

国外经典程序设计系列



Java 编程规范 (第三版)

The Java Language Specification

(Third Edition)

James Gosling

Bill Joy

[美]

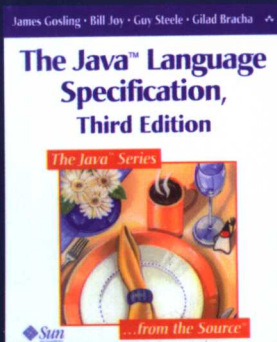
Guy Steele

著

Gilad Bracha

陈宗斌 沈金河

译



Java 之父的全新力作
Java 语言的标准参考书



中国电力出版社

www.infopower.com.cn

国外经典程序设计系列

Java 编程规范 (第三版)

The Java Language Specification

(Third Edition)

James Gosling

Bill Joy

[美]

Guy Steele

著

Gilad Bracha

陈宗斌 沈金河

译



中国电力出版社

www.infopower.com.cn

Java Language Specification, 3rd Edition (ISBN 0-321-24678-0)

James Gosling, Bill Joy, Guy Steele, Gilad Bracha

Copyright ©1996-2005 Sun Microsystems, Inc.

Original English Language Edition Published by Addison-Wesley.

All rights reserved.

Translation edition published by PEARSON EDUCATION ASIA LTD and CHINA ELECTRIC
POWER PRESS, Copyright © 2006.

本书翻译版由 Pearson Education 授权中国电力出版社独家出版、发行。
未经出版者书面许可, 不得以任何方式复制或抄袭本书的任何部分。

本书封面贴有 Pearson Education 防伪标签, 无标签者不得销售。
北京市版权局著作权合同登记号 图字: 01-2006-3388 号

图书在版编目 (CIP) 数据

Java 编程规范 (第三版) / (美) 高斯林 (Gosling, J.) 等编著; 陈宗斌, 沈金河译.

—北京: 中国电力出版社, 2006

(国际经典程序设计系列)

书名原文: Java Language Specification, 3rd Edition

ISBN 7-5083-4292-5

I.J... II.①高...②陈...③沈... III.JAVA 语言—程序设计 IV.TP312

中国版本图书馆 CIP 数据核字 (2006) 第 060159 号

丛 书 名: 国际经典程序设计系列

书 名: Java 编程规范 (第三版)

编 著: (美) James Gosling, Bill Joy, Guy Steele, Gilad Bracha

翻 译: 陈宗斌 沈金河

责任编辑: 牛贵华

出版发行: 中国电力出版社

地址: 北京市三里河路 6 号 邮政编码: 100044

电话: (010) 88515918 传 真: (010) 88518169

印 刷: 北京市铁成印刷厂

开本尺寸: 185×233 印 张: 29.25 字 数: 665 千字

书 号: ISBN 7-5083-4292-5

版 次: 2006 年 7 月北京第 1 版 2006 年 7 月第 1 次印刷

定 价: 49.80 元

版权所有 翻印必究

“当我说话时，”矮胖子轻蔑地说，“我爱那个字是什么意思，它便是什么意思，丝毫不差。”

爱丽丝抗议道：“问题是，你能随心所欲地为一个词下定义吗？”

矮胖子回答得倒也干脆，他说：“问题是，这只是谁说了算而已。”

——Lewis Carroll, 《镜中世界》

前 言

Java™编程语言最初被称为 Oak，由 James Gosling 设计，旨在用于嵌入式消费类电子产品应用程序上。使用几年之后，加上 Ed Frank、Patrick Naughton、Jonathan Payne 和 Chris Warth 的巨大贡献，该语言被重新定位为以 Internet 为目标，经过重新命名，从实质上加以修改，就形成了本书要讨论的 Java 语言。Java 语言的最终形式由 James Gosling、Bill Joy、Guy Steele、Richard Tuck、Frank Yellin 和 Arthur van Hoff 确定下来，在此过程中得到了 Graham Hamilton、Tim Lindholm 以及许多朋友及同事的大力帮助。

Java 编程语言是一种多用途的、并发的、基于类的、面向对象的编程语言，且特别设计为具有尽可能少的实现依赖性。它允许应用程序开发人员编写一次程序，然后可以在 Internet 上到处运行。

