

Java 语言 及编程技巧

姚庭宝 刘青宝 蒋 杰 编著

国防科技大学出版社

NATIONAL UNIVERSITY OF DEFENSE TECHNOLOGY PRESS

TP312
2081

Java 语言及编程技巧

姚庭宝 刘青宝 蒋 杰 编 著

国防科技大学出版社
湖南·长沙

图书在版编目(CIP)数据

Java 语言及编程技巧/姚庭宝,刘青宝,蒋杰编著. —长沙:国防科技大学出版社,2006.9
ISBN 7-81099-352-6

I. J… II. ①姚…②刘…③蒋… III. Java 语言—程序设计—高等学校—教材
IV. TP312

中国版本图书馆 CIP 数据核字(2006)第 080066 号

国防科技大学出版社出版发行

电话:(0731)4572640 邮政编码:410073

<http://www.gfkdcbs.com>

责任编辑:何晋 徐飞 责任校对:唐卫藏 肖滨

新华书店总店北京发行所经销

国防科技大学印刷厂印装

*

开本:787×1092 1/16 印张:27.5 字数:704 千

2006 年 9 月第 1 版第 1 次印刷 印数:1-3000 册

ISBN 7-81099-352-6/TP·44

*

定价:38.00 元

前 言

Java 是在 C、C++ 的基础上发展而成的又一种现代程序设计语言。Java 是新一代面向对象的程序设计语言。它也是一种卓越的 Internet 语言。Java 与 Internet 息息相关、交相辉映。Internet 使得 Java 能找到适宜的场合充分地施展其在程序设计方面的卓越才能,而 Java 则对 Internet 的进一步发展产生了深远的影响。Java 随着 Internet 和 Web 的飞速发展而不断改进与完善。

Java 设计团队在设计目标中考虑了一系列需加以实现的关键因素,如:简单、可移植、面向对象、解释型、分布式、高性能、健壮、多线程、安全、动态以及体系结构中立等,围绕着以与 Internet 紧密相关的安全性、可移植性问题为主要目标,并充分顾及其他诸因素,开发出一种更佳的、全面的编程语言和环境,用于创建更为简洁、更加可靠的软件。

Java 并非一种从头开始设计、研制的崭新的计算机语言。它大量地借鉴了 C 和 C++ 的诸多优良特性和思想,具有与 C 和 C++ 相似的概念构架。C、C++ 和 Java 的相继出现是现代程序设计的一个个里程碑。Java 继承了 C 的语法,Java 的面向对象模型参照了 C++。但 Java 摒弃了 C++ 中一些很少使用、难以理解、易于混淆的特性,使得 Java 比 C++ 更为简洁有效,更易写出优良的代码。另外,Java 的新增特性使其特别适合于网络编程和多线程编程,提供了一个强大的、有更多丰富类库可供利用的、逻辑一致的程序设计环境,使其可用于支持开发各类图形用户界面、多媒体、网络、数据库连接以及其他复杂功能的软件。Java 作为一种革命性的技术,它正引领着软件开发的方向。IT 行业对 Java 人才的需求正迅速地增长。

全书主要内容分为上、下两编。上编为《基础编》,共八章。上编较为详尽地阐述了 Java 的一系列核心概念、语言要素与基础知识,特别是面向对象技术的基本概念、特性与使用,介绍用 Java 进行程序设计特别是面向对象程序设计的基本方法与技巧,提供内容丰富的示例与完整程序加以佐证与应用,并推介良好编程风格与习惯的养成,以促进广大读者正确地领会 Java 功能强大、使用灵活以及与众不同的特色,逐步培养与掌握编写 Java 应用程序的诀窍,增强并提高用 Java 进行程序开发的能力。下编为《提高编》,共五章。下编较为深入地探讨 Java 所提供的一系列高级特性及其应用,以扩展 Java 的应用范围,提升 Java 的应用能力。下编所涉及的内容包括:使用 Swing 类库来创建具有图形用户界面的图形应用程序;选用各种布局管理器布置并美化图形用户界面;利用事件处理机制,在程序运行期间实现用户和图形用户界面实时交互应答,响应用户输入,进行相应的事件处理;在图形用户界面上显示矢量文本,绘制一些基本的、特定的图形,配置颜色,选择字体,以及在封闭图形内填充着色,等等;最后,还介绍怎样编写 Java 小应用程序——applet。applet 是作为 Web 页面的一部分运行在支持 Java 的浏览器

