

<<深入理解Java7>>

图书基本信息

书名：<<深入理解Java7>>

13位ISBN编号：9787111380399

10位ISBN编号：7111380398

出版时间：2012-5

出版时间：机械工业出版社

作者：成富

页数：467

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：<http://www.tushu007.com>

<<深入理解Java7>>

前言

前言：为什么要写这本书 我最早开始接触Java语言是在大学的时候。当时除了用Java开发一些小程序之外，就是用Struts框架开发Web应用。在后来的实习和工作中，我对Java的使用和理解更加深入，逐渐涉及Java相关的各种不同技术。使用Java语言的一个深刻体会是：Java语言虽然上手容易，但是要真正掌握并不容易。

Java语言对开发人员屏蔽了一些与底层实现相关的细节，但是仍然有很多内容对开发人员来说是很复杂的，这些内容恰好是容易出现错误的地方。

我在工作中就经常遇到与类加载器和垃圾回收相关的问题。

在解决这些问题的过程中，我积累了一些经验，遇到类似的问题可以很快地找到问题的根源。

同时，在解决这些实际问题的过程中，我意识到虽然可以解决某些具体的问题，但是并没有真正理解这些问题的解决办法背后所蕴含的基本原理，仍然还只是处于一个“知其然，不知其所以然”的状态。

于是我开始阅读Java相关的基础资料，包括Java语言规范、Java虚拟机规范、Java类库的源代码和其他在线资料等。

在阅读的基础上，编写小程序进行测试和试验。

通过阅读和实践，我对Java平台中的一些基本概念有了更加深入的理解。

从2010年开始，我对积累的相关知识进行了整理，在InfoQ中文站的“Java深度历险”专栏上发表出来，受到了一定的关注。

2011年7月，在时隔数年之后，Java的一个重大版本Java SE 7发布了。

在这个新的版本中，Java平台增加了很多新的特性。

在Java虚拟机方面，invokedynamic指令的加入使虚拟机上的动态语言的性能得到很大的提升。

这使得开发人员可以享受到动态语言带来的在提高生产效率方面的好处。

在Java语言方面，语言本身的进一步简化，使开发人员编写代码的效率更高。

在Java类库方面，新的IO库和同步实用工具类为开发人员提供了更多实用的功能。

从另外一个角度来说，Java SE 7是Oracle公司收购Sun公司之后发布的第一个Java版本，从侧面反映出了Oracle公司对Java社区的领导力，可以继续推动Java平台向前发展。

这可以打消企业和社区对于Oracle公司领导力的顾虑。

Java SE 7的发布也证明了基于JCP和OpenJDK的社区驱动模式可以很好地推动Java向前发展。

随着新版本的发布，肯定会有越来越多的开发人员想尝试使用Java SE 7中的新特性，毕竟开发者社区对这个新版本期待了太长的时间。

在Java程序中使用这些新特性，可以提高代码质量，提升工作效率。

Java平台的每个版本都致力于提高Java程序的运行性能。

随着新版本的发布，企业都应该考虑把Java程序的运行平台升级到最新的Java SE 7，这样可以享受到性能提升所带来的好处。

对于新的Java程序开发，推荐使用Java SE 7作为标准的运行平台。

本书将Java SE 7中的新特性介绍和对Java平台的深入探讨结合起来，让读者既可以了解最新版本的Java平台的新特性，又可以对Java平台的底层细节有更加深入的理解。

