



世纪高职高专规划教材
高等职业教育规划教材编委会专家审定

JAVA YUYAN CHENGXU SHEJI

Java语言程序设计

主 编	曹大有	臧芝玉
主 审	季昌武	
副主编	高晓黎	王 盟 宫文颖



北京邮电大学出版社
www.buptpress.com

21 世纪高职高专规划教材

高等职业教育规划教材编委会专家审定

Java 语言程序设计

主 编 曹大有 臧芝玉

主 审 季昌武

副主编 高晓黎 王 盟 宫文颖

北京邮电大学出版社

· 北 京 ·

内 容 简 介

本书根据普通高等教育“十一五”国家级规划教材的指导精神和高职高专学生的特点,从基础知识入手,结合实例讲解 Java 语言的编程技术和面向对象程序设计技术。书中首先介绍 Java 语言的发展历史和基础知识,然后借助于 Java 语言讲述了面向对象程序设计的基本原则和特点,并把这些原则和特点融入具体的 Java 程序中。最后介绍了 Java 编程的必备工具,这里包括常用类库、GUI、Swing 组件、网络编程、JDBC 编程、Servlet 编程、JSP 和 JavaBean 编程等。

全书内容丰富,结构清晰,所有例题均在 JDK 6 上调试通过。本书的每章都有项目实训和习题,以加深学生对知识的理解,巩固各阶段的学习内容。

本书可以作为高职高专、成人高等院校计算机及其相关专业的教材和教学参考书,也可作为广大 Java 程序设计爱好者的入门教材,对其他 Java 学习者也有一定的帮助。

图书在版编目(CIP)数据

Java 语言程序设计/曹大有,臧芝玉主编. —北京:北京邮电大学出版社,2009
ISBN 978-7-5635-2041-1

I. J… II. ①曹…②臧… III. JAVA 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2009)第 114902 号

书 名: Java 语言程序设计

作 者: 曹大有 臧芝玉

责任编辑: 王晓丹

出版发行: 北京邮电大学出版社

社 址: 北京市海淀区西土城路 10 号(邮编:100876)

发 行 部: 电话: 010-62282185 传真: 010-62283578

E-mail: publish@bupt.edu.cn

经 销: 各地新华书店

印 刷: 北京忠信诚胶印厂

开 本: 787 mm×1 092 mm 1/16

印 张: 16.75

字 数: 414 千字

印 数: 1—3 000 册

版 次: 2009 年 7 月第 1 版 2009 年 7 月第 1 次印刷

ISBN 978-7-5635-2041-1

定 价: 29.00 元

• 如有印装质量问题,请与北京邮电大学出版社发行部联系 •

编 委 会

主 编：曹大有(湖北鄖阳师范高等专科学校)

臧芝玉(辽宁大连商务职业学院)

主 审：季昌武(山东信息职业技术学院)

副主编：高晓黎(山东水利职业学院)

王 盟(山东东营职业学院)

宫文颖(辽宁信息职业技术学院)

委 员：(排名不分先后)

周天宏(湖北鄖阳师范高等专科学校)

魏本昌(湖北鄖阳师范高等专科学校)

刘耀钦(湖北鄖阳师范高等专科学校)

李建利(湖北三峡职业技术学院)

李成银(荆楚理工学院)

王 喻(湖北鄖阳师范高等专科学校)

前 言

一、关于本书

本书是根据普通高等教育“十一五”国家级规划教材的指导精神而编写的。

目前,全国各地高职高专院校普遍扩招,高职高专学生人数迅速增长,这也给他们的就业带来了巨大的压力。而当前高职高专学生的就业情况不容乐观,究其原因,所用教材与实际应用脱离是一大主要因素。针对现有教材质量较差、品种单一、版本陈旧、实用性和可操作性不强等原因,肩负着应用型人才培养的高职高专院校急需一系列符合当前教学改革需要的教材。

