

Textbook for
Programming in Java
(Second Edition)

Java程序设计
(第2版)

雍俊海 编著



清华大学出版社

Textbook for
Programming in Java
(Second Edition)

Java程序设计

(第2版)

雍俊海 编著

清华大学出版社

内 容 简 介

本书是“十二五”普通高等教育本科国家级规划教材《Java程序设计教程（第3版）》的简版，讲解Java程序设计的基本知识及其基本编程方法，包括Java语言的基础语法、结构化程序设计、面向对象程序设计、数组、向量、字符串、哈希表、泛型、枚举、异常处理、文件与数据流、图形用户界面程序设计、小应用程序和编程规范等。本书内容的安排和取舍以人们的学习与认知过程为基础，与社会的实际需求相匹配，每章都附有习题，而且在附录中包含图、表、例程以及类和接口的页码索引，在正文中采用特殊字体突出中心词，使读者轻松且愉悦地了解 and 掌握Java程序设计的知识和方法，并应用到实践中去。本书内容丰富、语言简练，而且提供了丰富的例程，既可以作为计算机专业和非计算机专业的基础教材以及Sun公司的SCJP（Java程序员认证）考试的辅导教材，也可以作为需要使用Java语言的工程人员和科技工作者的自学参考书。

本书封面贴有清华大学出版社防伪标签，无标签者不得销售。

版权所有，侵权必究。侵权举报电话：010-62782989 13501256678 13801310933

图书在版编目（CIP）数据

Java 程序设计 / 雍俊海编著. —2 版. —北京：清华大学出版社，2014

ISBN 978-7-302-36052-0

I. ①J… II. ①雍… III. ①JAVA 语言—程序设计 IV. ①TP312

中国版本图书馆 CIP 数据核字（2014）第 065920 号

责任编辑：魏江江 王冰飞

封面设计：傅瑞学

责任校对：李建庄

责任印制：王静怡

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 刷 者：三河市君旺印务有限公司

装 订 者：三河市新茂装订有限公司

经 销：全国新华书店

开 本：185mm×260mm 印 张：21.75 字 数：540 千字

版 次：2008 年 2 月第 1 版 2014 年 8 月第 2 版 印 次：2014 年 8 月第 1 次印刷

印 数：1~2000

定 价：39.50 元

产品编号：056515-01

前言

现代科学技术正在迅猛地发展,计算机信息技术在其中发挥着巨大的作用。现在,计算机技术已经渗透到各行各业,并推动着这些行业迅速发展,如何尽快地掌握计算机知识,学好一门计算机语言,已经成为人们面临的比较普遍的基本问题,本书就是在这种背景下编写的。

首先,Java 语言本身是一种可以满足这种需求的计算机语言。它比 C++ 计算机语言简单,去掉了 C++ 语言中一些不易理解或容易出错的概念和语法。此外,因为 Java 是一种较新的计算机语言,所以它在面向对象和多线程特性上比其他现有的计算机语言显得更纯粹一些。同时,Java 语言在网络、平台无关性和安全性方面的优点也比大部分计算机语言(如 C++ 语言)更为突出。这样,学习 Java 程序设计,应用 Java 语言实现算法也比较容易,从而节省了编程时间。同时,编写出来的 Java 代码比较容易得到复用和移植。

最初的 Java 程序设计教材是应选修我主讲的“Java 程序设计”课程的同学的要求而编写的。在 2003 至 2004 年期间,共有三百多名清华大学本科生选修该课程,其中很多同学通过清华大学的教学评估系统以及给我发 E-mail 等方式强烈要求我编写一本 Java 程序设计教材。为此,我编写了本教材的第 1 版(《Java 程序设计》雍俊海,北京:清华大学出版社,2004)。

我希望本教材能够给读者带来尽可能多的益处。对于学习而言,最重要的应当是对学习方法的引导,学习每门课程都有其内在的学习规律,顺应其规律,采用正确的学习方法一般会产生良好的学习效果。对于有些初学者而言,在最开始学习的时候,要把握学习规律常常有难度;要按学习规律进行学习,常常会有很多来自自身的阻力。如果能够克服上述不利因素并加以坚持,相信会有事半功倍的效果。我希望本教材能够为初学者适应这些学习规律添加一些辅助的约束力,从而帮助初学者克服来自自身的阻力。