读者对象及如何阅读本书 本书面向的主要读者是具备一定Java基础的开发人员和在校学生。

本书中不涉及Java的基本语法，因此不适合Java初学者阅读。

如果只对Java SE 7中的新特性感兴趣，可以阅读第1章到第6章；如果对Java中的特定主题感兴趣，可以根据目录有选择地阅读。

另外，第1章到第6章虽然以Java SE 7的新特性介绍为主，但是其中也穿插了对相关内容的深入探讨。

本书可分为三大部分：第一部分为Java SE 7新特性介绍，从第1章到第6章。

这部分详细地介绍了Java SE 7中新增的重要特性。

在对新特性的介绍中，也包含了对Java平台相关内容的详细介绍。

第二部分为Java SE 7的深入探讨，从第7章到第13章。

<<深入理解Java7>>

这部分着重讲解了Java平台上的底层实现，并对一些重要的特性进行了深入探讨。

这个部分所涉及的内容包括Java虚拟机、Java源代码和字节代码操作、Java类加载器、对象生命周期、多线程与并发编程实践、Java泛型和Java安全。

第三部分为Java SE 8的内容展望，即第14章。

这部分简要介绍了Java SE 8中将要增加的新特性。

本书还通过两个附录对OpenJDK（附录A）和Java语言的历史（附录B）进行了简要的介绍。

勘误和支持 由于作者的水平有限，编写时间仓促，书中难免会出现一些错误或者不准确的地方，恳请读者批评指正。

致谢 感谢InfoQ中文站和InfoQ编辑张凯峰先生。

这本书能够面世，得益于我在InfoQ中文站的“Java深度历险”专栏上发表的文章。

感谢机械工业出版社华章公司的编辑杨福川和姜影的辛勤工作，使得这本书能够最终顺利完成。

感谢家人和朋友对我的支持与帮助！

<<深入理解Java7>>

内容概要

本书是学习Java 7新功能和特性以及深入理解Java核心技术的最佳选择之一。经过近6年的等待，Java迎来了它的又一个历史性的版本—Java 7。Java 7在提高开发人员的生产效率、平台性能和模块方向上又迈进了一步，变得更加强大和灵活。本书不仅对Java 7的所有重要更新进行了全面的解读，而且还对Java平台的核心技术的底层实现进行了深入探讨，包含大量最佳实践。

全书的主要内容可分为三大部分：第一部分是1~6章，全面阐释Java 7在语法、JVM、类库和API等方面的所有重要新功能和特性，掌握这部分内容有助于大幅度提升编码效率和提高代码质量；第二部分是7~13章，对JVM、Java源代码和字节码操作、类加载器、对象生命周期、多线程、并发编程、泛型、安全等Java平台的核心技术进行了深入解析，掌握这部分内容有助于深入理解Java的底层原理；第三部分为第14章，是对Java 8的展望，简要介绍了Java 8中将要增加的新特性。

<<深入理解Java7>>

作者简介

成富 资深Java软件工程师，有多年Java企业级应用开发经验，对Java 7和Java平台的各项技术的底层原理有深入透彻的研究。曾就职于IBM中国研发中心，先后在IBM新技术学院和Lotus部门参与了多个重要产品的开发工作，现就职于新西兰PropellerHead公司。他是非常受欢迎的技术作家，在IBM developerWorks上发表中英文技术文章近30篇，获得了其颁发的“极具人气作者奖”；他还是知名技术网站InfoQ的专栏作家，撰写了“Java深度历险”专栏，共发表技术文章10余篇。此外，他还非常精通HTML 5、CSS 3、JavaScript等Web 2.0核心技术，实战经验丰富。

<<深入理解Java7>>

书籍目录

前言

Java的挑战与展望

第1章 Java 7语法新特性

- 1.1 Coin项目介绍
- 1.2 在switch语句中使用字符串
 - 1.2.1 基本用法
 - 1.2.2 实现原理
 - 1.2.3 枚举类型
- 1.3 数值字面量的改进
 - 1.3.1 二进制整数字面量
 - 1.3.2 在数值字面量中使用下划线
- 1.4 优化的异常处理
 - 1.4.1 异常的基础知识
 - 1.4.2 创建自己的异常
 - 1.4.3 处理异常
 - 1.4.4 Java 7的异常处理新特性
- 1.5 try-with-resources语句
- 1.6 优化变长参数的方法调用
- 1.7 小结

第2章 Java语言的动态性

- 2.1 脚本语言支持API
 - 2.1.1 脚本引擎
 - 2.1.2 语言绑定
 - 2.1.3 脚本执行上下文
 - 2.1.4 脚本的编译
 - 2.1.5 方法调用
 - 2.1.6 使用案例
- 2.2 反射API
 - 2.2.1 获取构造方法
 - 2.2.2 获取域
 - 2.2.3 获取方法
 - 2.2.4 操作数组
 - 2.2.5 访问权限与异常处理
- 2.3 动态代理
 - 2.3.1 基本使用方式
 - 2.3.2 使用案例
- 2.4 动态语言支持
 - 2.4.1 Java语言与Java虚拟机
 - 2.4.2 方法句柄
 - 2.4.3 invokedynamic指令
- 2.5 小结