Java 语言是一种面向对象的网络编程语言,其强大的网络功能和其他语言难以比拟的。Java 语言特别适合开发 Internet 应用程序。Java 程序具有与硬件无关、与平台无关、可靠性高、安全稳定和支持多线程等特点,在当今的计算机和网络技术中占据重要的地位。

以网络为中心的计算机是当今计算机发展的主流,网络程序设计已经成为应用程序设计的主体。Java 语言程序设计是网络应用的重要组成部分。此课程是计算机应用、计算机软件技术、计算机网络技术应用与服务专业学生的必修课。此课程与程序设计基础、网络数据库及网页制作课程相互联系,相互补充。它们分别给读者提供基于 Web 的 C/S 程序设计方法和技能,面向对象程序设计方法,网络数据库的设计和管理及网页制作技巧的学习。读者能够利用 Java 语言的程序设计将网络数据库、动态网页信息有机地结合起来,使技能得到综合、系统的训练和培养,成为社会专用人才。

二、本书结构

全书的安排如下。

第 1 章:Java 程序简介。主要包括 Java 语言的发展历史和特点。

第 2 章:Java 语言基础。主要包括数据类型、常量和变量、运算符和表达式以及流程控制结构等。

第 3 章:抽象、封装与类。主要包括面向对象程序设计的类的定义和类的设计及实现。

第 4 章:继承与多态。主要包括继承、多态、重载、接口和包的概念。

第 5 章:Java 常用工具类的介绍。主要包括基础类库、Applet 类、数组、向量、字符串类和异常类。

第 6 章:Java 的多线程机制与实现。主要包括多线程基础知识、创建线程、线程间通

信和线程的同步等。

第7章:图形用户界面设计与实现。主要包括图形坐标、图形绘制、文本、属性设置、组件介绍、布局设计、菜单设计、对话框设计等。

第8章:Java语言的高级编程。主要包括输入/输出的基础知识、使用URL定位资源、Socket通信、数据报通信、JDBC编程、Servlet编程、JSP编程和JavaBean编程等。

三、本书特点

本书系统、全面地研究和借鉴了国外相关教材先进的教学方法,结合国内同级高职高专院校教学实际和先进的教学成果,根据教育部“十一五”国家级规划教材应用型高职高专教育的指导思想编写,具有实用性和可操作性,与时俱进,与当前就业市场结合得更加紧密。

本书深入浅出地介绍了Java程序设计语言。语言通俗,系统性强,内容丰富、实用,文字叙述简明易懂,注重实用性和可操作性,使读者能轻松而快速地学好Java语言的语法知识及应用。为了巩固对每章的学习,每章最后都有实训项目和一定数量精心设计的练习题。

四、适用对象

本书在结构和内容的安排上注重深入浅出、循序渐进。可作为高职高专学校计算机专业的教材,也可作为Java开发人员的读物或参考资料。

本书第1章由魏本昌编写,第2章由曹大有编写,第3章由高晓黎编写,第4章由季昌武编写,第5章由刘耀钦编写,第6章由臧芝玉编写,第7由宫文颖编写,第8章由王盟编写。全书由曹大有统稿并校对。

