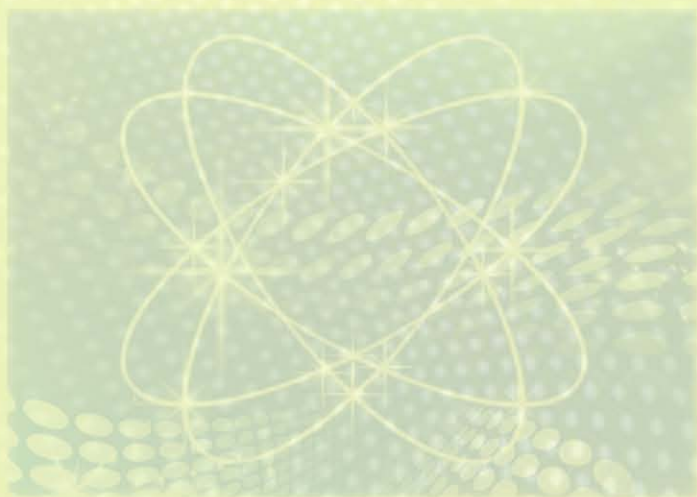


J A V A 程序设计研究

李红日 著



北京理工大学出版社

JAVA 程序设计研究

李红日 著

 **北京理工大学出版社**
BEIJING INSTITUTE OF TECHNOLOGY PRESS

版权专有 侵权必究

图书在版编目 (CIP) 数据

JAVA 程序设计研究 / 李红日著. —北京: 北京理工大学出版社, 2019. 9
ISBN 978 - 7 - 5682 - 7714 - 3

I. ①J… II ①李… III. ①JAVA 语言 - 程序设计 - 研究 IV. ①TP312. 8

中国版本图书馆 CIP 数据核字 (2019) 第 235227 号

出版发行 / 北京理工大学出版社有限责任公司

社 址 / 北京市海淀区中关村南大街 5 号

邮 编 / 100081

电 话 / (010) 68914775 (总编室)

(010) 82562903 (教材售后服务热线)

(010) 68948351 (其他图书服务热线)

网 址 / <http://www.bitpress.com.cn>

经 销 / 全国各地新华书店

印 刷 / 三河市华骏印务包装有限公司

开 本 / 787 毫米 × 1092 毫米 1/16

印 张 / 12

字 数 / 220 千字

版 次 / 2019 年 9 月第 1 版 2019 年 9 月第 1 次印刷

定 价 / 50.00 元

责任编辑 / 王玲玲

文案编辑 / 王玲玲

责任校对 / 周瑞红

责任印制 / 施胜娟

图书出现印装质量问题, 请拨打售后服务热线, 本社负责调换

前 言

Java 是由 Sun 公司于 1995 年 5 月推出的 Java 面向对象程序设计语言和 Java 平台的总称。由 James Gosling 和其同事共同研发，并在 1995 年正式推出。

Java 分为三个体系：JavaSE (J2SE) (Java 2 Platform Standard Edition, Java 平台标准版)；JavaEE (J2EE) (Java 2 Platform, Enterprise Edition, Java 平台企业版)；JavaME (J2ME) (Java 2 Platform Micro Edition, Java 平台微型版)。

Java 语言的语法与 C 语言及 C++ 语言的很接近，使大多数程序员很容易学习和使用。一方面，Java 丢弃了 C++ 中很少使用的、很难理解的、令人迷惑的那些特性，如操作符重载、多继承、自动的强制类型转换。特别地，Java 语言不使用指针，而是引用，并提供了自动废料收集功能，使程序员不必考虑内存管理的问题。

Java 语言提供类、接口和继承等原语，为简单起见，其只支持类之间的单继承，但支持接口之间的多继承，并支持类与接口之间的实现机制（关键字为 implements）。Java 语言全面支持动态绑定，而 C++ 语言只对虚函数使用动态绑定。总之，Java 语言是一个纯的面向对象程序设计语言。与那些解释型的高级脚本语言相比，Java 是高性能的。事实上，随着 JIT (Just - In - Time) 编译器技术的发展，Java 的运行速度越来越接近于 C++。