第3章 Java IO

- 3.1 流
 - 3.1.1 基本输入流
 - 3.1.2 基本输出流

<<深入理解Java7>>

- 3.1.3 输入流的复用
- 3.1.4 过滤输入输出流
- 3.1.5 其他输入输出流
- 3.1.6 字符流
- 3.2 缓冲区
 - 3.2.1 基本用法
 - 3.2.2 字节缓冲区
 - 3.2.3 缓冲区视图
- 3.3 通道
 - 3.3.1 文件通道
 - 3.3.2 套接字通道
- 3.4 NIO.2
 - 3.4.1 文件系统访问
 - 3.4.2 zipjar文件系统
 - 3.4.3 异步IO通道
 - 3.4.4 套接字通道绑定与配置
 - 3.4.5 IP组播通道
- 3.5 使用案例
- 3.6 小结
- 第4章 国际化与本地化
 - 4.1 国际化概述
 - 4.2 Unicode
 - 4.2.1 Unicode编码格式
 - 4.2.2 其他字符集
 - 4.2.3 Java与Unicode
 - 4.3 Java中的编码实践
 - 4.3.1 Java NIO中的编码器和解码器
 - 4.3.2 乱码问题详解
 - 4.4 区域设置
 - 4.4.1 IETF BCP 47
 - 4.4.2 资源包
 - 4.4.3 日期和时间
 - 4.4.4 数字和货币
 - 4.4.5 消息文本
 - 4.4.6 默认区域设置的类别
 - 4.4.7 字符串比较
 - 4.5 国际化与本地化基本实践
 - 4.6 小结
- 第5章 图形用户界面
 - 5.1 Java图形用户界面概述
 - 5.2 AWT
 - 5.2.1 重要组件类
 - 5.2.2 任意形状的窗口
 - 5.2.3 半透明窗口
 - 5.2.4 组件混合
 - 5.3 Swing
 - 5.3.1 重要组件类

<<深入理解Java7>>

- 5.3.2 JLayer组件和LayerUI类
- 5.4 事件处理与线程安全性
 - 5.4.1 事件处理
 - 5.4.2 事件分发线程
 - 5.4.3 SwingWorker类
 - 5.4.4 SecondaryLoop接口
- 5.5 界面绘制
 - 5.5.1 AWT中的界面绘制
 - 5.5.2 Swing中的绘制
- 5.6 可插拔式外观样式
- 5.7 JavaFX
 - 5.7.1 场景图
 - 5.7.2 变换
 - 5.7.3 动画效果
 - 5.7.4 FXML
 - 5.7.5 CSS外观描述
 - 5.7.6 Web引擎与网页显示
- 5.8 使用案例
- 5.9 小结
- 第6章 Java 7其他重要更新
 - 6.1 关系数据库访问
 - 6.1.1 使用try-with-resources语句
 - 6.1.2 数据库查询的默认模式
 - 6.1.3 数据库连接超时时间与终止
 - 6.1.4 语句自动关闭
 - 6.1.5 RowSet实现提供者
 - 6.2 java.lang包的更新
 - 6.2.1 基本类型的包装类
 - 6.2.2 进程使用
 - 6.2.3 Thread类的更新
 - 6.3 Java实用工具类
 - 6.3.1 对象操作
 - 6.3.2 正则表达式
 - 6.3.3 压缩文件处理
 - 6.4 JavaBeans组件
 - 6.4.1 获取组件信息
 - 6.4.2 执行语句和表达式
 - 6.4.3 持久化
 - 6.5 小结
- 第7章 Java虚拟机
 - 7.1 虚拟机基本概念
 - 7.2 内存管理
 - 7.3 引用类型
 - 7.3.1 强引用
 - 7.3.2 引用类型基本概念
 - 7.3.3 软引用
 - 7.3.4 弱引用