学习首先应当是“学以致用”,为此,我常常利用各种机会调研软件公司对 Java 程序设计的实际需求。如果能够从实际应用出发进行学习,那么应该会提高学习的效率。另外,学习过程的关键应当是实践,教材是实践的一种辅助工具,为此,本教材比较详细地讲解了 Java 语言编程环境的建

立过程,希望读者在开始学习Java语言的时候能够建立起Java语言编程环境。在教材每章的后面都有习题,对教材的习题,本书都没有提供答案,真诚地希望这些习题能够给读者增加一些自主性思考和实践练习的机会。自主性思考意味着应该通过自己的思考去理解Java语言并求解问题,而且不要拘泥于某一种答案,即可以采用多种不同的方法求解相同的问题。这似乎会增加学习时间和学习难度,但实际上一般会迅速提高学习的效率。在刚开始的时候,有些读者可能会不太习惯,但是如果坚持,那么实际上会迅速降低后续学习的难度,而且会使对Java语言的掌握更加牢固。另外,希望读者能够理解习题编写的初衷,即它的主要目的是加强具有自主性思考的实践,而不是习题答案本身。希望读者能够经常总结实践过程的收获,享受其中的成就感,即使无法最终求解问题。因为教材提供的习题有些少,所以我还整理了一本习题集——《Java程序设计习题集(含参考答案)》(雍俊海,北京:清华大学出版社,2006)。这本习题集对判断正误题、填空题和选择题基本上都给出了答案,但只给出了少量编程题的答案,这本习题集是本教材的一个有益的补充。如果读者需要阅读编程样例,那么教材已经提供了大量的例程,而且在建立Java语言编程环境之后,在Java系统的安装目录中也包含了一些例程。

在进行编程实践的时候,经常需要查阅在线帮助文档,而不是各种教材或参考书。这对很多初学者来说有较大的难度,其实难度主要来自于自身的惰性,要真正做到这一点,基本上没有任何客观条件上的难度。现在越来越多的在线帮助文档已经有了相应的中文版本,即使直接使用英文的在线帮助文档,它的词汇量也不大,而且语法结构比较简单。另外,理解这些词汇的关键是实践,即通过实践理解或加深理解各种中文或英文术语。为了强化读者对在线帮助文档的使用,在2004年出版的那本教材中,我基本上不将在线帮助文档的内容编写入教材,而是希望读者对照在线帮助文档进行教材的阅读。当然,这样有一个负面作用,那就是会使阅读教材的速度变慢。其实,这种“慢”属于“磨刀不误砍柴工”,会给读者以后的工作或学习带来较大的益处,一般会提高工作和学习的整体效率。在本教材中,应很多读者的要求,我将这些本来属于在线帮助文档的内容添加到本教材中,但是读者不仅不能忽略在线帮助文档,而且应当将重视程度提高到足够的高度。当然,本教材不是简单地翻译在线帮助文档,而是在该文档的基础上增加编程原理、经验和技巧的介绍,并且在内容上与在线帮助文档相比力求准确、简洁、易于理解。

本教材和2007年出版的教材《Java程序设计教程》(雍俊海,北京,清华大学出版社,2007)是姊妹篇,后者内容更为全面,体现出一定的手册特点。本教材内容是后者的节选,保留了Java语言核心部分,去掉了其中属于提高的部分内容。这两本教材都是在2004年出版的那本教材的基础上编写而成的,相对于2004年出版的那本教材,本教材删除了其中属于提高的部分,添加了泛型、枚举和哈希表等内容,同时添加了相关的在线帮助文档内容,从而方便读者阅读。为了方便读者查找知识点和中心内容,本教材通过加黑加粗加框的方式强调各部分内容的中心词以及各基本概念或定义的核心词,在附录中添加了图、表、例程以及类和接口的页码索引。同时,本教材继承了2004年出版的那本教材的一些特点,例如考虑了如何方便读者自学,希望各章内容的相关性尽可能地小。本教材的所有例程都在Java 7.0版本(也称为1.7版本)上编译运行。

