

<<Java完全探索 第2版>>

图书基本信息

书名：<<Java完全探索 第2版>>

13位ISBN编号：9787500645290

10位ISBN编号：7500645295

出版时间：2001-10

出版时间：中国青年出版社

作者：(美)Chuck Cavaness Geoff Friesen Brian Keeton

页数：930

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>

<<Java完全探索 第2版>>

内容概要

本书是关于Java语言的权威参考书，内容涵盖了Java语言的方方面面。

全书共分为五个部分。

在第一部分中，本书首先介绍了面向对象语言的设计方法和一些基本概念，使读者对Java语言的编程模式有了整体性的了解；接着以一个简单的Hello World程序为例子，使读者在整体上对Java编程有所了解；之后介绍了Java的基本数据类型、基本数据结构、基本类、基本接口和基本方法、表达式的使用方法以及线程的概念和使用方法等。

本书的第二部分对Java语言提供的丰富的图形显示方法集作了详尽的介绍。

首先介绍了Java基础类，接着介绍了AWT的使用方法，然后用两个创建图形用户界面的例子来具体说明，随之还介绍了Java语言的图像处理方法和用户与图形界面的交互。

本书第三部分介绍了Java语言的输入 / 输出方法。

在本书的第四部分和第五部分分别介绍了Java应用程序和数据库的交互与反射机制，以及JavaBean等内容。

对所有Java程序员而言、本书都是不容错过的最新、最详尽的参考书。

书籍目录

第一部分 Java语言

第1章 面向对象程序设计

1.1 面向对象程序设计：一种新的思维方式

1.2 程序设计简史

1.2.1 过程式语言

1.2.2 结构化开发

1.2.3 面向对象程序设计

1.3 关于对象

1.3.1 传统的程序设计

1.3.2 OOP方法

1.3.3 用继承机制扩展对象

1.4 多实体对象

1.5 模块化与代码组织

1.6 将对象关联到Java类

1.7 构建层次结构：OOP设计步骤

1.7.1 将问题分解成多个实体

1.7.2 在实体间寻找共同点

1.7.3 在实体间寻找差异

1.7.4 用抽象和继承机制设计层次结构

1.8 OOP与UML

1.8.1 UML简介

1.8.2 类图

1.8.3 顺序图

1.8.4 UML指导思想

1.9 Java是充满魔力的OOP语言吗?

第2章 HelloWorld：第一个Java程序

2.1 理解HelloWor1d程序

2.1.1 创建文件

2.1.2 编译代码

2.1.3 运行程序

2.2 理解HelloWorld

2.2.1 注释

2.2.2 声明类

2.2.3 main方法

2.2.4 显示到屏幕上

2.2.5 System.out和System.in

2.3 作为applet的HelloWorld

2.3.1 修改并编译源代码

2.3.2 创建HTML文件

2.3.3 在AppletViewer中运行程序

2.3.4 在Internet Explorer中运行HelloWorld

2.3.5 导入其他类

2.3.6 声明一个applet类

2.3.7 Applet的paint方法

2.3.8 applet短暂的生命

<<Java完全探索 第2版>>

2.4 保留字

2.5 Javal . 3 API

2.6 故障排除

第3章 数据类型与其他标记

3.1 Java数据类型

3.1.1 原始类型

3.1.2 引用类型

3.2 使用变量

3.2.1 声明变量

3.2.2 标识符：对变量的命名

3.3 布尔原始类型

3.4 整数类型

3.4.1 整数值的界限

3.4.2 声明整型变量

3.4.3 整数运算

3.5 运算符

3.5.1 算术运算符

3.5.2 赋值运算符

3.5.3 增量/减量运算符

3.6 字符变量

3.7 浮点变量

3.8 直接量：赋值

3.9 整型直接量

3.10 字符型直接量

3.11 浮点型直接量

3.12 字符串直接量

3.13 数组

3.14 不使用标记的输入元素

3.14.1 空格(whitespace)

3.14.2 注释

3.15 故障排除

第4章 方法与异常

4.1 方法声明

4.1.1 访问修饰符

4.1.2 修饰符(modifiers)