(比如 Microsoft 的 IE 浏览器)上的小型 Java 程序。applet 与 Java 应用程序是用户在 Java 编程上所采用的两种选择途径。

全书介绍了一百五十多个 Java 应用程序与 applet。它们将为广大读者展现一个功能强大、齐全、使用灵活、可靠,内容充实,技术先进,应用广泛的优秀程序设计语言——Java。

作者在编写本书的过程中,力求做到内容翔实,信息量大,结构严谨,编排合理,文字流畅,示例众多,便于自学,实用性强。文字叙述努力体现启发性、针对性、直观性、技巧性。每章末均安排有小结。作者还将编写与本书相配套的《Java 语言程序设计学习要点与习题解析》一书,提供大量能体现 Java 诸多特性的各类示例特别是程序示例,以帮助广大读者进一步培养、检验他们的理解力,增进独立编程以及上机操作的能力,提高并加强 Java 应用的整体水平。

作者在著述时,参阅了国内外一些有关 Java 语言及程序设计的专著、教材,受益匪浅。在此,谨向这些专著、教材的作者们表示钦佩并致谢。

国防科技大学信息系统与管理学院邓苏教材策划了本书的选题,在作者编写过程中一直给予支持与指点。作者谨予致谢。

对于本书的内容选择以及文字叙述上的不当之处,热诚欢迎广大读者包括大学生、研究生提出批评、建议,也衷心希望能得到各高校教师及各界专家、同行的指教与帮助。

姚庭宝

2006 年 6 月于国防科技大学

信息系统与管理学院

目 录

◁ 上编 基础编 ▷

第一章 Java 编程基础

1.1 关于 Java	(2)
1.2 Java 程序的开发过程	(3)
1.3 Java 程序的基本结构	(4)
1.4 字符集、标识符与关键字	(11)
1.5 基本数据类型	(14)
1.6 运算符与表达式	(23)
1.7 赋值语句及赋值表达式	(31)
1.8 程序设计风格	(36)
1.9 小结	(37)

第二章 程序控制结构

2.1 if 语句	(38)
2.2 switch 语句	(45)
2.3 while 语句	(47)
2.4 do - while 语句	(49)
2.5 for 语句	(53)
2.6 用于循环控制的几个特殊运算符	(56)
2.7 多重循环语句	(60)
2.8 从键盘输入字符	(67)
2.9 break 语句与 continue 语句	(70)
2.10 小结	(75)

第三章 数组与字符串

3.1 一维数组	(77)
3.2 多维数组	(94)

3.3 字符串与字符串类 String	(104)
3.4 命令行参数	(112)
3.5 小结	(113)

第四章 方法

4.1 方法概述	(114)
4.2 方法定义与方法调用	(115)
4.3 方法的嵌套调用和递归调用	(125)
4.4 方法参数	(135)
4.5 数组与字符串作为方法参数传递	(136)
4.6 方法重载	(147)
4.7 小结	(152)

第五章 类与对象

5.1 面向对象技术的基本特征	(153)
5.2 Java 中的类	(154)
5.3 类定义和对象声明	(156)
5.4 类的构造方法	(168)
5.5 this 引用	(174)
5.6 static 成员和 static 代码块	(178)
5.7 内部类	(184)
5.8 小结	

第六章 类的继承和多态性

6.1 类的继承特性——基类与派生类	(186)
6.2 类的继承层次结构	(192)
6.3 类继承层次结构下的构造方法	(195)
6.4 派生类与基类中同名成员的处理——同名覆盖	(204)
6.5 同名覆盖成员方法支持多态性和动态联编	(206)
6.6 使用抽象方法和抽象类	(209)
6.7 使用 final 方法和 final 类	(213)
6.8 小结	(216)