由于编写时间仓促,编者水平有限,书中如有疏漏和不足之处,敬请各位读者批评指正。

编 者

目 录

第 1 章 Java 程序简介	(1)
1.1 Java 语言的发展历史	(1)
1.2 Java 虚拟机及工作原理	(2)
1.2.1 Java 虚拟机	(2)
1.2.2 垃圾内存自动回收机制	(3)
1.2.3 代码安全性检查机制	(3)
1.3 Java 程序的开发环境及配置	(4)
1.3.1 J2SDK 的下载	(5)
1.3.2 J2SDK 的安装与配置	(5)
1.4 Java 程序的种类及简介	(8)
1.4.1 Java Application	(8)
1.4.2 JavaApplet	(11)
1.4.3 Servlet 的简介与开发环境	(15)
1.4.4 JSP 的提供与 Servlet 的比较	(15)
1.5 Java 语言的特点	(16)
实训 1	(18)
习题 1	(20)
第 2 章 Java 语言基础	(21)
2.1 Java 程序的构成	(21)
2.2 数据类型、变量和常量	(22)
2.2.1 数据类型	(22)
2.2.2 标识符、变量和常量	(23)
2.3 表达式	(28)
2.3.1 赋值与类型转换	(28)
2.3.2 算术运算	(29)
2.3.3 关系运算	(31)
2.3.4 逻辑运算	(32)
2.3.5 位运算	(34)
2.3.6 其他运算符	(34)

2.3.7 优先级与结合性	(35)
2.3.8 注释	(35)
2.4 控制语句	(36)
2.4.1 结构化程序设计	(36)
2.4.2 分支语句	(36)
2.4.3 循环语句	(39)
2.4.4 跳转语句	(42)
2.5 应用示例	(43)
实训 2	(47)
习题 2	(48)
第 3 章 抽象、封装与类	(50)
3.1 抽象与封装	(50)
3.1.1 抽象	(50)
3.1.2 封装	(51)
3.2 Java 类的设计	(53)
3.2.1 Java 的系统类	(53)
3.2.2 自定义类的设计	(54)
3.2.3 对象创建与类的构造函数	(55)
3.3 类的修饰符	(59)
3.3.1 abstract	(59)
3.3.2 final	(61)
3.4 域和方法	(62)
3.5 访问控制符	(65)
3.6 应用示例	(67)
实训 3	(71)
习题 3	(76)
第 4 章 继承与多态	(79)
4.1 继承的概念	(79)
4.2 Java 的继承	(79)
4.2.1 类的派生	(80)
4.2.2 域的继承与隐藏	(80)
4.2.3 方法的继承与覆盖	(83)
4.2.4 this 和 super	(84)
4.3 Java 的多态	(87)
4.4 Java 的重载	(87)
4.5 构造方法的继承与重载	(88)
4.6 包的创建与使用	(90)

4.6.1 包的创建	(90)
4.6.2 包的使用	(91)
4.7 接口	(93)
4.7.1 接口和多重继承	(93)
4.7.2 接口的声明	(94)
4.7.3 接口的实现	(94)
4.8 应用示例	(98)
实训 4	(101)
习题 4	(101)
第 5 章 Java 常用工具类的介绍	(104)
5.1 Java 语言基础类库	(104)
5.1.1 Object 类	(104)
5.1.2 基本数据类型类	(105)
5.1.3 Math 类与 System 类	(107)
5.2 Applet 类与 Applet 小程序	(110)
5.2.1 Applet 小程序的工作原理	(110)
5.2.2 Applet 类	(110)
5.2.3 HTML 文件参数的传递	(112)
5.3 Java 的数组	(113)
5.3.1 一维数组的声明和初始化	(114)
5.3.2 二维数组	(115)
5.3.3 多维数组	(116)
5.4 Java 的向量	(116)
5.4.1 向量与数组	(116)
5.4.2 向量类及其方法、属性	(116)
5.5 Java 的字符串	(119)
5.5.1 String 类	(119)
5.5.2 StringBuffer 类	(122)
5.5.3 Java 应用程序的命令行参数	(124)
5.6 Java 的异常处理	(125)
5.6.1 异常与异常类	(125)
5.6.2 异常的抛出	(127)
5.6.3 异常的处理	(129)
5.7 应用示例	(130)
实训 5	(134)
习题 5	(135)