4.1.3 返回信息

4.1.4 方法名

4.1.5 参数

4.1.6 异常

4.2 块与语句

4.2.1 标号语句

4.2.2 作用域

4.2.3 局部变量初始化

4.3 方法和UML顺序图

4.4 捕获并抛出异常

4.4.1 使用try / catch / finally

4.4.2 抛出异常

<<Java完全探索 第2版>>

- 4.4.3 组合方法
- 4.4.4 异常的类型
- 4.4.5 处理多个异常
- 4.4.6 创建自己的异常类
- 4.4.7 异常与返回值
- 4.5 Java的错误类
- 4.6 故障排除
- 第5章 使用表达式
 - 5.1 什么是表达式?
 - 5.2 表达式怎样求值
 - 5.2.1 运算符的结合性
 - 5.2.2 运算符的优先顺序
 - 5.2.3 求值顺序
 - 5.3 C程序员应特别注意的地方
 - 5.4 按位运算符
 - 5.5 移位运算符
 - 5.5.1 左移
 - 5.5.2 有符号右移
 - 5.5.3 无符号右移
 - 5.6 类型转换
 - 5.6.1 隐式转换
 - 5.6.2 使用类型转换运算符的显式转换
 - 5.6.3 字符转换
 - 5.7 支持字符串的特殊运算符
 - 5.8 故障排除
- 第6章 控制流
 - 6.1 控制运行流程
 - 6.2 布尔运算符
 - 6.2.1 关系运算符
 - 6.2.2 等价运算符
 - 6.3 逻辑表达式
 - 6.3.1 条件AND和条件OR运算符
 - 6.3.2 逻辑非运算符
 - 6.4 条件运算符
 - 6.5 控制流语句中的布尔运算
 - 6.6 控制流功能
 - 6.6.1 if语句
 - 6.6.2 if-else语句
 - 6.7 Switch语句
 - 6.8 循环语句
 - 6.8.1 while循环
 - 6.8.2 do循环
 - 6.8.3 for循环
 - 6.9 跳转语句
 - 6.9.1 break语句
 - 6.9.2 continue语句
 - 6.9.3 return语句

6.10 故障排除

第7章 类

7.1 什么是类?

7.2 为什么使用类?

7.3 Java中的类

7.4 声明一个类

7.4.1 访问修饰符

7.4.2 修饰符

7.4.3 类名

7.4.4 超类——扩展另一个类

7.5 变量——定义状态

7.5.1 访问修饰符

7.5.2 修饰符

7.5.3 实例域

7.5.4 类域

7.5.5 声明一个常量

7.5.6 实现一个枚举类型

7.6 方法——定义行为

7.6.1 构造器

7.6.2 重载方法

7.6.3 用方法来提供guarded访问

7.6.4 类方法

7.7 创建类的一个实例

7.7.1 new运算符

7.7.2 垃圾回收

7.8 引用类元素

7.9 Object类

7.9.1 对象相等

7.9.2 拷贝一个类实例

7.9.3 在finalize()方法中进行清扫工作

7.10 类型转换与转换引用类型

7.10.1 编译时与运行时的类型检查

7.10.2 使用instanceof运算符

7.11 类属类

7.11.1 定义类属类

7.11.2 使用类属类的理由

7.11.3 类属类如何工作

7.12 包

7.12.1 使用包来组织代码

7.12.2 包与你的文件系统

7.12.3 从另一个包中导入一个类

7.12.4 导入整个包

7.12.5 不经导入而使用类

7.12.6 Java .1ang的隐式导入

7.13 将原始类型包装在类中

7.14 使用标准的数学函数

7.15 建立UML类图

<<Java完全探索 第2版>>

7.16 故障排除

第8章 字符串和文本的使用

8.1 字符串介绍

8.2 使用字符串类

8.3 获得字符串对象信息

8.4 字符串比较

8.5 字符串提取

8.6 字符串操作

8.7 字符串连接

8.8 将对象转换成字符串

8.9 将原始类型转换成字符串

8.10 使用StringBuffer类

8.10.1 创建一个StringBuffer对象

8.10.2 在StringBuffer后面附加内容

8.10.3 转换一个StringBuffer对象为字符串

8.10.4 对StringBuffer对象进行操作

8.11 使用StringTokenizer类从文件中获得所有标记符

8.12 字符串和编译器

8.13 故障排除

第9章 接口

9.1 接口是什么?

