Java核心技术 卷1 基础知识（原书第9版）

译者序

前言

致谢

第1章 Java程序设计概述

1.1 Java程序设计平台

1.2 Java“白皮书”的关键术语

1.2.1 简单性

1.2.2 面向对象

1.2.3 网络技能

1.2.4 健壮性

1.2.5 安全性

1.2.6 体系结构中立

1.2.7 可移植性

1.2.8 解释型

1.2.9 高性能

1.2.10 多线程

1.2.11 动态性

1.3 Java applet与Internet

1.4 Java发展简史

1.5 关于Java的常见误解

第2章 Java程序设计环境

2.1 安装Java开发工具箱

2.1.1 下载JDK

2.1.2 设置执行路径

2.1.3 安装库源文件和文档

2.1.4 安装本书中的示例

2.1.5 导航Java目录

2.2 选择开发环境

2.3 使用命令行工具

2.4 使用集成开发环境

2.5 运行图形化应用程序

2.6 建立并运行applet

第3章 Java的基本程序设计结构

3.1 一个简单的Java应用程序

3.2 注释

3.3 数据类型

3.3.1 整型

3.3.2 浮点类型

3.3.3 char类型

3.3.4 boolean类型

3.4 变量

3.4.1 变量初始化

3.4.2 常量

3.5 运算符

3.5.1 自增运算符与自减运算符

3.5.2 关系运算符与boolean运算符

3.5.3 位运算符

3.5.4 数学函数与常量

3.5.5 数值类型之间的转换

3.5.6 强制类型转换

3.5.7 括号与运算符级别

3.5.8 枚举类型

3.6 字符串

3.6.1 子串

3.6.2 拼接

3.6.3 不可变字符串

3.6.4 检测字符串是否相等

3.6.5 空串与Null串

3.6.6 代码点与代码单元

3.6.7 字符串API

3.6.8 阅读联机API文档

3.6.9 构建字符串

3.7 输入输出

3.7.1 读取输入

3.7.2 格式化输出

3.7.3 文件输入与输出

3.8 控制流程

3.8.1 块作用域

3.8.2 条件语句

3.8.3 循环

3.8.4 确定循环

3.8.5 多重选择：switch语句

3.8.6 中断控制流程语句

3.9 大数值

3.10 数组

3.10.1 for each循环

3.10.2 数组初始化以及匿名数组

3.10.3 数组拷贝

3.10.4 命令行参数

3.10.5 数组排序

3.10.6 多维数组

3.10.7 不规则数组

第4章 对象与类

4.1 面向对象程序设计概述

4.1.1 类

4.1.2 对象

4.1.3 识别类

4.1.4　类之间的关系

4.2 使用预定义类

4.2.1 对象与对象变量

4.2.2 Java类库中的Gregorian-Calendar类

4.2.3 更改器方法与访问器方法

4.3 用户自定义类

4.3.1 Employee类

4.3.2 多个源文件的使用

4.3.3 剖析Employee类

4.3.4 从构造器开始

4.3.5 隐式参数与显式参数

4.3.6 封装的优点

4.3.7 基于类的访问权限

4.3.8 私有方法

4.3.9 final实例域

4.4 静态域与静态方法

4.4.1 静态域

4.4.2 静态常量

4.4.3 静态方法

4.4.4 工厂方法

4.4.5 main方法

4.5 方法参数

4.6 对象构造

4.6.1 重载

4.6.2 默认域初始化

4.6.3 无参数的构造器

4.6.4 显式域初始化

4.6.5 参数名

4.6.6 调用另一个构造器

4.6.7 初始化块

4.6.8 对象析构与finalize方法

4.7 包

4.7.1 类的导入

4.7.2 静态导入

4.7.3 将类放入包中

4.7.4 包作用域

4.8 类路径

4.9 文档注释

4.9.1 注释的插入

4.9.2 类注释

4.9.3 方法注释

4.9.4 域注释

4.9.5 通用注释

4.9.6 包与概述注释

4.9.7 注释的抽取

4.10 类设计技巧

第5章 继承

5.1 类、超类和子类

5.1.1 继承层次

5.1.2 多态

5.1.3 动态绑定

5.1.4 阻止继承：final类和方法

5.1.5 强制类型转换

5.1.6 抽象类

5.1.7 受保护访问

5.2 Object：所有类的超类

5.2.1 equals方法

