

21 CENTURY

21世纪全国高等职业院校 计算机应用技术专业
计算机网络技术专业 通用教材

21SHIJI QUANGUO GAODENG ZHIYEJISHU YUANXIAO JISUANJI YINGYONG
JISHU ZHUANYE JISUANJI WANGLUO JISHU ZHUANYE TONGYONGJIAOCAI

JAVA语言基础与实训

JAVA YUYANJICHU YU SHIXUN



山东科学技术出版社
www.lkj.com.cn

21 CENTURY

许文宪 主编

计算机应用技术专业
21世纪全国高等职业技术学院 计算机网络技术专业 通用教材
21SHIJI QUANGUO GAODENG ZHIYEJISHU YUANXIAO JISUANJI YINGYONG
JISHU ZHUANYE JISUANJI WANGLUO JISHU ZHUANYE TONGYONGJIAOCAI

JAVA语言基础与实训

JAVA YUYANJICHU YU SHIXUN

 山东科学技术出版社

图书在版编目 (CIP) 数据

Java 语言基础与实训/许文宪主编. —济南:山东科学技术出版社,2009

21 世纪全国高等职业技术学院 计算机应用技术专业
计算机网络技术专业 通用教材
ISBN 978 - 7 - 5331 - 4790 - 7

I. J... II. 许... III. Java 语言—程序设计—高等学校:技术学校—教材 IV. TP312

中国版本图书馆 CIP 数据核字(2007)第 128466 号

21 世纪全国高等职业技术学院 计算机应用技术专业
计算机网络技术专业 通用教材

Java 语言基础与实训

主编 许文宪

出版者:山东科学技术出版社

地址:济南市玉函路 16 号

邮编:250002 电话:(0531)82098088

网址:www.lkj.com.cn

电子邮件:sdkj@sdpress.com.cn

发行者:山东科学技术出版社

地址:济南市玉函路 16 号

邮编:250002 电话:(0531)82098071

印刷者:山东华鑫天成印刷有限公司

地址:潍坊市经济技术开发区

邮编:261000 电话:(0536)2250609

开本:787mm×1092mm 1/16

印张:15.5

版次:2009 年 8 月第 1 版第 1 次印刷

ISBN 978 - 7 - 5331 - 4790 - 7

定价:29.80 元

编审委员会

主 任 田幼勤 景 凯 赵成龙

副主任 包志家 杨志文 徐洪祥

委 员 (按姓氏笔画排序)

王建强 刘 刚 刘典型 许文宪

孙育新 苏 波 陈延祥 陈旭东

陈瑞斌 周连兵 周佩锋 彭 芸

葛良国

策 划 王洪胜

本书编审人员

主 编 许文宪

副主编 (按姓氏笔画排序)

刘 刚 刘典型 孙育新 李兴福

杨志文 苏 波 陈旭东 周连兵

徐 震

编 者 (按姓氏笔画排序)

亓俊红 王建强 王丹丹 史新鹏

江 磊 刘 军 刘 刚 刘典型

孙育红 许文宪 李兴福 陈旭东

陈延祥 苏 波 杨志文 张克瑜

张 倩 周连兵 徐 震 彭 芸

喻 云 翟俊华

内容 提要

本书是学习 Java 语言 (J2SE) 的基础教程, 共分为 10 个模块, 每个模块分为知识点、能力点、任务 (若干)、小结、思考与练习题、实训。内容涉及 Java 语言的运行环境、编程基础、面向对象编程、类库、异常处理、Applet、图形化设计、数据库操作、文件操作等。每个模块除了提供思考与练习题之外, 还提供了若干典型实训项目, 便于实践教学。全书的例题代码、参考答案, 均可从山东科学技术出版社网站上 (www.lkj.com.cn) 下载, 或通过 E-mail 发送。

本书以高职高专学生为主要读者对象, 可以作为高职高专计算机类和信息管理类专业的教材, 也可作为其他专业的选学教材和自学教材。

前言

preface

Java 语言作为一种纯面向对象的程序语言，近年来对计算机软件的发展产生了巨大的影响。目前，高职高专院校计算机专业以及相关专业的专业基本上都开设 Java 语言课程。