本书试图详细说明 Java 语言的语法和语义规范。我们试图在本书中说明每一种语言结构的行为，以便让所有实现都接受同一个程序。除了时间相关性或无法确定因素之外，如果有足够的时间和足够的内存空间，用 Java 编程语言所写的程序应该可以在所有机器和所有实现中计算出相同的结果。

我们相信 Java 编程语言是一种将被广泛使用的成熟语言。尽管如此，我们仍期望未来几年内该语言会有所进步。我们试图让它以完全与现有应用程序兼容的方式来管理这些演变。为了达到此目的，我们试图写一些相对新的语言版本。编译器和系统能够同时支持几个具有完全兼容性的版本。

Java 平台的许多研究和实验都在进行中。我们鼓励进行此项工作，并且将继续与外部组织协作开发来改进语言和平台。例如，我们已经收到几个有意义的参数化类型的提议。在技术上较难的领域中，尤其是最新的技术上，这种研究合作是很有必要的。

我们诚挚地感谢通过优秀的反馈、协助和鼓励对本书作出贡献的人。

尤其要感谢的是：Tom Cargill、Peter Deutsch、Paul Hilfinger、Masayuki Ida、David Moon、Steven Muchnick、Charles L. Perkins、Chris Van Wyk、Steve Vinoski、Philip Wadler、Daniel Weinreb 和 Kenneth Zadeck，他们完整、细心和缜密地对本书的草稿进行了审阅。我们由衷地感谢他们非常自愿的努力。

我们还要感谢 Stephen Adams、Bowen Alpern、Glenn Ammons、Leonid Arbuzov、Kim Bruce、Edwin Chan、David Chase、Pavel Curtis、Drew Dean、William Dietz、David Dill、Patrick Dussud、Ed Felten、John Giannandrea、John Gilmore、Charles Gust、Warren Harris、Lee Hasiuk、Mike Hendrickson、Mark Hill、Urs Hoelzle、Roger Hoover、Susan Flynn Hummel、

Christopher Jang、Mick Jordan、Mukesh Kacker、Peter Kessler、James Larus、Derek Lieber、Bill McKeeman、Steve Naroff、Evi Nemeth、Robert O'Callahan、Dave Papay、Craig Partridge、Scott Pfeffer、Eric Raymond、Jim Roskind、Jim Russell、William Scherlis、Edith Schonberg、Anthony Scian、Matthew Self、Janice Shepherd、Kathy Stark、Barbara Steele、Rob Strom、William Waite、Greg Weeks 和 Bob Wilson 对本书的审阅、提问、评论和建议（此人员名单是从我们的 E-mail 记录自动生成的。如果遗漏了任何人，我们对此深表歉意）。

在改进语言的定义和本书的演示文稿形式上，所有这些审阅者的反馈对我们来说都是非常宝贵的。我们感谢他们的辛勤劳动。本书中任何残留的错误——我们希望这会很少——都是我们而非他们的责任。

我们感谢 Francesca Freedman 和 Doug Kramer 对印刷和出版事务的协助。我们感谢 Adobe Systems Incorporated 的 Dan Mills 对寻找可能的字体提供的协助。

Sun 公司的许多同事在很多方面帮助我们。本系列丛书的编辑 Lisa Friendly 协调着我们与 Addison-Wesley 的关系。Susan Stambaugh 帮助管理着给审阅者的几百份草稿副本。我们收到了 Ben Adida、Ole Agesen、Ken Arnold、Rick Cattell、Asmus Freytag、Norm Hardy、Steve Heller、David Hough、Doug Kramer、Nancy Lee、Marianne Mueller、Akira Tanaka、Greg Tarsy、David Ungar、Jim Waldo、Ann Wollrath、Geoff Wyant 和 Derek White 有价值的帮助和技术上的建议。我们感谢 Alan Baratz、David Bowen、Mike Clary、John Doerr、Jon Kannegaard、Eric Schmidt、Bob Sproull、Bert Sutherland 和 Scott McNealy 的领导和鼓励。

在研究和验证遍布本书的引文的过程中，Columbia 大学的在线 Bartleby Library（在线网址为 <http://www.cc.columbia.edu/acis/bartleby/>）对我们来说非常宝贵。下面是一个例子：