5.2.2 相等测试与继承

5.2.3 hashCode方法

5.2.4 toString方法

5.3 泛型数组列表

5.3.1 访问数组列表元素

5.3.2 类型化与原始数组列表的兼容性

5.4 对象包装器与自动装箱

5.5 参数数量可变的方法

5.6 枚举类

5.7 反射

5.7.1 Class类

5.7.2 捕获异常

5.7.3 利用反射分析类的能力

5.7.4 在运行时使用反射分析对象

5.7.5 使用反射编写泛型数组代码

5.7.6 调用任意方法

5.8 继承设计的技巧

第6章 接口与内部类

6.1 接口

6.1.1 接口的特性

6.1.2 接口与抽象类

6.2 对象克隆

6.3 接口与回调

6.4 内部类

6.4.1 使用内部类访问对象状态

6.4.2 内部类的特殊语法规则

6.4.3 内部类是否有用、必要和安全

6.4.4 局部内部类

6.4.5 由外部方法访问final变量

6.4.6 匿名内部类

6.4.7 静态内部类

6.5 代理

第7章 图形程序设计

7.1 Swing概述

7.2 创建框架

7.3 框架定位

7.3.1 框架属性

7.3.2 确定合适的框架大小

7.4 在组件中显示信息

7.5 处理2D图形

7.6 使用颜色

7.7 文本使用特殊字体

7.8 显示图像

第8章 事件处理

8.1 事件处理基础

8.1.1 实例：处理按钮点击事件

8.1.2 建议使用内部类

8.1.3 创建包含一个方法调用的监听器

8.1.4 实例：改变观感

8.1.5 适配器类

8.2 动作

8.3 鼠标事件

8.4 AWT事件继承层次

第9章 Swing用户界面组件

9.1 Swing和模型-视图-控制器设计模式

9.1.1 设计模式

9.1.2 模型-视图-控制器模式

9.1.3 Swing按钮的模型-视图-控制器分析

9.2 布局管理概述

9.2.1 边框布局

9.2.2 网格布局

9.3 文本输入

9.3.1 文本域

9.3.2 标签和标签组件

9.3.3 密码域

9.3.4 文本区

9.3.5 滚动窗格

9.4 选择组件

9.4.1 复选框

9.4.2 单选按钮

9.4.3 边框

9.4.4 组合框

9.4.5 滑动条

9.5 菜单

9.5.1 菜单创建

9.5.2 菜单项中的图标

9.5.3 复选框和单选按钮菜单项

9.5.4 弹出菜单

9.5.5 快捷键和加速器

9.5.6 启用和禁用菜单项

9.5.7 工具栏

9.5.8 工具提示

9.6 复杂的布局管理

9.6.1 网格组布局

9.6.2 组布局

9.6.3 不使用布局管理器

9.6.4 定制布局管理器

9.6.5 遍历顺序

9.7 对话框

9.7.1 选项对话框

9.7.2 创建对话框

9.7.3 数据交换

9.7.4 文件对话框

9.7.5 颜色选择器

第10章 部署应用程序和applet

10.1 JAR文件

10.1.1 清单文件

10.1.2 可运行JAR文件

10.1.3 资源

10.1.4 密封

10.2 Java Web Start

10.2.1 沙箱

10.2.2 签名代码

10.2.3 JNLP API

10.3 applet

10.3.1 一个简单的applet

10.3.2 applet的HTML标记和属性

10.3.3 object标记

10.3.4 使用参数向applet传递信息

10.3.5 访问图像和音频文件

10.3.6 applet上下文

10.4 应用程序首选项存储

10.4.1 属性映射

10.4.2 Preferences API

第11章 异常、断言、日志和调试

11.1 处理错误

11.1.1 异常分类

11.1.2 声明已检查异常

11.1.3 如何抛出异常

11.1.4 创建异常类

11.2 捕获异常

11.2.1 捕获多个异常

11.2.2 再次抛出异常与异常链

11.2.3 finally子句

11.2.4 带资源的try语句

11.2.5 分析堆栈跟踪元素

11.3 使用异常机制的技巧

11.4 使用断言

11.4.1 启用和禁用断言

11.4.2 使用断言完成参数检查

11.4.3 为文档假设使用断言

11.5 记录日志

11.5.1 基本日志

11.5.2 高级日志

11.5.3 修改日志管理器配置

11.5.4 本地化

11.5.5 处理器

11.5.6 过滤器

