

高等学校“十一五”规划教材·计算机系列

Java 语言实用教程

主编 杨兆楠 张剑飞 主审 张 珑



哈尔滨工业大学出版社
HARBIN INSTITUTE OF TECHNOLOGY PRESS

高等学校“十一·五”规划教材·计算机系列

Java 语言实用教程

主 编 杨兆楠 张剑飞
副主编 王 洋 宋井峰
主 审 张 珑

哈尔滨工业大学出版社

内 容 简 介

本书从面向对象编程的思路及方法入手,循序渐进地介绍了 Java 语言作为面向对象的编程工具的各个方面,并在示例中使用 Java 语言的实用类,使读者能将 Java 语言自如地运用到面向对象的程序设计过程中。通过本书的学习,使读者成为一名彻底的面向对象的程序员。

本书在第 1~3 章介绍了 Java 语言的概况及特点,引入如何利用面向对象地思想进行程序开发;第 4 章介绍了 Java 语言如何实现流程控制;第 5 章介绍了面向对象的概念,并逐步深入地讲解了如何在程序设计中利用面向对象的优点;第 6、7 章介绍了 Java 的异常处理及 I/O 系统;第 8 章介绍了 Java 的 GUI 设计;第 9 章介绍了如何在 Java 程序中使用多线程编程;第 10 章介绍了 Java 的特色——网络编程。本书既可作为高等学校计算机专业的教材,也可作为其他相关专业学习 Java 语言的教材。

图书在版编目(CIP)数据

Java 语言实用教程/杨兆楠等主编. —哈尔滨:哈尔滨工业大学出版社,2009.2

高等学校“十一五”规划教材·计算机系列

ISBN 978-7-5603-2807-2

I. J… II. 杨… 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(2009)第 010140 号

责任编辑 王桂芝 贾学斌

出版发行 哈尔滨工业大学出版社

社 址 哈尔滨市南岗区复华四道街 10 号 邮编 150006

传 真 0451-86414749

网 址 <http://hitpress.hit.edu.cn>

印 刷 肇东粮食印刷厂

开 本 787mm×1092mm 1/16 印张 10.5 字数 265 千字

版 次 2009 年 3 月第 1 版 2009 年 3 月第 1 次印刷

书 号 ISBN 978-7-5603-2807-2

定 价 22.00 元

(如因印装质量问题影响阅读,我社负责调换)

“高等学校计算机类系列教材”编委会

主 任 王义和

编 委 (按姓氏笔画排序)

王建华	王国娟	孙惠杰	衣治安
许善祥	宋广军	李长荣	周 波
尚福华	胡 文	姜成志	郝维来
秦湘林	戚长林	梁颖红	

序

当今社会已进入前所未有的信息时代,以计算机为基础的信息技术对科学的发展、社会的进步,乃至一个国家的现代化建设起着巨大的推进作用。可以说,计算机科学与技术已不以人的意志为转移地对其他学科的发展产生了深刻影响。需要指出的是,学科专业的发展都离不开人才的培养,而高校正是培养既有专业知识、又掌握高层次计算机科学与技术的研究型人才和应用型人才最直接、最重要的阵地。

随着计算机新技术的普及和高等教育质量工程的实施,如何提高教学质量,尤其是培养学生的计算机实际动手操作能力和应用创新能力是一个需要值得深入研究的课题。

虽然提高教学质量是一个系统工程,需要进行学科建设、专业建设、课程建设、师资队伍建设、教材建设和教学方法研究,但其中教材建设是基础,因为教材是教学的重要依据。在计算机科学与技术的教材建设方面,国内许多高校都做了卓有成效的工作,但由于我国高等教育多模式和多层次的特点,计算机科学与技术日新月异的发展,以及社会需求的多变性,教材建设已不再是一蹴而就的事情,而是一个长期的任务。正是基于这样的认识和考虑,哈尔滨工业大学出版社组织哈尔滨工业大学、东北林业大学、大庆石油学院、哈尔滨师范大学、哈尔滨商业大学等多所高校编写了这套“高等学校计算机类系列教材”。此系列教材依据教育部计算机教学指导委员会对相关课程教学的基本要求,在基本体现系统性和完整性的前提下,以必须和够用为度,避免贪大求全、包罗万象,重在突出特色,体现实用性和可操作性。