他们以其他著作的丰富经验润色他们贫乏的书籍。

——Robert Burton（1576~1640）

我们非常感激在 Project Bartleby 工作的人，节省了我们大量的工作，并且重新激起了我们对沃尔特·惠特曼作品的欣赏之情。

感谢在写这本书时我们在整理工作中所用到的工具和服务：电话、旭夜快递、台式工作站、激光打印机、复印机、排版软件、字体库、电子邮件、WWW，当然还有 Internet。我们住在三个不同的州，但是仍能不费吹灰之力与其他人以及审阅者合作。荣誉给予那些常年努力不懈的、让这些优秀的工具和服务快速且可靠工作的成千上万的人们。

感谢 Addison-Wesley 的 Mike Hendrickson、Katie Duffy、Simone Payment 和 Rosa Aimée González 在本书付梓期间所给予的帮助、鼓励和耐心。我们还要感谢这些文字编辑。

感谢 Rosemary Simpson 在非常繁忙的日常工作中为本书制作了索引。

最后，我们非常感谢我们的家人和朋友在最近疯狂工作的一年里给予了我们爱和支持。

Brian Kernighan 和 Dennis Ritchie 在他们的《The C Programming Language》中写道，他们觉得 C 语言“随着经验的增长越来越好”。如果你喜欢 C，我们想你也喜欢 Java 编程

语言。我们希望它对你也一样好用。

James Gosling

加利福尼亚州，库珀蒂诺

Bill Joy

科罗拉多州，阿斯彭

Guy Steele

马萨诸塞州，切姆斯福德

1996 年 7 月

第二版前言

……金字塔已经岿然屹立一千年；而生物则要么进化，要么毁灭。
——Alan Perlis, 《Structure and Interpretation of Computer Programs》一书的前言

过去几年里, Java™编程语言获得了空前的成功。这种成功带来了挑战:随着用户的增长,对于语言和库的需求也出现了爆炸性的增长。为了应对这个挑战,语言得到了发展(幸运的是,不是爆炸性的),库也是如此。

本书第二版反映出了这些发展。它整合了 Java 编程语言自 1996 年本书第一版出版后所发生的改变。1997 年,这些大量的改变被写入了 Java 平台的 1.1 版本中,并考虑增加嵌入的类型声明。稍后的修改是浮点操作。另外,这一版本收编了包含方法搜索和二进制兼容性的重要说明和修正。

本书定义了目前现有的语言。Java 编程语言仍会继续发展。在写这本书时,就有一些正在进行的计划,通过 Java Community Process 利用泛型类型和断言扩展语言、优化内存模型,等等。但是,如果等到这些工作都有了结果之后再出版本书的第二版,肯定是不合适的。

现在 API 库的规范已经太大了,不适合纳入本书的范围,而且它们还在不断发展。因此,API 规范已从本书中删除。库规范可以在 <http://java.sun.com> 网站上找到;该规范目前仅仅专注于 Java 编程语言。

读者可以将对本书的意见发送至邮箱: jls@java.sun.com。要学习最新的 Java 2 平台或是下载最新的 Java 2 SDK 版本,请访问 <http://java.sun.com>。关于 Java 系列图书的更新信息,包括本书的勘误以及即将出版的书籍,请访问: <http://java.sun.com/Series>。

许多人对本书直接或间接地做出了贡献。Tim Lindholm 作为技术编辑贡献非常大,尤其是在浮点数问题上。本书如果没有他的话,将会缺少色彩。Lisa Friendly 作为本系列丛书的编辑,提供了很多的鼓励和建议,对此我非常感谢。

