

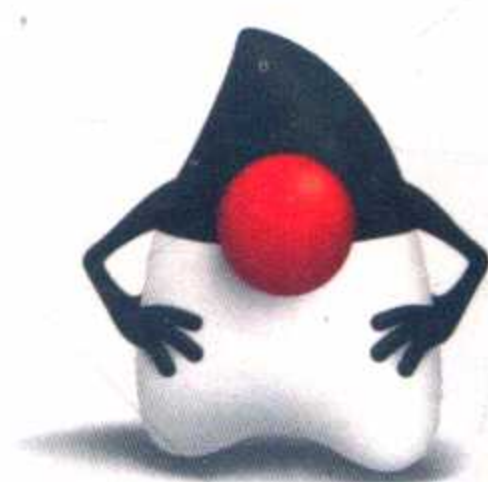
Mc
Graw
Hill
Education

ORACLE®



完全更新为Java SE 9(JDK9)

Java: A Beginner's Guide, Seventh Edition



Java

编程入门官方教程(第7版)

创建、编译和运行Java 9程序!

[美] Herbert Schildt 著

左 雷 译

Oracle
Press™

Mc
Graw
Hill
Education

清华大学出版社

Java 编程入门

官方教程

(第7版)

[美] Herbert Schildt 著

左 雷 译

清华大学出版社

北 京

Herbert Schildt

Java: A Beginner's Guide, Seventh Edition

EISBN: 978-1-259-58931-7

Copyright © 2018 by McGraw-Hill Education.

All Rights reserved. No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including without limitation photocopying, recording taping, or any database, information or retrieval system, without the prior written permission of the publisher.

This authorized Chinese translation edition is jointly published by McGraw-Hill Education and Tsinghua University Press Limited. This edition is authorized for sale in the People's Republic of China only, excluding Hong Kong, Macao SAR and Taiwan.

Translation copyright © 2018 by McGraw-Hill Education and Tsinghua University Press Limited.

版权所有。未经出版人事先书面许可，对本出版物的任何部分不得以任何方式或途径复制或传播，包括但不限于复印、录制、录音，或通过任何数据库、信息或可检索的系统。

本授权中文简体字翻译版由麦格劳-希尔(亚洲)教育出版公司和清华大学出版社有限公司合作出版。此版本经授权仅限在中国大陆区域销售，不能销往中国香港、澳门特别行政区和中国台湾地区。

版权©2018 由麦格劳-希尔(亚洲)教育出版公司与清华大学出版社有限公司所有。

北京市版权局著作权合同登记号 图字：01-2018-0331

本书封面贴有 McGraw-Hill Education 公司防伪标签，无标签者不得销售。

版权所有，侵权必究。侵权举报电话：010-62782989 13701121933

图书在版编目(CIP)数据

Java 编程入门官方教程(第7版)/(美)赫伯特·希尔德特(Herbert Schildt)著；左雷译。—北京：清华大学出版社，2018

书名原文：Java: A Beginner's Guide, Seventh Edition

ISBN 978-7-302-49932-9

I. ①J… II. ①赫… ②左… III. ①JAVA 语言—程序设计—教材 IV. ①TP312.8

中国版本图书馆 CIP 数据核字(2018)第 064681 号

责任编辑：王 军 韩宏志

封面设计：牛艳敏

版式设计：思创景点

责任校对：曹 阳

责任印制：李红英

出版发行：清华大学出版社

网 址：<http://www.tup.com.cn>, <http://www.wqbook.com>

地 址：北京清华大学学研大厦 A 座

邮 编：100084

社 总 机：010-62770175

邮 购：010-62786544

投稿与读者服务：010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈：010-62772015, zhiliang@tup.tsinghua.edu.cn

印 装 者：清华大学印刷厂

经 销：全国新华书店

开 本：190mm×260mm

印 张：30

字 数：992 千字

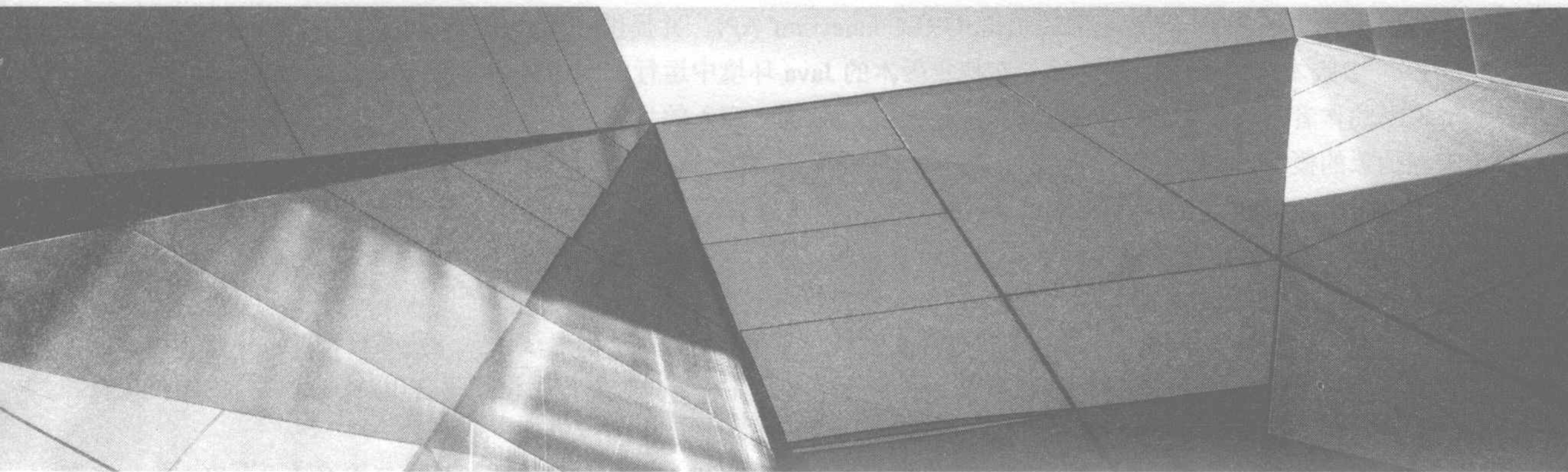
版 次：2018 年 6 月第 1 版

印 次：2018 年 6 月第 1 次印刷

印 数：1~5500

定 价：89.00 元

产品编号：078952-01



译者序

Java 是一种可以撰写跨平台应用程序的面向对象的程序设计语言。Java 技术具有卓越的通用性、高效性、平台移植性和安全性，广泛应用于 PC、数据中心、游戏控制台、科学超级计算机、移动电话和互联网，同时拥有全球最大的开发者专业社群。Java 是印度尼西亚爪哇岛的英文名称，因盛产咖啡而闻名。Java 语言中的许多库类名称多与咖啡有关：如 JavaBeans(咖啡豆)、NetBeans(网络豆)以及 ObjectBeans(对象豆)等。Sun 和 Java 的标识也正是一杯正冒着热气的咖啡。