9.1.1 并不是每一个类本身都有一个接口

9.1.2 多接口

9.2 定义一个接口

9.2.1 声明

9.2.2 接口程序体

9.2.3 标记接口

9.3 实现一个接口

9.3.1 方法重载

9.3.2 修饰符

9.3.3 参数列表

9.3.4 程序体

9.3.5 异常

9.3.6 接口方法冲突

9.4 接口引用

9.4.1 访问常量

9.4.2 设计一个接口胜于实现一个接口

9.5 在一个接口和一个抽象类之间选择

9.6 在UML中描述接口

9.7 故障排除

第10章 数据结构与Java设施

10.1 Collection和Utility类

10.2 Collection结构

10.3 Collection结构的接口

10.3.1 Collection接口

10.3.2 List接口

10.3.3 Set接口和SortedSet接口

<<Java完全探索 第2版>>

- 10.3.4 Map接口和SonedMap接口
- 10.4 基本类和接口
 - 10.4.1 Vector类
 - 10.4.2 ENUMERATION接口
 - 10.4.3 STACK
 - 10.4.4 Hashtable
- 10.5 通用实现类
 - 10.5.1 ArrayList类
 - 10.5.2 LinkedList类
 - 10.5.3 HashMap
 - 10.5.4 TreeMap
 - 10.5.5 WedkHashMap
 - 10.5.6 HashSet
 - 10.5.7 TreeSet
- 10.6 Co11ection排序
 - 10.6.1 COMPARABLE接口
 - 10.6.2 Comparator接口
 - 10.6.3 Co11ections.sort()的使用
- 10.7 Co11ection结构重述
 - 10.7.1 Iterator接口
 - 10.7.2 ListIterator接口
 - 10.7.3 有效搜索
- 10.8 Co11ection功能和约束
 - 10.8.1 方法和域
 - 10.8.2 Singleton方法
 - 10.8.3 同步化的Co11ection结构
 - 10.8.4 不可变Collection结构
- 10.9 使用哪种Co11ection类?
- 10.10 Array的功能
 - 10.10.1 Array类
 - 10.10.2 搜索Array
 - 10.10.3 两个Array的比较
 - 10.10.4 填充Array
 - 10.10.5 排序Array
- 10.11 使用Properties类配置应用程序属性
 - 10.11.1 设置Propedies对象
 - 10.11.2 查询Properties对象
 - 10.11.3 存储和获取Properties对象
- 10.12 Date类型
 - 10.12.1 Date类
 - 10.12.2 Date对象比较
 - 10.12.3 将Date类型转换为String类型
 - 10.12.4 改变Date属性
- 10.13 BitSet类
- 10.14 随机数的产生
- 10.15 使用Observer / Observab1e观察状态变化
 - 10.15.1 Observable类

<<Java完全探索 第2版>>

10.15.2 Observer接口

10.16 故障排除

第11章 线程

11.1 什么是线程?

11.2 为什么使用线程?

11.3 如何制作线程类?