第 6 章 Java 的多线程机制与实现	(136)
6.1 线程的概念与生命周期	(136)
6.1.1 程序、进程与线程	(136)
6.1.2 Java 中的线程	(137)
6.2 线程的创建	(137)
6.2.1 通过 Thread 类创建线程	(138)
6.2.2 通过 Runnable 接口实现线程	(139)
6.3 线程的状态与控制	(140)
6.3.1 线程的生命周期与状态	(140)
6.3.2 线程的方法与控制	(142)
6.4 线程的优先级与调度	(145)
6.5 线程的同步	(147)
6.6 线程的交互	(150)
6.7 应用示例	(153)
实训 6	(158)
习题 6	(159)
第 7 章 图形用户界面设计与实现	(160)
7.1 图形用户界面创建	(160)
7.1.1 屏幕坐标系	(160)
7.1.2 建立窗口的 Frame 类	(160)
7.2 用户自定义成分的实现	(161)
7.2.1 绘制图形	(162)
7.2.2 显示文字	(163)
7.2.3 控制颜色	(163)
7.2.4 显示图像	(165)
7.2.5 实现动画	(166)
7.3 Java 的事件处理机制	(167)
7.3.1 概念	(167)
7.3.2 可用的事件监听者和事件种类	(168)
7.3.3 事件及其响应	(170)
7.3.4 Java 中事件处理方式	(171)
7.4 Java 容器的布局设计	(172)
7.4.1 BorderLayout	(172)
7.4.2 FlowLayout	(173)
7.4.3 CardLayout	(174)
7.4.4 GridLayout	(176)
7.5 常用 AWT 组件介绍	(177)
7.5.1 标签、按钮	(178)

7.5.2 文本框、文本域	(180)
7.5.3 复选按钮、单选按钮和列表框	(181)
7.5.4 滚动条、画布和 Panel	(186)
7.6 常用 Swing 组件的介绍	(188)
7.6.1 JApplet	(190)
7.6.2 JButton	(190)
7.6.3 JSlider	(191)
7.6.4 JPasswordField	(192)
7.6.5 JTabbedPane	(193)
7.7 菜单的定义与使用	(195)
7.8 应用示例	(197)
7.8.1 动作事件应用举例	(197)
7.8.2 选项事件应用举例	(198)
7.8.3 调整事件应用举例	(200)
7.8.4 键盘事件应用举例	(201)
7.8.5 鼠标事件应用举例	(201)
实训 7	(202)
习题 7	(203)
第 8 章 Java 语言的高级编程	(205)
8.1 网络编程	(205)
8.1.1 基于 Socket 类和 ServerSocket 类的网络编程	(205)
8.1.2 基于 UDP 的网络编程	(209)
8.1.3 DatagramSocket 和 DatagramPacket 类	(210)
8.1.4 基于 URL 类的网络编程	(213)
8.2 数据库编程	(218)
8.2.1 JDBC 编程技术概述	(218)
8.2.2 使用 JDBC 访问数据库	(221)
8.2.3 查询数据	(224)
8.2.4 更新数据	(225)
8.2.5 添加数据	(226)
8.2.6 删除数据	(227)
8.3 流式输入/输出与文件处理编程	(227)
8.3.1 输入/输出类库	(227)
8.3.2 标准输入/输出	(228)
8.3.3 文件操作	(229)
8.4 Servlet 编程	(232)
8.4.1 Servlet 概述	(232)
8.4.2 在 Resin 运行 Servlet	(234)
8.4.3 Servlet 的常用类及接口	(235)

8.5 JSP 编程	(239)
8.5.1 JSP 概述	(239)
8.5.2 JSP 的运行过程	(240)
8.5.3 JSP 指令与隐藏对象	(241)
8.6 JavaBean 编程	(243)
8.6.1 JavaBean 概述	(243)
8.6.2 JavaBean 设计目标及其实现手段	(244)
8.6.3 JavaBean 的属性、事件和方法	(244)
8.6.4 创建一个 JavaBean	(245)
8.6.5 JavaBean 程序结构要点	(246)
8.7 应用示例	(247)
实训 8	(250)
习题 8	(253)
参考文献	(254)