本教材既可以作为计算机专业和非计算机专业的基础教材,也可以作为需要使用计算机的工程人员和科技工作者的自学参考书。本教材在编写与出版的过程中得到了许多朋友

的帮助,这里一并表示诚挚的谢意,其中,读者与选修我所负责的课程的同学起到非常重要的作用,他们的建议和批评意见是教材发生变化的最重要的外在因素,这里再次对他们表示诚挚的谢意,真诚地希望读者能够轻松并且愉悦地掌握 Java 程序设计,也希望自己能做得更好。清华大学的研究生杜敏、范怀宇、高扬、高跃、李勇、林鸿维、刘倩欣、刘曙、刘永宾、卢新来、潘峰、宋征轩、孙学卫、佟强、汪亚君、王天兴、王维勃、王治中、夏雨、许嵩罡、余忠冕、喻晓峰、张佳、张楠、张怡文和赵宏星等同学参与了本教材的核对工作,本教材也凝聚了他们的劳动结晶,在此表示感谢。欢迎广大读者特别是讲授此课程的老师对本教材进行批评和指正,我真诚地希望这本教材能够给读者带来轻松和快乐,而我也会不断为此努力。目前,我的个人主页是 <http://cgcad.thss.tsinghua.edu.cn/~yongjh/chn/index.html>,E-mail 是 yongjunhai@tsinghua.org.cn,真诚地欢迎广大读者提出各种建设性意见。

本教材还配有网站,网址是 <http://cgcad.thss.tsinghua.edu.cn/~yongjh/java/index.html>。如果出现网络不稳定或无法连接的现象,请过几天重新连接或者访问清华大学出版社的网站,清华大学出版社网站的网址是 <http://www.tup.tsinghua.edu.cn/>。

通过这两个网站,读者可以下载与本教材相关的一些资料,例如:

- (1) 本教材中用到的所有例程;
- (2) 本教材的课件(可能会不断更新)。

雍俊海
2014 年 5 月于清华园

CONTENTS

目录

第1章 绪论	1
1.1 Java 语言的历史简介	1
1.2 Java 语言的特点	2
1.3 Java 开发环境的建立	3
1.4 Java 程序及其执行过程	13
1.4.1 开发 Java 程序的工作流程	13
1.4.2 Java 程序的工作原理	21
1.5 本章小结	22
习题	23
第2章 结构化程序设计	24
2.1 标识符和关键字	24
2.2 基本数据类型、直接量和变量	28
2.2.1 基本数据类型	28
2.2.2 直接量	30
2.2.3 变量	31
2.3 运算符	34
2.3.1 算术运算符	35
2.3.2 关系运算符	37
2.3.3 布尔逻辑运算符	37
2.3.4 位运算符	39
2.3.5 赋值类运算符	42
2.3.6 条件运算符	43
2.3.7 其他运算符	43

2.4 控制结构	44
2.4.1 if 语句和 if...else 语句	44
2.4.2 switch 语句	46
2.4.3 for 语句	48
2.4.4 while 语句	50
2.4.5 do...while 语句	51
2.4.6 break 语句	52
2.4.7 continue 语句	54
2.5 结构化程序设计方法	57
2.6 本章小结	59
习题	60
第 3 章 面向对象程序设计	63
3.1 类、域、方法和实例对象	63
3.2 继承性	71
3.3 多态性	76
3.3.1 静态多态性	76
3.3.2 动态多态性	78
3.4 包	81
3.5 封装性	86
3.6 修饰词 abstract、static 和 final	89
3.6.1 修饰词 abstract	89
3.6.2 修饰词 static	90
3.6.3 修饰词 final	95
3.7 接口	95
3.8 内部类	97
3.9 变量作用域范围与参数传递方式	106
3.9.1 变量作用域范围	106
3.9.2 方法调用的值传递方式	110
3.10 递归方法	114
3.11 面向对象程序设计的基本思想	116
3.12 本章小结	120
习题	120
第 4 章 数组、字符串、向量和哈希表	123
4.1 数组	123
4.1.1 一维数组	124
4.1.2 多维数组	126
4.2 字符串和字符串缓冲区	133