11.4 扩展Thread

11.5 实现Runnable

11.6 线程的生命周期

11.7 使用Thread定位变量

11.8 改变线程优先级

11.9 线程同步

线程间通信

11.10 改变线程运行状态

11.11 线程组

11.12 获取当前运行线程数量

11.13 线程让位

11.14 守护线程

11.15 使用Timer类和Timer Task类

11.16 在程序中安全使用线程

11.17 故障排除

第二部分 用户界面

第12章 Java基础类介绍

12.1 什么是JFC

12.1.1 AWT

12.1.2 Swing

12.1.3 辅助功能

12.1.4 Java 2D

12.1.5 拖放

12.2 JFC的历史

12.2.1 AWTI.o

12.2.2 Internet基础类

12.2.3 应用基础类

12.2.4 Java基础类

12.2.5 将来的展望

12.3 JFC的应用程序

12.3.1 尝试JFC的AWT应用程序

12.3.2 尝试JFC的Swing应用程序

12.4 Applet

12.4.1 什么是Applet

12.4.2 Applet和WorldWide Web

12.4.3 AWT applet

12.4.4 AWT AppletHTML

12.4.5 Swingapplet

12.4.6 SwingApplet HTML

12.4.7 体系结构

12.4.8 Applet和应用程序

<<Java完全探索 第2版>>

12.5 故障排除

第13章 AWT基础

13.1 什么是AWT

13.2 图形

13.2.1 画图的基础

13.2.2 显示文本

13.2.3 绘制形状

13.2.4 实用类和接口

13.2.5 实用方法

13.3 字体和颜色

13.3.1 认识字体

13.3.2 创建字体

13.3.3 字体度量

13.3.4 颜色描述

13.3.5 系统颜色

13.4 图像

13.4.1 载入和绘制图像

13.4.2 动画

13.4.3 产生图像

13.4.4 消费图像

13.4.5 过滤图像

13.5 故障排除

第14章 使用AWT方式创建GUI

14.1 窗口工具集介绍

14.2 使用组件、容器和布置管理器创建GUI

14.2.1 系统工具集

14.2.2 对等实体、重量组件和轻量组件

14.2.3 组件研究

14.2.4 容器研究

14.2.5 布局管理器研究

14.3 使用事件和监听器创建GUI

14.3.1 事件研究

14.3.2 监听器研究

14.4 定制窗口工具集

14.4.1 创建自己的容器

14.4.2 创建自己的布局管理器

14.5 故障排除

第15章 Swing简介

15.1 什么是Swing

15.2 比较Swing和AWT

15.3 混合重量和轻量的组件

15.4 从AWT到Swing的转换

15.5 故障排除

第16章 使用Swing方式创建GUI

16.1 Swing工具集

16.1.1 模型视图控件器(Model—View—Contro11er)结构

16.1.2 可插入的外观和感觉

<<Java完全探索 第2版>>

16.2 使用组件、容器和布局管理器创建GUI

16.2.1 研究组件

16.2.2 研究容器

16.2.3 研究布局管理器

16.3 使用事件和监听器创建GUI

16.4 附加功能

16.4.1 行为

16.4.2 边

16.4.3 键盘输入和捆绑

16.4.4 绘制

16.4.5 属性

16.4.6 线程问题

16.4.7 定时器

16.5 故障排除

第17章 辅助功能

17.1 什么是辅助功能

17.2 辅助功能是如何工作的

17.3 实用工具(Uilities)

17.3.1 AccessibilityMonitor

17.3.2 AWTMonitor

17.3.3 Ferret

17.3.4 JavaMonitor

17.3.5 Linker

17.3.6 Monkey

17.4 Robot

17.5 故障排除

第18章 Java 2D

18.1 什么是Java 2D

18.1.1 Graphics2D

18.1.2 坐标系统

18.1.3 环境、设备和配置

18.1.4 多屏幕环境(Multiple-ScreenEnvironments)

18.1.5 API

18.2 着色101

18.2.1 着色提示属性

18.2.2 Stroke属性

18.2.3 Paint属性

18.2.4 Transform属性

18.2.5 裁剪路径属性

18.2.6 合成属性

18.3 图像、文本和带缓冲的图像(Shapes、Text And Buffered Images)

18.3.1 图形(Shapes)

18.3.2 文本(Text)

18.3.3 带缓冲区的图像(Buffered Image)

18.4 打印

18.4.1 打印结构(The Printing Framework)

18.4.2 工作控制和提交(Job Control And Rendering)

<<Java完全探索 第2版>>

- 18.4.3 操纵Printable类型打印(PrintingWith Printable)
- 18.4.4 操纵Pageable类型打印(PrintingWithPageable)
- 18.4.5 快速打印和脏打印(Quick And Dirty Printing)
- 18.5 故障排除

第19章 拖放

- 19.1 什么是拖放
- 19.2 数据传送
 - 19.2.1 DataFlavors
 - 19.2.2 Transferables
 - 19.2.3 剪贴板所有者
 - 19.2.4 剪贴板
- 19.3 拖放
- 19.4 故障排除

第20章 Java媒体框架

- 20.1 什么是JMF .
 - 20.1.1 JMF的下载与安装
 - 20.1.2 API
 - 20.1.3 分层的体系结构
 - 20.1.4 媒体流
 - 20.1.5 时基和时钟
 - 20.1.6 管理器
- 20.2 播放媒体
- 20.3 处理媒体
- 20.4 捕捉媒体
- 20.5 故障排除

第三部分 I/O

第21章 流、文件和串行化

- 21.1 什么是流
- 21.2 使用流类
- 21.3 用字节流读写数据
 - 21.3.1 InputStream类
 - 21.3.2 OutputStream类
 - 21.3.3 读写字节数组
 - 21.3.4 读写文件
 - 21.3.5 流缓冲
 - 21.3.6 过滤流
 - 21.3.7 打印流
- 21.4 重定向标准输入 / 输出
- 21.5 使用Reader和Writer
 - 21.5.1 使用BufferedReader和BufferedWriter
 - 21.5.2 使用LineNumberReader
 - 21.5.3 使用InputStreamReader和OutputStreamReader
 - 21.5.4 PrintWriter类
- 21.6 有关文件的工作
- 21.7 创建文件创建临时文件
- 21.8 读写文件
 - 21.8.1 随机文件访问