我们认为，面面俱到地讲述 Java 语言所涉及的内容，对高职高专的学生是不适合的，也是教学计划所不允许的。应该注意到，在开始 Java 语言课程之前，学生面向对象程序设计知识很弱且不系统。根据 Java 语言 (J2SE) 的内容和高职高专的教学特点，学习应该分为三个层次：基础部分、开发工具与专题编程部分、项目开发部分。Java 语言课程基础部分的教学要求是，具备使用 Java 进行程序设计的基本能力，为 Java 进一步深入学习或专题学习奠定基础；强化培养面向对象的思维方式，为学习后继课程提供必要的知识准备。

本书强调实用性和易学性。在知识内容上，以“基本够用、适当扩展”为原则，注重应用能力的培养；在知识讲述上，尽量采用高职高专学生能够理解的叙述方式，力求通俗易懂，逻辑严谨。全书将 Java 语言 (J2SE) 的基础知识划分为十个模块，每个模块包含若干相对独立的任务，涉及运行环境、编程环境、面向对象编程、类库、异常处理、Applet、图形化设计、数据库操作、文件操作等。每个模块除了提供思考与练习题之外，还提供了若干典型实训项目，每个实训项目都列出了实训目的、实训内容和简要提示，教师可以从中选择进行实训教学。附录中给出了全部实训的参考答案，供指导教师参考之用。可以从山东科学技术出版社网站 (www.lkj.com.cn) 下载实训答案的源代码，也可以通过 E-mail 向我们直接索取 (xwxian@126.com)。可同时提供的还包括全书的例题代码、思考与练习题的参考答案。

本书作为三年制或二年制的高职高专教材，建议讲授学时为 48~64 学时，上机实习为 32 学时。

由于 Java 语言发展迅速，作者水平有限，书中难免会有缺点和错误，敬请广大读者特别是讲授此课程的老师批评指正。您在使用过程中发现的问题和提出的建议，请随时发送到邮箱 (lkj_whs2008@126.com)，以便我们今后改进，在此表示衷心的感谢！

编者

模块一 准备 Java 运行环境

知识点	1
能力点	1
任务一 认识 Java 语言	1
任务二 安装 Java 环境	5
任务三 学习最简单的 Java 程序	7
任务四 了解 Java 开发工具	11
小结	17
思考与练习题	17
实训	17

模块二 掌握 Java 编程基础

知识点	19
能力点	19
任务一 学习 Java 语言基础	19
任务二 使用分支控制流程	27
任务三 使用循环控制流程	31
小结	35
思考与练习题	35
实训	36

模块三 认识面向对象编程

知识点	37
能力点	37
任务一 了解面向对象编程的基本思想	37
任务二 学习类的使用	41
任务三 使用构造函数	49
任务四 理解对象的引用	52
任务五 组织类的结构	58
小结	66
思考与练习题	67
实训	68

模块四 深入面向对象编程

知识点	70
能力点	70
任务一 实现继承	70
任务二 理解隐藏与覆盖	84
任务三 使用重载	89
任务四 认识抽象类和接口	93
小结	101
思考与练习题	101
实训	103

模块五 熟悉 Java 类库

知识点	105
能力点	105
任务一 使用字符串	105
任务二 了解数据类型类	114
任务三 使用数组	118
任务四 使用 System 类和 Math 类	127
小结	134
思考与练习题	135
实训	136

模块六 处理程序异常

知识点	138
能力点	138
任务一 捕获并处理异常	138
任务二 抛出异常	143
任务三 自定义异常	145
小结	146
思考与练习题	146
实训	147

模块七 学用 Applet

知识点	148
能力点	148
任务一 认识 Applet	148
任务二 播放声音	154
任务三 绘制图像	156

小结	160
思考与练习题	160
实训	161

模块八 设计图形化程序

知识点	162
能力点	162
任务一 了解基本组件	162
任务二 使用布局管理器	169
任务三 如何处理事件	172
任务四 使用选择控件和选择事件	179
任务五 使用菜单和其他常用事件	184
任务六 了解事件适配器	193
小结	196
思考与练习题	197
实训	198