第 1 章

Java 程序简介

Java 是一种跨平台的、适合于分布式计算机环境的面向对象程序设计语言,同时也是一种开发环境和应用环境。目前 Java 还处在快速的发展当中,新的特性和应用仍在不断出现。

1.1 Java 语言的发展历史

Java 是一种计算机程序设计语言,既可以用来编写嵌入到网页文件中运行的 Java Applet 小应用程序,也可以用来编写独立运行的 Java Application 应用程序。它是目前最流行的程序设计语言之一。

1991 年,美国 Sun Microsystems 公司成立 First Person 研究小组,用以开发手机、PDA 以及机顶盒等消费类电子产品的应用程序。消费类电子产品种类繁多,即使是同一类消费电子产品所采用的处理芯片和操作系统也不相同,存在着跨平台的问题。Sun 公司的研究人员最先考虑是否可以采用 C++ 语言来编写消费类电子产品的应用程序。经实验表明,对于消费类电子产品而言,C++ 语言因过于复杂和庞大而不适合,安全性也不令人满意。于是,由 James Gosling 领导的研究小组决定着手设计和开发一种新的程序语言,称之为 Oak。该语言采用了许多 C 语言的语法,提高了安全性并且是面向对象的语言,遗憾的是 Oak 语言并没有被 Sun 公司和消费类家电公司所接受。

1995 年,随着因特网(Internet)的兴起,人们迫切需要一个小巧的、经得起考验的、跨平台的语言来开发网络应用程序。Sun 公司发现 Oak 语言所具有的跨平台、面向对象、安全性高等特点非常符合因特网的需要,于是改进了该语言的设计,使之能适应网络程序的开发。

据说印度尼西亚有一个著名的生产咖啡的岛屿叫 Java,其咖啡也叫 Java,该语言的研究人员经常在 Sun 公司门口的咖啡店饮用这种咖啡,并一起讨论问题。于是,研究人员给这种新的开发语言起名为 Java,其寓意是为世人端上一杯香浓醇厚的热咖啡。最终,Sun 公司给该语言取名为 Java 语言,造就了一代成功的编程语言环境。

1996 年,Java 编译器的第一版发布并发放许可。1997 年,Sun 公司推出 Java 1.1(JDK 1.1),国际标准化组织批准 Java 规范。在 JDK 1.1 的时代,主要用于一般消费性电子产品等,其嵌入式系统是 Personal Java 与 Embedded Java。此二者并无明确的界线,大致上来说,运算资源、内存以及显示装置比较丰富者使用 Personal Java,例如 Ser-Top Box、视讯电话等;反之,资源较有限者使用 Embedded Java,例如呼叫器、移动电话等。

JDK 1.2 版于 1998 年底公开。基于市场行销的考虑, Sun 在 JDK 1.2 版公开后即将 Java 改名为 Java 2, 将 JKD 改名为 Java 2 Software Development Kit, 简称 J2SDK。

Java 2 改变了先前 Personal Java 与 Embedded Java 的分法, 分成 Java 2 Platform Enterprise Edition(简称 J2EE), Java 2 Platform Micro Edition(简称 J2ME), Java 2 Platform Standard Edition(简称 J2SE)。

J2EE 适用于服务器, 目前已经成为企业运算、电子商务等领域中相当热门的技术。如 IBM、Oracle、BEA、HP 等 29 家公司运用 J2EE 技术推出了自己的产品, 其中尤以 BEA 公司的 Weblogic 和 IBM 公司的 Websphere 最为著名。与 J2EE 技术相抗衡的是微软的 .NET。J2EE 具有多层体系结构, 高可用性、安全性、可靠性和可伸缩性。它运行在 Java 虚拟机(JVM, Java Virtual Machine)上。

