

<<Java核心技术卷1>>

图书基本信息

书名：<<Java核心技术卷1>>

13位ISBN编号：9787115188335

10位ISBN编号：7115188335

出版时间：2008-11

出版时间：人民邮电出版社

作者：（美）Cay S.Horstmann Gary Cornel

页数：836

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>

## 前言

1995年底, Java语言在Internet舞台上一亮相便名声大噪。其原因在于它将有希望成为通过信息将用户联系起来的“万能胶”, 而不论这些信息来自于Web服务器、数据库、信息提供商, 还是任何其他渠道。

事实上, 就发展前景而言, Java的地位是独一无二的。

它是一种完全可以信赖的程序设计语言, 得到了除微软之外所有厂家的认可。

其固有的可靠性与安全性不仅令Java程序员放心, 也令使用Java程序的用户放心。

Java内建了对网络编程、数据库连接、多线程等高级程序设计任务的支持。

1995年以来, SUN Microsystems公司已经发布了Java开发工具箱 (Java Development Kit) 的7个主要版本。

在过去的11年中, 应用程序接口 (API) 已经从200个类扩展到3000个类, 并覆盖了用户界面构建、数据库管理、国际化、安全性以及XML处理等各个不同的领域。

本书是《Java核心技术》第8版的卷I。

自《Java核心技术》出版以来, 每个新版本都尽可能快地跟上Java开发工具箱发展的步伐, 并重新改写部分内容, 以适应Java的最新特性。

在这一版中, 已经反映了Java标准版 (Java SE 6) 的特性。

与前几版一样, 本版仍然将读者群定位在那些打算将Java应用到实际工程项目中的程序设计人员。

本书假设读者是一名具有程序设计语言 (除Java之外) 坚实背景知识的程序设计人员, 并且不希望书中充斥着玩具式的示例 (诸如烤面包机、动物园的动物或神经质的跳动文本)。

这些内容绝对不会在本书中出现。

本书的目标是让读者充分地理解书中介绍的Java语言及Java类库的相关特性, 而不会产生任何误解。

在本书中, 我们选用大量的示例代码演示所讨论的每一个语言特性和类库特性。

我们有意使用简单的示例程序以突出重点, 然而, 其中的大部分既不是赝品也没有偷工减料。

它们将成为读者自己编写代码的良好开端。

## 内容概要

《Java核心技术卷1：基础知识(第8版)(英文版)》第1版出版以来，一直备受广大Java程序设计人员的青睐，畅销不衰，是Java经典书籍。

第8版针对Java SE 6平台进行了全面更新，囊括了Java 2平台、标准版(J2SE)的全部基础知识，提供了大量完整且具有实际意义的应用实例，详细介绍了Java语言基础知识、面向对象程序设计、接口与内部类、事件监听器模型、Swing图形用户界面程序设计、打包应用程序、异常处理、登录与调试、泛型程序设计、集合框架、多线程等内容。

《Java核心技术卷1：基础知识(第8版)(英文版)》示例程序经过精心设计，不但具有实用价值，而且易于阅读理解，可以作为初学者自己编写程序的良好开端，也能够帮助程序员快速地了解Java SE 6的新特性，或迅速从其他语言转向Java语言。