本书通过对 Java 编程语言的全面介绍，引导读者快速掌握 Java 编程语言的核心内容并学会灵活运用所学的语言知识及面向对象的编程思想。全书共分 13 个部分，内容包括 Java 语言概述、面向对象编程设计、多线程设计、Java 集合框架设计、反射、Socket 编程设计、JSP、正则表达式与 XML 操作、JavaScript、Servlet、Ajax、算法与数据结构及 Oracle 数据库的研究。

尽管作者在写作过程中付出了极大的努力，但由于水平所限，书中难免存在不足之处，恳请读者批评指正。

李红日

目 录

第一章 Java 语言概述	(1)
第一节 Java 语言诞生背景	(1)
第二节 Java 程序设计解析	(10)
第二章 面向对象编程设计的研究	(21)
第一节 接口和抽象类	(21)
第二节 重载和覆盖	(26)
第三节 类的执行顺序	(29)
第四节 字符串类	(31)
第三章 多线程设计研究	(33)
第一节 线程概念	(33)
第二节 Thread 与 Runnable	(34)
第三节 线程调度	(35)
第四节 案例	(40)
第四章 Java 集合框架设计研究	(42)
第一节 Java 的集合框架	(42)
第二节 ArrayList 与 LinkedList	(43)
第三节 集合排序	(46)
第四节 负载因子	(51)
第五章 反射的研究	(53)
第一节 Java 的类加载机制和反射机制	(53)
第二节 使用 Class 类	(55)
第三节 使用反射获得对象类型	(56)
第四节 Class 类的方法	(58)
第六章 Socket 编程的研究	(64)
第一节 网络基础	(64)
第二节 TCP 与 UDP	(65)
第三节 Socket 编程	(66)
第四节 使用 Socket 传输文件	(69)
第七章 JSP 的研究	(74)
第一节 B/S 模式与 Web 基本原理	(74)
第二节 JSP 相关知识	(75)
第三节 会话跟踪	(78)

第四节 JSP 案例	(82)
第八章 正则表达式与 XML 操作的研究	(86)
第一节 正则表达式	(86)
第二节 Java 解析 XML	(88)
第三节 用 DOM 解析 XML 文档	(90)
第四节 用 SAX 解析 XML 文档	(92)
第九章 JavaScript 的研究	(95)
第一节 框架访问	(95)
第二节 列表框级联	(98)
第三节 图片轮换	(101)
第四节 使用 jQuery 简化 DOM 操作	(104)
第十章 Servlet 的研究	(108)
第一节 Servlet 的基本架构	(108)
第二节 Servlet 的生命周期	(109)
第三节 Servlet 案例	(111)
第四节 MVC	(115)
第十一章 AJAX 的研究	(118)
第一节 AJAX 的基本原理	(118)
第二节 两种提交方式	(119)
第三节 AJAX 案例	(124)
第十二章 算法与数据结构的研究	(132)
第一节 用递归法计算 n!	(132)
第二节 用递归法解决汉诺塔问题	(134)
第三节 线性表及抽象数据类型	(137)
第四节 线性表的顺序存储	(143)
第五节 线性表的链式存储	(148)
第十三章 Oracle 数据库的研究	(162)
第一节 使用游标处理 SELECT 语句返回的数据	(162)
第二节 使用 REF 游标查看员工或部门信息	(164)
第三节 使用程序包	(165)
第四节 使用触发器监控表操作	(168)
参考文献	(170)

第一章

Java 语言概述

第一节 Java 语言诞生背景

一、Java 的历史

Java 来自 Sun 公司的 Green 的项目。为了给家用电子消费产品开发一个分布式代码系统，从而对冰箱、电视机等家用电器进行控制，Sun 公司基于 C++ 开发了一种新语言 Oak（Java 的前身）。Oak 是一种用于网络的精巧而安全的语言，Sun 公司曾以此投标一个交互式电视项目，但被 SGI 打败了。就在此时，Mark Ardreessen 开发的 Mosaic 和 Netscape 启发了 Oak 项目组成员，他们用 Java 编制了 HotJava 浏览器，得到了 Sun 公司首席执行官 ScottMcNealy 的支持，促使 Java 进军 Internet。

