第7章 备选JVM语言 170		
7.1	Java 太笨？纯粹诽谤	170
7.1.1	整合系统	171
7.1.2	函数式编程的基本原理	172
7.1.3	映射与过滤器	173
7.2	语言生态学	174
7.2.1	解释型与编译型语言	175
7.2.2	动态与静态类型	175
7.2.3	命令式与函数式语言	176
7.2.4	重新实现的语言与原生语言	176
7.3	JVM上的多语言编程	177
7.3.1	为什么要用非Java语言	178
7.3.2	崭露头角的语言新星	179
7.4	如何挑选称心的非Java语言	180
7.4.1	低风险	181
7.4.2	与Java的交互操作	181
7.4.3	良好的工具和测试支持	182
7.4.4	备选语言学习难度	182
7.4.5	使用备选语言的开发者	182
7.5	JVM对备选语言的支持	183
7.5.1	非Java语言的运行时环境	183
7.5.2	编译器小说	184
7.6	小结	185
第8章 Groovy：Java的动态伴侣 187		
8.1	Groovy入门	189
8.1.1	编译和运行	189
8.1.2	Groovy控制台	190
8.2	Groovy 101：语法和语义	191
8.2.1	默认导入	192

8.2.2	数字处理	192
8.2.3	变量、动态与静态类型、作用域	193
8.2.4	列表和映射语法	195
8.3	与Java的差异——新手陷阱	196
8.3.1	可选的分号和返回语句	196
8.3.2	可选的参数括号	197
8.3.3	访问限定符	197
8.3.4	异常处理	198
8.3.5	Groovy中的相等	198
8.3.6	内部类	199
8.4	Java不具备的Groovy特性	199
8.4.1	GroovyBean	199
8.4.2	安全解引用操作符	200
8.4.3	猫王操作符	201
8.4.4	增强型字符串	201
8.4.5	函数字面值	202
8.4.6	内置的集合操作	203
8.4.7	对正则表达式的内置支持	204
8.4.8	简单的XML处理	205
8.5	Groovy与Java的合作	207
8.5.1	从Groovy调用Java	207
8.5.2	从Java调用Groovy	208
8.6	小结	211
第9章	Scala：简约而不简单	212
9.1	走马观花Scala	213
9.1.1	简约的Scala	213
9.1.2	match表达式	215
9.1.3	case类	217
9.1.4	actor	218
9.2	Scala能用在我的项目中吗	219
9.2.1	Scala和Java的比较	219
9.2.2	何时以及如何开始使用Scala	220
9.2.3	Scala可能不适合当前项目的迹象	220
9.3	让代码因Scala重新绽放	221
9.3.1	使用编译器和REPL	221
9.3.2	类型推断	222
9.3.3	方法	223
9.3.4	导入	224
9.3.5	循环和控制结构	224
9.3.6	Scala的函数式编程	225
9.4	Scala对象模型：相似但不同	226
9.4.1	一切皆对象	226
9.4.2	构造方法	228
9.4.3	特质	228
9.4.4	单例和伴生对象	230
9.4.5	case类和match表达式	232
9.4.6	警世寓言	234
9.5	数据结构和集合	235
9.5.1	List	235

9.5.2	Map	238
9.5.3	泛型	239
9.6	actor介绍	242
9.6.1	代码大舞台	242
9.6.2	用mailbox跟actor通信	243
9.7	小结	244
第10章	Clojure：更安全地编程	245
10.1	Clojure介绍	245
10.1.1	Clojure的Hello World	246
10.1.2	REPL入门	247
10.1.3	犯了错误	248
10.1.4	学着去爱括号	248
10.2	寻找Clojure：语法和语义	249
10.2.1	特殊形式新手营	249
10.2.2	列表、向量、映射和集	250
10.2.3	数学运算、相等和其他操作	252
10.3	使用函数和循环	253
10.3.1	一些简单的Clojure函数	253
10.3.2	Clojure中的循环	255
10.3.3	读取器宏和派发器	256
10.3.4	函数式编程和闭包	257
10.4	Clojure序列	258
10.4.1	懒序列	260
10.4.2	序列和变参函数	261
10.5	Clojure与Java的互操作	262
10.5.1	从Clojure中调用Java	262
10.5.2	Clojure值的Java类型	263
10.5.3	使用Clojure代理	264
10.5.4	用REPL做探索式编程	264
10.5.5	在Java中使用Clojure	265
10.6	Clojure并发	265
10.6.1	未来式与并行调用	266
10.6.2	ref形式	267
10.6.3	代理	271
10.7	小结	272
第四部分	多语种项目开发	
第11章	测试驱动开发	274
11.1	TDD概览	275
11.1.1	一个测试用例	276
11.1.2	多个测试用例	280
11.1.3	深入思考红—绿—重构循环	282
11.1.4	JUnit	283
11.2	测试替身	285
11.2.1	虚设对象	286
11.2.2	存根对象	287
11.2.3	伪装替身	290
11.2.4	模拟对象	295
11.3	ScalaTest	296
11.4	小结	298