(1)在体现科学性、系统性的同时,突出实用性,以适应当前 IT 技术的发展,满足 IT 业的需求。

(2)教材内容简明扼要、通俗易懂,融入大量具有启发性的综合性应用实例,加强了实践部分。

本系列教材的编者大都是长期工作在教学第一线的优秀教师。他们具有丰富的教学经验,了解学生的基础和需要,指导过学生的实验和毕业设计,参加过计算机应用项目的开发,所编教材适应性好、实用性强。

这是一套能够反映我国计算机发展水平,并可与世界计算机发展接轨,且适合我国高等学校计算机教学需要的系列教材。因此,我们相信,这套教材会以适用于提高广大学生的计算机应用水平为特色而获得成功!

王 毅 和

2008年1月

前 言

Java 语言是目前最流行的软件开发语言之一,由 Sun 公司推出,被广泛应用于桌面应用程序及网络程序设计中。Java 建立在 C 和 C++ 之上,所以熟悉 C 和 C++ 的程序员能方便地运用 Java 进行编程。Java 摒弃了 C 和 C++ 许多不合理的特性,具有简单高效、面向对象、不依赖于机器结构的特点,支持多线程、分布式应用。Java 在可移植性、健壮性、安全性方面大大优于已存在的其他编程语言。Java 提供了丰富的类库,可广泛用于面向对象事件处理、企业级应用开发、Internet 系统管理、网页设计、可视化软件开发等。所有这些都奠定了 Java 作为网络应用开发的首选工具。

现在很多大学的计算机及信息管理专业都开设有关面向对象理论及 Java 语言方面的课程,但“Java 语言”与“面向对象”在教学中并没有被完全整合在一起,很多程序员还在用“面向过程”的思维使用着“面向对象”的工具。这就是本书的特点,在编写过程中,更注重“面向对象”的思想与“Java 语言”的结合,使初学者能顺利地过渡到“面向对象”的思维,并能用 Java 语言解决问题。

全书内容共分 10 章。第 1 章介绍 Java 语言的历史及发展概况、Java 语言的特点及开发环境;第 2 章介绍 Java 语言程序的构成,引入如何利用面向对象的思想进行程序开发,讲解了 Java 语言源程序的编写、编译、运行过程;第 3 章介绍 Java 语言的基础知识,包括数据类型、运算符与表达式等;第 4 章介绍 Java 语言如何实现流程控制;第 5 章介绍面向对象的概念,并逐步深入地讲解了如何在程序设计中利用面向对象的优点;第 6 章介绍 Java 语言的异常处理机制;第 7 章介绍 Java I/O 系统;第 8 章介绍 Java 的 GUI 设计;第 9 章介绍如何在 Java 程序中使用多线程编程;第 10 章介绍利用 Java 的网络包进行网络程序设计。

本书由黑龙江科技学院杨兆楠、齐齐哈尔大学张剑飞任主编,黑龙江科技学院王洋、宋井峰任副主编,其中第 1~3 章由王洋编写,第 4~6 章由杨兆楠编写,第 7~8 章由宋井峰编写,第 9~10 章由张剑飞编写,全书由杨兆楠统稿,哈尔滨师范大学张琬主审。

限于编者水平,本书难免存在一些疏漏和不当之处,敬请读者批评指正。

编 者

2009 年 1 月

目 录

第 1 章 Java 语言概述	1
1.1 什么是 Java 语言	1
1.2 Java 语言的特点	2
1.2.1 简单性	2
1.2.2 面向对象	2
1.2.3 平台无关性	3
1.2.4 分布式	4
1.2.5 多线程	4
1.2.6 安全性和稳定性	4
1.3 Java 的开发环境	5
1.3.1 JDK 工具集	5
1.3.2 Java 集成开发工具	6
小 结	8
习 题	9
第 2 章 Java 程序的构成	10
2.1 Java 应用程序	10
2.1.1 Java 程序入门	10
2.1.2 源程序的编辑	12
2.1.3 字节码的编译生成	13
2.1.4 字节码的解释与运行	15
2.1.5 使用 NetBeans 开发 Java 应用程序	16
2.2 小应用程序	18
2.2.1 源程序的编辑与编译	18
2.2.2 代码嵌入	19
2.2.3 Applet 的运行	20
2.2.4 在 NetBeans 中编写 Applet	22
小 结	22
习 题	22
第 3 章 Java 程序设计基础	23
3.1 数据类型、变量与常量	23
3.1.1 数据类型	23