<<Java完全探索 第2版>>

21.8.2 文件的安全性

21.9 目录操作

退出时删除文件

21.10 对象流

21.11 故障排除

第22章 对象串行化

22.1 什么是对象串行化

22.1.1 对象串行化如何工作

22.1.2 用对象引用处理对象

22.2 对象串行化的例子

22.3 读写自己的对象

22.4 定制对象串行化

使用Transient关键字

22.5 使用Externalizable接口

22.6 故障排除

第23章 通信和组网

23.1 网络通信概述

23.2 TCP/IP体系结构

23.2.1 TCP/IP协议

23.2.2 网际协议(Internet Protocol.IP)

23.2.3 传输控制协议(TCP)

23.3 和Socket一起工作

23.3.1 什么是一个Socket

23.3.2 Client Socket

23.3.3 服务器Sockets

23.4 使用Socket时压缩数据

23.5 使用Socket通过网络传送对象

23.6 使用Socket传送对象

23.7 获得主机名和IP地址信息

23.8 用户数据报协议(UDP)

23.8.1 UDP Socket特性

23.8.2 JavaUDP类

23.8.3 一个数据报的例子

23.8.4 多点传送

23.9 WorldWideWeb概述

23.10 理解请求 / 应答机制

23.11 使用URL类

23.11.1 什么是URL

23.11.2 创建一个URL对象

23.11.3 创建一个URL连接

23.11.4 对一个URL链接读和写

23.12 一个用来建立一个WWW图表的例子

23.13 HTTP和SSL协议

23.14 Java安全端口扩展包概述

23.15 故障排除

第24章 国际化使用

24.1 什么是国际化

<<Java完全探索 第2版>>

24.2 国际化的Java支持

24.3 Locale类

被支持的现场

24.4 对位置敏感的数据打包

24.4.1 数据捆绑

24.4.2 ListResourceBundles

24.4.3 PropertyResourceBundle

24.5 日历和时区支持

Calendar类

24.6 国际化格式化

24.6.1 Format类

24.6.2 日期和时间格式化

24.6.3 数据格式化

24.6.4 消息格式化

24.7 字符集转换器

24.8 Reader和Writer类

24.9 Java.text包

24.10 一个例子：国际化测试

24.11 故障排除

第25章 JAR文件

25.1 使用JAR文件

25.1.1 绑定

25.1.2 压缩

25.1.3 可移植性

25.1.4 版本信息

25.1.5 安全性

25.1.6 减少下载时间

25.2 ZIP格式与JAR格式之间的差异

何时使用JAR文件

25.3 创建和浏览JAR文件

25.3.1 列出档案文件内容

25.3.2 从文件中解压文件

25.3.3 更新JAR文件

25.4 显示说明文件

25.4.1 绑定为JAR文件的应用程序

25.4.2 包封装

25.4.3 包的版本信息

25.4.4 下载扩展

25.5 读写JAR文件—

25.6 在Classpath中加入JAR文件

25.7 JAR文件与安全性

25.7.1 私钥、公钥及证书

25.7.2 Keytool

25.7.3 jarsigner

25.8 何时使用JAR文件

25.9 故障排除

第四部分数据库

<<Java完全探索 第2版>>

第26章 JDBC 2.0 基本概述

- 26.1 什么是JDBC
- 26.2 JDBC 2.0概述
- 26.3 JDBC驱动程序
 - 26.3.1 Type I
 - 26.3.2 Type II
 - 26.3.3 Type III
 - 26.3.4 Type IV
- 26.4 建立数据库
建立范例数据库模式
- 26.5 安装和使用JDBC驱动程序
- 26.6 JDBC URL
- 26.7 连接数据库
 - 26.7.1 载入驱动程序
 - 26.7.2 建立连接
- 26.8 使用JDBC读取数据
 - 26.8.1 Statement类
 - 26.8.2 ResultSet类
- 26.9 用JDBC写数据
- 26.10 使用已准备的语句
- 26.11 使用事务
- 26.12 从数据库中获得元数据
- 26.13 使用现有的存储过程
- 26.14 故障排除

第27章 高级JDBC 2.0

- 27.1 JDBC 2.0 API中的新特性
 - 27.1.1 使用可滚动的ResultSet
 - 27.1.2 使用可修改的ResultSet
 - 27.1.3 修改的批处理
- 27.2 使用JDBC可选包
 - 27.2.1 行集
 - 27.2.2 连接池
 - 27.2.3 分布事务
 - 27.2.4 数据源
- 27.3 理解从对象到关系的映射机制
- 27.4 故障排除