Java 自面世后就非常流行，发展迅速，对 C++ 语言形成有力冲击。在全球云计算和移动互联网的产业环境下，Java 更具备了显著优势和广阔前景。Java 的特性如下：

- (1) Java 语言是易学的；
- (2) Java 语言是强制面向对象的；
- (3) Java 语言是分布式的；
- (4) Java 语言是健壮的；
- (5) Java 语言是安全的；
- (6) Java 语言是体系结构中立的；
- (7) Java 语言是可移植的；
- (8) Java 语言是解释型的；
- (9) Java 性能较高；
- (10) Java 语言是原生支持多线程的；
- (11) Java 语言是动态的。

Java 8 发布三年多之后，发布了 Java 9。这个新版本主要有如下 9 个新功能：

- **Java 平台级模块系统：**这是一套全新的模块系统，模块允许更好地强制结构化封装应用程序，并明确依赖。
- **Linking：**可以创建针对应用程序进行优化的最小运行时映像。
- **JShell：**这是一个交互式 Java REPL，它的即时反馈使它成为探索 API 和尝试语言特性的好工具。
- **改进的 Javadoc：**它现在支持在 API 文档中进行搜索。Javadoc 的输出也兼容 HTML5 标准。每个 Javadoc 页面都包含有关 JDK 模块类或接口来源的信息。
- **集合工厂方法：**可在代码中创建一个集合，并直接用一些元素填充它。
- **改进的 Stream API：**可在集合上建立用于转换的管道。Optional 和 Stream 之间的结合也得到了改进。

- **私有接口方法：**可以避免代码的重复，但不会成为公共 API 的一部分。
- **HTTP/2：**用于代替老旧的 'HttpURLConnection' API，并提供对 WebSocket 和 HTTP/2 的支持。
- **多版本兼容 JAR：**能创建仅在特定版本的 Java 环境中运行库程序时选择使用的 class 版本。

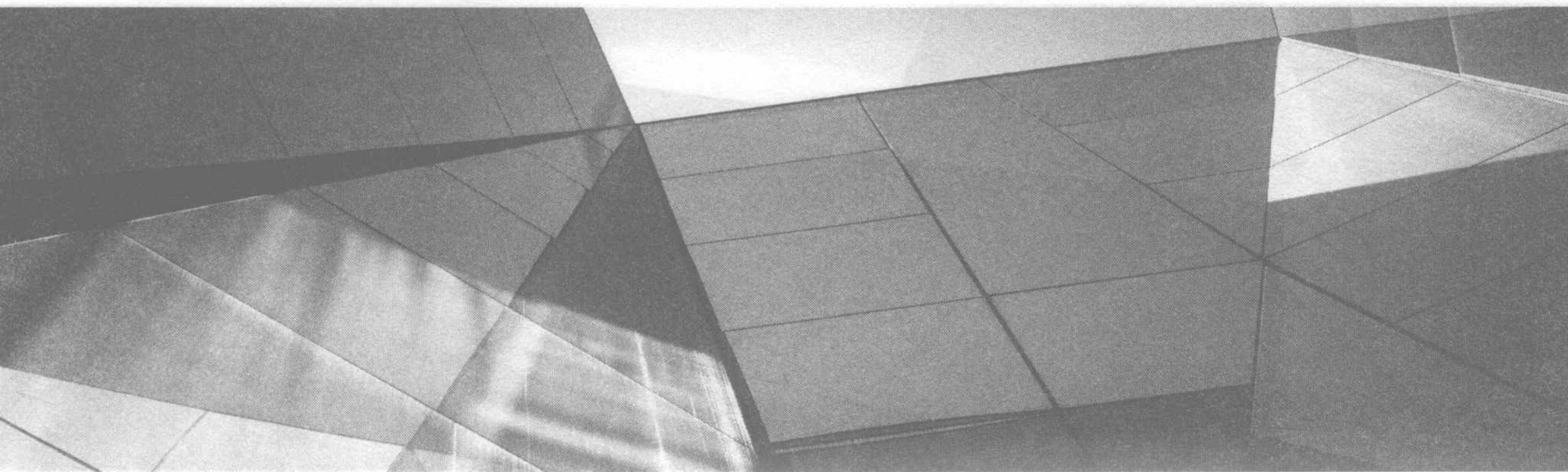
本书的作者拥有多年教学经验，对 Java 语言有独到、深入的见解，以通俗易懂、短小而直接的示例解释一个个晦涩抽象的概念。本书是 Java 技术经典参考书，多年畅销不衰，第 7 版在保留以前版本风格的基础上，涵盖 Java 开发平台标准版 Java SE 9 的基础知识，本书共 17 章，主题包括 Java 基础、数据类型、运算符、程序控制语句、类、对象、方法、继承、包、接口、异常处理、使用 I/O、多线程程序设计、枚举、自动装箱、静态导入、注解、泛型、lambda 表达式、方法引用、模块、Swing 和 JavaFX 等。

本书内容详实、深入浅出，附有大量程序实例，包含 Java 语言基础语法以及一些高级特性，极具实用价值，是 Java 初学者和 Java 程序员的必备参考书，也是高等院校讲授面向对象程序设计语言以及 Java 语言的绝佳教材和参考书。

这里要感谢清华大学出版社的编辑们，他们为本书的翻译投入了巨大的热情并付出了很多心血。没有她们的帮助和鼓励，本书不可能顺利付梓。本书主要由左雷翻译，参与翻译的还有陈妍、何美英、陈宏波、熊晓磊、管兆昶、潘洪荣、曹汉鸣、高娟妮、王燕、谢李君、李珍珍、王璐、王华健、柳松洋、曹晓松、陈彬、洪妍、刘芸、邱培强、高维杰、张素英、颜灵佳、方峻、顾永湘、孔祥亮。