## 书籍目录

Contents1 AN INTRODUCTION TO JAVAJava As a Programming PlatformThe Java “ White Paper ”  
 Buzzwords Simple Object Oriented Network-Savvy Robust Secure Architecture Neutral Portable  
 Interpreted High Performance Multithreaded DynamicJava Applets and the InternetA Short History of  
 JavaCommon Misconceptions about Java2 THE JAVA PROGRAMMING ENVIRONMENTInstalling the Java  
 Development Kit Downloading the JDK Setting the Execution Path Installing the Library Source and  
 Documentation Installing the Core Java Program Examples Navigating the Java DirectoriesChoosing a  
 Development EnvironmentUsing the Command-Line Tools Troubleshooting HintsUsing an Integrated  
 Development Environment Locating Compilation ErrorsRunning a Graphical ApplicationBuilding and Running  
 Applets3 FUNDAMENTAL PROGRAMMING STRUCTURES IN JAVA  
 A Simple Java ProgramCommentsData Types Integer Types Floating-Point Types The char Type The boolean  
 TypeVariablesInitializing Variables ConstantsOperators Increment and Decrement Operators Relational and  
 boolean Operators Bitwise Operators Mathematical Functions and Constants Conversions between Numeric  
 Types Casts Parentheses and Operator Hierarchy Enumerated TypesStrings Substrings Concatenation  
 Strings Are Immutable Testing Strings for Equality Code Points and Code Units The String API Reading  
 the On-Line API Documentation Building StringsInput and Output Reading Input Formatting Output  
 File Input and OutputControl Flow Block Scope Conditional Statements Loops Determinate Loops  
 Multiple Selections—The switch Statement Statements That Break Control FlowBig NumbersArrays The  
 “ for each ” Loop Array Initializers and Anonymous Arrays Array Copying Command-Line Parameters  
 Array Sorting Multidimensional Arrays Ragged Arrays4 OBJECTS AND CLASSESIntroduction to  
 Object-Oriented Programming Classes Objects Identifying Classes Relationships between ClassesUsing  
 Predefined Classes Objects and Object Variables The GregorianCalendar Class of the Java Library Mutator  
 and Accessor MethodsDefining Your Own Classes An Employee Class Use of Multiple Source Files  
 Dissecting the Employee Class First Steps with Constructors Implicit and Explicit Parameters Benefits of  
 Encapsulation Class-Based Access Privileges Private Methods Final Instance FieldsStatic Fields and Methods  
 Static Fields Static Constants Static Methods Factory Methods The main MethodMethod  
 ParametersObject Construction Overloading Default Field Initialization Default Constructors Explicit  
 Field Initialization Parameter Names Calling Another Constructor Initialization Blocks Object Destruction  
 and the finalize MethodPackages Class Importation Static Imports Addition of a Class into a Package  
 Package ScopeThe Class Path Setting the Class PathDocumentation Comments Comment Insertion Class  
 Comments Method Comments Field Comments General Comments Package and Overview Comments  
 Comment ExtractionClass Design Hints5 INHERITANCEClasses, Superclasses, and Subclasses Inheritance  
 Hierarchies Polymorphism Dynamic Binding Preventing Inheritance: Final Classes and Methods Casting  
 Abstract Classes Protected AccessObject: The Cosmic Superclass The equals Method Equality Testing and  
 Inheritance The hashCode Method The toString MethodGeneric Array Lists Accessing Array List Elements  
 Compatibility between Typed and Raw Array ListsObject Wrappers and AutoboxingMethods with a Variable  
 Number of ParametersEnumeration ClassesReflection The Class Class A Primer on Catching Exceptions  
 Using Reflection to Analyze the Capabilities of Classes Using Reflection to Analyze Objects at Runtime  
 Using Reflection to Write Generic Array Code Method Pointers!Design Hints for Inheritance6  
 INTERFACES AND INNER CLASSESInterfaces Properties of Interfaces Interfaces and Abstract  
 ClassesObject CloningInterfaces and CallbacksInner Classes Use of an Inner Class to Access Object State  
 Special Syntax Rules for Inner Classes Are Inner Classes Useful? Actually Necessary? Secure? Local Inner  
 Classes Accessing final Variables from Outer Methods Anonymous Inner Classes Static Inner ClassesProxies  
 Properties of Proxy Classes7 GRAPHICS PROGRAMMINGIntroducing SwingCreating a FramePositioning  
 a Frame Frame Properties Determining a Good Frame SizeDisplaying Information in a ComponentWorking  
 with D ShapesUsing ColorUsing Special Fonts for TextDisplaying Images8 EVENT HANDLINGBasics of Event