David Bowen 是第一个建议我关心 Java 平台规范的人。我很感谢他引导我进入这个丰富的领域。

感谢 Java 编程语言中的嵌套类型之父 John Rose,在我试图详细准确地解释嵌套类型时,他给予了我无限的亲切和支持。

很多人对第二版提供了很有价值的意见。特别感谢 Ergnosis 的 Roly Perera 以及 Leonid

Arbouzov 和他在 Novosibirsk 的 Sun Java 平台一致性小组的同事: Konstantin Bobrovsky、Natalia Golovleva、Vladimir Ivanov、Alexei Kaigorodov、Serguei Katkov、Dmitri Khukhro、Eugene Latkin、Ilya Neverov、Pavel Ozhdikhin、Igor Pyankov、Viatcheslav Rybalov、Serguei Samoilidi、Maxim Sokolnikov 和 Vitaly Tchaiko。他们完整地阅读了早期的草稿,对大大提高本书的准确性有很大的帮助。

非常感激 Martin Odersky、Andrew Bennett 以及 Sun javac 编译器小组过去和现在的成员: Iris Garcia、Bill Maddox、David Stoutamire 和 Todd Turnidge。他们非常努力地确保参考实现符合本规范。还有许多有意义的技术交流,我感谢他们和我在 Sun 公司的其他同事: Lars Bak、Joshua Bloch、Cliff Click、Robert Field、Mohammad Gharahgouzloo、Ben Gomes、Steffen Grarup、Robert Griesemer、Graham Hamilton、Gordon Hirsch、Peter Kessler、Sheng Liang、James McIlree、Philip Milne、Srdjan Mitrovic、Anand Palaniswamy、Mike Paleczny、Mark Reinhold、Kenneth Russell、Rene Schmidt、David Ungar、Chris Vick 和 Hong Zhang。我的经理 Tricia Jordan 是耐心、体量和理解下属的典范。还要感谢 Java 2 标准版的主管 Larry Abrahams 对这个工作的支持。

下列这些人都提供了对本书很有用的意见: Godmar Bak、Hans Boehm、Philippe Charles、David Chase、Joe Darcy、Jim des Rivieres、Sophia Drossopoulou、Susan Eisenbach、Paul Haahr、Urs Hoelzle、Bart Jacobs、Kent Johnson、Mark Lillibridge、Norbert Lindenberg、Phillipe Mulet、Kelly O'Hair、Bill Pugh、Cameron Purdy、Anthony Scian、Janice Shepherd、David Shields、John Spicer、Lee Worall 和 David Wragg。

Suzette Pelouch 和 Doug Kramer、Atul Dambalkar 一起,利用他们的 FrameMaker 专业知识,为本书索引提供了非常宝贵的帮助。Addison-Wesley 的 Mike Hendrickson 和 Julie Dinicola 相当友好,他们对本书的出版给予了无限的帮助。

就我个人来说,我要感谢我的妻子 Weihong 对我的爱和支持。

最后,衷心感谢我的合作者 James Gosling、Bill Joy 和 Guy Steele 邀请我参加这项工作。这是件相当愉快且荣耀的事情。

Gilad Bracha
加利福尼亚州, 洛斯阿尔托斯
2000 年 4 月

这是为女性准备的辞典。
男性辞典大致相同,但不完全相同。注意,这一段是最为关键的不同。

谨慎选择吧!

——Milorad Pavic,《哈扎尔辞典》女性版

第三版前言

本书的这个版本代表该语言的历史中一组最大的变革。该语言近来把泛型、注释、断言、自动装箱、拆箱、枚举类型、foreach 循环、可变元数方法和静态导入等技术都添加到了其中。除了断言之外，其他所有的技术都是在 2004 年秋季的 5.0 版本上新增的。

本书的第三版反映了这些发展。它集成了自 2000 年第二版发布以来对 Java 编程语言所做的所有变革。

该语言在过去 4 年里获得了长足的发展。不幸的是，浓缩一种商业上成功的编程语言的精华是不现实的——而只会使它增长得越来越大。在兼容性约束下和广泛的使用和用户冲突要求下，管理这种增长是一个很大的挑战。我只能希望借助本规范成功地满足这项挑战；时间会告诉我们一切。

读者可以把对本规范的评论发送至：jls@java.sun.com。要了解关于 Java 平台的最新信息，或者下载最新的 J2SE 版本，请访问 <http://java.sun.com>。可以在 <http://java.sun.com/Series> 上找到关于 Java 系列（包括本书第三版的勘误表以及即将出版的书籍）的更新信息。