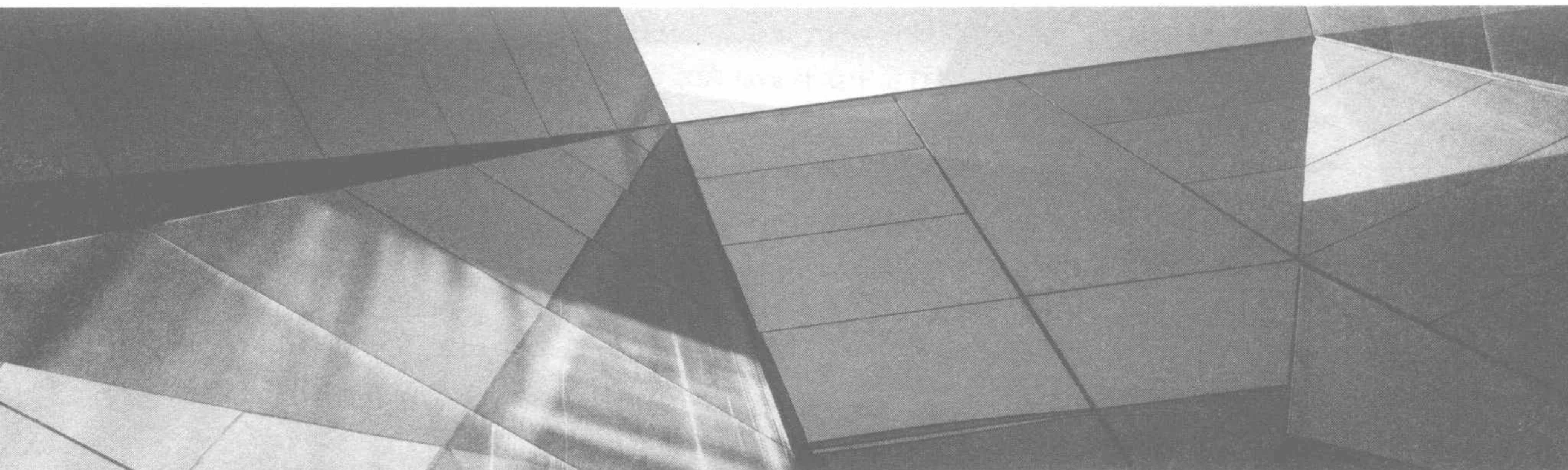
对于这本经典之作，译者本着“诚惶诚恐”的态度，在翻译过程中力求“信、达、雅”，但鉴于译者水平有限，错误和失误在所难免，如有任何意见和建议，请不吝指正。

译者



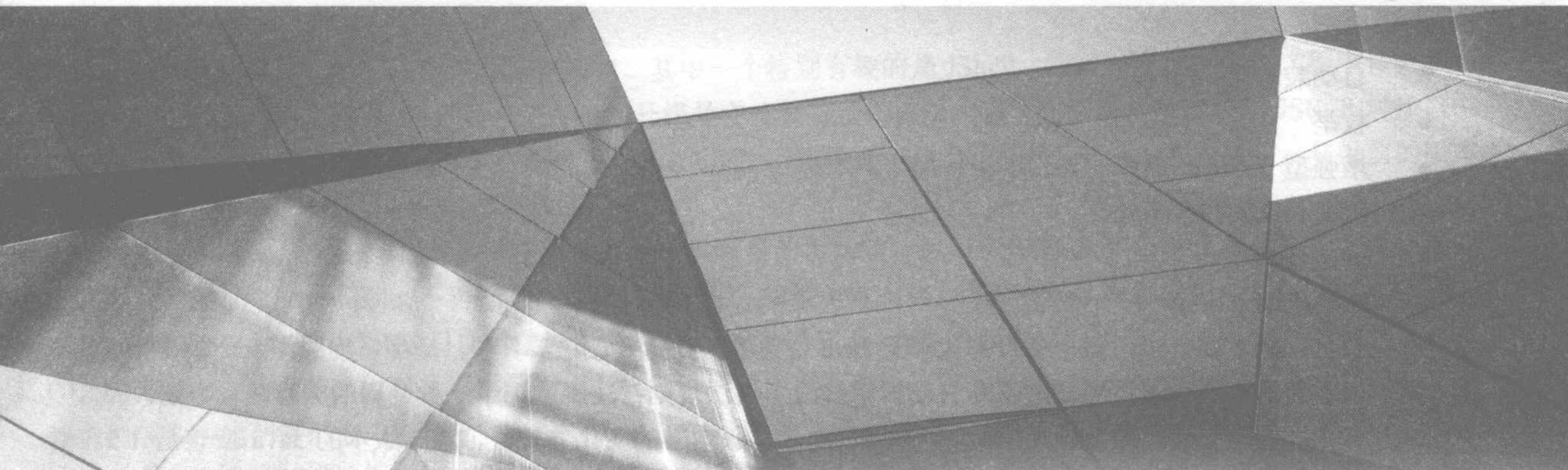
作者简介

畅销书作家 Herbert Schildt 是 Java 语言的权威，他撰写程序设计图书超过三十年，在全世界销售了数百万册，并被翻译成了多种语言。他撰写了众多关于 Java 的图书，包括 *Java: The Complete Reference*、*Herb Schildt's Java Programming Cookbook*、*Introducing JavaFX 8 Programming* 和 *Swing: A Beginner's Guide*，还撰写了一些有关 C、C++ 和 C# 方面的书籍。虽然他对计算机的方方面面都很感兴趣，但主要关注点是计算机语言。Schildt 获得了伊利诺伊大学的学士和硕士学位。



技术编辑简介

Danny Coward 博士在 Java 平台的各个版本上都工作过。他将 Java servlet 的定义引入 Java EE 平台的第一个版本中，将 Web 服务引入 Java ME 平台，并提出了 Java SE 7 的战略和规划。他开发了 JavaFX 技术，最近，他还为 Java EE 7 标准设计了 Java WebSocket API。从用 Java 编写代码到与行业专家设计 API，到他作为 Java Community Process Committee 主管的多年来，他对 Java 技术的多个方面都具有独到的见解。另外，他还是 *Java WebSocket Programming* 和 *Java EE: The Big Picture* 两本书的作者。Coward 博士拥有牛津大学数学系的学士、硕士和博士学位。



前言

本书旨在帮助你学习 Java 程序设计的基础知识，采用循序渐进的教学方法，安排了许多示例、自测题和编程练习。本书不需要读者具备编程经验，而是从最基础的知识，从如何编译并运行一个 Java 程序开始讲起。然后讨论构成 Java 语言核心的关键字、功能和结构。还介绍 Java 的一些最重要高级功能，如多线程编程、泛型和 lambda 表达式和模块。此外，本书还介绍 Swing 基础和 JavaFX。学完本书后，读者将牢固掌握 Java 编程精髓。

值得说明的是，本书只是学习 Java 的起点。Java 不仅是一些定义语言的元素，还包括扩展的库和工具来帮助开发程序。要想成为顶尖的 Java 程序员，就必须掌握这些知识。读者在学习完本书后，就有了足够的知识来继续学习 Java 的其他方面。