3.1.2 标识符与关键字	24
3.1.3 变量与常量	25
3.2 运算符与表达式	28
3.2.1 表达式	28
3.2.2 运算符	28
3.2.3 运算符的优先级与结合性	33
3.3 数据操作实例	34
小 结	34
习 题	35
第 4 章 结构化程序设计	36
4.1 选择结构	36
4.1.1 if - then 语句	36
4.1.2 if - then - else 语句	36
4.1.3 switch 语句	37
4.2 循环结构	39
4.2.1 while and do - while 语句	39
4.2.2 for 循环	40
4.3 数组	41
4.3.1 一维数组	41
4.3.2 二维数组	43
4.3.3 字符数组	46
4.4 字符串	46
4.4.1 String 类	46
4.4.2 StringBuffer 和 StringBuilder 类	48
小 结	49
习 题	49
第 5 章 面向对象的特征	51
5.1 面向对象的基本特征	51
5.1.1 封装	51
5.1.2 继承	52
5.1.3 多态	52
5.2 对象和类	52
5.2.1 对象和类的概念	52
5.2.2 类的定义	53
5.2.3 对象的声明、创建与使用	55
5.2.4 构造方法	56
5.2.5 方法重载(Overloading)	57

5.2.6	关键字 this	58
5.3	类的继承	59
5.3.1	继承的定义	59
5.3.2	属性隐藏和方法覆盖(Override)	61
5.3.3	关键字 super	62
5.3.4	类的装载过程	63
5.4	Java 的修饰符	65
5.4.1	Java 的包(Package)	65
5.4.2	访问限定修饰符	66
5.4.3	static 修饰符	68
5.4.4	final 修饰符	70
5.5	抽象类和接口	70
5.5.1	抽象类的概念和基本特征	70
5.5.2	接口的概念和基本特征	71
5.5.3	抽象类和接口的异同	72
5.6	多态与动态绑定	75
	小 结	76
	习 题	76
第 6 章	异常处理	79
6.1	异常的概念	79
6.2	异常的分类层次	80
6.3	处理异常过程: try、catch 和 finally	81
6.4	自定义异常	82
6.4.1	自定义异常类设计	82
6.4.2	throw 操作	82
6.4.3	方法的异常声明 throws	83
	小 结	83
	习 题	83
第 7 章	Java 输入输出	85
7.1	输入输出的基本概念	85
7.1.1	数据流	85
7.1.2	数据流的类	85
7.1.3	InputStream	86
7.1.4	OutputStream	87
7.2	File 类与外部文件处理	87
7.2.1	File 类的主要方法	88
7.2.2	文件流(File Streams)的创建	88

7.3	过滤器流(Filter Streams)	90
7.3.1	数据流(Data Streams)	91
7.3.2	打印流(Print Streams)	94
7.3.3	缓冲流(Buffered Streams)	96
7.4	RandomAccessFile 类	97
7.5	控制台文本输入输出	99
	小 结	99
	习 题	100
第8章	Java GUI 设计	101
8.1	GUI 程序设计概述	101
8.1.1	应用程序界面的发展	101
8.1.2	图形用户界面	101
8.1.3	GUI 设计准则	101
8.2	GUI 程序体验	102
8.3	Java GUI 设计	104
8.3.1	Java GUI 工具	104
8.3.2	Swing 组件的应用	106
8.3.3	Java 语言的事件触发机制	110
8.3.4	利用 NetBeans 设计 GUI 程序	114
	小 结	115
	习 题	116
第9章	Java 多线程	117
9.1	Java 线程的概念	117
9.1.1	进程	117
9.1.2	线程	117
9.2	多线程设计	118
9.2.1	线程对象	118
9.2.2	创建和启动线程	118
9.2.3	线程的状态	119
9.2.4	线程操作函数	119
9.3	线程的同步	122
9.3.1	多线程程序可能出现的问题	122
9.3.2	同步方法	123
9.4	多线程的 GUI 程序	124
	小 结	126
	习 题	126