Handling Example: Handling a Button Click Becoming Comfortable with Inner Classes Creating Listeners  
Containing a Single Method Call Example: Changing the Look and Feel Adapter Classes Actions Mouse  
Events The AWT Event Hierarchy Semantic and Low-Level Events 9 USER INTERFACE COMPONENTS  
WITH SWING Swing and the Model-View-Controller Design Pattern Design Patterns The  
Model-View-Controller Pattern A Model-View-Controller Analysis of Swing Buttons Introduction to Layout  
Management Border Layout Grid Layout Text Input Text Fields Labels and Labeling Components  
Password Fields Text Areas Scroll Panes Choice Components Checkboxes Radio Buttons Borders  
Combo Boxes Sliders Menus Menu Building Icons in Menu Items Checkbox and Radio Button Menu  
Items Pop-Up Menus Keyboard Mnemonics and Accelerators Enabling and Disabling Menu Items  
Toolbars Tooltips Sophisticated Layout Management The Grid Bag Layout Group Layout Using No  
Layout Manager Custom Layout Managers Traversal Order Dialog Boxes Option Dialogs Creating Dialogs  
Data Exchange File Dialogs Color Choosers 10 DEPLOYING APPLICATIONS AND APPLETS JAR Files  
The Manifest Executable JAR Files Resources Sealing Java Web Start The Sandbox Signed Code The  
JNLP API Applets A Simple Applet The Applet HTML Tag and Its Attributes The object Tag Use of  
Parameters to Pass Information to Applets Accessing Image and Audio Files The Applet Context Storage of  
Application Preferences Property Maps The Preferences API 11 EXCEPTIONS, LOGGING, ASSERTIONS,  
AND DEBUGGING Dealing with Errors The Classification of Exceptions Declaring Checked Exceptions  
How to Throw an Exception Creating Exception Classes Catching Exceptions Catching Multiple Exceptions  
Rethrowing and Chaining Exceptions The finally Clause Analyzing Stack Trace Elements Tips for Using  
Exceptions Using Assertions Assertion Enabling and Disabling Using Assertions for Parameter Checking  
Using Assertions for Documenting Assumptions Logging Basic Logging Advanced Logging Changing the  
Log Manager Configuration Localization Handlers Filters Formatters A Logging Recipe Debugging Tips  
Using a Console Window Tracing AWT Events Letting the AWT Robot Do the Work Using a Debugger 12  
GENERIC PROGRAMMING Why Generic Programming? Who Wants to Be a Generic  
Programmer? Definition of a Simple Generic Class Generic Methods Bounds for Type Variables Generic Code and  
the Virtual Machine Translating Generic Expressions Translating Generic Methods Calling Legacy  
Code Restrictions and Limitations Type Parameters Cannot Be Instantiated with Primitive Types Runtime  
Type Inquiry Only Works with Raw Types You Cannot Throw or Catch Instances of a Generic Class Arrays of  
Parameterized Types Are Not Legal You Cannot Instantiate Type Variables Type Variables Are Not Valid in  
Static Contexts of Generic Classes Beware of Clashes After Erasure Inheritance Rules for Generic Types Wildcard  
Types Supertype Bounds for Wildcards Unbounded Wildcards Wildcard Capture Reflection and Generics  
Using Class Parameters for Type Matching Generic Type Information in the Virtual Machine 13  
COLLECTIONS Collection Interfaces Separating Collection Interfaces and Implementation Collection and  
Iterator Interfaces in the Java Library Concrete Collections Linked Lists Array Lists Hash Sets Tree Sets  
Object Comparison Queues and Deques Priority Queues Maps Specialized Set and Map Classes The  
Collections Framework Views and Wrappers Bulk Operations Converting between Collections and  
Arrays Algorithms Sorting and Shuffling Binary Search Simple Algorithms Writing Your Own  
Algorithms Legacy Collections The Hashtable Class Enumerations Property Maps Stacks Bit Sets 14  
MULTITHREADING What Are Threads? Using Threads to Give Other Tasks a Chance Interrupting  
Threads Thread States New Threads Runnable Threads Blocked and Waiting Threads Terminated  
Threads Thread Properties Thread Priorities Daemon Threads Handlers for Uncaught  
Exceptions Synchronization An Example of a Race Condition The Race Condition Explained Lock Objects  
Condition Objects The synchronized Keyword Synchronized Blocks The Monitor Concept Volatile  
Fields Deadlocks Lock Testing and Timeouts Read/Write Locks Why the stop and suspend Methods Are  
Deprecated Blocking Queues Thread-Safe Collections Efficient Maps, Sets, and Queues Copy on Write Arrays  
Older Thread-Safe Collections Callables and Futures Executors Thread Pools Scheduled Execution  
Controlling Groups of Tasks Synchronizers Semaphores Countdown Latches Barriers Exchangers

Synchronous Queues    Example: Pausing and Resuming an Animation    Threads and Swing    Running  
Time-Consuming Tasks    Using the Swing Worker    The Single-Thread Rule    AppendixIndex

## <<Java核心技术卷1>>

### 编辑推荐

《Java核心技术卷1：基础知识(第8版)(英文版)》是Java实际项目应用的权威指导书。

《Java核心技术卷1：基础知识(第8版)(英文版)》作者Cay S. Horstmann是圣何塞州立大学计算机科学系教授、Java的倡导者。

Garv Cornell拥有20余年程序设计专业课程教学经验，撰写过多本专著，曾获得Jolt大奖提名。

《Java核心技术卷1：基础知识(第8版)(英文版)》针对Java SE 6平台进行了全面更新，并通过大量经过测试的示例说明了最重要的语言特性和类库特性。

这些示例程序经过精心设计，不但具有实用价值，而且易阅读和理解，可以作为读者自己编写程序的良好开端。

本卷详细介绍以下内容：    Java语言基础知识；    Swing图形用户界面程序设计；    泛型程序设计；    面向对象程序设计；    打包应用程序；    集合框架；    接口与内部类；    异常处理；    多线程。

    事件监听器模型；    登录与调试。

<<Java核心技术卷1>>

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>