0.1 Java 的发展历程

只有少数几种编程语言对程序设计带来过根本性影响。其中，Java 的影响由于迅速和广泛而格外突出。可以毫不夸张地说，1995 年 Sun 公司发布的 Java 1.0 给计算机程序设计领域带来了一场变革。这场变革迅速地把 Web 转变成了一个高度交互的环境，也给计算机语言的设计设置了一个新标准。

多年来，Java 不断发展、演化和修订。和其他语言加入新功能的动作迟缓不同，Java 一直站在计算机程序设计语言的前沿，部分原因是其变革的文化，部分原因是它所面对的变化。Java 已经做过或大或小的多次升级。

第一次主要升级是 Java 1.1 版，这次升级比较大，加入了很多新的库元素，修订了处理事件的方式，重新配置了 1.0 版本的库中的许多功能。

第二个主要版本是 Java 2，它代表 Java 的第二代，标志着 Java “现代化”的到来。Java 2 第一个发布的版本号是 1.2。Java 2 在第一次发布时使用 1.2 版本号看上去有些奇怪。原因在于，该号码最初指 Java 库的内部版本号，后来就泛指整个版本号了。Java 2 被 Sun 重新包装为 J2SE(Java 2 Platform Standard Edition)，并且开始把版本号应用于该产品。

Java 的下一升级是 J2SE 1.3，它是 Java 2 版本首次较大的升级。它增强了一些已有的功能，并且精简了开发环境。J2SE 1.4 进一步增强了 Java。该版本包括一些重要的新功能，如链式异常、基于通道的 I/O 以及 assert 关键字。

Java 的下一版本是 J2SE 5，它是 Java 的第二次变革。以前的几次 Java 升级提供的改进虽然重要，但都是增量式的，而 J2SE 5 却从该语言的作用域、功能和范围等方面提供了根本性改进。为帮助理解 J2SE 5 的修改程度，下

面列出了 J2SE 5 中的一些主要新功能：

- 泛型
- 自动装箱/自动拆箱
- 枚举
- 增强型 for-each 形式的 for 循环
- 可变长度实参(varargs)
- 静态导入
- 注解(annotation)

这些条目都是重要升级，每个条目都代表了 Java 语言的一处重要改进。其中，泛型、增强型 for 循环和可变长度实参引入了新的语法元素；自动装箱和自动拆箱修改了语法规则；注解增加了一种全新的编程注释方法。

这些新功能的重要性反映在使用的版本号“5”上。从版本号的变化方式看，这一版本的 Java 应该是 1.5。由于新功能和变革如此之多，常规的版本号升级(从 1.4 到 1.5)已无法标识实际的变化，因此 Sun 决定使用版本号 5，以强调发生了重要改进。因此，当前的版本称为 J2SE 5，开发工具包称为 JDK 5。但是，为了保持和以前的一致性，Sun 决定使用 1.5 作为内部版本号，也称为开发版本号。J2SE 5 中的“5”称为产品版本号。

之后发布的 Java 版本是 Java SE 6，Sun 再次决定修改 Java 平台的名称，把“2”从版本号中删除了。因此，Java 平台现在的名称是 Java SE，官方产品名称是 Java Platform Standard Edition 6，对应的 Java 开发工具包称为 JDK 6。和 J2SE 5 一样，Java SE 6 中的“6”是指产品的版本号，内部的开发版本号是 1.6。

Java SE 6 建立在 J2SE 5 的基础之上，做了进一步的增强和改进。Java SE 6 并没有对 Java 语言本身添加较大的功能，而是增强了 API 库，添加了多个新包，改进了运行时环境。它在漫长的生命周期(Java 术语)内经历了一些更新，添加了一些升级功能。总之，Java SE 6 进一步巩固了 J2SE 5 建立的领先地位。

Java 的下一版本是 Java SE 7，对应的 Java 开发工具包称为 JDK 7，内部版本号是 1.7。Java SE 7 是 Oracle 收购 Sun Microsystems 之后发布的第一个主版本。Java SE 7 包含许多新功能，对语言和 API 库做了许多增强。Java SE 7 添加的最重要功能是在 Project Coin 中开发的那些功能。Project Coin 的目的是确保把对 Java 语言所做的很多小改动包含到 JDK 7 中，其中包括：

- 现在 String 可控制 switch 语句。
- 二进制整型字面值。
- 在数值字面值中使用下划线。
- 新增一种称为 try-with-resources 的 try 语句，支持自动资源管理。
- 构造泛型实例时，通过菱形运算符使用类型推断。
- 增强了异常处理，可以使用单个 catch 捕获两个或更多个异常(多重捕获)，并且对重新抛出的异常可以进行更好的类型检查。

可以看到，虽然 Project Coin 中的功能被视为小改动，但是“小”这个词实在不能体现它们所带来的好处。特别是，try-with-resources 语句会对大量代码的编写方式产生深远影响。

Java 的下一版本是 Java SE 8，对应的开发工具包是 JDK 8，内部的开发版本号是 1.8。JDK 8 表示这是对 Java 语言的一次重大升级，因为本次升级包含了一种意义深远的新语言功能：lambda 表达式。lambda 表达式的影响深远：不但改变了概念化的编程方式，而且改变了 Java 代码的编写方式。使用 lambda 表达式，可以简化并减少创建某个结构所需的源代码量。另外，使用 lambda 表达式还可将新的运算符-> 和一种新的语法元素引入 Java 语言中。

除了 lambda 表达式，JDK 8 中还新增了一些其他重要功能。例如，从 JDK 8 开始，通过接口可以为指定的方法定义默认实现。JDK 8 也捆绑了对 JavaFX、Java 的新 GUI 框架的支持。期待 JavaFX 不久便能在几乎所有的 Java 应用程序中扮演重要的角色，并且最终替代用于大多数基于 GUI 项目的 Swing。总之，Java SE 8 主要版本扩展了 Java 语言的功能，并且改变了 Java 代码的编写方式，带来的影响足够深远。

0.2 Java SE 9

Java 的最新版本是 Java SE 9，对应的开发工具包是 JDK 9，内部的开发版本号也是 9。JDK 9 表示这是对 Java 语言的一次重大升级，合并了对 Java 语言及其库的重大改进。主要的新功能是模块，它允许指定构成应用程序的代码之间的关系和依赖。模块还给 Java 的访问控制功能添加了另一种方式。包括模块导致了一个新的语法元素、