11.5.7 格式化器

11.5.8 日志记录说明

11.6 调试技巧

11.7 GUI程序排错技巧

11.8 使用调试器

第12章 泛型程序设计

12.1 为什么要使用泛型程序设计

12.2 定义简单泛型类

12.3 泛型方法

12.4 类型变量的限定

12.5 泛型代码和虚拟机

12.5.1 翻译泛型表达式

12.5.2 翻译泛型方法

12.5.3 调用遗留代码

12.6 约束与局限性

12.6.1 不能用基本类型实例化类型参数

12.6.2 运行时类型查询只适用于原始类型

12.6.3 不能创建参数化类型的数组

12.6.4 Varargs警告

12.6.5 不能实例化类型变量

12.6.6 泛型类的静态上下文中类型变量无效

12.6.7 不能抛出或捕获泛型类的实例

12.6.8 注意擦除后的冲突

12.7 泛型类型的继承规则

12.8 通配符类型

12.8.1 通配符的超类型限定

12.8.2 无限定通配符

12.8.3 通配符捕获

12.9 反射和泛型

12.9.1 使用Class参数进行类型匹配

12.9.2 虚拟机中的泛型类型信息

第13章 集合

13.1 集合接口

13.1.1 将集合的接口与实现分离

13.1.2 Java类库中的集合接口和迭代器接口

13.2 具体的集合

13.2.1 链表

13.2.2 数组列表

13.2.3 散列集

13.2.4 树集

13.2.5 对象的比较

13.2.6 队列与双端队列

13.2.7 优先级队列

13.2.8 映射表

13.2.9 专用集与映射表类

13.3 集合框架

13.3.1 视图与包装器

13.3.2 批操作

13.3.3 集合与数组之间的转换

13.4 算法

13.4.1 排序与混排

13.4.2 二分查找

13.4.3 简单算法

13.4.4 编写自己的算法

13.5 遗留的集合

13.5.1 Hashtable类

13.5.2 枚举

13.5.3 属性映射表

13.5.4 栈

13.5.5 位集

第14章 多线程

14.1 什么是线程

14.2 中断线程

14.3 线程状态

14.3.1 新创建线程

14.3.2 可运行线程

14.3.3 被阻塞线程和等待线程

14.3.4 被终止的线程

14.4 线程属性

14.4.1 线程优先级

14.4.2 守护线程

14.4.3 未捕获异常处理器

14.5 同步

14.5.1 竞争条件的一个例子

14.5.2 竞争条件详解

14.5.3 锁对象

14.5.4 条件对象

14.5.5 synchronized关键字

14.5.6 同步阻塞

14.5.7 监视器概念

14.5.8 Volatile域

14.5.9 final变量

14.5.10 原子性

14.5.11 死锁

14.5.12 线程局部变量

14.5.13 锁测试与超时

14.5.14 读/写锁

14.5.15 为什么弃用stop和suspend方法

14.6 阻塞队列

14.7 线程安全的集合

14.7.1 高效的映射表、集合和队列

14.7.2 写数组的拷贝

14.7.3 较早的线程安全集合

14.8 Callable与Future

14.9 执行器

14.9.1 线程池

14.9.2 预定执行

14.9.3 控制任务组

14.9.4 Fork-Join框架

14.10 同步器

14.10.1 信号量

14.10.2 倒计时门栓

14.10.3 障栅

14.10.4 交换器

14.10.5 同步队列

14.11 线程与Swing

14.11.1 运行耗时的任务

14.11.2 使用Swing工作线程

14.11.3 单一线程规则

附录　Java关键字

Java核心技术 卷II 高级特性

译者序

前言

第1章　流与文件

　1.1　流

　1.1.1　读写字节

　1.1.2　完整的流家族

　1.1.3　组合流过滤器

　1.2　文本输入与输出

　1.2.1　如何写出文本输出

　1.2.2　如何读入文本输入

　1.2.3　以文本格式存储对象

　1.2.4　字符集

　1.3　读写二进制数据

　1.4　ZIP文档

　1.5　对象流与序列化

　1.5.1　理解对象序列化的文件格式

　1.5.2　修改默认的序列化机制

　1.5.3　序列化单例和类型安全的枚举

　1.5.4　版本管理

　1.5.5　为克隆使用序列化

　1.6　操作文件