第 10 章	Java 网络编程	127
10.1	Java 网络编程的基本概念	127
10.1.1	通信与协议	127
10.1.2	套接字编程概述	128
10.2	Java 网络包(java.net)	129
10.2.1	套接字类(Socket)	129
10.2.2	服务器套接字类(ServerSocket)	129
10.3	使用 TCP 协议的 Socket 网络编程	130
10.3.1	一对一的 Socket C/S 通讯	130
10.3.2	综合示例:一对多通讯的实现	135
10.4	使用 UDP 协议的 Socket 网络编程	141
10.4.1	数据报套接字类 DatagramSocket、数据报类 DatagramPacket	142
10.4.2	服务器端实现	142
10.4.3	客户端实现	143
10.5	URL 类编程	145
10.5.1	URL 类简介	145
10.5.2	构造 URL 类对象	145
10.5.3	URL 类获取 URL 特征的主要方法	146
10.5.4	获取 URL 对象内容的方法	147
10.6	URLConnection 类的使用	147
10.6.1	利用 URLConnection 类对象读取 URL 内容	148
10.6.2	利用 URLConnection 类对象向 URL 对象发送服务请求及参数利用	148
小 结		150
习 题		150
参考文献		151

第1章

Java 语言概述

本章重点:熟悉 Java 语言的历史和发展;Java 语言的特点;Java 开发环境 JDK。

本章难点:Java 语言的特点。

1.1 什么是 Java 语言

Java 语言是一种计算机高级语言,也是一种极富创造力的计算平台。它是 Sun Microsystems 于 1995 年推出的,最初称为 Oak 语言,同年被重命名为 Java 编程语言。

Java 技术为用户带来了无数令人兴奋的可能性,它几乎使所有应用程序(包括游戏、工具及信息程序和服务)都能在任何计算机或设备上运行。从桌面 PC 到移动手持设备和移动电话,今天,Java 技术已经无处不在。

Java 起源于 Sun 公司的一个叫“Green”的项目,其目的是开发嵌入家用电器的分布式软件系统,使电器更加智能化。Green 项目一开始准备采用 C++ 语言,但是考虑到 C++ 语言太复杂,而且安全性差,于是决定基于 C++ 语言开发一种新的 Oak 语言(即 Java 语言的前身)。

Oak 语言是一种适用于网络编程的既精巧又安全的语言,它保留了许多 C++ 语言的语法,但去除了明确的资源引用、指针算法与操作符重载等潜在的危险特性。Oak 语言具有与硬件无关的特性,制造商只需要更改芯片,就可以将烤面包机上的程序代码移植到微波炉或其他电器上,而不必改变软件,这就大大降低了开发成本。当 Oak 语言成熟时,全球 Internet 也在迅速发展。

1994 年 Sun 公司的开发小组认识到 Oak 语言非常适合于 Internet 编程,用其编写了早期的 Web 浏览器,称为 WebRunner,后改名为 HotJava,展示了 Oak 语言作为 Internet 开发工具的能力。

1995 年 Oak 语言更名为 Java 语言(以下简称为 Java)。

1996 年 Sun 公司发布 JDK 1.0,计算机产业的各大公司(包括 IBM、Apple、DEC、Adobe、Silicon Graphics、HP、Oracle、Toshiba 和 Microsoft 等)相继从 Sun 公司购买了 Java 技术许可证,开发相应的产品。

1998 年 Sun 公司发布了 JDK 1.2(从这个版本开始的 Java 技术都称为 Java 2)。Java 2 不仅兼容于智能卡和小型消费类设备,还兼容于大型服务器系统,它使软件开发商、服务提供商和设备制造商更加容易抢占市场。这一开发工具极大地简化了编程人员编制企业级 Web 应用的工作,把一次编程多处使用的诺言应用到服务器领域。

1999 年 Sun 公司把 Java 2 技术分成 J2SE、J2EE 和 J2ME。其中 J2SE 是指从 1.2 版本开始的 Java 开发工具包(Java Development Kit, JDK),它为创建和运行 Java 程序提供了最基本的环

境。J2EE 和 J2ME 建立在 J2SE 的基础上, J2EE 为分布式的企业应用提供了开发和运行环境, 而 J2ME 为嵌入式应用(比如运行在手机里的 Java 程序)提供了开发和运行环境。