4.2.1 String	133
4.2.2 StringBuffer	145
4.3 向量	151
4.4 哈希表	156
4.4.1 哈希表的基本原理	156
4.4.2 Hashtable	157
4.5 本章小结	160
习题	161
第5章 泛型和枚举	164
5.1 泛型	164
5.2 枚举	170
5.3 本章小结	174
习题	174
第6章 异常处理	175
6.1 异常及其种类	175
6.2 异常的产生	177
6.3 异常的处理	177
6.4 本章小结	183
习题	183
第7章 文件与数据流	185
7.1 输入流与输出流	185
7.1.1 InputStream和FileInputStream	185
7.1.2 OutputStream和FileOutputStream	189
7.1.3 PrintStream	192
7.1.4 带缓存的输入流和输出流	196
7.1.5 标准输入/输出流的重定向	199
7.2 随机访问文件	201
7.3 读写器	205
7.3.1 Reader和Writer	205
7.3.2 FileReader和FileWriter	206
7.3.3 带缓存的读写器	208
7.3.4 PrintWriter	211
7.3.5 从控制台窗口读入数据	214
7.4 文件	217
7.5 本章小结	221
习题	221

第 8 章 Swing 图形用户界面程序设计	223
8.1 组件和容器	223
8.1.1 整体介绍	223
8.1.2 JFrame 和 JLabel	224
8.1.3 JTextField 和 JPasswordField	228
8.1.4 JButton、JCheckBox 和 JRadioButton	230
8.1.5 JComboBox、JList、JTextArea 和 JScrollPane	234
8.1.6 JPanel	238
8.2 布局管理器	239
8.2.1 FlowLayout 和 GridLayout	240
8.2.2 BorderLayout	242
8.2.3 BoxLayout	244
8.2.4 CardLayout	246
8.2.5 组合布局方式	248
8.3 事件处理及其模型	250
8.3.1 事件处理模型	250
8.3.2 鼠标事件处理和自定义绘制方法	257
8.4 菜单	267
8.5 本章小结	276
习题	278
第 9 章 小应用程序	280
9.1 源程序	280
9.1.1 生命周期	280
9.1.2 图形用户界面	284
9.2 网页标记	289
9.2.1 采用 APPLET 网页标记	289
9.2.2 归档文件	291
9.2.3 小应用程序参数	295
9.3 应用程序与小应用程序	297
9.4 本章小结	303
习题	303
第 10 章 编程规范	305
10.1 程序编写规范	305
10.1.1 命名规范	305
10.1.2 排版规范	308
10.1.3 语句	310

10.1.4 文件组织	312
10.2 文档注释	314
10.3 本章小结	324
习题	324
附录 A 图的索引	325
附录 B 表的索引	328
附录 C 例程索引	329
附录 D 类和接口索引	331
参考文献	334

绪 论

第1章

自从 1946 年第一台 ENIAC 计算机在美国宾夕法尼亚州 (Pennsylvania) 诞生以来, 计算机产业的发展速度以及计算机向其他领域渗透的速度都远远出乎人们的意料, 现在, 它已经成为各行各业的基本工具。在这期间, 计算机语言本身也在飞速发展, 其发展方向之一就是使计算机语言越来越接近于人们的思维习惯。按照这种发展方向来分, 计算机语言可以分为第一代 (机器) 语言、第二代 (低级) 语言和第三代 (高级) 语言。这种发展方向使得程序越来越容易编写、阅读、维护、复用和移植。Java 语言就是这样发展起来的一种高级语言, 易学易用并迅速受到推崇。目前, Java 语言已经成为最常用的计算机语言之一。本章简单介绍 Java 的历史和特点以及从建立 Java 环境到运行 Java 程序的整个流程。