　1.6.1　Path

　1.6.2　读写文件

　1.6.3　复制、移动和删除文件

　1.6.4　创建文件和目录

　1.6.5　获取文件信息

　1.6.6　迭代目录中的文件

　1.6.7　ZIP文件系统

　1.7　内存映射文件

　1.7.1　缓冲区数据结构

　1.7.2　文件加锁机制

　1.8　正则表达式

第2章　XML

　2.1　XML概述

　2.2　解析XML文档

　2.3　验证XML文档

　2.3.1　文档类型定义

　2.3.2　XML Schema

　2.3.3　实用示例

　2.4　使用XPath来定位信息

　2.5　使用命名空间

　2.6　流机制解析器

　2.6.1　使用SAX解析器

　2.6.2　使用StAX解析器

　2.7　生成XML文档

　2.7.1　不带命名空间的文档

　2.7.2　带命名空间的文档

　2.7.3　写出文档

　2.7.4　示例：生成SVG文件

　2.7.5　使用StAX写XML文档

　2.8　XSL转换

第3章　网络

　3.1　连接到服务器

　3.1.1　套接字超时

　3.1.2　因特网地址

　3.2　实现服务器

　3.2.1　为多个客户端服务

　3.2.2　半关闭

　3.3　可中断套接字

　3.4　获取Web数

　3.4.1　URL和URI

　3.4.2　使用URLConnection获取信息

　3.4.3　提交表单数据

　3.5　发送E-mail

第4章　数据库编程

　4.1　JDBC的设计

　4.1.1　JDBC驱动程序类型

　4.1.2　JDBC的典型用法

　4.2　结构化查询语言

　4.3　JDBC配置

　4.3.1　数据库URL

　4.3.2　驱动程序JAR文件

　4.3.3　启动数据库

　4.3.4　注册驱动器类

　4.3.5　连接到数据库

　4.4　执行SQL语句

　4.4.1　管理连接、语句和结果集

　4.4.2　分析SQL异常

　4.4.3　组装数据库

　4.5　执行查询操作

　4.5.1　预备语句

　4.5.2　读写LOB

　4.5.3　SQL转义

　4.5.4　多结果集

　4.5.5　获取自动生成键

　4.6　可滚动和可更新的结果集

　4.6.1　可滚动的结果集

　4.6.2　可更新的结果集

　4.7　行集

　4.7.1　构建行集

　4.7.2　被缓存的行集

　4.8　元数据

　4.9　事务

　4.9.1　保存点

　4.9.2　批量更新

　4.9.3　高级SQL类型

　4.10　Web与企业应用中的连接管理

第5章　国际化

　5.1　Locales

　5.2　数字格式

　5.3　日期和时间

　5.4　排序

　5.4.1　排序强度

　5.4.2　分解

　5.5　消息格式化

　选择格式

　5.6　文本文件和字符集

　源文件的字符编码

　5.7　资源包

　5.7.1　定位资源包

　5.7.2　属性文件

　5.7.3　包类

　5.8　一个完整的例子

第6章　高级Swing

　6.1　列表

　6.1.1　JList构件

　6.1.2　列表模式

　6.1.3　插入和移除值

　6.1.4　值的绘制

　6.2　表格

　6.2.1　简单表格

　6.2.2　表格模型

　6.2.3　对行和列的操作

　6.2.4　单元格的绘制和编辑

　6.3　树

　6.3.1　简单的树

　6.3.2　节点枚举

　6.3.3　绘制节点

　6.3.4　监听树事件

　6.3.5　定制树模型

　6.4　文本构件

　6.4.1　文本构件中的修改跟踪

　6.4.2　格式化的输入框

　6.4.3　JSpinner构件

　6.4.4　用JEditorPane显示HTML

　6.5　进度指示器

　6.5.1　进度条

　6.5.2　进度监视器

　6.5.3　监视输入流的进度

　6.6　构件组织器和装饰器

　6.6.1　分割面板

　6.6.2　选项卡面板

　6.6.3　桌面面板和内部框体

　6.6.4　级联与平铺

　6.6.5　否决属性设置

第7章　高级AWT

　7.1　绘图操作流程

　7.2　形状

　7.3　区域

　7.4　笔划

　7.5　着色

　7.6　坐标变换

　7.7　剪切

　7.8　透明与组合

　7.9　绘图提示

　7.10　图像的读取器和写入器

　7.10.1　获得图像文件类型的读取器和写入器

　7.10.2　读取和写入带有多个图像的文件

　7.11　图像处理

　7.11.1　构建光栅图像

