第一图书网, tushu007.com

<<Java脚本编程>>

图书基本信息

书名：<<Java脚本编程>>

13位ISBN编号：9787111238492

10位ISBN编号：7111238494

出版时间：2008-5

出版时间：机械工业出版社

作者：波斯安耐克

页数：308

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

内容概要

本书讲解了脚本语言的基本概念和使用方法，概括了Java开发人员可以使用的解决方案，并探讨了在Java应用程序中应用脚本语言的用例和设计模式。
内容分为五部分：第一部分介绍脚本语言的基本特征及适合用脚本语言的应用程序；第二部分介绍Java平台实际的技术和解决方案，详细讲解了Groovy脚本语言；第三部分介绍脚本语言在实际项目中的使用；第四部分介绍Java平台的脚本编程规范；第五部分提供了关于文中涉及的技术的安装和使用细节。
本书内容丰富，讲解清晰，适合作为软件开发人员的参考书。

作者简介

Dejan Bosanac是一名专业的软件开发人员和技术顾问。
他致力于各种技术的整合和互用，尤其是与Java和Web相关的应用。
他花了几年的时间开发复杂的软件项目，从高流量的网站到企业级应用程序。
他是JSR 223专家组的成员之一。

<<Java脚本编程>>

书籍目录

第一部分　第1章 脚本简介　1.1 背景　1.2 脚本语言的定义　1.2.1 编译器与解释器　1.2.2 产品中的源代码　1.2.3 类型策略　1.2.4 数据结构　1.2.5 代码作为数据　1.2.6 小结　1.3 脚本语言和虚拟机　1.4 脚本和系统编程的对比　1.4.1 运行时性能　1.4.2 开发速度　1.4.3 健壮性　1.4.4 维护　1.4.5 极限编程　1.5 混合法　1.6 一个脚本案例　1.7 小结　第2章 适用脚本语言的应用程序　2.1 组装　2.1.1 UNIX Shell语言　2.1.2 Perl　2.1.3 Tcl　2.2 原型　2.3 定制　2.4 软件开发支持　2.4.1 项目构建　2.4.2 测试　2.5 运维与管理　2.6 用户界面编程　2.7 用例2　2.7.1 Web应用程序　2.7.2 脚本和UNIX　2.7.3 游戏中的脚本　2.8 其他特征　2.8.1 可嵌入　2.8.2 可扩展　2.8.3 易于学习和使用　2.9 小结第二部分　第3章 JVM内部的脚本语言　3.1 帽底乾坤　3.2 脚本语言概念　3.3 BeanShell　3.3.1 入门　3.3.2 基本语法　3.3.3 松类型的语法　3.3.4 语法风格　3.3.5 命令　3.3.6 方法　3.3.7 对象　3.3.8 实现接口　3.3.9 嵌入Java　3.4 Jython　3.4.1 入门　3.4.2 基本语法　3.4.3 使用Java　3.4.4 实现接口　3.4.5 异常处理　3.4.6 嵌入Java　3.4.7 小结　3.5 Rhino　3.5.1 入门　3.5.2 使用Java　3.5.3 实现接口　3.5.4 JavaAdapter　3.5.5 嵌入Java　3.5.6 Host Object　3.5.7 小结　3.6 Groovy　3.7 其他脚本语言　3.7.1 JRuby　3.7.2 Tcl/Java　3.7.3 JudoScript　3.7.4 ObjectScript　3.8 小结　第4章 Groovy　4.1 为什么需要Groovy　4.2 安装　4.3 运行Groovy脚本　4.3.1 用交互式的Shell　4.3.2 用交互式的控制台　4.3.3 执行脚本文件　4.4 编译Groovy脚本　4.4.1 依赖　4.4.2 Classpath　4.4.3 Ant Task　4.5 脚本结构　4.6 语言语法　4.6.1 Java兼容性　4.6.2 语句　4.6.3 松类型　4.6.4 类型技巧　4.6.5 