Java 技术包括 Java 语言和 Java Media APIs、Security APIs、Management APIs、Java Applet、Java RMI、JavaBean、JavaOS、Java Servlet、JavaServer Page 及 JDBC 等。

Java 技术的发展历程简述如下：

1990 年，Sun 公司的 James Gosling 领导的小组设计了一种平台独立的语言 Oak，主要用于为各种家用电器编写程序。

1995 年 1 月，Oak 被改名为 Java。

1995 年 5 月 23 日，Sun 公司在 Sun World'95 上正式发布 Java 和 HotJava 浏览器。

1995 年 8 月至 12 月，Netscape 公司、Oracle 公司、Borland 公司、SGI 公司、Adobe 公司、IBM 公司、AT&T 公司、Intel 公司获得 Java 许可证。

1996 年 1 月，Sun 公司宣布成立新的业务部门——JavaSoft 部，以开发、销售并支持基于 Java 技术的产品，由 Alan Baratz 任总裁。同时，推出 Java 开发工具包 JDK（Java Development Kit）1.0，为开发人员提供用来编制 Java 应用软件所需的工具。

1996 年 2 月，Sun 公司发布 Java 芯片系列，包括 PicoJava、MicroJava 和

UltraJava, 并推出 Java 数据库连接 JDBC (Java Database Connectivity)。

1996 年 3 月, Sun 公司推出 Java WorkShop。

1996 年 4 月, 微软公司、SCO 公司、苹果电脑公司、NEC 公司等获得 Java 许可证。Sun 公司宣布允许苹果电脑、惠普、日立、IBM、微软、Novell、SGI、SCO、Tandem 等公司将 Java 平台嵌入其操作系统中。

1996 年 5 月, 惠普公司、Sybase 公司获得 Java 许可证。北方电讯公司宣布把 Java 技术和 Java 微处理器应用到其下一代电话机中的计划。5 月 29 日, Sun 公司在旧金山举行第一届 JavaOne 世界 Java 开发者大会, 业界人士踊跃参加。Sun 公司在大会上推出一系列 Java 平台新技术。

1996 年 8 月, Java WorkShop 成为 Sun 公司通过互联网提供的第一个产品。

1996 年 9 月, Addison - Wesley 和 Sun 公司推出 Java 虚拟机规范和 Java 类库。

1996 年 10 月, 德州仪器等公司获得 Java 许可证。Sun 公司提前完成 JavaBean 规范并发布, 同时, 发布第一个 Java JIT (Just - In - Time) 编译器, 并打算在 Java WorkShop 和 Solaris 操作系统中加入 JIT。10 月 29 日, Sun 公司发布 Java 企业计算技术, 包括 JavaStation 网络计算机、65 家公司发布的 85 个 Java 产品及应用、7 个新的 Java 培训课程及 Java 咨询服务、基于 Java 的 Solstice 互联网邮件软件、新的 Java 开发者支持服务、HotJava Views 演示、Java Tutor、Java Card API 等。Sun 公司宣布完成 Java Card API 规范, 这是智能卡使用的第一个开放 API。Java Card API 规范将把 Java 能力赋予全世界亿万张智能卡。

1996 年 11 月, IBM 公司获得 JavaOS 和 HotJava 许可证。Novell 公司获得 Java WorkShop 许可证。Sun 公司和 IBM 公司宣布双方就提供 Java 化的商业解决方案达成一项广泛协议, IBM 公司同意建立第一个 Java 检验中心。

1996 年 12 月, Xerox 等公司获得 Java 或 JavaOS 许可证。Sun 公司发布 JDK1.1、Java 商贸工具包、JavaBean 开发包及一系列 Java APIs; 推出一个新的 JavaServer 产品系列, 其中包括 Java Web Server、Java NC Server 和 JavaServer Toolkit。Sun 公司发布 100% 纯 Java 计划, 得到百家公司的支持。