本规范建立于许多人的努力之上，包括 Sun 公司内部和外部的人员。

最至关重要的贡献是由那些把规范变成真实软件的人做出的。这些人中首屈一指的是 javac 的维护者，javac 是 Java 编程语言的参考编译器。

Neal Gafter 是在集成和产品化这里描述的重大变革这个重要时期的“javac 先生”。Neal 的贡献和生产力可以被公正地描述为如史诗般的宏伟。如果没有他，我们简直不能完成这项任务。此外，他的见解和技能对设计所有的新语言特性做出了巨大的贡献。对该语言的这个版本来说，没有哪个人应该得到比他更多的赞誉——但是，对于其缺点的任何指责都应该落到我本人以及许多 JSR 专家组的成员身上！

Neal 着手研究了新的挑战，Peter von der Ahé 接替和继承了他的工作，他继续改进和加强了实现。在 Neal 参与进来之前，在前一版本完成时，Bill Maddox 负责 javac，并且他通过他们早期的工作发掘了诸如泛型和断言之类的特性。

另一个值得单独提及的人是 Joshua Bloch。Josh 参与了无休止的语言设计讨论，在多个专家组担任主席，并且为 Java 平台做出了关键的贡献。公平地讲，Josh 和 Neal 对本书的关心超过了我本人！

本规范的许多部分都是由多个不同的专家组在 Java 社区组织的框架中开发的。

最普及的语言变革集合是 JSR-014 的成果：添加泛型到 Java 编程语言中。JSR-04 专家

组的成员有：Norman Cohen、Christian Kemper、Martin Odersky、Kresten Krab Thorup、Philip Wadler 和我本人。在早期阶段，成员还包括 Sven-Eric Panitz 和 Steve Marx。所有这些人都值得对他们的参与表示谢意。

JSR-014 代表一项空前的努力，在非常严格的兼容性要求之下，从根本上扩展广泛使用的编程语言的类型系统。经过一段漫长且费力的设计和实现过程之后，我们到达了当前的语言扩展。远在 JSR 发起泛型研究以前，Martin Odersky 和 Philip Wadler 就已经创建了一种称为 Pizza 的实验语言，用于探索涉及的思想。1998 年春天，David Stoutamire 和我本人基于这些思想开始与 Martin 和 Phil 合作，这就产生了 GJ。当 JSR-014 专家组把各个成员召集在一起时，就把 GJ 选作扩展 Java 编程语言的基础。Martin Odersky 实现了 GJ 编译器，并且他的实现变成了 javac 的基础（javac 开始于 JDK 1.3，尽管直到 JDK 1.5 才启用泛型）。

泛型类型系统的核心的理论基础在很大程度上归功于 Martin Odersky 和 Phil Wadler 的专业知识。后来，利用通配符扩展了该系统。这些基于 Atsushi Igarashi 和 Mirko Viroli 的工作，它本身构建于 Kresten Thorup 和 Mads Torgersen 的早期工作之上。通配符最初是作为 Sun 和 Aarhus 大学之间合作的一部分而设计和实现的。Neal Gafter 和我本人代表 Sun，连同 Erik Ernst 和 Mads Torgersen 以及代表 Aarhus 的 Peter von der Ahé 和 Christian Plesner-Hansen 一起参与了这次合作。感谢 Ole Lehrmann-Madsen 鼓励和支持那项工作。

Joe Darcy 和 Ken Russell 实现了对泛型反射的大部分特定支持。Neal Gafter、Josh Bloch 和 Mark Reinhold 做了大量的工作来泛型化 JDK 库。

有些人必须得到表扬，他们对泛型设计的评论导致了显著的差别。Alan Jeffrey 通过指出原始类型系统中的细微缺陷，从而对 JSR-14 做出了至关重要的贡献。Bob Deen 建议为低有界通配符使用“? super T”语法。

JSR-201 包括一系列变革：自动装箱、枚举、foreach 循环、可变元数方法和静态导入。JSR-201 专家组的成员包括：Cédric Beust、David Biesack、Joshua Bloch（联合主席）、Corky Cartwright、Jim des Rivieres、David Flanagan、Christian Kemper、Doug Lea、Changshin Lee、Tim Peierls、Michel Trudeau 和我本人（联合主席）。枚举和 foreach 循环主要是由 Josh Bloch 和 Neal Gafter 设计的。如果没有 Neal 专门的设计工作，可变元数方法将永远不会加入到该语言中（这里没有提及实现它们的一些小问题）。