String　4.6.6 GString　4.6.7 正则表达式　4.6.8 集合　4.6.9 逻辑分支　4.6.10 循环　4.6.11 类　4.6.12 操作符重载　4.6.13 GroovyBean　4.6.14 闭包　4.7 系统操作　4.7.1 文件　4.7.2 进程　4.8 嵌入Java　4.9 安全性　4.10 小结　第5章 高级的Groovy编程　5.1 GroovySQL　5.1.1 groovy.sql.Sql　5.1.2 groovy.sql.DataSet　5.2 Groovlet　5.3 Groovy模板　5.4 GroovyMarkup　5.4.1 groovy.xml.MarkupBuilder　5.4.2 groovy.util.NodeBuilder　5.4.3 groovy.xml.SaxBuilder　5.4.4 groovy.xml.DomBuilder　5.4.5 groovy.xml.Namespace　5.4.6 groovy.util.BuilderSupport　5.5 Groovy和Swing　5.5.1 TableLayout　5.5.2 TableModel　5.6 小结　第6章 Bean Scripting Framework　6.1 Bean Scripting Framework简介　6.2 入门　6.3 基本概念　6.3.1 架构　6.3.2 脚本语言的注册　6.3.3 管理器和引擎初始化　6.3.4 使用脚本　6.4 使用脚本文件　6.5 方法和函数　6.5.1 call()　6.5.2 apply()　6.6 数据绑定　6.6.1 注册Bean　6.6.2 声明Bean　6.7 编译　6.8 应用程序　6.8.1 JSP　6.8.2 Xalan?J（XSLT）　6.9 小结第三部分　第7章 在Java实践脚本　7.1 单元测试　7.1.1 JUnit基础知识　7.1.2 GroovyTestCase类　7.1.3 断言方法　7.1.4 测试套件　7.1.5 用脚本作为单元测试案例　7.1.6 小结　7.2 交互式调试　7.3 构建工具　7.3.1 BSF支持　7.3.2 GroovyMarkup（AntBuilder）　7.3.3 小结　7.4 Shell Scripting　7.4.1 Classpath　7.4.2 实例　7.5 管控和管理　7.6 小结　第8章 脚本模式　8.1 脚本化组件模式　8.1.1 问题　8.1.2 解决方案　8.1.3 结果　8.1.4 范例代码　8.1.5 相关模式　8.2 中介者模式（胶合代码模式）　8.2.1 问题　8.2.2 解决方案　8.2.3 结果　8.2.4 范例代码　8.2.5 相关模式　8.3 脚本对象工厂模式　8.3.1 问题　8.3.2 解决方案　8.3.3 结果　8.3.4 范例代码　8.3.5 相关模式　8.4 观察者（广播）模式　8.4.1 问题　8.4.2 解决方案　8.4.3 结果　8.4.4 范例代码　8.4.5 相关模式　8.5 扩展点模式　8.5.1 问题　8.5.2 解决方案　8.5.3 结果　8.5.4 范例代码　8.5.5 相关模式　8.6 Active File模式　8.6.1 问题　8.6.2 解决方案　8.6.3 结果　8.6.4 范例代码　8.7 小结第四部分　第9章 Scripting API　9.1 动机和历史　9.2 简介　9.3 入门　9.4 架构　9.5 发现机制　9.6 引擎元数据　9.7 创建和注册脚本引擎　9.7.1 创建方法　9.7.2 注册方法　9.8 执行求值　9.9 ScriptException　9.10 绑定　9.10.1 引擎范围　9.10.2 全局范围　9.10.3 脚本上下文　9.11 代码生成　9.11.1 输出语句　9.11.2 方法调用语法　9.11.3 程序　9.12 其他引擎接口　9.12.1 可调用　9.12.2 可编译