模块九 操作数据库

知识点	199
能力点	199
任务一 了解 JDBC	199
任务二 连接数据库	204
任务三 查询数据库	209
任务四 处理数据	213
小结	219
思考与练习题	219
实训	220

模块十 文件操作

知识点	221
能力点	221
任务一 管理目录与文件	221
任务二 使用字节流	227
任务三 使用过滤流	231
任务四 读写随机文件	234
小结	238
思考与练习题	238
实训	239

参考文献

模块一 准备 Java 运行环境

知 识 点

Java 语言的特点、Java 程序运行原理。安装配置 Java 运行环境。Java 程序基本结构,编写简单 Java 应用程序的过程。

能 力 点

理解 Java 程序的运行原理。完成 Java 系统的安装配置。熟悉 Java 应用程序的开发过程。

●●● 任务一 认识 Java 语言

【了解任务】了解 Java 语言的发展历史,理解 Java 语言平台无关、面向对象、安全性、分布式、多线程、自动垃圾回收等特点,理解 Java 语言的运行机制:通过虚拟机解释运行字节码文件。

一、Java 语言概述

1. 发展历史

1991 年,Sun 公司成立 Green 项目组,为家用电子消费市场开发分布式系统,目标是可以利用网络远程控制家用电器。鉴于家用电器的制造标准各异,项目组希望新系统具有独立于软件平台的特征,并且安全易用。开始时项目组采用 C++ 语言进行系统开发,但是由于 C++ 语言的复杂性和难以满足安全性要求,最后不得不放弃 C++,转而研究设计出了一套新的与设备无关的程序设计语言。这就是 Java 语言的前身,被命名为 Oak(橡树)。

Oak 语言的发展曾一度遇到很大的问题。20 世纪 90 年代中期,WWW 的影响在 Internet 网上越来越大,WWW 浏览器开始在市场上出现。Green 项目组基于 Oak 满足网络应用所要求的平台独立性、系统可靠性和安全性等,于 1994 年用 Oak 设计了一个称为 WebRunner(后来称为 HotJava)的 WWW 浏览器。1995 年 5 月 23 日,Sun 公司正式发布了 Java 和 HotJava 两项产品,标志着 Java 语言的诞生。1995 年 8 月,Netscape 公司第一个将 Java 解释器集成到它的 Navigator 浏览器中,接着,Microsoft 公司在 Internet Explorer 浏览器中认可了 Java 语言。Java 语言开始了自己的发展历程。

1999 年 6 月,SUN 公司发布了 J2SE、J2EE、J2ME 三个版本(常被称为 Java2),分别用于不同的领域。J2SE(Java Standard Edition,Java 标准版)用于工作站、PC 机,为桌面开发和低端商务应用提供了 Java 标准平台;J2EE(Java Enterprise Edition,Java 企业版)用于服务器,构建可扩展的企业级 Java 平台;J2ME(Java Micro Edition,Java 微型版)用于消费性电子产品和嵌入式设备,构建嵌入式 Java 电子平台。2004 年 9 月发布了 J2SE1.5,为了表

示这个版本在 Java 语言发展史上的重要性,J2SE 1.5 更名为 J2SE 5.0。2005 年 6 月,SUN 公司发布 J2SE 6.0,称为 Java SE 6。同时,Java 各种版本的命名中取消数字“2”:J2SE 更名为 Java SE(Java Platform,Standard Edition Java SE,Java 平台标准版),J2EE 更名为 Java EE(Java Platform,Enterprise Edition,Java 平台企业版),J2ME 更名为 Java ME(Java Platform,Micro Edition Java ME,Java 平台微型版)。

2. 语言特点

Java 语言是当今流行的面向对象程序设计语言,它集中体现了当代软件技术的成果。具有分布式、安全性,又可以实现多线程,更主要的是它与平台无关,解决了困扰软件界多年的软件移植问题。

(1)面向对象 面向对象(object-oriented)程序设计模式是近代软件工业的一种革新,它提供软件的弹性度(flexibility)、模块化(modularity)与重复使用率(resability),降低开发与成本。Java 是完全对象化的程序语言,编程重点在于产生对象、操作对象以及如何使对象能一起协调工作,以实现程序的功能。