Josh Bloch 自己勇敢地承担了 JSR-175 的职责，即把注释添加到该语言中。JSR-175 专家组的成员包括：Cédric Beust、Joshua Bloch（主席）、Ted Farrell、Mike French、Gregor Kiczales、Doug Lea、Deeptendu Majunder、Simon Nash、Ted Neward、Roly Perera、Manfred Schneider、Blake Stone 和 Josh Street。Neal Gafter 作为主要的贡献者，也照例工作在最前沿。

本版本的另一个变革是完全修订了 Java 内存模型，它是由 JSR-133 承担的。JSR-133 专家组的成员包括：Hans Boehm、Doug Lea、Tim Lindholm（联合主席）、Bill Pugh（联合主席）、Martin Trotter 和 Jerry Schwarz。内存模型的主要技术作家是 Sarita Adve、Jeremy Manson 和 Bill Pugh。本书中关于 Java 内存模型的章节事实上几乎全都是他们的工作，只是做了少许编辑上的修订。Joseph Bowbeer、David Holmes、Victor Luchangco 和 Jan-Willem

Maessen 也做出了重大贡献。第 12 章中论及终结 (finalization) 的关键几节内容在很大程度上也应归功于这项工作，特别是 Doug Lea 的工作。

许多人对本版本提供了有价值的评论。

我希望向 Archibald Putt 表达我的谢意，他对本书提供了见解和鼓励。他的著作总是鼓舞人心。再一次感谢 Joe Darcy，他引荐了我们，提出了许多有用的意见，并且在数值问题和十六进制值的设计方面做出了他的独特贡献。

Sun 的许多同事 (过去的和现在的) 提供了有用的反馈和讨论，并且以无数种方式帮助了本书的出版，他们是：Andrew Bennett、Martin Buchholz、Jerry Driscoll、Robert Field、Jonathan Gibbons、Graham Hamilton、Mimi Hills、Jim Holmlund、Janet Koenig、Jeff Norton、Scott Seligman、Wei Tao 和 David Ungar。

特别感谢我的经理 Laurie Tolson，感谢她在我编写这些规范的漫长过程中所提供的支持。

对本规范提供了有价值意见的还有：Scott Annanian、Martin Bravenboer、Bruce Chapman、Lawrence Gonsalves、Tim Hanson、David Holmes、Angelika Langer、Pat Lavarre、Phillipe Mulet 和 Cal Varnson。

Addison-Wesley 的 Ann Sellers、Greg Doench 和 John Fuller 极其耐心地保证了本书的问世，而不计较本书多次错过了截稿日期。

与以往一样，我感谢我的妻子 Weihong 和我的儿子 Teva 的支持和协作。

Gilad Bracha

加利福尼亚州，洛斯阿尔托斯

2005 年 1 月

目 录

前言	
第二版前言	
第三版前言	
第 1 章 简介	1
1.1 示例程序	4
1.2 符号	4
1.3 预定义类和接口的关系	4
1.4 参考文献	5
第 2 章 语法	7
2.1 与环境无关的语法	7
2.2 词法语法	7
2.3 语义语法	7
2.4 语法符号	8
第 3 章 词法结构	11
3.1 Unicode	11
3.2 词法转换	12
3.3 Unicode 转义符	12
3.4 行终止符	13
3.5 输入元素和标记	14
3.6 空白	15
3.7 注释	15
3.8 标识符	16
3.9 关键字	17
3.10 字面值	18
3.11 分隔符	26
3.12 运算符	26
第 4 章 类型、值和变量	27
4.1 各种类型和值	28