1997 年 1 月, SAS 等公司获得 Java 许可证。Sun 公司交付完善的 JavaBean 开发包, 这是在确定其规范后不到 8 个月内完成的。

1997 年 2 月, Sun 公司和 ARM 公司宣布同意使 JavaOS 运行在 ARM 公司的 RISC 处理器架构上。Informix 公司宣布在其 Universal Server 和其他数据库产品上支持 JDK1.1。Netscape 公司宣布其 Netscape Communicator 支持所有 Java 化的应用程序和核心 API。

1997 年 3 月, 惠普公司获得 Java WorkShop 许可证, 用于其 HP - UX 操作系统。西门子、AG 公司等获得 Java 许可证。日立半导体公司、Informix 公司等获得 JavaOS 许可证。Novell 公司获得 Java Studio 许可证。Sun 公司发售了 JavaOS

1.0 操作系统，这是一种在微处理器上运行 Java 环境的最小、最快的方法，可提供给 JavaOS 许可证持有者使用；还发售了 HotJava Browser 1.0，这是一种 Java 浏览器，可以方便地按需编制专用的信息应用软件，如客户自助台和打上公司牌号的网络应用软件。1996 年 6 月，Sun 公司发布 JSP1.0，同时推出 JDK1.3 和 Java Web Server 2.0。1999 年 11 月，Sun 公司发布 JSP1.1，同时推出 JSWDK1.0.1 和 Java Servlet 2.2。2000 年 9 月，Sun 公司发布 JSP1.2 和 Java Servlet 2.3 API。

二、Java 的现状

Java 是 Sun 公司推出的新一代面向对象程序设计语言，特别适用于 Internet 应用程序开发，它的平台无关性直接威胁到 Wintel 的垄断地位，这表现在以下几个方面：计算机产业的许多大公司购买了 Java 许可证，包括 IBM、苹果、DEC、Adobe、SiliconGraphics、惠普、Oracle、东芝及微软。这一点说明，Java 已得到了业界的认可。众多的软件开发商开始支持 Java 软件产品。例如 Inprise 公司的 JBuilder、Sun 公司自己做的 Java 开发环境 JDK 与 JRE。Sysbase 公司和 Oracle 公司均已支持 HTML 和 Java。

Intranet 正在成为企业信息系统最佳的解决方案，而其中 Java 将发挥不可替代的作用。Intranet 的目的是将 Internet 用于企业内部的信息类型，它的优点是便宜、易于使用和管理。用户不管使用何种类型的机器和操作系统，界面是统一的 Internet 浏览器，而数据库、Web 页面、Applet、Servlet、JSP 则储存在 Web 服务器上，无论是开发人员还是管理人员或是用户，都可以受益于该解决方案。

三、Java 的特点

1. Java 语言的优点

Java 语言是一种优秀的编程语言。它最大的优点就是与平台无关，在 Windows 9X、Windows NT、Solaris、Linux、MacOS 及其他平台上，都可以使用相同的代码。“一次编写，到处运行”的特点，使其在互联网上被广泛采用。

由于 Java 语言的设计者们十分熟悉 C++ 语言，所以，在设计时很好地借鉴了 C++ 语言。可以说，Java 语言是一种比 C++ 语言“还面向对象”的编程语言。Java 语言的语法结构与 C++ 语言的语法结构十分相似，这使得 C++ 程序员学习 Java 语言更加容易。

当然，如果仅仅是对 C++ 的改头换面，那么就不会有今天的“Java 热”了。Java 语言提供的一些有用的新特性，使得使用 Java 语言比 C++ 语言更容易写出“无错代码”。

这些新特性包括：

(1) 提供了对内存的自动管理，程序员无须在程序中进行分配、释放内存，那些可怕的内存分配错误不会再打扰设计者了。

(2) 去除了 C++ 语言中的令人费解、容易出错的“指针”，用其他方法进行弥补。

(3) 避免了赋值语句（如 $a = 3$ ）与逻辑运算语句（如 $a == 3$ ）的混淆。

(4) 取消了多重继承这一复杂的概念。

Java 语言的规范是公开的，可以在 <http://www.sun.com> 上找到它，阅读 Java 语言的规范是提高技术水平的好方法。

2. Java 语言的关键特性

Java 语言有许多有效的特性，吸引着程序员们，最主要的有以下几个：

(1) 简洁有效。

Java 语言是一种相当简洁的“面向对象”程序设计语言。Java 语言省略了 C++ 语言中所有的难以理解、容易混淆的特性，例如头文件、指针、结构、单元、运算符重载、虚拟基础类等。它更加严谨、简洁。