几个新关键字和各种工具改进添加到 Java 中。模块还对 API 库有深远的影响，因为在 JDK 9 中，库包现在组织为模块。

除了模块之外，JDK 9 还包括几个新功能。其中一个特别有趣的是 JShell，它是一个支持交互式程序体验和学习工具(JShell 简介见附录 D)。另一个有趣的升级是支持私有接口方法。包含它们进一步增强了 JSK 8 对接口中默认方法的支持。JDK 9 给 javadoc 工具添加了搜索功能，还添加了一个新的标记@index 来支持它。与以前的版本一样，JDK 9 包含对 Java API 库的许多更新和改进。

作为一般规则，在任何 Java 版本中，都有最受人瞩目的新功能。但 JDK 9 废弃了 Java 高度配置的一个方面：applet。从 JDK 9 开始，applet 不再推荐在新项目中使用。如第 1 章所述，因为 applet 需要浏览器支持(和其他因素)，JDK 9 废弃了整个 applet API。目前，建议使用 Java Web Start 在 Internet 上部署应用程序(Java Web Start 简介见附录 C)。因为 applet 被设置为过时，不推荐用于新代码，所以本书不再讨论它。但是对 applet 感兴趣的读者可以在本书以前的版本中找到它们的内容。

总之，JDK 9 继续 Java 的革新，确保 Java 成为用户所期望的充满活力且敏捷的语言。本书中的内容已更新至 Java SE 9 版本(JDK 9)，包括一些新的功能、更新和其他内容。

0.3 本书的组织结构

本书采用教程式的组织结构，每一章都建立在前面的基础之上。本书共分 17 章，每一章讨论一个有关 Java 的主题。本书的特色就在于包含许多便于读者学习的特色内容。

- **关键技能与概念：**每一章都首先介绍一些该章中要介绍的重要技能。
- **自测题：**每一章都有自测题，测试读者学习到的知识。答案在附录 A 中提供。
- **专家解答：**每一章中都穿插一些“专家解答”，以一问一答的形式介绍补充知识和要点。
- **编程练习：**每一章中都包含一两个编程练习，以帮助读者将学习到的知识应用到实践中。很多这样的练习都是实际的示例，读者可以将其用作自己的程序的起点。

0.4 本书不需要读者具备编程经验

本书假定读者没有任何编程经验。如果读者没有编程经验，阅读本书是正确的选择。如果读者有过一些编程经验，在阅读本书时可以加快速度。但是要记住，Java 在几个重要的地方与其他一些流行的计算机语言不同，所以不要急于下结论。因此，即使读者是经验丰富的程序员，也仍然建议仔细阅读本书。

0.5 本书需要的软件环境

要编译和运行本书提供的所有程序，需要获得 Oracle 提供的最新版本的 Java Development Kit (JDK)。在撰写本书时，最新版本为 JDK 9，这是 Java SE 9 使用的 JDK 版本。本书第 1 章介绍如何获得 Java JDK。

如果读者使用早期版本的 Java，也仍然可以阅读本书，只是无法编译和运行使用了 Java 新功能的程序。

0.6 不要忘记 Web 上的代码

本书所有示例和编程项目的源代码都可以免费从 Web 网址 www.oraclepressbooks.com 下载，也可以扫描本书封底二维码下载。

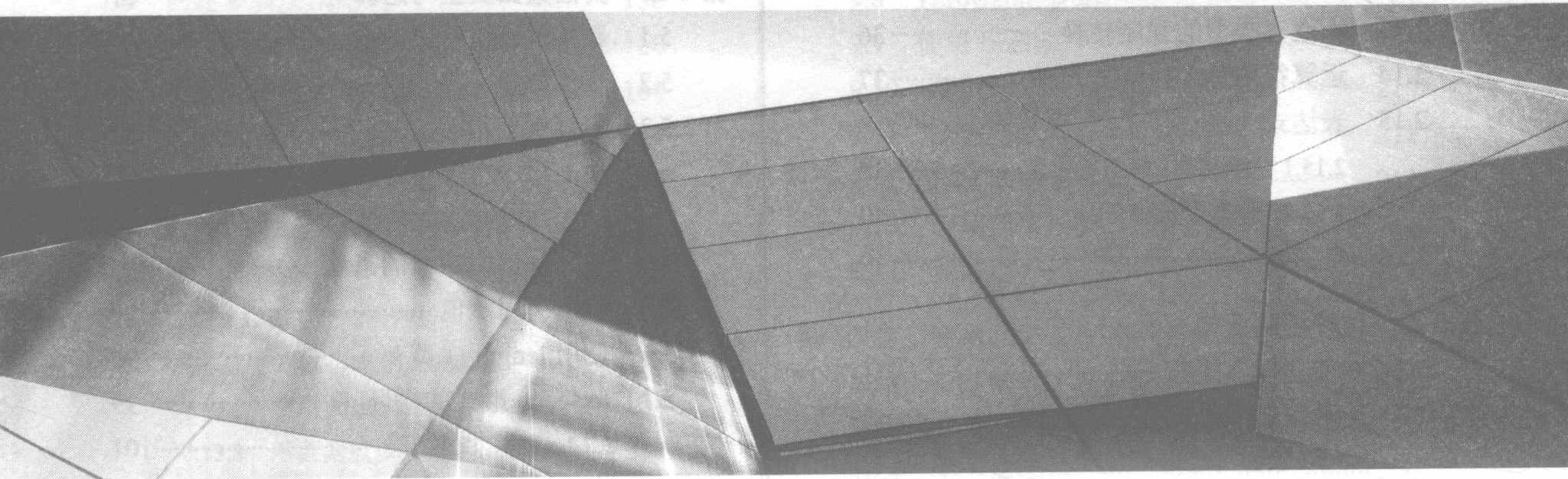
0.7 特别感谢

特别感谢本书的技术编辑 Danny Coward。他对本书提出了许多宝贵的建议和意见，对此十分感谢。

0.8 进一步学习

本书是引导读者进入 Herbert Schildt 系列编程图书的大门，对下面的一些书你也会感兴趣：

- Java: The Complete Reference*
- Herb Schildt's Java Programming Cookbook*
- The Art of Java*
- Swing: a Beginner's Guide*
- Introducing JavaFX 8 Programming*



目 录

- 第 1 章 Java 基础 1
 - 1.1 Java 的起源 1
 - 1.2 Java 对 Internet 的贡献 2
 - 1.2.1 Java 简化了基于 Web 的编程 3
 - 1.2.2 Java applet 3
 - 1.2.3 安全性 3
 - 1.2.4 可移植性 3
 - 1.3 Java 的魔法：字节码 3
 - 1.4 超越 applet 4
 - 1.5 Java 的主要术语 4
 - 1.6 面向对象程序设计 5
 - 1.6.1 封装 5
 - 1.6.2 多态性 6
 - 1.6.3 继承 6
 - 1.7 获得 Java 开发工具包 6
 - 1.8 第一个简单的程序 7
 - 1.8.1 输入程序 7
 - 1.8.2 编译程序 8
 - 1.8.3 逐行分析第一个程序 8
 - 1.9 处理语法错误 9
 - 1.10 第二个简单程序 10
 - 1.11 另一种数据类型 11
 - 1.12 两条控制语句 13
 - 1.12.1 if 语句 13
 - 1.12.2 for 循环语句 14
 - 1.13 创建代码块 15
 - 1.14 分号和定位 16