第七章 包与接口

7.1 包	(217)
7.2 包和成员访问控制机制	(221)

7.3 使用 import 语句导入包	(225)
7.4 接口	(228)
7.5 使用接口引用	(231)
7.6 扩展接口——接口继承链	(235)
7.7 小结	(237)

第八章 异常处理

8.1 Java 编程中的错误	(238)
8.2 Java 异常处理机制——使用 try 块-throw 语句-catch 块	(244)
8.3 再次引发异常	(251)
8.4 使用 finally 块	(253)
8.5 异常的传播	(257)
8.6 使用 throws 子句	(261)
8.7 创建异常派生类	(265)
8.8 小结	(268)

◁ 下编 提高编 ▷

第九章 使用 Swing 创建用户界面

9.1 Swing 和 AWT	(272)
9.2 组件与容器	(273)
9.3 窗口和帧	(274)
9.4 在帧内添加按钮组件和标签组件	(279)
9.5 在帧内添加文本框组件和文本区域组件	(283)
9.6 在帧内添加滚动条组件	(287)
9.7 在帧内添加复选框组件和单选按钮组件	(291)
9.8 在帧内添加下拉式列表组件和组合框组件	(293)
9.9 小结	(298)

第十章 使用布局管理器布置用户界面

10.1 Java 中的布局管理器	(299)
10.2 使用顺序布局管理器	(300)
10.3 使用网格布局管理器	(305)
10.4 使用边框布局管理器	(307)
10.5 综合地使用多个布局管理器	(311)

10.6 使用卡片布局管理器	(315)
10.7 使用网格袋布局管理器	(317)
10.8 小结	(326)

第十一章 事件处理——响应用户输入

11.1 事件处理模式	(327)
11.2 行为事件处理	(329)
11.3 调整事件处理	(336)
11.4 选项事件处理	(341)
11.5 键盘事件处理	(344)
11.6 鼠标事件处理	(346)
11.7 窗口事件处理	(350)
11.8 事件监听器的适配器类	(355)
11.9 小结	(359)

第十二章 图形编程

12.1 矢量文本	(360)
12.2 2D 图形类	(363)
12.3 直线和矩形	(364)
12.4 椭圆和圆	(367)
12.5 颜色配置与填充着色	(371)
12.6 字体选择	(377)
12.7 多边形和弧线	(381)
12.8 饼图	(390)
12.9 小结	(395)

第十三章 Java 小应用程序——applet

13.1 创建小应用程序	(397)
13.2 使用 HTML 标记和属性——HTML 文本文件	(401)
13.3 将 Java 应用程序转换为 applet	(406)
13.4 在 applet 中接收并处理参数	(417)
13.5 小结	(424)

附录一 常用字符 ASCII 代码对照表	(426)
----------------------------	-------

附录二 运算符的优先级和结合性	(427)
-----------------------	-------

主要参考书目	(429)
--------------	-------

上编 基础编

上编较为详尽地阐述了 Java 的一系列核心概念、语言要素与基础知识,特别是面向对象技术的基本概念、特性与使用,介绍用 Java 进行程序设计特别是面向对象程序设计的基本方法与技巧,提供内容丰富的示例与完整程序加以佐证与应用,并推介良好编程风格与习惯,以促进广大读者正确地领会 Java 功能强大、使用灵活以及与众不同的特色,逐步培养与掌握编写 Java 应用程序的诀窍,增强并提高用 Java 进行程序开发的能力。殷切期望广大读者在阅读与理解的过程中,能够勤思索,多实践,先模仿,再创新;主动进取,促进思维,举一反三,灵活运用。不仅着力于培养与提高使用 Java 这一优秀语言工具进行程序设计的兴趣,而且还需追求充分挖掘 Java 强大的潜在能力,不断加强用计算机编程解决实际应用问题的本领。必须紧紧抓住实践环节,坚持不懈,反复演练,通过经常性的上机实践对所编写的 Java 应用程序进行编译、调试、运行、检验、巩固所学概念与知识,考核创新意识。在实践中领悟真谛,在实践中拓展技能。这必将为进一步学习下编《提高编》打下较为坚实的基础。