(2) 可移植性。

对于一个程序员而言，写出来的程序如果不需修改就能够同时在 Windows、MacOS、UNIX 等平台上运行，简直就是梦寐以求的好事！而 Java 语言就让这个原本遥不可及的事越来越近了。使用 Java 语言编写的程序，只要做较少的修改，甚至有时根本不需修改，就可以在不同平台上运行了。

(3) 面向对象。

“面向对象”是软件工程学的一次革命，大大提升了人类的软件开发能力，是一个伟大的进步，是软件发展的一个重大的里程碑。在过去的 30 年间，“面向对象”有了长足的发展，充分体现了其自身的价值，到现在已经形成了一个包含了“面向对象的系统分析”“面向对象的系统设计”“面向对象的程序设计”的完整体系。所以作为一种现代编程语言，是不能够偏离这一方向的，Java 语言也不例外。

3. 解释型

Java 语言是一种解释型语言，相对于 C/C++ 语言来说，用 Java 语言写出来的程序效率低，执行速度慢。但它正是通过在不同平台上运行 Java 解释器，对 Java 代码进行解释，来实现“一次编写，到处运行”的宏伟目标的。为了达到目标，牺牲效率还是值得的，况且，现在的计算机技术日新月异，运算速度也越来越快，用户是不会感到太慢的。

4. 适合分布式计算

Java 语言具有强大的、易于使用的联网能力，非常适合开发分布式计算的程序。Java 应用程序可以像访问本地文件系统那样通过 URL 访问远程对象。使用 Java 语言编写 Socket 通信程序十分简单，使用它比使用任何其他语言都简单。并且它还十分适用于公共网关接口（CGI）脚本的开发，另外，还可以使用 Java 小应用程序（Applet）、Java 服务器页面（Java Server Page, JSP）、Servlet 等手段来

构建更丰富的网页。

5. 拥有较好的性能

正如前面所述，由于 Java 是一种解释型语言，所以它的执行效率相对就会慢一些，但由于 Java 语言采用了两种手段，其性能较好。

(1) Java 语言源程序编写完成后，先使用 Java 伪编译器进行伪编译，将其转换为中间码（也称为字节码）后再解释。

(2) 提供了一种“准实时”（Just-in-Time, JIT）编译器，当需要更快的速度时，可以使用 JIT 编译器将字节码转换成机器码，然后将其缓冲下来，这样速度就会更快。

6. 健壮、防患于未然

Java 语言在伪编译时，做了许多早期潜在问题的检查，并且在运行时又做了一些相应的检查，可以说是一种最严格的“编译器”。它的这种“防患于未然”的手段将许多程序中的错误扼杀在“摇篮”之中。经常有许多在其他语言中必须通过运行才会暴露出来的错误，在 Java 中在编译阶段就被发现了。

另外，在 Java 语言中还具备了許多保证程序稳定、健壮的特性，有效地减少了错误，这样使 Java 应用程序更加健壮。

7. 具有多线程处理能力

线程，是一种轻量级进程，是现代程序设计中必不可少的一种特性。多线程处理能力使程序具有更好的交互性、实时性。Java 在多线程处理方面性能超群，具有让设计者惊喜的强大功能，并且在 Java 语言中进行多线程处理很简单。