4.2	基本类型和值	28
4.3	引用类型和值	35
4.4	类型变量	39
4.5	参数化类型	41
4.6	类型擦除	45
4.7	可具体化的类型	45
4.8	原生类型	46
4.9	交集类型	49
4.10	子类型化	49
4.11	在何处使用类型	51
4.12	变量	53
第 5 章	转换和提升	60
5.1	转换的种类	62
5.2	赋值转换	71
5.3	方法调用转换	76
5.4	字符串转换	77
5.5	强制转换	77
5.6	数值提升	82
第 6 章	名称	85
6.1	声明	86
6.2	名称和标识符	86
6.3	声明的作用域	88
6.4	成员和继承	92
6.5	确定名称的含义	95
6.6	访问控制	104
6.7	完全限定的名称和规范名称	109
6.8	命名约定	110
第 7 章	包	116
7.1	包成员	116
7.2	包的主机支持	117
7.3	编译单元	119
7.4	包声明	119
7.5	导入声明	121
7.6	顶级类型声明	126
7.7	惟一的包名称	128

第 8 章 类	130
8.1 类声明	131
8.2 类成员	143
8.3 字段声明	147
8.4 方法声明	159
8.5 成员类型声明	180
8.6 实例初始化语句	181
8.7 静态初始化语句	181
8.8 构造函数声明	182
8.9 枚举	189
第 9 章 接口	197
9.1 接口声明	197
9.2 接口成员	200
9.3 字段（常量）声明	201
9.4 抽象方法声明	203
9.5 成员类型声明	205
9.6 注释类型	206
9.7 注释	213
第 10 章 数组	219
10.1 数组类型	219
10.2 数组变量	220
10.3 数组创建	221
10.4 数组访问	221
10.5 数组：一个简单的示例	221
10.6 数组初始化语句	222
10.7 数组成员	223
10.8 数组的 Class 对象	224
10.9 字符的数组不是一个 String	224
10.10 数组存储异常	224
第 11 章 异常	226
11.1 异常的起因	227
11.2 异常的编译时检查	227
11.3 异常处理	229
11.4 异常的示例	231
11.5 异常层次结构	232

第 12 章 执行	234
12.1 虚拟机启动.....	234
12.2 加载类和接口	236
12.3 链接类和接口	237
12.4 初始化类和接口	239
12.5 创建新的类实例	243
12.6 类实例的终结	246
12.7 卸载类和接口	249
12.8 程序退出	250
第 13 章 二进制兼容性	251
13.1 二进制的形式	252
13.2 二进制兼容性是什么，不是什么	255
13.3 包的演变	255
13.4 类的演变	256
13.5 接口的演变.....	269
第 14 章 块和语句	271
14.1 语句的正常结束和突然结束.....	271
14.2 块	272
14.3 本地类声明.....	273
14.4 局部变量声明语句	274
14.5 语句.....	278
14.6 空语句	279
14.7 标签语句	280
14.8 表达式语句.....	280
14.9 if 语句	281
14.10 assert 语句	282
14.11 switch 语句	285
14.12 while 语句.....	288
14.13 do 语句	289
14.14 for 语句	291
14.15 break 语句.....	294
14.16 continue 语句	296
14.17 return 语句	297
14.18 throw 语句	298
14.19 synchronized 语句	299
14.20 try 语句.....	300

14.21 不可到达语句	305
第 15 章 表达式	309
15.1 计算、表示和结果	309
15.2 变量作为值	310
15.3 表达式的类型	310
15.4 精确浮点数表达式	310
15.5 表达式和运行时检查	311
15.6 计算的正常和突然结束	312
15.7 求值顺序	313
15.8 主表达式	317
15.9 类实例创建表达式	320
15.10 数组创建表达式	325
15.11 字段访问表达式	329
15.12 内存调用表达式	332
15.13 数组访问表达式	362
15.14 后缀表达式	365
15.15 一元运算符	366
15.16 强制转换表达式	369
15.17 乘法运算符	370
15.18 加运算符	373
15.19 移位运算符	377
15.20 关系运算符	378
15.21 相等运算符	380
15.22 位和逻辑运算符	382
15.23 条件与运算符&&	383
15.24 条件或运算符 	383
15.25 条件运算符?:	384
15.26 赋值运算符	385
15.27 表达式	396
15.28 常量表达式	396
第 16 章 明确赋值	398
16.1 明确赋值和表达式	402
16.2 语明确赋值和语句	406
16.3 明确赋值和参数	412
16.4 明确赋值和数组初始化方法	412
16.5 明确赋值和枚举常量	412