上编主要内容

- 第一章 Java 编程基础
- 第二章 程序控制结构
- 第三章 数组与字符串
- 第四章 方法
- 第五章 类与对象
- 第六章 类的继承和多态性
- 第七章 包与接口
- 第八章 异常处理

第一章 Java 编程基础

Java 是在 C、C++ 的基础上发展而成的又一种现代程序设计语言。它已成为一种卓越的 Internet 语言,正广为流行,并不断完善。

对于初次接触并下决心学习 Java 语言及程序设计的读者来说,本章内容是入门开篇,是进入 Java 神奇殿堂的基石。本章通过展示一些简单示例介绍 Java 程序的基本结构,以及构成 Java 程序若干最基本、最常用的元素概念及使用规则。

本章主要内容包括:关于 Java;Java 程序的开发过程;Java 程序的基本结构;字符集、标识符与关键字;基本数据类型;运算符与表达式;赋值语句及赋值表达式;程序设计风格。

1.1 关于 Java

Java 是新一代面向对象的程序设计语言。它也是一种卓越的 Internet 语言。Java 的出现促进了程序设计思维的变化、程序设计技术的改进和运算环境的改变。它建立了一种新的计算机语言标准,体现了计算机语言的革新,改变了软件的开发方式。Java 作为一种革命性的技术,它正引领着软件开发的方向。

Java 与 Internet 息息相关、交相辉映。Internet 使得 Java 能找到适宜的场合充分施展其在程序设计方面的卓越才能,而 Java 则对 Internet 的进一步发展产生了深远的影响。Java 随着 Internet 和 Web 的飞速发展而不断改进与完善。如今,Java 不仅是一种优秀的 Internet 主流编程语言,而且已经发展成为当今广泛流行并应用的、具有超强功能的编程平台。

Java 设计团队在设计目标中考虑了一系列需加以实现的关键因素,如:简单、可移植、面向对象、解释型、分布式、高性能、健壮、多线程、安全、动态以及体系结构中立,围绕着以与 Internet 紧密相关的安全性、可移植性问题为主要目标,并充分顾及其他诸因素,开发出一种更佳的、全面的编程语言和环境,用于创建更为简洁、更加可靠的软件。经历了近十年的不断发展与完善,如今 Java 语言已经被推向计算机语言设计的前端,牢固地站稳了脚跟,成为当今最为流行的计算机语言之一,足以在 Internet、Web 以及其他前沿领域展现其最佳的技术和功能。

Java 并非一种从头开始设计、研制的崭新的计算机语言。它大量地借鉴了 C 和 C++ 的诸多优良特性和思想,具有与 C 和 C++ 相似的概念构架。C 语言是国际上广泛流行的一种结构化语言。C 语言具有很强的功能和高度的灵活性,能够提供丰富的数据类型、广泛使用的软件以及一组很丰富的、供计算和数据处理的运算符。在相当长的时期内,C 语言业已成为工业界所选用的系统实现语言。然而,C 语言是一种面向过程的编程语言,已不能满足运用面向对象程序设计方法开发现代软件的需求。C++ 语言应运而生。C++ 语言是在 C 语言基础上,扩充了面向对象的程序设计发展而成的一种全新的计算机高级语言。它是 C 语言的有机继承

与发展。采用 C++ 语言编程能够更自然地体现抽象数据类型的概念,从而更本质地描述数据结构 and 算法,在设计、开发系统软件与应用软件中,展现了强大的生命力。Java 与 C、C++ 具有直接的联系。C、C++ 和 Java 的相继出现是现代程序设计的一个个里程碑。Java 继承了 C 的语法,Java 的面向对象模型参照了 C++。但 Java 摒弃了 C++ 中一些很少使用、难以理解、易于混淆的特性,使得 Java 比 C++ 更为简洁有效,更易写出优良的代码。另外,Java 的新增特性使其特别适合于网络编程和多线程编程,提供了一个强大的、有更多丰富类库可供利用的、逻辑一致的程序设计环境,使其可用于支持开发各类图形用户界面、多媒体、网络、数据库连接以及其他复杂功能的软件。