1.1 Java 语言的历史简介

Java 语言是一种很新的计算机语言, 它的历史很短。Java 语言的前身是 Oak 计算机语言。1991 年, Sun 公司为了占领智能消费型电子产品的市场, 资助了一个“绿色项目”。这个项目是由 James Gosling 负责的, 主要开发用于智能消费型电子产品的语言, 即 Oak 语言。Oak 语言是在 C 和 C++ 计算机语言的基础上进行简化和改进的一种语言。项目进行不久, Sun 公司意识到已经存在一种叫做 Oak 的计算机语言。于是, Sun 公司重新给自己开发的这种语言命名为 Java 计算机语言。这样, James Gosling 就成为 Java 语言的创始人。

但 Java 语言很快就遇到了一些困难, 因为 Sun 公司发现智能消费型电子产品的发展并没有预想的那样快, 而且当时 Sun 公司在竞争一个大项目时失败了, Sun 公司差一点就要取消这个“绿色项目”。到 1993 年, Sun 公司重新分析市场需求, 认为网络具有很好的发展前景, 而且 Java 语言似乎非常适合网络编程, 于是 Sun 公司将 Java 语言的应用背景转向网络市场, 为网页增加“动态的内容”。

Sun 公司的这次市场策略转变是非常成功的。1995 年, 当 Sun 公司在“Sun

World 95”大会上第一次正式公布 Java 语言时,立即引起了巨大的轰动,因为那时正是网络“泡沫经济”的时代,网络正处于“狂热”的时期。Java 语言为网络的发展开辟了一个新纪元。同年,Java 语言就被计算机杂志《PC Magazine》评为 1995 年十大优秀科技产品之一(当年计算机产品就此一项入选)。微软公司总裁比尔·盖茨当时的一句话“Java 语言是有史以来最卓越的计算机程序设计语言”也是当时人们对 Java 语言的普遍评价。许多计算机公司都开始支持和开发 Java 产品,其中包括 IBM 公司、Apple 公司和 Oracle 公司等。1996 年, Sun 公司专门成立 Javasoft 分公司来发展 Java, Java 从此得到了迅猛的发展和广泛的应用,这种速度是前所未有的。

1999 年, Sun 公司重新组织 Java 平台的集成方法,加强 Java 企业级应用平台的功能。目前, Java 程序可以支持智能消费型电子产品的开发,各种应用程序的开发(包括个人应用程序和企业级的应用程序),尤其是网络程序的开发,Java 语言拥有“互联网上的世界语”的美称。

2009 年 4 月 20 日,甲骨文(Oracle)公司和 Sun 公司召开新闻发布会,正式宣布甲骨文将以每股 9.5 美元的现金收购 Sun 公司,交易总价值达到 74 亿美元。甲骨文公司表示,他们之所以收购 Sun 公司主要是看重了 Sun 公司的两大软件资产——Java 和 Solaris,非常希望这次收购能够让 Java 的发展上一个新的台阶。

1.2 Java 语言的特点

Java 语言的特点与其历史发展是相关的。它之所以能够受到如此众多的好评以及拥有如此迅猛的发展速度与其语言本身的特点是分不开的,其主要特点如下所述。

(1) **简单性**: 从 Java 语言的发展史可以了解到 Java 语言是在 C 和 C++ 计算机语言的基础上进行简化和改进的一种新型计算机语言,它去掉了 C 和 C++ 中最难正确应用的指针和最难理解的多重继承技术等内容,通过垃圾自动回收机制简化了程序内存管理,统一了各种数据类型在不同操作系统平台上所占用的内存大小。Java 程序的简单性是其得以迅速普及的最重要的原因之一。

(2) **网络特性**: Java 语言正是因为其对互联网的良好支持而受到推崇并得以迅速推广的,Java 语言是目前对网络支持最全面、与网络关系最密切的计算机语言之一。