8. 具有较高的安全性

由于 Java 语言在设计时，在安全性方面考虑很仔细，做了许多探究，使得 Java 语言成为目前最安全的一种程序设计语言。尽管 Sun 公司曾经许诺过：“通过 Java 可以轻松构建出防病毒、防黑客的系统。”但“世界上没有绝对的安全”这一真理是不会因为许诺而失灵验的。就在 JDK（Java Development Kit）1.0 发布不久后，美国 Princeton（普林斯顿）大学的一组安全专家发现了 Java 1.0 安全特性中的第一例错误。从此，Java 安全方面的问题开始被关注。不过至今所发现的安全隐患都微不足道，并且 Java 开发组还宣称，他们对系统安全方面的 bug 非常重视，会对这些被发现的 bug 立即进行修复。同时，由于 Sun 公司开放了 Java 解释器的细节，所以有助于通过各界力量，共同发现、防范、消除这些安全隐患。

9. 是一种动态语言

Java 是一种动态的语言，这表现在以下两个方面：

- (1) 在 Java 语言中，可以简单、直观地查询运行时的信息；
- (2) 可以将新代码加入一个正在运行的程序中。

10. 是一种中性结构

“Java 编译器生成的是一种中性的对象文件格式。”也就是说，Java 编译器通过伪编译后，将生成一个与任何计算机体系系统无关的“中性”的字节码。这种中性结构其实并不是 Java 首创的，在 Java 出现之前，UCSD Pascal 系统就已在一种商业产品中做到了这一点，另外，在 UCSD Pascal 之前也有这种方式的先例，在 Niklaus Wirth 实现的 Pascal 语言中就采用了这种降低一些性能，换取更好的可移植性和通用性的方法。

Java 的这种字节码经过了許多精心的设计，使得其能够很好地兼容于当今大多数流行的计算机系统，在任何机器上都易于解释，易于动态翻译成机器代码。

四、Java 虚拟机 (JVM)

Java 虚拟机 (JVM) 是可运行 Java 代码的假想计算机。只要根据 JVM 规范描述将解释器移植到特定的计算机上，就能够保证经过编译的任何 Java 代码在该系统上运行。如图 1-1 所示。

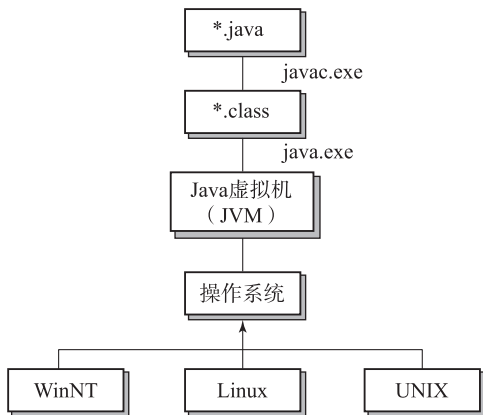


图 1-1 Java 虚拟机

从图 1-1 中不难看出 Java 实现可移植性的原因，只要在操作系统 (WinNT、Linux、UNIX) 中植入 JVM，Java 程序就具有可移植性，也符合 Sun 公司提出的口号“WriteOnce, Run Anywhere” (“一次编写，处处运行”)。

目前，Java 技术的架构包括以下三个方面：

(1) J2EE (Java 2 Platform Enterprise Edition, 企业版)，是以企业为环境而开发应用程序的解决方案。

(2) J2SE (Java 2 Platform Stand Edition, 标准版)，是桌面开发和低端商务应用的解决方案。

(3) J2ME (Java 2 Platform Micro Edition, 小型版)，是致力于消费产品和嵌入式设备的最佳解决方案。

五、JDK 的安装及环境变量的配置

要开发 Java 程序，首先必须要配置好环境变量，而 Java 的运行环境的配置比较麻烦，相信有些读者也会有这种体会，下面来看一下 JDK 的安装过程。在这里 JDK 选用的是 J2SDK1.4.2 版本。