第五部分组件开发

第28章 反射

- 28.1 什么是反射
- 28.2 反射API中的关键类
 - 28.2.1 构造类
 - 28.2.2 Method类
 - 28.2.3 field类
- 28.3 安全考虑
- 28.4 创建一个只知道构造器清单的类
- 28.5 检查一个类的方法
 - 28.5.1 获得一个方法的清单

<<Java完全探索 第2版>>

- 28.5.2 使用getDeclaredMethod()方法来调用一个方法
- 28.5.3 调用将参数作为原语类型接受的方法
- 28.6 访问一个类的声明域
- 28.7 应用方法包
- 28.8 使用代理实现一个接口
- 28.9 故障排除
- 第29章 JavaBeans
 - 29.1 自包含组件
 - 29.2 组件模型中的重要概念
 - 29.2.1 组件域或属性
 - 29.2.2 组件方法或函数
 - 29.2.3 组件间通信和事件
 - 29.2.4 状态持续和存储
 - 29.3 设计一个JavaBean的基本问题
 - 29.3.1 指定Bean的属性
 - 29.3.2 指定Bean产生和响应的事件
 - 29.3.3 属性、方法和事件开放
 - 29.3.4 最初的属性值和Bean定制
 - 29.4 创建和使用属性
 - 29.4.1 单值属性
 - 29.4.2 带索引的属性
 - 29.4.3 绑定属性
 - 29.5 使用事件和别的组件通信
 - 29.5.1 多点传送事件
 - 29.5.2 单点传送事件
 - 29.5.3 事件适配器
 - 29.6 内部透视：创建和使用BeanInfo类
 - 29.7 定制：提供定制的Property Editors和GUI接口
 - 29.7.1 PropertyEditors和Propety EditorManager
 - 29.7.2 定制编辑器
 - 29.7.3 在非GUI环境中提供选择性的行为
 - 29.8 Bean打包
 - 29.9 Enterprise JavaBean
 - 29.9.1 将用户应用程序进行划分
 - 29.9.2 服务器组件
 - 29.10 故障排除
- 附录
 - 附录A 安装SDK并开始应用
 - A.1 获得SDKI.3SE
 - A.2 安装SDKI.3SE
 - A.2.1 为Windows安装SDK
 - A.2.2 连接多个安装文件
 - A.2.3 开始Windows安装程序
 - A.2.4 为So1ads安装SDK
 - A.2.5 使用自解压安装格式
 - A.2.6 为Linux安装JDK

<<Java完全探索 第2版>>

- A . 2 . 7 使用RPM安装格式
- A . 3 测试Java编译器和Java虚拟机(JVM)
- A . 3 . 1 PATH环境变量
- A . 3 . 2 CLASSPATH环境变量

附录B SDK工具

- B . 1 SDK工具介绍
- B . 2 AppleViewer
- B . 3 java—Java解释器
- B . 4 javaC—Java编译器
- B . 5 javaP—Java分解器
- B . 6 javadoc工具(文档产生器)
- B . 7 job—Java调试器

附录C SDK 1.3性能提高

- C . 1 通常的性能提高
 - C . 1 . 1 减少和字符串相关的存储器脚印
 - C . 1 . 2 JAR文件存储器使用减少
 - C . 1 . 3 Bighteger中的性能增强
 - C . 1 . 4 装入Swing类
- C . 2 HotSpot技术介绍
- C . 3 什么使HotSpot更好
 - C . 3 . 1 实时自适应编译
 - C . 3 . 2 方法内联
 - C . 3 . 3 改进和重新设计的对象设置
 - C . 3 . 4 快速和完全准确的garbage收集
 - C . 3 . 5 快速线程同步
- C . 4 JavaHotSpot客户机虚拟机Java应用程序发射器
- C . 5 JavaHotSpot服务器虚拟机2 . 0
- C . 6 Aplet应用改进
 - C . 6 . 1 Aplet缓存控制
 - C . 6 . 2 Aplet支持类的快速装入
 - C . 6 . 3 JAR索引
 - C . 6 . 4 扩展部分的自动安装

附录D Java资源

- D . 1 站点
- D . 2 新闻组
- D . 3 邮件列表

<<Java完全探索 第2版>>

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>