　7.11.2　图像过滤

　7.12　打印

　7.12.1　图形打印

　7.12.2　打印多页文件

　7.12.3　打印预览

　7.12.4　打印服务程序

　7.12.5　流打印服务程序

　7.12.6　打印属性

　7.13　剪贴板

　7.13.1　用于数据传递的类和接口

　7.13.2　传递文本

　7.13.3　Transferable接口和数据风格

　7.13.4　构建一个可传递的图像

　7.13.5　通过系统剪贴板传递Java对象

　7.13.6　使用本地剪贴板来传递对象引用

　7.14　拖放操作

　7.14.1　Swing对数据传递的支持

　7.14.2　拖曳源

　7.14.3　放置目标

　7.15　平台集成

　7.15.1　闪屏

　7.15.2　启动桌面应用程序

　7.15.3　系统托盘

第8章　JavaBean构件

8.1　为何使用Bean

　8.2　编写Bean的过程

　8.3　使用Bean构造应用程序

　8.3.1　将Bean打包成JAR文件

　8.3.2　在开发环境中组合Bean

　8.4　Bean属性与事件的命名模式

　8.5　Bean属性的类型

　8.5.1　简单属性

　8.5.2　索引属性

　8.5.3　绑定属性

　8.5.4　约束属性

　8.6　BeanInfo类

　8.7　属性编辑器

　8.8　定制器

　8.9　JavaBean持久化

　8.9.1　JavaBean持久化可用于任何数据

　8.9.2　一个JavaBean持久化的完整示例

第9章　安全

　9.1　类加载器

　9.1.1　类加载器的层次结构

　9.1.2　将类加载器作为命名空间

　9.1.3　编写你自己的类加载器

　9.2　字节码校验

　9.3　安全管理器与访问权限

　9.3.1　Java平台安全性

　9.3.2　安全策略文件

　9.3.3　定制权限

　9.3.4　实现权限类

　9.4　用户认证

　9.5　数字签名

　9.5.1　消息摘要

　9.5.2　消息签名

　9.5.3　校验签名

　9.5.4　认证问题

　9.5.5　证书签名

　9.5.6　证书请求

　9.6　代码签名

　9.6.1　JAR文件签名

　9.6.2　软件开发者证书

　9.7　加密

　9.7.1　对称密码

　9.7.2　密钥生成

　9.7.3　密码流

　9.7.4　公共密钥密码

第10章　脚本、编译与注解处理

　10.1　Java平台的脚本

　10.1.1　获取脚本引擎

　10.1.2　脚本赋值与绑定

　10.1.3　重定向输入和输出

　10.1.4　调用脚本的函数和方法

　10.1.5　编译脚本

　10.1.6　一个示例：用脚本处理GUI事件

　10.2　编译器API

　10.2.1　编译便捷之法

　10.2.2　使用编译工具

　10.2.3　一个示例：动态Java代码生成

　10.3　使用注解

　10.4　注解语法

　10.5　标准注解

　10.5.1　用于编译的注解

　10.5.2　用于管理资源的注解

　10.5.3　元注解

　10.6　源码级注解处理

　10.7　字节码工程

第11章　分布式对象

　11.1　客户与服务器的角色

　11.2　远程方法调用

　11.3　RMI编程模型

　11.3.1　接口与实现

　11.3.2　RMI注册表

　11.3.3　部署程序

　11.3.4　记录RMI活动的日志

　11.4　远程方法中的参数和返回值

　11.4.1　传递远程对象

　11.4.2　传递非远程对象

　11.4.3　动态类加载

　11.4.4　具有多重接口的远程引用

　11.4.5　远程对象与equals、hashCode和clone方法

　11.5　远程对象激活

第12章　本地方法

　12.1　从Java程序中调用C函数

　12.2　数值参数与返回值

　12.3　字符串参数

　12.4　访问域

　12.4.1　访问实例域

　12.4.2　访问静态域

　12.5　编码签名

　12.6　调用Java方法

　12.6.1　实例方法

　12.6.2　静态方法

　12.6.3　构造器

　12.6.4　替代方法调用

　12.7　访问数组元素

　12.8　错误处理

　12.9　使用调用API

　12.10　完整的示例：访问Windows注册表

　12.10.1　Windows注册表概述

　12.10.2　访问注册表的Java平台接口

　12.10.3　以本地方法方式实现注册表访问函数