安装分为两个步骤：

(1) 首先准备好 JDK 的安装文件 `j2sdk-1_4_0_03-windows-i586`，如图 1-2 和图 1-3 所示。



图 1-2 启动 JDK 安装程序

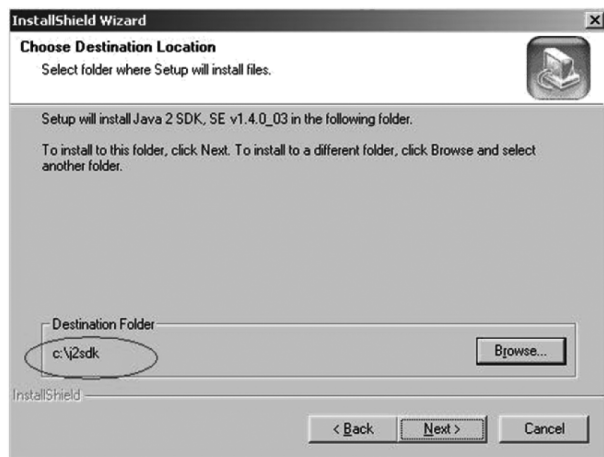


图 1-3 将安装路径设置为 `c:\j2sdk`

(2) 配置环境变量 `path`。

之后根据默认的设置安装即可。

从图 1-1 可以看出，在编译 Java 程序时，需要用到 `javac` 这个命令，执行

Java 程序时，需要 java 这个命令，而这两个命令并不是 Windows 自带的，所以使用它们时，需要配置好环境变量，这样就可以在任何目录下使用这两个命令了。

设置环境变量的步骤为：

①在“我的电脑”上右击，选择“属性”→“高级”→“环境变量”→“Path”，如图 1-4 所示。



图 1-4 系统环境变量的配置

②在 Path 后面加上 c:\j2sdk\bin，c:\j2sdk 是安装 JDK 的路径，如图 1-5 所示。

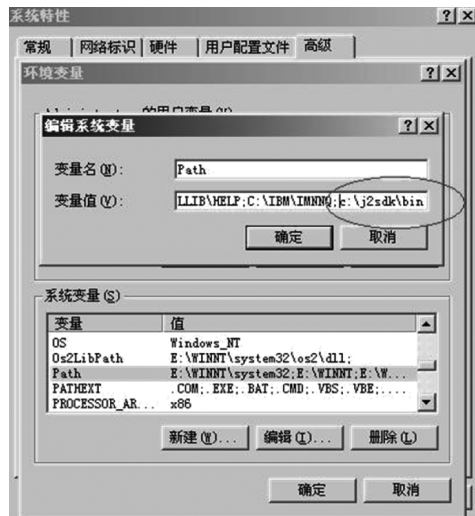


图 1-5 添加环境变量

这样就可以在任何目录下使用 javac 和 java 这两个命令了。

六、编写第一个 Java 程序

Java 程序分为两种形式：一种是网页上使用的 Java Applet 程序（Java 小程序），另一种是 Java Application 程序（即 Java 应用程序），本书主要讲解的是 Java Application 程序。

范例：Hello.java

```
01 public class Hello
02 {
03     //是程序的起点,所有程序由此开始运行
04     public static void main(String args[])
05     {
06         //此语句表示向屏幕上打印"Hello World!"字符串
07         System.out.println("Hello World!");
08     }
09 }
```

将上面的程序保存为 Hello.java 文件，并在命令行中输入“javac Hello.java”，没有错误后，输入“java Hello”。

输出结果：

```
Hello World!
```

程序说明：

在所有的 Java Application 程序中，所有程序都是从 public static void main (String args[]) 开始运行的，刚接触的读者可能会觉得有些难记，在后面的章节中会详细给读者讲解 main 方法的各个组成部分。

上面的程序如果暂时不明白，也没有关系，读者只要将程序都输入，之后按照步骤编译、执行就可以了。这里只是让读者对 Java Application 程序有初步印象，因为以后所有的内容都将围绕 Java Application 程序进行。

七、classpath 的指定

Java 中可以使用 set classpath 命令指定 java 类的执行路径。下面通过一个实验来了解 classpath 的作用，假设这里的 Hello.class 类位于 C 盘下。