(3) **面向对象**: 由于 Java 语言是一种新型计算机语言,没有兼容过程式计算机语言的负担,所以 Java 语言在面向对象的特性上比 C++ 语言更为彻底。面向对象模型是一种模拟人类社会和人解决实际问题的模型,因此更符合人们的思维习惯,而且容易扩充和维护。它的缺点是程序在开发的过程中常常会变得越来越大。

(4) **平台无关性/可移植性**: Java 语言的设计目标是让其程序不用修改就可以在任何一种计算机平台上运行。解决异构操作系统的兼容性问题是一个很艰巨的任务, Sun 公司提供的 Java 语言也没有完全做到这一点。在 Java 语言的说明书中, Sun 公司用权重(weight)的轻重来表示其提供的类或成员方法与计算机平台的相关性大小。不过总的来说,Java 语言在这一方面是做得最好的计算机语言。

(5) **鲁棒性**: 鲁棒性指的是程序执行的**稳定性**,也称为**健壮性**。Java 语言设计者在设计 Java 语言的过程中一直考虑如何减少编写程序的过程中可能产生的错误。Java 在编译和

执行的过程中都会进行比较严格的检查,以减少错误的发生。Java 语言的垃圾自动回收机制和异常处理机制在很大程度上提高了程序的鲁棒性。另外,Java 语言的简单性同时也在一定程度上保证了程序的鲁棒性。

(6) **安全性**: 在网络上运行的 Java 程序是符合网络安全协议的。在执行 Java 程序的过程中,Java 虚拟机对程序的安全性进行检测。一般来说,Java 程序是安全的,它不会访问或修改不允许访问的内存或文件。

(7) **多线程性**: 多线程性主要用来处理复杂事务或需要并行的事务,组成 Java 虚拟机的各个程序本身一般也采用多线程机制。采用多线程机制是提高程序运行效率的一种方法,不过,增加了程序的设计难度。

(8) **解释性**: Java 语言是一种解释执行的语言。这是 Java 语言的一个缺点,因为解释执行的语言相比编译执行的语言(如 C 和 C++ 语言)需要额外的语句解释步骤及其运行时间。

总而言之,Java 语言是一种易学好用、健壮性好,但采用解释执行的计算机语言。用 Java 语言编写程序一般会比用其他计算机语言编写程序花费的时间更少,而且调试所需的时间一般也会较短。对于计算机初学者或正打算开始学习一门计算机语言的工程师或教学科研工作者来说,选择 Java 程序设计进行学习是一个很好的方案。

1.3 Java 开发环境的建立

要学好任何一门计算机语言,都必须加强实践,即编写程序解决各种实际问题。只有多练习,勇于尝试,并善于总结,才能真正掌握和精通一门计算机语言。要练习,首先需要建立起 Java 的开发环境。

要在一台计算机上编写和运行 Java 程序,首先应该在这台计算机上建立起 Java 开发环境。**建立 Java 开发环境**就是在计算机上安装 Java 开发工具包并在计算机中设置相应的参数,使得 Java 开发工具包可以在计算机中顺利地得到正确的运行。甲骨文(Oracle)公司推出的开发工具包主要可以分为三类,即 J2SE(JavaTM 2 Platform, Standard Edition)、J2EE(JavaTM 2 Platform, Enterprise Edition)和 J2ME(JavaTM 2 Platform, Micro Edition)。其中,J2SE 是用于工作站和个人计算机(简称 PC 机)的标准开发工具包,J2EE 是用于企业级开发的工具包,J2ME 用于开发智能消费型电子产品(如移动电话和汽车导航系统等)。另外,甲骨文(Oracle)公司还推出了 JavaFX,JavaFX 是面向设计人员的脚本编程语言,进一步降低了二维动画和网页等设计难度。本书介绍的是基于 J2SE 的 Java 程序设计。因为 J2SE 的早期版本一直被称为 JDK(Java Developer's Kit),所以在甲骨文(Oracle)公司的文档或网页中 J2SE 通常也被称为 JDK。J2SE 还有一个别名,就是 Java SE(Java, Standard Edition)。J2SE 的版本编号常常会让初学者感到非常困惑,J2SE 目前存在两个版本命名体系,例如,J2SE 7.0 版本与 J2SE 1.7 版本实际上是同一个版本。因此,J2SE 7、Java SE 7、JDK 7、J2SE 1.7、Java SE 1.7 和 JDK 1.7 完全是同一个东西。**建立基于 J2SE 的 Java 开发环境的步骤**如下:

- (1) 下载 J2SE 安装程序;
- (2) 运行 J2SE 安装程序,安装 J2SE;

(3) 设置环境变量运行路径 (path) 和类路径 (classpath);

(4) 下载 J2SE 的在线帮助文档。

J2SE 安装程序可以从甲骨文 (Oracle) 公司的网站 (<http://www.oracle.com>) 下载。在下载时用户要注意自己计算机的操作系统类型, 下载的安装程序应当与自己计算机的操作系统相匹配, 而且版本一般选择最新的。安装程序下载完后就可以运行了, 安装过程只要遵循安装程序提供的指示进行就可以了。在安装完成之后会进入步骤 (3), 这个步骤的目标是给计算机设置 Java 工具包运行的环境变量, 即运行路径 (path) 和类路径 (classpath), 其中, **运行路径 (path) 变量**记录的是各个运行程序所在的路径, 系统会根据这个变量的值来查找运行程序。因此, 在运行路径 (path) 变量中加上 J2SE 运行程序所在的路径, 可以使得在运行 J2SE 程序时不必输入全路径名。**类路径 (classpath) 环境变量**通常用来记录当前路径和 J2SE 类库所在的路径, 这是 J2SE 需要的一个环境变量。在 J2SE 类库中包含 J2SE 系统提供的各种软件包, 其中包括各个类和接口等。在设置好类路径 (classpath) 环境变量之后, 可以在程序中直接使用在当前路径和 J2SE 类库所在路径下的各种 Java 软件包的类或者接口等, 通常将类路径 (classpath) 的值设置为**当前路径** (用一个点 “.” 表示) 和 J2SE 类库所在的路径。在设置运行路径 (path) 和类路径 (classpath) 变量时, 在相邻两个路径之间用分号 (在 Windows 系列的操作系统中) 或者冒号 (在 Linux 或 UNIX 操作系统中) 隔开。以 Windows 系列的操作系统为例, 本节假设 J2SE 的安装路径是 “C:\j2sdk” (如果实际的安装路径不是 “C:\j2sdk”, 则用实际的安装路径代替 “C:\j2sdk”), 需要**设置的环境变量及其值**分别为:

```
path=%path%;C:\j2sdk\bin
classpath=.;C:\j2sdk\lib
```

其中, “%path%” 表示环境变量 path 原有的值, “.” 表示当前路径, “C:\j2sdk\bin” 是 Java 虚拟机的各个运行程序所在的路径, “C:\j2sdk\lib” 是 J2SE 类库所在的路径。实际的具体设置步骤在不同的操作系统中会有所不同, 而且有多种实现方法。无论是在什么操作系统下或采用什么方法, 只要给计算机系统正确地设置上面的两个环境变量就可以了。下面分别介绍在操作系统 Microsoft Windows 8/7/XP/NT/2000/Me/98 以及在 Linux 或 UNIX 操作系统下设置这两个环境变量的方法。

1. 在 Microsoft Windows 8 下设置运行路径 (path) 和类路径 (classpath) 的步骤

这里首先介绍如何在 Windows 8 的桌面上显示 “计算机” 的图标, 需要进入 Windows 8 的桌面, 右击 Windows 8 桌面的空白处, 弹出如图 1.1 所示的右键菜单, 然后选择其中的 “个性化” 命令, 将打开如图 1.2 所示的 “个性化” 窗口。

如图 1.2 所示, 单击其中的 “更改桌面图标” 选项, 将弹出如图 1.3 所示的 “桌面图标设置” 对话框。选中其中的 “计算机” 桌面图标选项, 则在 “计算机” 桌面图标选项左侧的方框内会出现对钩的符号, 如图 1.3 所示。依次单击对话框中的 “应用” 和 “确定” 按钮, 这时将在 Windows 8 的桌面上出现 “计算机” 的图标。