- 1.15 缩进原则 16
- 1.16 Java 关键字 18
- 1.17 Java 标识符 18
- 1.18 Java 类库 19
- 1.19 自测题 19
- 第 2 章 数据类型与运算符 21
 - 2.1 数据类型为什么重要 21
 - 2.2 Java 的基本类型 21
 - 2.2.1 整数类型 22
 - 2.2.2 浮点型 23
 - 2.2.3 字符型 23
 - 2.2.4 布尔类型 24
 - 2.3 字面值 26
 - 2.3.1 十六进制、八进制和二进制
字面值 26
 - 2.3.2 字符转义序列 26
 - 2.3.3 字符串字面值 27
 - 2.4 变量详解 28
 - 2.4.1 初始化变量 28
 - 2.4.2 动态初始化 28
 - 2.5 变量的作用域和生命期 28
 - 2.6 运算符 30
 - 2.7 算术运算符 30
 - 2.8 关系运算符和逻辑运算符 32
 - 2.9 短路逻辑运算符 33
 - 2.10 赋值运算符 34
 - 2.11 速记赋值 34

2.12	赋值中的类型转换	35
2.13	不兼容类型的强制转换	36
2.14	运算符的优先级	37
2.15	表达式	38
2.15.1	表达式中的类型转换	38
2.15.2	间距和圆括号	40
2.16	自测题	40
第 3 章	程序控制语句	41
3.1	从键盘输入字符	41
3.2	if 语句	42
3.2.1	嵌套 if 语句	43
3.2.2	if-else-if 阶梯状结构	44
3.3	switch 语句	44
3.4	for 循环	48
3.4.1	for 循环的一些变体	50
3.4.2	缺失部分要素的 for 循环	50
3.4.3	无限循环	51
3.4.4	没有循环体的循环	51
3.4.5	在 for 循环内部声明循环控制变量	52
3.4.6	增强型 for 循环	52
3.5	while 循环	52
3.6	do-while 循环	54
3.7	使用 break 语句退出循环	57
3.8	将 break 语句作为一种 goto 语句使用	58
3.9	使用 continue 语句	61
3.10	嵌套循环	65
3.11	自测题	65
第 4 章	类、对象和方法	67
4.1	类的基础知识	67
4.1.1	类的基本形式	67
4.1.2	定义类	68
4.2	如何创建对象	70
4.3	引用变量和赋值	70
4.4	方法	71
4.5	从方法返回值	73
4.6	返回值	73
4.7	使用形参	75
4.8	构造函数	81
4.9	带形参的构造函数	82
4.10	深入介绍 new 运算符	83
4.11	垃圾回收	83
4.12	this 关键字	84
4.13	自测题	85

第 5 章	其他数据类型与运算符	87
5.1	数组	87
5.2	多维数组	91
5.3	不规则数组	91
5.3.1	三维或更多维的数组	92
5.3.2	初始化多维数组	93
5.4	另一种声明数组的语法	93
5.5	数组引用赋值	94
5.6	使用 length 成员	95
5.7	for-each 形式的循环	99
5.7.1	迭代多维数组	101
5.7.2	应用增强型 for 循环	102
5.8	字符串	102
5.8.1	构造字符串	102
5.8.2	操作字符串	103
5.8.3	字符串数组	104
5.8.4	字符串是不可变的	105
5.8.5	使用 String 控制 switch 语句	106
5.9	使用命令行实参	106
5.10	位运算符	107
5.10.1	位运算符的与、或、异或和非	108
5.10.2	移位运算符	111
5.10.3	位运算符的赋值速记符	112
5.11	?运算符	114
5.12	自测题	115
第 6 章	方法和类详解	117
6.1	控制对类成员的访问	117
6.2	向方法传递对象	121
6.3	返回对象	124
6.4	方法重载	125
6.5	重载构造函数	129
6.6	递归	132
6.7	理解 static 关键字	134
6.8	嵌套类和内部类	138
6.9	varargs	140
6.9.1	varargs 基础	141
6.9.2	重载 varargs 方法	143
6.9.3	varargs 和歧义	144
6.10	自测题	145
第 7 章	继承	147
7.1	继承的基础知识	147
7.2	成员访问与继承	149
7.3	构造函数和继承	151
7.4	使用 super 调用超类构造函数	152
7.5	使用 super 访问超类成员	155

7.6 创建多级层次结构	158	9.8 Throwable 详解	207
7.7 何时调用构造函数	160	9.9 使用 finally	208
7.8 超类引用和子类对象	161	9.10 使用 throws 语句	210
7.9 方法重写	164	9.11 新增的 3 种异常功能	211
7.10 重写的方法支持多态性	166	9.12 Java 的内置异常	212
7.11 为何使用重写方法	167	9.13 创建异常子类	213
7.12 使用抽象类	170	9.14 自测题	217
7.13 使用 final	173	第 10 章 使用 I/O	219
7.13.1 使用 final 防止重写	173	10.1 Java 的 I/O 基于流	219
7.13.2 使用 final 防止继承	174	10.2 字节流和字符流	219
7.13.3 对数据成员使用 final	174	10.3 字节流类	220
7.14 Object 类	175	10.4 字符流类	220
7.15 自测题	175	10.5 预定义流	221
第 8 章 包和接口	177	10.6 使用字节流	221
8.1 包	177	10.6.1 读取控制台输入	222
8.1.1 定义包	177	10.6.2 写入控制台输出	223
8.1.2 寻找包和 CLASSPATH	178	10.7 使用字节流读写文件	223
8.1.3 一个简短的包示例	178	10.7.1 从文件输入	223
8.2 包和成员访问	179	10.7.2 写入文件	226
8.3 理解被保护的成员	181	10.8 自动关闭文件	227
8.4 导入包	182	10.9 读写二进制数据	229
8.5 Java 的类库位于包中	183	10.10 随机访问文件	232
8.6 接口	184	10.11 使用 Java 字符流	234
8.7 实现接口	185	10.11.1 使用字符流的控制台输入	235
8.8 使用接口引用	187	10.11.2 使用字符流的控制台输出	237
8.9 接口中的变量	192	10.12 使用字符流的文件 I/O	237
8.10 接口能够扩展	193	10.12.1 使用 FileWriter	238
8.11 默认接口方法	193	10.12.2 使用 FileReader	238
8.11.1 默认方法的基础知识	194	10.13 使用 Java 的类型封装器转换数值	
8.11.2 默认方法的实际应用	195	字符串	239
8.11.3 多继承问题	196	10.14 自测题	246
8.12 在接口中使用静态方法	196	第 11 章 多线程程序设计	247
8.13 私有接口方法	197	11.1 多线程的基础知识	247
8.14 有关包和接口的最后思考	198	11.2 Thread 类和 Runnable 接口	248
8.15 自测题	198	11.3 创建一个线程	248
第 9 章 异常处理	199	11.4 创建多个线程	255
9.1 异常的层次结构	199	11.5 确定线程何时结束	257
9.2 异常处理基础	200	11.6 线程的优先级	259
9.2.1 使用关键字 try 和 catch	200	11.7 同步	262
9.2.2 一个简单的异常示例	200	11.8 使用同步方法	262
9.3 未捕获异常的结果	202	11.9 同步语句	264
9.4 使用多个 catch 语句	203	11.10 使用 notify()、wait() 和 notifyAll()	
9.5 捕获子类异常	204	的线程通信	266
9.6 try 代码块可以嵌套	205	11.11 线程的挂起、继续执行和停止	270
9.7 抛出异常	206	11.12 自测题	274