在 D 盘下的命令行窗口中执行下面的指令：

```
set classpath = c:
```

之后在 D 盘根目录下执行 java Hello 程序，如图 1-6 所示。

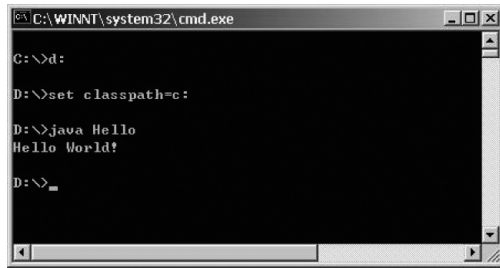


图 1-6 输出结果

由上面的输出结果可以发现，虽然在 D 盘中并没有 Hello.class 文件，但却也可以用 java Hello 执行 Hello.class 文件，之所以会有这种结果，就是因为 D 盘中使用了 set classpath 命令，将类的查找路径指向了 C 盘，所以，在运行时，会从 C 盘下开始查找。

第二节 Java 程序设计解析

本节介绍了修饰符、关键字及一些基本的数据类型。通过简单的范例，让读者了解到检测与提高程序可读性的方法，以培养读者正确的程序编写观念与习惯。

一、一个简单的例子

首先来看一个简单的 Java 程序。在介绍程序的内容之前，先简单回顾一下第一节讲解的例子，之后再来看下面这个程序，试试能否看出它是在做些什么事情。

代码范例 1.1:

```
01 //TestJava1.1.java,Java 的简单范例
02 public class TestJava1.1
03 {
04     public static void main(String args[])
05     {
06         int num;        //声明一个整型变量 num
07         num = 3;        //将整型变量赋值为 3
08         //输出字符串,这里用 "+" 号连接变量
09         System.out.println("这是数字" + num);
10         System.out.println("我有" + num + "本书!");
11     }
12 }
```

输出结果:

```
这是数字 3
我有 3 本书!
```

从上面的输出结果中可以看出 `System.out.println()` 的作用, 就是输出括号内所包含的文字, 至于 `public`、`class`、`static`、`void` 这些关键字的意思, 将在以后的章节中再做更深入一层的探讨。

程序说明:

(1) 第 1 行为程序的注释, Java 语言的注释是以 “//” 标志开始的, 注释有助于对程序的阅读与检测, 被注释的内容在编译时不会被执行。

(2) 第 2 行 `public class TestJava1.1` 中的 `public` 与 `class` 是 Java 的关键字, `class` 为 “类” 的意思, 后面接上类名称, 在本程序中取名为 `TestJava1.1`。`public` 则是用来表示该类为公有, 也就是在整个程序里都可以访问到它。

需要注意的是, 如果将一个类声明成 `public`, 则也要将文件名称取成和这个类一样的名称, 如图 1-7 所示。本例中的文件名为 `TestJava1.1.java`, 而 `public` 之后所接的类名称也为 `TestJava1.1`。也就是说, 在一个 Java 文件里, 最多只能有一个 `public` 类, 否则, 文件便无法命名。

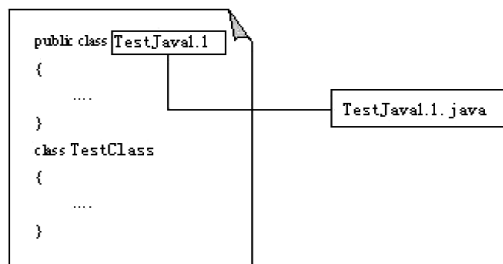


图 1-7 如果将类声明成 `public`, 则也要将文件名称取成和这个类一样的名称

(3) 第 4 行 `public static void main(String args[])` 为程序运行的起点。第 4 ~ 10 行的功能类似于一般程序语言中的函数 (function), 但在 Java 中称为 `method` (方法), 因此 C 语言里的 `main()` 函数 (主函数), 在 Java 中则被称为 `main()method` (主方法)。

(4) `main()method` 的主体 (body) 从第 5 行的左大括号 “{” 起, 到第 11 行的右大括号 “}” 为止。每一个独立的 Java 程序一定要有 `main()method` 才能运行, 因为它是程序开始运行的起点。

(5) 第 6 行 “`int num;`” 的作用是声明 `num` 为一个整数类型的变量。在使用变量之前, 必须先声明其类型。

(6) 第 7 行 “`num = 3;`” 为一赋值语句, 即把整数 2 赋给存放整数的变量 `num`。