在 Java 的发展历程中先后出现过多个版本。它们均是由 Sun Microsystems 公司发布的。以下是 Java 简要编年史。

1995 年 6 月,Java 问世。

1995 年秋,Java 1.0。

1997 年春,Java 1.1。

1998 年夏秋,Java 1.2。更名为 Java 2 SDK 1.2。SDK 系 Software Developer's Kit 即软件开发人员工具包的英文词头缩略语。简称 Java 2 1.2。

2000 年秋,Java 2 1.3。

2002 年春,Java 2 1.4。

目前,Sun Microsystems 公司已推出了 Java 2 1.5 试用版。

当前最流行的 Java 2 版本是运行在 Java 2 编程平台上的标准版(Java 2 Platform, Standard Edition, J2SE)。它服务的对象主要是公司、科研单位、政府机构中的应用软件开发人员。此外还有两个版本:企业版与小型版。Java 2 平台企业版(Java 2 Platform, Enterprise Edition, J2EE)主要服务于企业系统开发人员,用于开发大型的、计算密集型的工程项目。Java 2 平台小型版(Java 2 Platform, Micro Edition, J2ME)主要服务于信息家电、个人数字助理等应用程序开发的需求。

本书所使用的 Java SDK 版本是 J2SE 1.4.2 版。

学习 Java 语言及编程技巧不一定非要有 C/C++ 的编程经验。然而,对于具有一些 C/C++ 编程经验的人员,在学习 Java 的过程中,将会发现 Java 与 C/C++ 有很多相似之处,并且 Java 要比 C++ 更为简洁、方便使用、不容易出错。因而,C/C++ 的编程经验将为尽快进入 Java 殿堂扫除障碍。本书不以具有 C/C++ 编程经验为前提,采用从头介绍 Java 诸多语言特性、列举大量示例、陆续展示 Java 编程技巧的方法,以使广大初学者能够易于入门、循序渐进,扎扎实实地学通、掌握 Java 语言及编程技巧。

1.2 Java 程序的开发过程

学习 Java 语言就是为了能顺利地使用 Java 的诸多优良特性,编写 Java 源程序并上机调试,以解决具体应用问题。Java 语言和其他流行的计算机程序设计语言一样,按其规定的构成和格式以及所提供的语句形式,由程序员编写程序,进行人与计算机的交互通信,实现指定的功能。Java 使用计算机能够识别和检验(通过 Java 编译器)的、类似于日常英语的专用词汇、术语及规则。它们构成了 Java 的语法。程序员使用它们编写 Java 程序时,可不必关心它们究竟是如何在计算机内部转换并实现的,而只需致力于正确地理解和使用这些词汇、术

语并严格遵循既定的规则即可。

一个 Java 程序的简要开发过程是

- (1)编写 Java 程序;
- (2)录入并编辑 Java 程序;
- (3)编译 Java 程序;
- (4)运行 Java 程序;
- (5)调试 Java 程序。

在开始编写 Java 程序之前,需要在所使用的计算机上正确地安装用于编辑、编译、运行 Java 程序的软件及相应的开发工具。本书使用当今流行的 Java 2 1.4.2 版本,在 Windows 2000 系统上进行工作。使用 Windows 的记事本来录入并编辑 Java 程序,在 MS-DOS 命令行上编译并运行 Java 程序。Windows 记事本是一个简单的文本编辑器,用以处理纯文本文件。MS-DOS 命令行使用 MS-DOS 命令。Java 编译器(名为 javac.exe 的可执行文件)对 Java 程序(通常称为 Java 源程序,放置于相应的 Java 源文件中)进行编译,得到一个相应的字节码文件。字节码文件是一个独立于平台的、专用的二进制码文件。该文件可由 Java 解释器进行逐句解释并执行 Java 程序。Java 程序的运行与所处位置与环境有关。在本地机比如 Windows 2000 系统上运行的 Java 程序,称为 Java 应用程序。在 Internet 的 Web 页面上运行的 Java 程序,称为 Java 小应用程序,简称小应用程序。它有个名称,为 applet。对于 Java 应用程序,需使用 Java 解释器(名为 java.exe 的可执行文件)来解释执行经编译生成的 Java 字节码文件。而对于小应用程序 applet,则必须通过支持 Java 标准的浏览器比如 Microsoft Internet Explorer(IE)来运行。在本书上编将集中精力于如何编写并运行 Java 应用程序。而在本书下编将有专门一章介绍如何编写并运行 Java 小应用程序。以下所述 Java 程序即指 Java 应用程序。