第 12 章 枚举、自动装箱、静态导入和注解..... 275

12.1 枚举..... 275

12.2 Java 语言中的枚举是类类型..... 277

12.3 values()和 valueOf()方法..... 277

12.4 构造函数、方法、实例变量和枚举..... 278

12.5 枚举继承 enum..... 279

12.6 自动装箱..... 284

12.7 类型封装器..... 284

12.8 自动装箱的基础知识..... 286

12.9 自动装箱和方法..... 286

12.10 发生在表达式中的自动装箱/自动拆箱..... 287

12.11 静态导入..... 289

12.12 注解(元数据)..... 290

12.13 自测题..... 292

第 13 章 泛型..... 295

13.1 泛型的基础知识..... 295

13.2 一个简单的泛型示例..... 296

13.2.1 泛型只能用于引用类型..... 298

13.2.2 泛型类型是否相同基于其类型实参..... 298

13.2.3 带有两个类型形参的泛型类..... 299

13.2.4 泛型类的一般形式..... 300

13.3 约束类型..... 300

13.4 使用通配符实参..... 302

13.5 约束通配符..... 304

13.6 泛型方法..... 306

13.7 泛型构造函数..... 308

13.8 泛型接口..... 308

13.9 原类型和遗留代码..... 313

13.10 使用菱形运算符进行类型推断..... 315

13.11 擦除特性..... 315

13.12 歧义错误..... 316

13.13 一些泛型限制..... 316

13.13.1 类型形参不能实例化..... 316

13.13.2 对静态成员的限制..... 317

13.13.3 泛型数组限制..... 317

13.13.4 泛型异常限制..... 318

13.14 继续学习泛型..... 318

13.15 自测题..... 318

第 14 章 lambda 表达式和方法引用..... 319

14.1 lambda 表达式简介..... 319

14.1.1 lambda 表达式的基础知识..... 320

14.1.2 函数式接口..... 320

14.1.3 几个 lambda 表达式示例..... 322

14.2 块 lambda 表达式..... 325

14.3 泛型函数式接口..... 326

14.4 lambda 表达式和变量捕获..... 330

14.5 从 lambda 表达式中抛出异常..... 331

14.6 方法引用..... 332

14.6.1 静态方法的方法引用..... 332

14.6.2 实例方法的方法引用..... 334

14.7 构造函数引用..... 336

14.8 预定义的函数式接口..... 338

14.9 自测题..... 339

第 15 章 模块..... 341

15.1 模块基础..... 341

15.1.1 简单的模块示例..... 342

15.1.2 编译、运行第一个模块示例..... 345

15.1.3 requires 和 exports..... 346

15.2 java.base 和平台模块..... 346

15.3 旧代码和未命名的模块..... 347

15.4 导出到特定的模块..... 347

15.5 使用 requires transitive..... 348

15.6 使用服务..... 351

15.6.1 服务和提供程序的基础知识..... 351

15.6.2 基于服务的关键字..... 351

15.6.3 基于模块的服务示例..... 352

15.7 其他模块功能..... 356

15.7.1 open 模块..... 357

15.7.2 opens 语句..... 357

15.7.3 requires static..... 357

15.8 继续模块的学习..... 357

15.9 自测题..... 358

第 16 章 Swing 介绍..... 359

16.1 Swing 的起源和设计原则..... 359

16.2 组件和容器..... 360

16.2.1 组件..... 361

16.2.2 容器..... 361

16.2.3 顶级容器窗格..... 361

16.3 布局管理器..... 362

16.4 第一个简单的 Swing 程序..... 362

16.5 Swing 事件处理..... 365

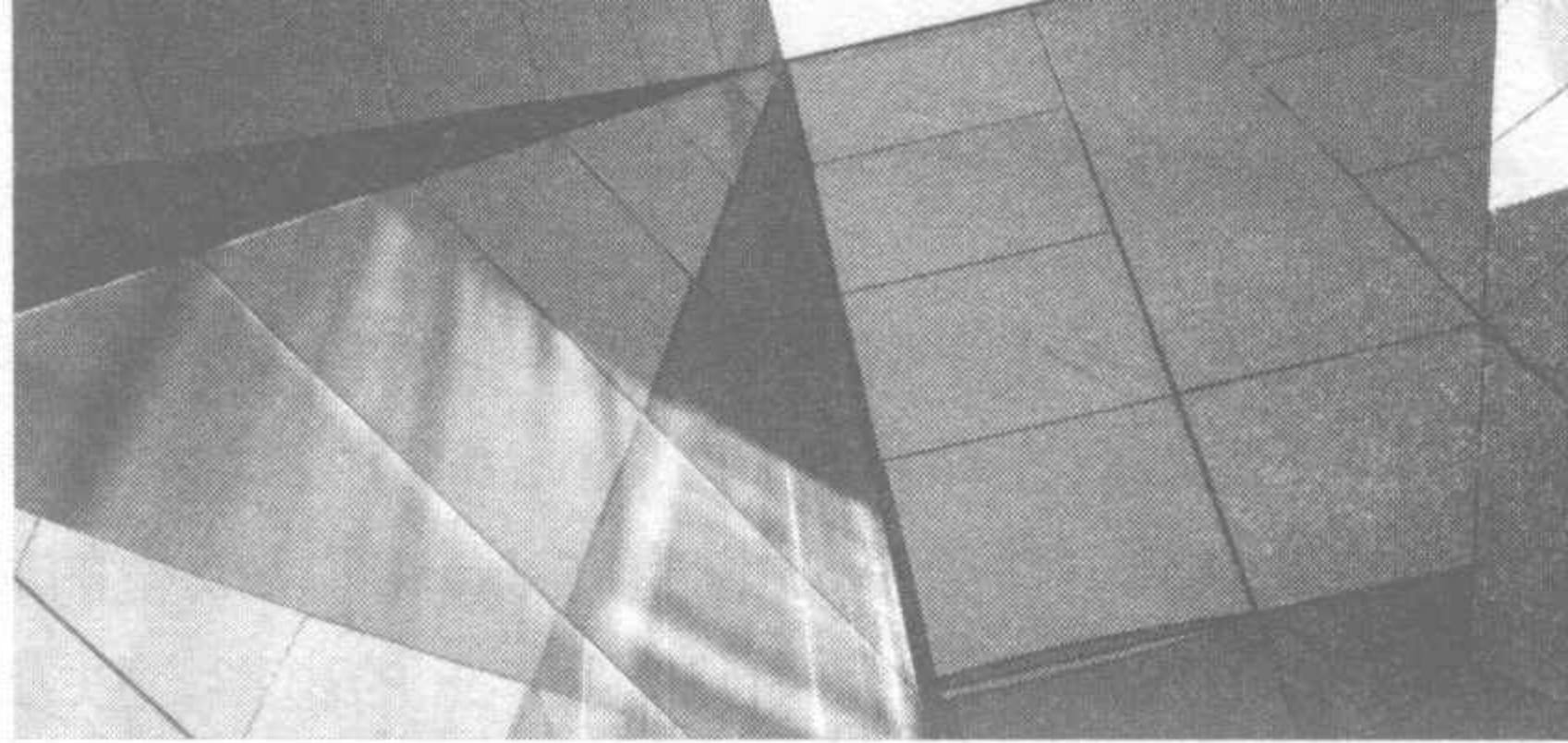