第12章	构建和持续集成	300
12.1	与Maven 3相遇	302
12.2	Maven 3入门项目	303
12.3	用Maven 3构建Java7developer项目	305
12.3.1	POM	305
12.3.2	运行示例	311
12.4	Jenkins：满足CI需求	314
12.4.1	基础配置	315
12.4.2	设置任务	316
12.4.3	执行任务	319
12.5	Maven和Jenkins代码指标	320
12.5.1	安装Jenkins插件	321
12.5.2	用Checkstyle保持代码一致性	322
12.5.3	用FindBugs设定质量标杆	323
12.6	Leiningen	325
12.6.1	Leiningen入门	326
12.6.2	Leiningen的架构	326
12.6.3	Hello Lein	327
12.6.4	用Leiningen做面向REPL的TDD	329
12.6.5	用Leiningen打包和部署	330
12.7	小结	332
第13章	快速Web开发	333
13.1	Java Web框架的问题	334
13.1.1	Java编译为什么不好	335
13.1.2	静态类型为什么不好	335
13.2	选择Web框架的标准	336
13.3	Grails入门	338
13.4	Grails快速启动项目	338
13.4.1	创建域对象	340
13.4.2	测试驱动开发	340
13.4.3	域对象持久化	342
13.4.4	创建测试数据	343
13.4.5	控制器	343
13.4.6	GSP/JSP页面	344
13.4.7	脚手架和UI的自动化创建	346
13.4.8	快速周转的开发	347
13.5	深入Grails	347
13.5.1	日志	347
13.5.2	GORM：对象关系映射	348
13.5.3	Grails插件	349
13.6	Compojure入门	350
13.6.1	Hello Compojure	350
13.6.2	Ring和路由	352
13.6.3	Hiccup	353
13.7	我是不是一只水獭	353
13.7.1	项目设置	354
13.7.2	核心函数	357
13.8	小结	359
第14章	保持优秀	361

14.1	对Java 8的期待	361
14.1.1	lambda表达式（闭包）	362
14.1.2	模块化（拼图Jigsaw）	363
14.2	多语言编程	365
14.2.1	语言的互操作性及元对象协议	365
14.2.2	多语言模块化	366
14.3	未来的并发趋势	367
14.3.1	多核的世界	367
14.3.2	运行时管理的并发	367
14.4	JVM的新方向	368
14.4.1	VM的合并	368
14.4.2	协同程序	369
14.4.3	元组	370
14.5	小结	372
附录A	java7developer：源码安装	373
附录B	glob模式语法及示例	380
附录C	安装备选JVM语言	382
附录D	Jenkins的下载和安装	388
附录E	java7developer：Maven POM	390

《Java程序员修炼之道》

精彩短评

《Java程序员修炼之道》

精彩书评

- 1、介绍java7特性的书，章节组织的还算合理。可惜每一章内容都没有什么深入的介绍。比如说forkjoin框架，应该是一个Java7中引入的很重要的并发编程框架，才讲了两页，看完之后还是不会用，只好再去看oracle的文档去。
- 2、这本书更像是一本java的学习地图，涉及的点挺多，有JDK7的一些新特性，TDD，JVM内存回收以及JVM上运行的非JAVA语言的介绍（Groovy,Scala,Closure）等，但是讲的都不深入，偏入门级的。唯一觉得介绍的还比较好的是Scala和Closure的篇幅，但作为早已经了解过这些语言的人来说，就没有那么有吸引力了。而且这部分的介绍Programming系列图书介绍的更好，如《Programming Scala》等。总而言之，不是Java初学者，不推荐看。

章节试读

1、《Java程序员修炼之道》的笔记-第67页

本章主要讲Java5之后的现代并发，同时给传统多线程同步互锁之类的并发编程起了个名字叫块结构并发。

2、《Java程序员修炼之道》的笔记-第233页

一个非常强的理论：不可能在没出现的源码中弄出bug来。

3、《Java程序员修炼之道》的笔记-初始Java7

- 1.更强的数值文本表示法
- 2.异常处理改善
- try-with-resource
- 3.钻石语法

4、《Java程序员修炼之道》的笔记-第260页

lazy 在大部分计算机科学的书籍当中应翻译为惰性
如：
惰性求值
惰性序列

5、《Java程序员修炼之道》的笔记-第95页

代码清单4-14在new LinkedTransferQueue时试图传入一个int参数，难道是界限？但是我看了下JDK没这个构造函数签名。
Javadoc 也说这种Queue是无界的。
所以代码有误。

6、《Java程序员修炼之道》的笔记-第178页

从多语言使用角度切分系统架构的多语言编程金字塔自顶向下分别是：特定领域(语言DSL)层、动态语言层、稳定(静态语言)层。Java适合于最下面的稳定层。

7、《Java程序员修炼之道》的笔记-第304页

模因（Meme）也称弥，是文化资讯传承单位。
瞬间上纲上线到传播学了。

《Java程序员修炼之道》

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.tushu111.com