J2ME 适用于消费类电子产品, 为消费类电子产品提供一个 Java 的运行平台, 使得 Java 程序能够在手机、机顶盒、PDA 等产品上运行。

J2SE 适用于一般的计算机, 主要目的是为台式机和 workstation 提供一个开发和运行的平台, 其中包含了基本 Java 的核心应用编程接口(API, Application Programming Interface), 实际上 J2EE 和 J2ME 中都可能使用到 J2SE 中的 API。在学习 Java 的过程中, 主要是采用 J2SE 来进行开发, 它也是运行在 JVM 上, 目前的最新版本是 J2SDK 1.6。时至今日, J2SE 已经发展为一个覆盖面广、效率高、易用性强的技术平台。

1.2 Java 虚拟机及工作原理

1.2.1 Java 虚拟机

Java 虚拟机(JVM)是软件模拟的计算机, 可以在任何处理器上(无论是在计算机中还是在其他电子设备中)安全并且兼容地执行保存在 .class 文件中的字节码。Java 虚拟机的“机器码”保存在 .class 文件中, 有时也可以称之为字节码文件。

Java 程序的跨平台主要是指字节码文件可以在任何具有 Java 虚拟机的计算机或者电子设备上运行, Java 虚拟机中的 Java 解释器负责将字节码文件解释成为特定的机器码进行运行。Java 源程序需要通过编译成为 .class(字节码文件), Java 程序的编译和执行过程如图 1-1 所示。

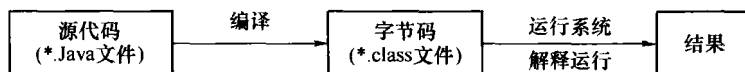


图 1-1

但是, Java 虚拟机的建立对不同的软硬件平台作专门的实现, 既要考虑处理器的型号, 也要考虑操作系统的种类。虚拟机的原理如图 1-2 所示, 目前在 SPARC 结构、X86 结构、MIPS 和 PPC 等嵌入式处理芯片上, 及 Unix、Linux、Windows 和部分实时操作系统上都有 Java 虚拟机的实现。

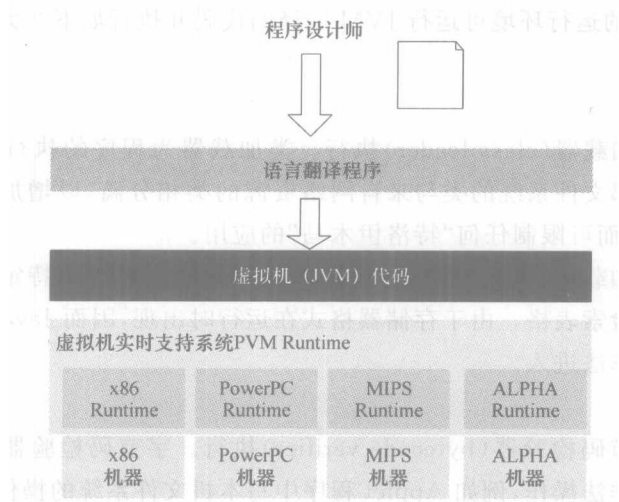


图 1-2

1.2.2 垃圾内存自动回收机制

在程序的执行过程中,部分内存在使用过后就处于废弃状态,如果不及时进行内存的回收,就会导致内存泄漏,进而导致系统崩溃。

在 C++ 语言中是由程序员进行内存回收的,程序员需要在编写程序时把不再使用的对象内存释放掉。但是这种人为的管理内存释放的方法却往往由于程序员的疏忽而使内存无法回收,同时也增加了程序员的工作量。而在 Java 运行环境中,始终存在着一个系统级的线程,专门跟踪内存的使用情况,定期检测出不再使用的内存,并进行自动回收,避免了内存的泄漏,也减轻了程序员的工作量。

