

# 前 言

## 致读者

1995年底，Java语言在Internet舞台一亮相便名声大噪。其原因在于它将有成为连接用户与信息的万能胶，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。它是一种完全可信赖的程序设计语言，得到了除微软之外的所有厂家的认可。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版的卷一。自《Java核心技术》出版以来，每个新版本都尽可能地跟上Java开发工具箱发展的步伐，而且每一版都重新改写了部分内容，以便适应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地跳动文本）。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。

我们假定读者愿意（甚至渴望）学习Java提供的所有高级特性。本书将详细介绍下列内容：

- 面向对象程序设计
- 反射与代理
- 接口与内部类
- 事件监听器模型
- 使用Swing UI工具箱进行图形用户界面设计
- 异常处理
- 泛型程序设计
- 集合框架
- 并行操作

随着Java类库的爆炸式增长，一本书无法涵盖程序员需要了解的所有Java特性。因此，我们决定将本书分为两卷。卷I（本书）集中介绍Java的基本概念以及图形用户界面程序设计的基础知识。卷II——高级特性，涉及企业特性以及高级的用户界面程序设计。其中包含下列内容：

- 文件与流
- 数据库
- 分布式对象
- 高级GUI组件

- 本地方法
- XML处理
- 网络编程
- 高级图形
- 国际化
- JavaBean
- 注释

在这一版中，我们对两卷中的内容进行了调整，特别是，鉴于多线程的日趋重要，我们将它编入卷I中，并以摩尔定律作为结尾。

在编写本书的过程中，难免出现错误和不准确之处。我们很想知道这些错误，当然，也希望同一个问题只被告知一次。我们在网页<http://horstmann.coni/coreJava.html>中以列表的形式给出了常见的问题、bug修正和出错位置。在勘误页（建议先阅读一遍）最后附有用来报告bug并提出修改意见的表单。如果我们不能回答每一个问题或没有及时回复，请不要失望。我们会认真地阅读所有的来信，感谢您的建议使本书后续的版本更清晰、更有指导价值。

## 关于本书

第1章概述Java与其他程序设计语言不同的性能。解释这种语言的设计初衷，以及在哪些方面达到了预期的效果。然后，简要叙述Java诞生和发展的历史。

第2章详细地论述如何下载和安装JDK以及本书的程序示例。然后，通过编译和运行三个典型的Java程序（一个控制台应用、一个图形应用、一个applet），指导读者使用简易的JDK、可启用Java的文本编辑器以及一个Java IDE。

第3章开始讨论Java 语言。这一章涉及的基础知识有变量、循环以及简单的函数。对于C或C++程序员来说，学习这一章的内容将会感觉一帆风顺，因为这些语言特性的语法本质上与C语言相同。对于没有C语言程序设计背景，但使用过其他程序设计语言（Visual Basic）的程序员，仔细地阅读这一章是非常必要的。

第4章介绍面向对象程序设计（Object-Oriented Programming, OOP）是当今程序设计的主流，而Java 是完全面向对象的。本章将介绍面向对象两个基本成分中最重要的——封装，以及Java 语言实现封装的机制，即类与方法。除了Java 语言规则之外，还对如何正确地进行OOP设计给出了忠告。最后，介绍奇妙的Javadoc 工具，它将代码注释转换为超链接的网页。熟悉C++的程序员可以快速地浏览这一章，而没有面向对象程序设计背景的程序员，应在进一步学习Java之前花一些时间了解OOP的有关概念。

第5章介绍类与封装仅仅是OOP中的一部分，本章将介绍另一部分——继承。继承使程序员可以使用现有的类，并根据需要进行修改。这是Java程序设计中的基础。Java中的继承机制与C++的继承机制十分相似。C++程序员只需关注两种语言的不同之处即可。

第6章展示如何使用Java的接口。接口可以让你的理解超越第5章的简单继承模型。掌握接口的使用将可以获得Java完全的面向对象程序设计的能力。本章还将介绍Java的一个有用的技术特性——内部类。内部类可以使代码更清晰、更简洁。

第7章开始细致地讨论应用程序设计。每一个Java程序员都应该了解一些图形用户界面程序设计的知识，本卷中包含了其中的基本内容部分。本章将展示如何制作窗口、如何在窗口中绘图、如何用几何图形作画、如何用多种字体格式化文本以及如何显示图像。

第8章详细讨论AWT（Abstract Window Toolkit）的事件模型。我们将介绍如何编写代码来响

应鼠标点击或敲击键盘等事件。同时，还将介绍如何处理基本的GUI元素，比如：按钮和面板。

第9章详细讨论Swing GUI 工具箱。Swing工具箱允许建立一个跨平台的图形用户界面。本章将介绍如何建立各种各样的按钮、文本组件、边界、滑块、列表框、菜单以及对话框等等。一些更高级的组件将在卷II中讨论。

第10章阐述如何部署自己编写的应用程序或applet。在这里将描述如何将应用程序打包到JAR 文件中，以及如何使用Java的Web Start 机制在Internet上发布应用程序。最后，将解释Java部署之后如何存储、检索配置信息。

第11章讨论异常处理，即Java的健壮机制，它用于处理调试好的程序可能出现的意外的情况。异常提供了一种将正常的处理代码与错误处理代码分开的有效手段。当然，即使程序包含处理所有异常情况的功能，依然有可能无法按照预计的方式工作。这一章的后半部分，将给出大量的实用调试技巧。最后，讲述如何使用各种工具完成一个示例程序。

第12章概要介绍泛型程序设计，这是Java SE5.0的一项重要改进。泛型程序设计使得程序拥有更好的可读性和安全性。在这里，将展示如何使用强类型机制，而舍弃不安全的强制类型转换，以及如何处理与旧版本Java兼容而带来的复杂问题。

第13章介绍Java平台的集合框架。当需要将大量对象收集到一起，并在过后要对它们进行检索时，可能会想要使用集合，这是目前最为合适的，它取代了将这些元素放置在数组中。本章将介绍如何使用预先建立好的标准集合。

第14章是本书的最后一章。在这章中，将介绍多线程，这是一种可以让程序任务并行执行的特性（线程是程序中的控制流），并阐述如何建立线程、如何处理线程的同步问题。从Java SE 5.0开始，多线程有了很大的改进，本章将介绍所有这些新的机制。

附录列出了Java语言的保留字。

## 约定

本书使用以下图标表示特殊内容。



注释：这个图标表示为正文提供的“注释”信息。



提示：这个图标表示为正文提供的“提示”信息。



警告：这个图标表示为正文提供的“警告”信息。



C++注释：在本书中有许多用来解释Java与C++的不同的C++注释。对于没有C++程序设计背景，或者不擅长C++程序设计的程序员，可以跳过这些注释。



### 应用程序编程接口

Java带有一个很大的程序设计库，即应用程序编程接口（API）。第一次调用API时，将会在这一节的结尾给出一个概要描述，并标有API图标。这些描述十分通俗易懂，希望能够比联机API文档提供更多的信息。我们在每一个API注释特性上都标记了版本号，用来提示那些不希

望使用Java “ 风险 ” 版本的读者。

程序源代码按照下列格式给出：

Listing 2-4 Wel comeAppl et. Java

## 示例代码

本书的网站<http://www.phptr.conl/coreJava> 以压缩的形式提供了书中的所有示例代码。可以用相应的解压缩程序或者用Java 开发工具箱中的jar实用程序展开这个文件。有关安装Java 开发工具箱和示例程序的详细信息请参看第2章。