<<深入理解Java7>>

- 7.3.5 幽灵引用
- 7.3.6 引用队列
- 7.4 Java本地接口
 - 7.4.1 JNI基本用法
 - 7.4.2 Java程序中集成CC++代码
 - 7.4.3 在CC++程序中启动Java虚拟机
- 7.5 HotSpot虚拟机
 - 7.5.1 字节代码执行
 - 7.5.2 垃圾回收
 - 7.5.3 启动参数
 - 7.5.4 分析工具
 - 7.5.5 Java虚拟机工具接口
- 7.6 小结
- 第8章 Java源代码和字节代码操作
 - 8.1 Java字节代码格式
 - 8.1.1 基本格式
 - 8.1.2 常量池的结构
 - 8.1.3 属性
 - 8.2 动态编译Java源代码
 - 8.2.1 使用javac工具
 - 8.2.2 Java编译器API
 - 8.2.3 使用Eclipse JDT编译器
 - 8.3 字节代码增强
 - 8.3.1 使用ASM
 - 8.3.2 增强代理
 - 8.4 注解
 - 8.4.1 注解类型
 - 8.4.2 创建注解类型
 - 8.4.3 使用注解类型
 - 8.4.4 处理注解
 - 8.5 使用案例
 - 8.6 小结
- 第9章 Java类加载器
 - 9.1 类加载器概述
 - 9.2 类加载器的层次结构与代理模式
 - 9.3 创建类加载器
 - 9.4 类加载器的隔离作用
 - 9.5 线程上下文类加载器
 - 9.6 Class.forName方法
 - 9.7 加载资源
 - 9.8 Web应用中的类加载器
 - 9.9 OSGi中的类加载器
 - 9.9.1 OSGi基本的类加载器机制
 - 9.9.2 Equinox框架的类加载实现机制
 - 9.9.3 Equinox框架嵌入到Web容器中
 - 9.10 小结
- 第10章 对象生命周期

<<深入理解Java7>>

- 10.1 Java类的链接
- 10.2 Java类的初始化
- 10.3 对象的创建与初始化
- 10.4 对象终止
- 10.5 对象复制
- 10.6 对象序列化
 - 10.6.1 默认的对象序列化
 - 10.6.2 自定义对象序列化
 - 10.6.3 对象替换
 - 10.6.4 版本更新
 - 10.6.5 安全性
 - 10.6.6 使用Externalizable接口
- 10.7 小结
- 第11章 多线程与并发编程实践
 - 11.1 多线程
 - 11.1.1 可见性
 - 11.1.2 Java内存模型
 - 11.1.3 volatile关键词
 - 11.1.4 final关键词
 - 11.1.5 原子操作
 - 11.2 基本线程同步方式
 - 11.2.1 synchronized关键词
 - 11.2.2 Object类的wait、notify和notifyAll方法
 - 11.3 使用Thread类
 - 11.3.1 线程状态
 - 11.3.2 线程中断
 - 11.3.3 线程等待、睡眠和让步
 - 11.4 非阻塞方式
 - 11.5 高级实用工具
 - 11.5.1 高级同步机制
 - 11.5.2 底层同步器
 - 11.5.3 高级同步对象
 - 11.5.4 数据结构
 - 11.5.5 任务执行
 - 11.6 Java SE 7新特性
 - 11.6.1 轻量级任务执行框架forkjoin
 - 11.6.2 多阶段线程同步工具
 - 11.7 ThreadLocal类
 - 11.8 小结
- 第12章 Java泛型
 - 12.1 泛型基本概念
 - 12.2 类型擦除
 - 12.3 上界和下界
 - 12.4 通配符
 - 12.5 泛型与数组
 - 12.6 类型系统
 - 12.7 覆写与重载