16.5.1 事件..... 365

16.5.2 事件源..... 365

16.5.3 事件监听器..... 366

16.5.4 事件类和监听器接口..... 366

16.6	使用 JButton	366	17.6.1	事件基础	389
16.7	使用 JTextField	369	17.6.2	按钮控件简介	390
16.8	使用 JCheckBox	371	17.6.3	演示事件处理和按钮	390
16.9	使用 JList	374	17.7	其他 3 个 JavaFX 控件	392
16.10	使用匿名内部类或 lambda 表达式 来处理事件	380	17.7.1	CheckBox	392
16.11	自测题	381	17.7.2	ListView	396
第 17 章	JavaFX 简介	383	17.7.3	TextField	399
17.1	JavaFX 的基本概念	383	17.8	效果和变换简介	401
17.1.1	JavaFX 包	384	17.8.1	效果	401
17.1.2	Stage 和 Scene 类	384	17.8.2	变换	403
17.1.3	节点和场景图	384	17.8.3	演示效果和变换	403
17.1.4	布局	384	17.9	进一步学习	406
17.1.5	Application 类和生命周期方法	384	17.10	自测题	406
17.1.6	启动 JavaFX 应用程序	385	附录 A	自测题答案	409
17.2	JavaFX 应用程序的骨架	385	附录 B	使用 Java 的文档注释	439
17.3	编译和运行 JavaFX 程序	387	附录 C	Java Web Start 概述	445
17.4	应用程序线程	387	附录 D	JShell 简介	451
17.5	使用简单的 JavaFX 控件 Label	387	附录 E	更多 Java 关键字	459
17.6	使用按钮和事件	389			



第1章 Java 基础

关键技能与概念

- 了解 Java 的历史和基本原理
- 理解 Java 对 Internet 的贡献
- 理解字节码的重要性
- 了解 Java 的术语
- 理解面向对象程序设计的基本原理
- 创建、编译和运行一个简单的 Java 程序
- 使用变量
- 使用 if 和 for 控制语句
- 创建代码块
- 理解如何定位、缩进和终止语句
- 了解 Java 关键字
- 理解 Java 标识符的规则

在计算领域，很少有技术具有 Java 这样的影响力。它在 Web 的早期创立，帮助构造了 Internet 的现代形式，包括客户端和服务端。它的创新功能改进了编程的艺术和科学，在计算机语言设计方面设立了新标准。围绕着 Java 的前向思考文化确保它一直充满活力。在计算界经常进行快速、频繁的变化，简言之，Java 不仅是世界上最重要的计算机语言，也是改革编程的一种力量，在此过程中也改变了世界。

尽管 Java 是常与 Internet 编程相关的语言，但它并不限于此。Java 是一门强大、功能全面、通用的编程语言。因此对于编程新手，Java 是一种绝佳的学习语言。今天，要成为职业程序员就意味着要具备使用 Java 编程的能力，它就是这么重要。在本书的课程中，你将学习必备的 Java 技能。

本章旨在向你介绍 Java，包括它的历史、设计原理和一些最重要的特性。目前，学习程序设计语言最大的难点是语言的各部分之间不是相互孤立的，而是相互关联的。这种相互关联性在 Java 中尤为突出。事实上，只讨论 Java 的一个方面，而不涉及其他部分是非常困难的。为了帮助克服这一困难，本章对 Java 的几个特性进行了简单概述，其中包括 Java 程序的基本形式、一些基本的控制结构和运算符。对于这些内容我们并不进行深入讨论，而只是关注一下 Java 程序共有的一些概念。

1.1 Java 的起源

Java 是 1991 年由 Sun Microsystems 公司的 James Gosling、Patrick Naughton、Chris Warth、Ed Frank 和 Mike Sheridan 共同构想成果。这门语言最初名为 Oak，于 1995 年更名为 Java。多少有些让人吃惊的是，设计 Java 的最初动力并不是源于 Internet，而是为了开发一种独立于平台的语言，使其能够用于创建内嵌于不同家电设备(如烤箱、微波炉和遥控器)的软件。你可能已猜到，不同类型的 CPU 都可以用作控制器。麻烦在于当时多数的计算机语言都是针对特定的目标而设计的，例如 C++。

虽然任何类型的 CPU 或许都能编译 C++ 程序，然而这需要 CPU 有完整的 C++ 编译器。而开发编译器的成本很高，并且很耗时。为了找到更好的解决方法，Gosling 和其他人尝试开发一种可移植的跨平台语言，使该语言生成