1.3 Java 程序的基本结构

从本节开始,将先从几个简单的 Java 程序示例入手,以使初学 Java 欲入门的读者对 Java 程序的基本结构有一个概貌性的了解。以后将由简到繁,逐步深入,陆续引出一些新的特性、规则,以解决一个个新的问题。

1.3.1 第一个 Java 程序

例 1 编写、编辑、编译并运行一个简短的 Java 程序。

```
// program Welcome.java
class Welcome
{
    public static void main(String args[])
    {
        System.out.println("Welcome to Java world!");
    }
}
```

这是个符合语法规则的 Java 程序。尽管它非常简短,但却包含了一般 Java 程序所共有

的若干特性。不妨称它为 Java 程序 Welcome。

这个 Java 程序所要实现的功能是往控制台即显示器上输出一行指定的文字信息。

下面较为细致地逐行分析这第一个 Java 程序。一共八行源代码。

程序第一行是注释行。它由“//”开始,后跟注释内容。注释内容由程序员根据需要书写。一般是为了提高程序的可读性而使用注释,故注释内容以能有助于理解程序为主。Java 编译器将忽略注释的内容。本程序中的注释仅给出了为该 Java 程序所应取的源文件名——Welcome.java。

程序第二行用于声明创建一个名为 Welcome 的新类。class 即表示类。它是 Java 的一个关键字。关键字是 Java 中具有特定含义的标识符。每个 Java 程序均系由一个到多个类所组成。类是 Java 具有面向对象特性的基本封装单元。Java 程序中的任一部分都必须包含在类中。本程序仅声明了一个类。Welcome 是为该类所起的类名。在 Java 中,选取类名需关注英文字母字符的大、小写。它们是互有区别的。故类名 Welcome 不等同于 welcome 或者 WELCOME。这是 Java 的一个特性:对英文字母字符大、小写是敏感的。

程序第三行至第八行是类 Welcome 的类定义。类定义自左花括号符“{”开始,至右花括号符“}”结束。其间的所有内容,均属于类 Welcome。它们将是类 Welcome 的成员。

程序第四行至第七行定义了类 Welcome 中唯一的、名为 main 的成员方法,或简称方法。这是本程序的主体部分。方法 main 是类 Welcome 中的一个成员。一个类所包含的方法给出了需实现的某项功能。Java 程序中的方法类似于某些流行程序设计语言比如 C 中的函数。对方法进行定义之后,可以在程序运行中对该方法进行调用。名为 main 的方法是 Java 程序的一个特殊的方法。一旦程序开始运行,Java 解释器就会自动地去寻找方法 main 并调用该方法。因此,它是 Java 程序的入口。通常在 Java 程序中,都具有方法 main 的代码。

程序第四行是方法 main 的首部。请记住它的书写形式。在本书往下相当长的篇幅里,每一个 Java 程序都将有这样一行

```
public static void main(String args[])
```