<<深入理解Java7>>

- 12.7.1 覆写对方法类型签名的要求
- 12.7.2 覆写对返回值类型的要求
- 12.7.3 覆写对异常声明的要求
- 12.7.4 重载
- 12.8 类型推断和<>操作符
- 12.9 泛型与反射API
- 12.10 使用案例
- 12.11 小结
- 第13章 Java安全
 - 13.1 Java安全概述
 - 13.2 用户认证
 - 13.2.1 主体、身份标识与凭证
 - 13.2.2 登录
 - 13.3 权限控制
 - 13.3.1 权限、策略与保护域
 - 13.3.2 访问控制权限
 - 13.3.3 特权动作
 - 13.3.4 访问控制上下文
 - 13.3.5 守卫对象
 - 13.4 加密与解密、报文摘要和数字签名
 - 13.4.1 Java密码框架
 - 13.4.2 加密与解密
 - 13.4.3 报文摘要
 - 13.4.4 数字签名
 - 13.5 安全套接字连接
 - 13.5.1 SSL协议
 - 13.5.2 HTTPS
 - 13.6 使用案例
 - 13.7 小结
- 第14章 超越Java 7
 - 14.1 lambda表达式
 - 14.1.1 函数式接口
 - 14.1.2 lambda表达式的语法
 - 14.1.3 目标类型
 - 14.1.4 词法作用域
 - 14.1.5 方法引用
 - 14.1.6 接口的默认方法
 - 14.2 Java平台模块化
 - 14.3 Java SE 8的其他更新
 - 14.4 小结
- 附录A OpenJDK
- 附录B Java简史 4483.4 加密与解密、报文摘要和数字签名
 - 13.4.1 Java密码框架
 - 13.4.2 加密与解密
 - 13.4.3 报文摘要
 - 13.4.4 数字签名
 - 13.5 安全套接字连接

<<深入理解Java7>>

- 13.5.1 SSL协议
- 13.5.2 HTTPS
- 13.6 使用案例
- 13.7 小结
- 第14章 超越Java 7
 - 14.1 lambda表达式
 - 14.1.1 函数式接口
 - 14.1.2 lambda表达式的语法
 - 14.1.3 目标类型
 - 14.1.4 词法作用域
 - 14.1.5 方法引用
 - 14.1.6 接口的默认方法
 - 14.2 Java平台模块化
 - 14.3 Java SE 8的其他更新
 - 14.4 小结
- 附录A OpenJDK
- 附录B Java简史

章节摘录

版权页： 1.1 Coin项目介绍 在介绍具体的新特性之前，有必要介绍一下Coin项目。

OpenJDK中的Coin项目的目的是维护对Java语言所做的语法增强。

在Coin项目开始之初，曾经广泛地向社区征求提议。

在短短的一个月时间内就收到了近70条提议。

最后有9条提议被列入考虑之中。

在这9条提议中，有6条成为Java 7的一部分，剩下的2条提议会在Java 8中重新考虑，还有1条提议被移到其他项目中实现。

这6条被接纳的提议除了本章会介绍的在switch语句中使用字符串、数值字面量的改进、优化的异常处理、try-with-resources语句和优化变长参数的方法调用之外，还包括第12章中会介绍的简化泛型类创建的""操作符。

在Java 8中考虑的2条提议则分别是集合类字面量和为List和Map提供类似数组的按序号的访问方式。

和其他对Java平台所做的修改一样，Coin项目所建议的修改也需要通过JCP来完成。

这些改动以JSR 334 (Small Enhancements to the Java™ Programming Language) 的形式提交到JCP。

1.2在SWitch语句中使用字符串 对于switch语句，开发人员应该都不陌生。

大部分编程语言中都有类似的语法结构，用来根据某个表达式的值选择要执行的语句块。

对于switch语句中的条件表达式类型，不同编程语言所提供的支持是不一样的。

对于Java语言来说，在Java 7之前，switch语句中的条件表达式的类型只能是与整数类型兼容的类型，包括基本类型char、byte、short和int，与这些基本类型对应的封装类Character、Byte、Short和Integer，还有枚举类型。

这样的限制降低了语言的灵活性，使开发人员在需要根据其他类型的表达式来进行条件选择时，不得不增加额外的代码来绕过这个限制。

为此，Java 7放宽了这个限制，额外增加了一种可以在switch语句中使用的表达式类型，那就是很常见的字符串，即String类型。

1.2.1 基本用法 在基于Java 7的代码中使用这个新特性非常简单，因为这个新特性并没有改变switch的语法含义，只是多了一种开发人员可以选择的条件判断的数据类型。

但是这个简单的新特性却带来了重大的影响，因为根据字符串进行条件判断在开发中是很常见的。

考虑这样一个应用情景，在程序中需要根据用户的性别来生成合适的称谓，比如男性就使用"×××先生"，女性就使用"×××女士"。

判断条件的类型可以是字符串，如"男"表示男性，"女"表示女性。

不过这在Java 7之前的switch语句中是行不通的，之前只能添加额外的代码先将字符串转换成整数类型

。

<<深入理解Java7>>

媒体关注与评论

备受推崇：Java 7六年磨一剑，它在前一个版本上发生了很大的变化，引入了非常多激动人心的新特性和新功能，在语法、Java虚拟机、I/O、安全性、并发和国际化等方面都有重要的更新。作为国内第一本Java 7方面的专著，本书全面介绍了这些重要的更新，对有一定开发经验的Java程序员来说非常宝贵，能极大地降低他们的学习成本。