但是 Java 的垃圾内存自动回收机制也存在缺点,即由于始终存在这个系统级的线程在运行,将会影响到 Java 应用的运行性能。

1.2.3 代码安全性检查机制

图 1-3 显示了 Java 虚拟机的工作及其加强代码安全性的方法。

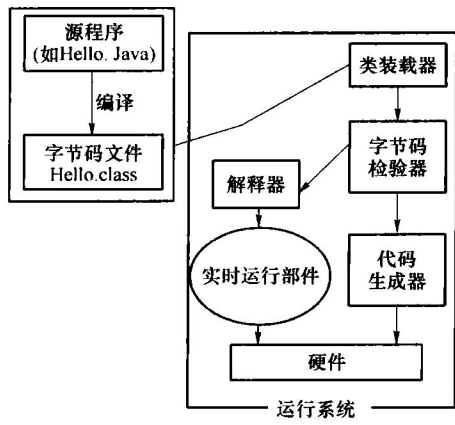


图 1-3

一个 Java 应用的运行环境可运行 JVM 编译的代码并执行如下 3 大任务:加载代码、检验代码和执行代码。

1. 加载代码

加载代码由类加载器(class loader)执行。类加载器为程序的执行加载所需要的全部类。类加载器将局部文件系统的类与来自网络资源的类相分离,以增加安全性。由于局部类总是首先加载,因而可限制任何“特洛伊木马”的应用。

当全部的类被加载后,可执行文件的存储器格式被确定,这样,特定的存储地址被分配给符号引用并创建检索表格。由于存储器格式在运行时出现,因而 Java 解释增加保护以防止对限制代码区的非法进入。

2. 检验代码

检验代码由字节码检验器(bytecode verifier)执行。字节码检验器检查该类文件的代码中是否存在某些非法操作,例如 Applet 程序中写本机文件系统的操作。

字节码检验器对程序代码进行 4 遍检验,这可以保证代码符合 JVM 规范并且不破坏系统的完整性。如果检验器在 4 遍检验后未返回出错信息,则下列各点可被保护:①类符合 JVM 规范的类文件格式;②无访问限制违例;③代码未引起操作栈上溢或下溢;④所有操作代码的参数类型总是正确的;⑤无非法数据转换发生,如将整数转换为对象引用;⑥对象域访问是合法的。所有源于网络的类文件都要经过字节码检验器。

3. 执行代码

执行代码由运行时的解释器执行。如果字节码检验器检验通过,由 Java 解释器负责把该类文件解释成为机器码进行执行。Java 虚拟机采用的是“沙箱”运行模式,即把 Java 程序的代码和数据都限制在一定内存空间里执行,不允许程序访问该内存空间外的内存,如果是 Applet 程序,还不允许访问客户端机器的文件系统。

1.3 Java 程序的开发环境及配置

最新的 Java 开发工具集 JDK 可以免费下载,其中包括了 Java API 的全套文档资料。除了 JDK 之外,目前有很多公司已经成功地开发了集成化的 Java 开发环境,其中包括 Eclipse、JBuilder、NetBeans。本书各章示例都在 JDK 下开发。JDK 虽然无可视化集成编程环境,但是对开发基础性、通用性软件有其独到的优势。对于编程初学者,开始在 JDK 环境下学习 Java,有利于理解 Java 程序的开发过程及培养学生面向对象的编程思想。Eclipse 之类的可视化集成开发环境是面向 Java 软件工程开发者的工具,方便开发者重用 Java 各种可重用类,开发速度快,特别是图形用户界面的编制给程序开发者带来许多方便。但是 Eclipse、JBuilder 不能很好地体现面向对象的教学思想,选择它们做初学 Java 面向对象程序设计的环境不是很适宜。初学者往往对可视化集成开发环境下生成的一大堆代码难以理解,因此,Java 语言学习者一开始就要重视培养编程思想,重视 Java 基本语法的理解,重视面向对象程序设计思想的训练,这样才能用好可视化集成开发工具。