其中,public、static、void 都是 Java 关键字。关键字 public 是一个访问指示符。它规定了一个类或类成员的访问控制属性。public 意即“公共的”,在此它表明方法 main 可被类以外的代码所调用。关键字 static 从字面上来说是“静态的”,在 Java 中它指明方法 main 是一个类方法。类方法的特点是该方法允许在类的对象创建之前被调用。这对于方法 main 来说是十分必要的,因为 Java 解释器将在创建任何类对象之前调用方法 main。故方法 main 亦称为主方法。关键字 void 的字面含义是“空的”,在此它表明当调用方法 main 时不返回任何值。在方法名 main 之后用圆括号对括起来的部分是方法 main 的形式参数表。Java 中的方法与函数相似,可以是不含参数的,也可以是带有参数的。前者称为无参方法,后者称为有参方法或称带参方法。形式参数表 String args[] 表明允许方法 main 带有一个取名为 args 的形式参数,它是一个 String 类型的数组对象,可用以存储若干字符序列。这里,“[]”指示这是一个数组。数组是具有相同类型对象的集合。在运行程序时,Java 解释器如果发现在命令行中附有若干命令行参数,它就会将这些参数作为字符串,分别存放于数组对象 args 相应的诸元素中,以供程序中使用。本程序没有用到这一信息。但是,必须书写这样的形式参数表。请注意,String 的第一个字母是大写的,不能写成 string。Java 允许在方法 main 中使用形式参数表的另一种书写方式:String[] args。它与 String args[] 是数组声明时的两种等价形式。此外,数组对象名并非必须取作 args。比如,亦可写成 String

arguments[]。此时,形式参数取名为 arguments。

程序第五行至第七行是方法 main 的主体,称为方法体。main 的方法体自左花括号符“{”开始,至右花括号符“}”结束。其间的代码用以实现调用该方法时所完成的功能。在本程序中,第六行的 Java 语句是一个输出语句,这也是本程序中实现指定功能的关键语句。在语句

```
System.out.println("Welcome to Java world!");
```

中, System 是 Java 中一个预定义类的类名。Java 包含了数以百计的预定义类,在程序中恰当地使用其中的一些预定义类,可以实现各种各样的功能。这些预定义类组成 Java 2 类库,可供程序员选用。System 类定义了若干与系统相关的内容,使用它可实现对系统的访问。out 是 System 类中的一个变量,它表示与控制台相连的输出流。System.out 是一个封装控制台输出的对象。这里的控制台即指计算机的显示器。因此,控制台输出将显现在显示屏上。println 是 System.out 这一对象中内置的一个方法。println 系 print line 的缩略形式,表示将输出一行信息。System.out.println 表示对方法 println 的调用,故需使用圆括号对括起来。它实现的功能是将包含在双引号符“”之间的字符序列(不包含双引号符本身)按原样显现在显示屏上,并且回车换行,即将光标停留在下一行即新行的起始位置。注意该语句的末尾有个分号符“;”。在 Java 中,分号符是语句终止符,它是每一个语句的必要组成部分。Java 中的每一个语句都需以分号符结尾。本程序中的语句,实际上是一个方法调用语句。由于输出内容,故称输出语句。

对于上述简短的 Java 程序,可使用 Windows 系统上的“记事本”录入并编辑,然后将程序源代码保存于一个磁盘文本文件之中。称该磁盘文本文件为 Java 源文件。保存 Java 源文件需遵循一定的规则。主要有两点:一是源文件的命名规则。强烈建议源文件名应取得与包含方法 main 的类名完全一致,对于本程序而言,即为 Welcome。二是源文件必须使用 .java 为文件扩展名。这些是 Java 编译器所要求的。因此,保存本 Java 程序的源文件其全名应是 Welcome.java。假定此源文件保存于 Windows 系统的目录 C:\yao 之下。

下面需编译、运行该 Java 程序。Java 编译器在任何时候都要求并期待一个符合语法规则的完整程序。对于这样的程序,将能顺利地通过编译并可执行。否则,Java 编译器将指出编译出错信息,并期待予以改正。只有改正了程序中所有语法错误,使其通过编译,才能开始执行程序。

为能开始编译上述 Java 程序,先要退出记事本,进而使用 Windows 系统上的“命令提示符”,打开 MS-DOS 命令行窗口,并将当前目录更换为 C:\yao:

```
C:\> cd yao ↵
```

```
C:\yao>
```

这里,命令 cd 系 change directory 的词头缩略语,即改变目录。长下划线表示需从键盘输入的内容。↵表示回车。

Java 编译器是名为 javac.exe 的可执行文件。输入下述命令行以对 Java 源文件 Welcome.java 进行编译:

```
C:\yao> javac Welcome.java ↵
```