21 世纪以来 随着 Web 技术成为展示和操作数据的事实标准, 企业利用 J2EE 平台对原来分散的子系统进行整合。尽管应用整合可以通过多种手段来实现, 但 J2EE 出现后, 因其天生具备良好的开放性和可扩展性, 使之在应用整合和开发的过程中发挥了愈来愈显著的优势。J2EE 逐渐成为开发商创建电子商务应用的事实标准。

Java 的公用规范(Publicly Available Specification, PAS)在 1997 年被国际标准化组织(ISO)认定, 这是 ISO 第一次破例接受一个具有商业色彩的公司作为公用规范 PAS 的提交者。总之, 面向对象的 Java 语言具备一次编程、任何地方均可运行的能力, 这使其成为服务提供商和系统集成商用以支持多种操作系统和硬件平台的首选解决方案。Java 作为软件开发的一种革命性的技术, 其地位已被确定。如今, Java 技术已被列为当今世界信息技术的主流之一。

1.2 Java 语言的特点

Java 语言作为网络编程语言已经被广泛使用, 是一种新的计算概念。具体来说, Java 语言具有如下特征: 简单性、面向对象、分布式、解释型、安全稳定、平台无关、可移植、高性能、多线程、动态性等。下面我们将重点介绍 Java 语言的简单性、面向对象、平台无关、分布式、多线程、安全性和稳定性等特征。

1.2.1 简单性

Java 语言简单有效, 使用者可以很容易地进行程序设计, 而不需要很长的培训期。首先, Java 语法格式与 C++ 相似, 这使得大多数熟悉 C++ 的程序员很容易掌握 Java 编程技术; 其次, Java 摒弃了 C++ 中容易引发程序错误的部分, 如指针和内存管理等; 再次, Java 提供了丰富的类库, Java 类库提供的基本功能使编程人员能够快速有效地开发应用程序, 从而简化开发过程。

1.2.2 面向对象

Java 语言最显著的特点就是它是真正面向对象的语言。在 Java 中, 面向对象编程的基本元素是对象, 它把变量和方法从功能上封装成一个可重用、动态装载的整体。类是对象的抽象集合, 同一类的对象有同样的操作方法和性质。一个类建立后, 只要增加一些功能就可以产生新类, 这就是封装性、继承性等性质的含义。

面向对象其实是现实世界模型的自然延伸, 现实世界中任何实体都可以看作是对象。对象之间通过消息相互作用。另外, 现实世界中任何实体都可归属于某类事物, 任何对象都是某一类事物的实例。如果说传统的面向过程的编程语言是以过程为中心、以算法为驱动的话, 面向对象的编程语言则是以对象为中心、以消息为驱动的。用公式表示面向过程的编程语言为

$$\text{程序} = \text{算法} + \text{数据}$$

面向对象的编程语言则可以表示为

$$\text{程序} = \text{对象} + \text{消息}$$

所有面向对象的编程语言都支持三个概念: 封装、多态性和继承, Java 也不例外。现实世

界中的对象均有属性和行为,映射到计算机程序上,属性则表示对象的数据,行为表示对象的方法(其作用是处理数据或同外界交互)。

1. 封装

所谓封装,就是用一个自主式框架把对象的数据和方法联在一起形成一个整体,即类,类是支持封装的手段,是封装的基本单位。在 Java 中绝大部分成员是对象,只有简单的数字类型、字符类型和布尔类型除外,而对于这些类型,Java 也提供了相应的对象类型,以便与其他对象进行交互操作。

2. 多态性

多态性就是多种表现形式,具体来说,可以用“一个对外接口,多个内在实现方法”表示。举一个例子,我们需要一个方法完成输出数据的功能,对输出不同类型的数据其实使用了不同的方法(函数名相同,参数类型不同),这就是多态性的一种形式。

3. 继承

继承是指一个类直接使用另一个类的属性和方法。事实上,我们遇到的很多实体都有继承的含义。例如,若把汽车看成一个类型,它可以分成多个子类型,如卡车、公共汽车等。这些子类都具有汽车的特性,因此,汽车是它们的“父亲”,而这些子类则是汽车的“孩子”。Java 提供给用户一系列类(Class),Java 的类有层次结构,子类可以继承父类的属性和方法。与其他一些面向对象编程语言不同,Java 只支持单一继承。