在 Windows 8 的桌面上右击 “计算机” 图标, 弹出如图 1.4 所示的右键菜单。然后选择其中的 “属性” 命令, 打开如图 1.5 所示的 “系统” 窗口。

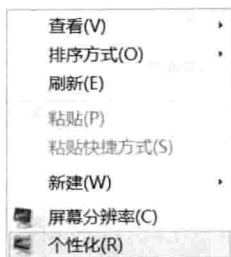


图 1.1 Windows 8 桌面的右键菜单



图 1.2 Windows 8 的“个性化”窗口



图 1.3 Windows 8 的“桌面图标设置”对话框

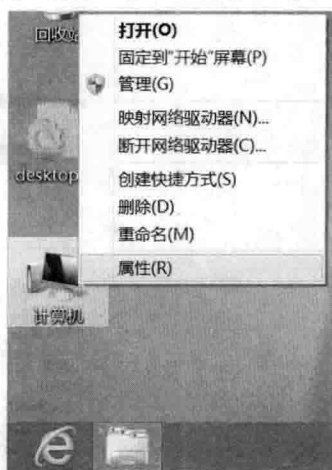


图 1.4 Windows 8 桌面上的“计算机”图标的右键菜单

在如图 1.5 所示的“系统”窗口中单击“高级系统设置”选项，会弹出如图 1.6 所示的“系统属性”对话框。

在“系统属性”对话框中选择“高级”选项卡，这时可以看到“环境变量”按钮，单击“环境变量”按钮，会弹出如图 1.7 所示的“环境变量”对话框。

在“环境变量”对话框中，可以分别为用户变量表和系统变量表设置或添加运行路径(path)和类路径(classpath)两个变量。如果在变量表中已经添加了路径(path)变量或类路径(classpath)变量，则在如图 1.7 所示的“环境变量”对话框中先在变量表中选中该变量，再单击变量表下方的“编辑”按钮。这里需要注意，Windows 系统通常不区分大小写，因此，path 变量、Path 变量、PATH 变量均是同一个变量。在单击“编辑”按钮之后，将弹出如图 1.8 所示的“编辑系统变量”对话框。如果是编辑路径(path)变量，则在原有



图 1.5 Windows 8 的“系统”窗口

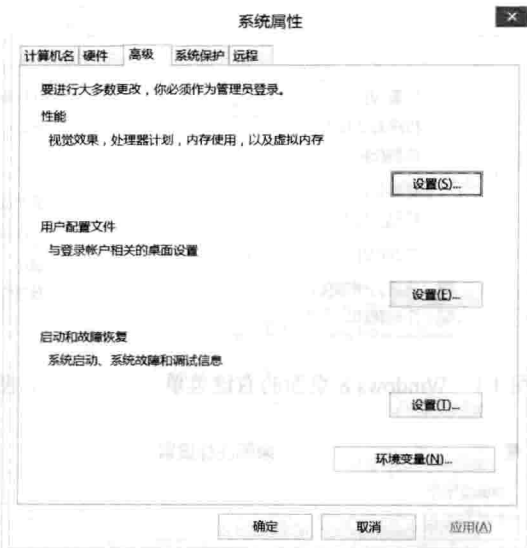


图 1.6 Windows 8 的“系统属性”对话框



图 1.7 Windows 8 的“环境变量”对话框

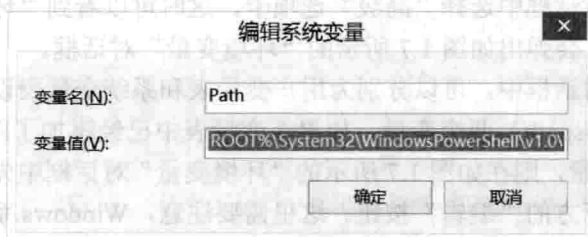


图 1.8 Windows 8 的“编辑系统变量”对话框