C:\yao>

如果 Java 编译器在编译中发现 Java 程序有错,则将会给出出错信息。这里没有任何信息输出,表示编译成功。此时,在当前目录 C:\yao 之下生成了一个全名为 Welcome.class 的字节码文件。该字节码文件的文件名与源文件名以及源文件中 Java 程序的类名一致,为 Welcome,而文件扩展名是 .class,以表明这是一个类文件。Java 字节码文件 Welcome.class 并非一个可执行文件。字节码是一种特殊形式的二进制代码,它属于“中间代码”,而不是其他流行程序设计语言经编译生成的机器代码。机器代码是能由计算机系统直接识别的一种二进制代码。Java 字节码文件需要由 Java 解释器(名为 java.exe 的可执行文件)来逐行解释执行。Java 解释器实现将字节码解释为本机二进制机器代码。对于本程序示例,应输入下述命令行,以获得输出结果:

C:\yao> java Welcome ↵

Welcome to Java world!

C:\yao>

请注意,在命令行中应使用 Java 字节码文件 Welcome.class 中不带文件扩展名的文件名 Welcome。Java 解释器所需要的是 Java 程序中包含方法 main 的类名,本程序即是 Welcome。如果在命令行中不慎带上了文件扩展名 .class,那么系统将会报告出错信息,从而不能获得所期待的那一行文字信息。

1.3.2 几个简单的 Java 程序

例 2 Java 程序 Welcome1

```
// program Welcome1.java
class Welcome1
{
    public static void main(String args[])
    {
        System.out.print("Welcome ");
        System.out.print("to ");
        System.out.print("Java world!\n");
    }
}
```

Java 程序 Welcome1 与例 1 中的 Java 程序 Welcome 相比较,出现两个新的元素。一个新元素是使用了 System.out 这一对象中另一个内置方法 print 的调用。对方法 println 和方法 print 的调用都可将括在双引号之间的字符序列输出到控制台。其区别是,前者在输出内容的同时自动加上了回车换行符,从而在输出字符序列之后有换行动作发生。而后者仅是输出字符序列。这样,执行本程序中的三个输出语句,其输出字符序列将连接在同一行上。第二个新元素是在第三个语句中,使用了一个特殊形式的字符 \n(记作'\n'),它是 Java 中的一个转义符,其作用是回车换行。转义符以反斜线字符“\”开始。

编译、运行 Java 程序 Welcome1

C:\yao> javac Welcome1.java ↵


```
C:\yao> java Welcome1 ↵
```

```
Welcome to Java world!
```

```
C:\yao>
```

可见,运行 Java 程序 Welcome1 获得与运行 Java 程序 Welcome 相同的输出结果。

输出语句

```
System.out.print("Java world!\n");
```

亦可用

```
System.out.println("Java world!");
```

代替。

例 3 Java 程序 Ratio 实现:分别给两个浮点类型变量 a、b 赋值,计算它们的比率,赋予浮点类型变量 ratio,并以直观形式输出结果信息。

```
// program Ratio.java
```

```
class Ratio
```

```
{
```

```
    public static void main(String args[])
```

```
    {
```

```
        double a,b,ratio;
```

```
        a=-7.5;
```

```
        b=2.0;
```

```
        ratio=a/b;
```

```
        System.out.println("a/b="+a+"/"+b+"="+ratio);
```

```
    }
```

```
}
```

编译、运行 Java 程序 Ratio

```
C:\yao> javac Ratio.java ↵
```

```
C:\yao> java Ratio ↵
```

```
a/b=-7.5/2.0=-3.75
```

```
C:\yao>
```

Java 程序 Ratio 有些新意。在方法 main 的方法体中,出现了几个新的元素。

第一个新元素是程序第六行。这是个变量声明语句,它是 main 方法体中的声明部分。在 Java 程序中会出现一些变量。对这些变量在使用前必须先行声明。double 是 Java 关键字,它表明后面的变量是双精度浮点类型,或简称双精度实型,而 a、b、ratio 均是变量名。故该语句声明变量 a、b、ratio 均为双精度实型变量。Java 编译器会为每一个变量按其所属数据类型分配相应内存单元。

第二个新元素是程序第七行至第九行。这是三个同样类型的语句——赋值语句。前两个