1.2.3 平台无关性

Java 的平台无关性是指用 Java 编写的应用程序不用修改就可以在不同的软硬件平台上运行。平台无关有两种:源代码级和目标代码级。C 和 C++ 具有一定程度的源代码级平台无关,表明用 C 或 C++ 写的应用程序不用修改,只需重新编译就可以在不同平台上运行。

Java 主要靠 Java 虚拟机(Java Virtual Machine, JVM)在目标代码级实现平台无关性。JVM 是一种抽象机器,它附着在具体操作系统之上,本身具有一套虚拟机器指令,并有自己的栈、寄存器组等。JVM 通常是在软件上而不是在硬件上实现(目前, Sun 公司已经设计实现了 Java 芯片,主要使用在网络计算机(Network Computer, NC)上。另外, Java 芯片的出现也会使 Java 更容易嵌入到家用电器中)。JVM 是 Java 与操作平台无关的基础,在 JVM 上,有一个 Java 解释器用来解释 Java 编译器编译后的程序。Java 编程人员在编写完软件后,通过 Java 编译器将 Java 源程序编译为 JVM 的字节代码。任何一台机器只要配备了 Java 解释器,就可以运行这个程序,而不管这种字节码是在何种平台上生成的。另外, Java 采用的是基于 IEEE 标准的数据类型,通过 JVM 保证数据类型的一致性,也确保了 Java 的平台无关性。

Java 的平台无关性具有深远的意义。首先,它使得编程人员所梦寐以求的事情(开发一次软件在任意平台上运行)变成事实,这将大大加快和促进软件产品的开发。其次, Java 的平台无关性正好迎合了“网络计算机”思想。如果大量常用的应用软件(如字处理软件等)都用 Java 重新编写,并且放在某个 Internet 服务器上,那么具有 NC 的用户将不需要占用大量空间安装软件,他们只需要一个 Java 解释器,每当需要使用某种应用软件时,下载该软件的字节代码即可,运行结果也可以发回服务器。目前,已有数家公司开始使用这种新型的计算模式构筑自己的企业信息系统。

1.2.4 分布式

分布式包括数据分布和操作分步。数据分布是指数据可以分散在网络的不同主机上,操作分布是指把一个计算分散在不同主机上处理。

Java 支持 WWW 客户机/服务器计算模式,因此,它支持这两种分布性。对于前者,Java 提供了一个叫做 URL 的对象,利用这个对象,可以打开并访问具有相同 URL 地址的对象,访问方式与访问本地文件系统相同。对于后者,Java 的 Applet 小程序可以从服务器下载到客户端,即部分计算在客户端进行,提高系统执行效率。

Java 提供了一整套网络类库,开发人员可以利用类库进行网络程序设计,方便实现 Java 的分布式特性。

1.2.5 多线程

所谓多线程是指在一个程序中可以同时执行一个以上的线程(Thread),也就是通常所说的并行执行多个任务。线程与进程(Process)相似,也是执行中的程序,但线程数据较少,多个线程可以共享一组系统资源。系统处理线程的负荷要比处理进程小得多。多线程的优点是可以合理调配多个任务,交互式响应性能较好,并有实时特性。比如,打印任务可能需要较长时间,如果程序是单线程的,那么不打印完就不能做其他事情。采用多线程方法后,一个线程负责打印,而另一个线程继续做其他事情,这样就不需要漫长的等待了。

Java 则加入了多线程功能,它完成了其他语言难以实现的数据同步化过程,避免了资源冲突,这是 Java 又一个突出的优点。

1.2.6 安全性和稳定性

Java 最初的设计目的是应用于电子类消费产品,因此要求较高的可靠性。Java 虽然源于 C++,但它消除了许多 C++ 的不可靠因素,防止更多的错误在程序中出现。具体有以下几条:

- (1) Java 是强类型语言,要求显式的方法声明,这保证了编译器可以发现方法调用错误,保证程序更加可靠;
- (2) Java 不支持指针,这杜绝了对内存的非法访问;
- (3) Java 的自动垃圾回收机制防止了内存丢失等动态内存分配导致的问题;
- (4) Java 解释器运行时进行检查,可以发现数组和字符串访问的越界错误;
- (5) Java 提供了异常处理机制,程序员可以把一组错误代码放在一起,这样可以简化错误处理任务,便于恢复。