此外，本书还探讨了各项Java技术的底层原理，对于想深入学习Java的读者尤为有价值。

强烈推荐！

——51CTO (www.51cto.com) 中国领先的IT技术网站 长期以来，Java一直雄踞TIOBE编程语言排行榜的首位，它拥有数量最庞大的开发者社区。

Java的每一次版本更新都会对语言、虚拟机和API类库等进行重要更新，Java 7这个版本尤为明显。

Java 7的一个显著改变就是它提供的新特性能让开发者通过简化代码来提升开发效率和编码质量。

本书以“深入理解”为宗旨，对Java 7中新增的内容和之前版本中已经非常成熟的核心技术都进行了非常深入的讨论，是系统学习Java 7和Java技术进阶的必备参考书。

书中还给出了非常多的代码示例，可以帮助读者更好地在实际中应用这些知识。

——秦小波 资深Java技术专家 著有畅销书《设计模式之禅》和《编写高质量代码：改善Java程序的151个建议》 学习一门技术，要想做到“知其然”并不难，难的是要“知其所以然”，而且这更重要，也是考查一个技术人员真实实力的标杆。

想跨越这个标杆，除了自己付出时间和精力之外，高人的点拨和指导更能起到事半功倍之效。

成富是InfoQ中文站的资深专栏作者，从读者对他的“Java深度历险”系列技术文章的反馈可以看出：他不仅能帮你“知其然”，而且还能通过深入浅出的讲解让你“知其所以然”。

他的这本书一定能让你的Java技能更上一层楼。

——郑柯 InfoQ中文站总编辑

<<深入理解Java7>>

编辑推荐

《深入理解Java 7:核心技术与最佳实践》是学习Java 7新功能和特性以及深入理解Java核心技术的最佳选择之一。

<<深入理解Java7>>

名人推荐

Java 7六年磨一剑，它在前一个版本上发生了很大的变化，引入了非常多激动人心的新特性和新功能，在语法、Java虚拟机、I/O、安全性、并发和国际化等方面都有重要的更新。

作为国内第一本Java 7方面的专著，本书全面介绍了这些重要的更新，对有一定开发经验的Java程序员来说非常宝贵，能极大地降低他们的学习成本。

此外，本书还探讨了各项Java技术的底层原理，对于想深入学习Java的读者尤为有价值。

强烈推荐！

——51CTO (www.51cto.com) 中国领先的IT技术网站 长期以来，Java一直雄踞TIOBE编程语言排行榜的首位，它拥有数量最庞大的开发者社区。

Java的每一次版本更新都会对语言、虚拟机和API类库等进行重要更新，Java 7这个版本尤为明显。

Java 7的一个显著改变就是它提供的新特性能让开发者通过简化代码来提升开发效率和编码质量。

本书以“深入理解”为宗旨，对Java 7中新增的内容和之前版本中已经非常成熟的核心技术都进行了非常深入的讨论，是系统学习Java 7和Java技术进阶的必备参考书。

书中还给出了非常多的代码示例，可以帮助读者更好地在实际中应用这些知识。

——秦小波 资深Java技术专家著有畅销书《设计模式之禅》和《编写高质量代码：改善Java程序的151个建议》 学习一门技术，要想做到“知其然”并不难，难的是要“知其所以然”，而且这更重要，也是考查一个技术人员真实实力的标杆。

想跨越这个标杆，除了自己付出时间和精力之外，高人的点拨和指导更能起到事半功倍之效。

成富是InfoQ中文站的资深专栏作者，从读者对他的“Java深度历险”系列技术文章的反馈可以看出：他不仅能帮你“知其然”，而且还能通过深入浅出的讲解让你“知其所以然”。

他的这本书一定能让你的Java技能更上一层楼。

——郑柯InfoQ中文站总编辑

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:<http://www.tushu007.com>