1.3.1 J2SDK 的下载

Java 是由 Sun 公司开发的,从 Sun 公司的 Internet 站点可以找到最新版本的 JDK。目前 JDK 的最高版本 JDK1.6.0 的下载网址是:<http://java.sun.com/>。

1.3.2 J2SDK 的安装与配置

以 Windows 系统为例,JDK 的安装与环境配置如下:

在 Windows 下,直接运行下载的可执行文件 jdk-1-6-0-windows-i586.exe,安装到一个目录,例如将其安装到 C:\j2sdk 目录下面。安装后,其中 lib 目录下是 Java 库文件,bin 目录下是执行文件和动态连接库文件,bemo 目录下是 Applet 的实例。图 1-4 是安装后的目录结构图。

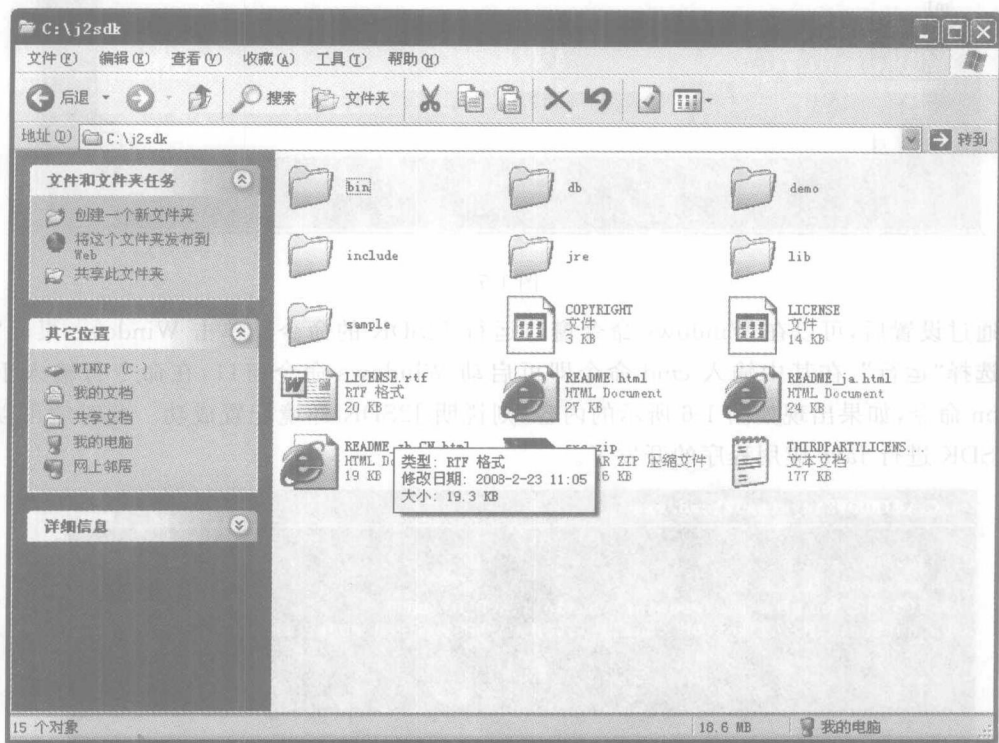


图 1-4

配置环境变量步骤如下:

从桌面上选择“我的电脑”(右键)→“属性”→“高级”→“环境变量”,进入环境变量对话框,在其中选择“系统变量”的“新建”;在变量名中输入 classpath,变量值中输入. 号;然后选择确定。再次选择“系统变量”的“新建”,在变量名中输入 path,变量值中输入 C:\j2sdk\bin,然后选择确定。在配置完成后,要重新启动计算机,环境变量配置才能生效,或者通过 sysedit 命令直接编辑系统自动批处理文件 autoexec. bat,如图 1-5 所示。在该文件中加入以下两行内容,也可以完成对运行 JDK 环境的配置。

```
path=C:\j2sdk\bin
```