由于 Java 主要用于网络应用程序的开发,因此对安全性有较高的要求。如果没有安全保证,用户从网络下载程序执行则非常危险。Java 通过自己的安全机制防止了病毒程序的产生和下载程序对本地系统的威胁和破坏。当 Java 字节码进入解释器时,首先必须经过字节码校验器的检查;然后,Java 解释器将决定程序中类的内存布局;随后,类装载器负责把来自网络的类装载到单独的内存区域,避免应用程序之间相互干扰破坏;最后,客户端用户还可以限制从网络上装载的类只能访问某些文件系统。上述几种机制结合起来,使得 Java 成为一种安全可靠的编程语言。

1.3 Java 的开发环境

Java 程序的开发与其他高级语言的开发过程类似,首先按照 Java 的语法编写 Java 源程序,然后经过 Java 的编译器进行编译,再对编译生成的字节码文件进行解释执行。Java 把系统类库、编译及解释执行的工具集成在一个工具集中,称为 JDK(Java Development Kit)。另外,为了用户更方便地进行程序的编写及调试,Sun 及其他一些组织提供了不同的集成开发环境(Integrated Developing Environment, IDE),在这些 IDE 中都提供拼写检查、代码自动完成、关键字特殊显示、第三方插件等功能。

1.3.1 JDK 工具集

对于初学者来说,不需要了解 JDK 提供的所有功能,只需了解编译、执行命令即可。初学者可以通过任何文本编辑器(如:Windows 记事本、UltraEdit、Editplus、FrontPage 以及 Dreamweaver 等)编写 Java 源文件,然后在 DOS 状态下通过 javac 命令将 Java 源程序编译成字节码,最后通过 java 命令来解释执行编译后的 Java 字节码文件。Java 初学者一般都采用这种开发工具。

JDK 可以从 Sun 公司提供的网站 <http://java.sun.com> 免费下载。在这里可以下载不同版本,适用于不同操作系统的 JDK。例如,可以下载适用于 Windows 操作系统的 jdk-6u11-windows-i586-p.exe,运行该文件,按照安装向导进行安装,不需要改变安装路径。安装的过程分为两部分,首先安装 JDK,然后安装 Java 运行时环境(Java Runtime Environment, JRE)。JRE 是运行 Java 程序所必需的环境的集合,包含 JVM 标准实现及 Java 核心类库。有了 JRE 便可以运行网页中的 Applet 了,所以,有可能您的计算机中之前就已经安装了 JRE,安装后如图 1.1 所示。

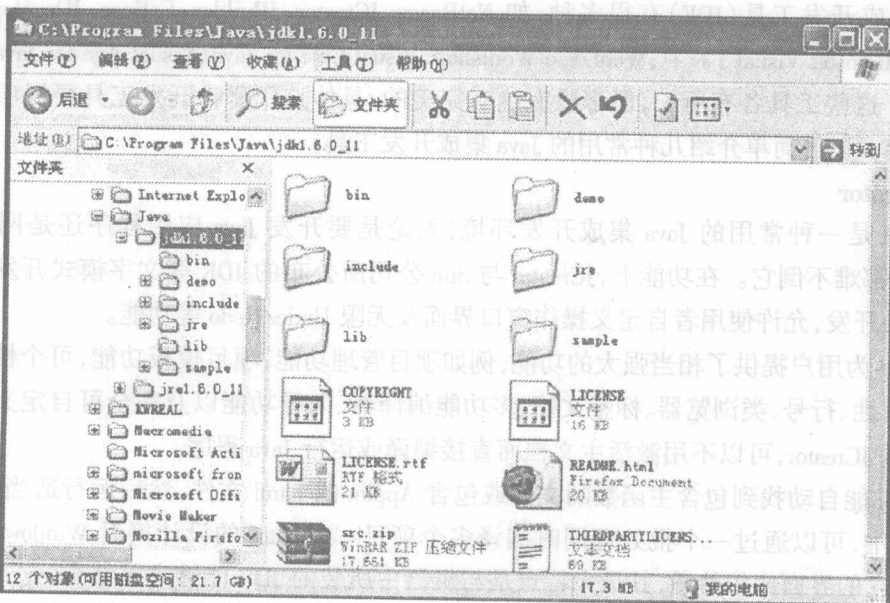


图 1.1 JDK 的安装