9.13 线程 9.14 动态绑定 9.15 小结 第10章 Web Scripting Framework 10.1 架构 10.1.1 上下文 10.1.2 Servlet 10.1.3 交互 10.2 入门 10.3 配置 10.3.1 取消脚本 10.3.2 脚本路径 10.3.3 脚本方法 10.3.4 语言许可 10.3.5 显示结果 10.4 绑定 10.4.1 应用程序 10.4.2 请求 10.4.3 响应 10.4.4 Servlet 10.5 include方法 10.6 forward方法 10.7 会话共享 10.8 语言标签 10.9 线程问题 10.10 架构挑战 10.10.1 Java与PHP应用程序的整合 10.10.2 PHP Web应用程序中的Java业务逻辑 10.10.3 Java Web应用程序中的PHP视图 10.11 小结第五部分附录A Groovy的安装附录B Groovy的IDE支持附录C 安装JSR 223

章节摘录

第1章　脚本简介　　本书的主题结合了脚本技术与Java平台。
它描述了Java开发人员可以用来创建一个更加强大的开发环境的项目，以及使脚本变得有用的一些实践。
在开始讨论Java领域中的脚本应用程序之前，先整体概括一下脚本背后的一些理论，以及它在信息技术基础结构中的用途。
这是本书开头两章的主题，这样，可以更好地了解脚本技术，以及这项技术在Java平台内部起着什么样的作用。
首先，我们必须定义什么是脚本语言，并描述它们的特征。
它们的特征很大程度上决定了它们可以（应该）用在哪些场景中。
在本章中，我解释了术语脚本语言的含义，并讲解了它们的基本特征。
本章最后讨论了脚本语言与系统编程语言之间的区别，以及这些区别如何使它们符合开发中的某些场景。
1.1　背景　　脚本语言的定义很含糊，有时候与脚本语言在现实中的使用方式不太一致，因此，最好整体概括一些关于编程与计算的基础概念。
这样可为定义脚本语言和讨论它们的特征提供必要的基础。
让我们从头开始。
处理器执行机器指令，它操作处理器的寄存器或者外部存储器中的数据。
简单来说，机器指令由一系列二进制数字（很多0和1）组成，是特定于它在其中运行的特殊处理器的。
机器指令由告诉处理器它应该执行什么操作的操作码和表示操作应该在其中执行的数据的操作数组成。
例如，将一个寄存器中包含的值添加到另一个寄存器中这样一个简单的操作。
现在让我们假设一个简单的处理器，包含8位指令集，其中前5位表示操作码（比如，00111为“加”的寄存器值），且寄存器都以3位的模式进行处理。
我们可以将这个简单的实例写成如下：　　00111 001 010　　在这个例子中，我用了001和010对处理器的寄存器数字1和2（相应为Rl和R2）进行操作。
这个基础的计算方法已经广为人知几十年，我相信你对它也很熟悉。
各种处理器对于它们指令集的外观有着不同的策略（RISC或者CISC架构），但是从软件开发人员的角度来说，唯一重要的是，处理器只能执行二进制指令。
无论使用什么编程语言，最终的应用程序都是通过处理器执行的　　一系列机器指令。
随着时间而改变的是人们如何创建执行机器指令的顺序。
这个有序的机器指令称为计算机程序。
由于硬件的成本变得越来越低，且功能越来越强，因此用户的期望在增长。
软件开发作为一门学科，其目标就是提供一些机制，使开发人员用与以前同样的（或者更少的）工作，就能编写出更加复杂的应用程序。
一个特定处理器的指令集称为它的机器语言。
机器语言属于第一代编程语言。
以这种方式编写的程序通常非常快，因为它们针对特殊处理器的架构进行了优化。
但是除了这点好处之外，人类却很难（即便不是不可能的话）用机器语言编写大型且安全的应用程序，因为人类不善于处理大量的0和1。
为了解决这个问题，开发人员们开始为二进制数创建符号，这样就引人了汇编语言。
汇编语言是第二代编程语言。
汇编语言中的指令就是机器指令的上一级，它们在该级别中用容易记的关键字（如ADD、SUB等等）替代了二进制数。
如此一来，你就可以用汇编语言将前面那个简单的指令实例重写成如下：　　ADO R1，R2　　在这

个例子中，ADD关键字表示指令的操作码，R1和R2定义操作中涉及的寄存器。
即使只看这个简单的实例，也可以看出汇编语言显然使程序变得更容易让人阅读，从而能够创建更复杂的应用程序。
　　尽管第二代语言更加面向人类了，但它们无论如何都没有扩展处理器的功能。
　　到了高级语言，它允许开发人员用更高级别的语义形式表达自己的想法，这些语言称作第三代编程语言。
高级语言提供了各种功能强大的循环、数据结构、对象等等，使得利用它们创建许多应用程序变得更加容易。
　　随着时间的推移，各种不同的高级编程语言也相继出现了，它们的特征也大不相同。
其中有些特征将编程语言归类为脚本（或者动态）语言，就像我们在接下来的小节中将看到的。
　　对于编程语言如何在主机上执行的问题上也有区别。
通常，编译器将高级语言构造翻译成内存中的机器指令。
虽然以这种方式编写的程序最初与用汇编语言编写的程序相比，效率稍微低一些，因为早期的编译器不能有效地使用系统资源，但随着时间的推移，编译器和机器都得到了改进，使得系统编程语言比汇编语言更高级。
最后，高级语言在各种开发领域中变得普及起来，从业务应用程序和游戏到通信软件和操作系统实现。
　　但是，将高级别的语义转换成机器指令还有另外一种方法，也就是当它们被执行的时候解释它们。
这样，应用程序就以它们原始的形式存在于脚本中，并且这种构造由名为解释器的程序在运行时进行转换。
一般而言，你是在执行解释器，它读取应用程序的语句然后执行它们。
这类语言称为脚本或者动态语言，提供了一种比系统编程语言更高级别的抽象，我们将在本章稍后详细讨论它们。
　　包含这些特征的语言天生就适合于某些任务，例如过程自动化、系统管理和将现有的软件组件整合在一起；简而言之，由于系统编程语言引入的严格语法和约束，在任何地方都成了开发人员与他们的工作之间的障碍。
对脚本语言常用场景的描述，是第2章的重点。
　　但是所有这些与Java开发人员有什么关系呢？
为了回答这个问题，让我们首先简要地概括一下Java平台的历史。
由于平台变得越来越多样化，也为开发人员编写能在大多数现有系统中运行的软件增加了难度。
就在这时候，Sun公司开发了Java语言，它具有“一次编写，随处运行”的简单性。

编辑推荐

《Sun公司核心技术丛书·Java脚本编程语言、框架与模式》用大量代码实例来阐述重要的概念，示范了在现实Java工程中运行的脚本。

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