(2)简单化 一是 Java 语言的语法结构类似于 C/C++,熟悉 C/C++ 的程序设计人员不会对它感到陌生。二是 Java 省略了 C++ 的复杂特性,增加了实用功能,如不再支持运算符重载、多级继承,将指针改为引用等,使得程序开发变得简单而可靠。三是 Java 的系统非常小,其基本解释支持部分只占 40KB,附加上基本类库和线程支持也只需增加 175 KB。

(3)平台无关性 平台无关性是指 Java 能运行于不同的系统平台。Java 引进虚拟机概念,Java 虚拟机(Java Virtual Machine,称为 JVM)建立在硬件和操作系统之上,用于实现对 Java 字节码文件的解释和执行,为不同平台提供统一的 Java 接口。这使得 Java 应用程序可以跨平台运行,非常适合网络应用。

(4)安全性 安全性是网络应用系统必须考虑的重要问题。Java 设计的目的是提供一个网络/分布式的计算环境,因此,Java 特别强调安全性。Java 程序运行之前会利用字节确认器进行代码的安全检查,确保程序不会存在非法访问本地资源、文件系统的可能,保证了程序在网络间传送运行的安全性。

(5)分布式应用 Java 为程序开发提供了 java.net 包,该包提供了一组类,使程序开发者可以轻易实现基于 TCP/IP 的分布式应用系统。此外,Java 还提供了专门针对互联网应用的一整套类库,供开发人员进行网络程序设计。

(6)多线程 Java 语言内置了多线程控制,可使用户程序并行执行。利用 Java 的多线程编程接口,开发人员可以方便地写出多线程的应用程序。Java 语言提供的同步机制可保证各线程对共享数据的正确操作。在硬件条件允许的情况下,这些线程可以直接分布到各个 CPU 上,充分发挥硬件性能,提高程序执行效率。

(7)自动垃圾回收 Java 中对象的建立和存储都是在内存堆栈上面进行的,当一个对象没有任何引用的时候,Java 会自动删除这个对象所占用的空间,释放其内存,以避免内存泄漏。这种机制称为内存空间的自动垃圾回收机制。

二、面向对象程序设计的概念

面向对象程序设计(Object - Oriented Programming, OOP)是近代软件设计领域的一大革命。在大型系统的开发中, OOP 技术使得程序结构简单, 相互协作容易, 更重要的是提高了程序的重复使用率, 并降低了维护成本。Java 语言是一种完全面向对象的程序设计语言, 它通过类、对象等概念来组织和构建整个程序, 因此, 理解面向对象的基本概念是学习 Java 的前提和基础。

1. 类和对象

所谓对象就是真实世界中的实体, 它是一种具体的概念。对象具有的特征是: ① 一个名字用来标识该对象; ② 一组属性用来描述其特征; ③ 一组方法用来实现其行为。

例如, 一个具体的灯泡, 其形状、亮度等特征就是它的属性, 其变亮、变暗等行为就是它的方法。再如, 一个叫“张三”的人, 其属性包括肤色、身高、体重等固有特征, 其方法包括吃饭、休息、成长等行为功能。

类是具备某些共同特征的对象集合, 用程序设计的语言来说, 类是一种抽象的数据类型。例如, 我们把“一辆轿车”、“一辆公交车”、“一辆卡车”分别看成三个相互独立的对象, 这三个对象之间有一些共同的特点, 如它们都有大小、自重、轮子、方向盘等基本特征, 都有启动、加速、减速、刹车、转向等基本行为。将这些特点抽象到一起, 就得到了一个集合的概念“汽车类”。

类是对对象的抽象, 而对象是对类的具体化。面向对象的程序设计方法, 首先是进行合理的抽象, 抽取客观实体的共同属性和方法, 并根据抽象出的结果设计可重复使用的类; 然后, 通过所设计的类去定义一个个的对象(称为类的实例化); 最后, 按照问题的要求, 操作对象实现软件的功能。

2. 类的封装、继承和多态

在面向对象的程序设计中, 描述对象特征的属性由变量来表示, 也称为数据; 描述对象行为的方法由函数来实现。将数据及对数据的操作方法捆绑在一起就构成了类, 这就是所谓的封装, 它是面向对象程序设计的重要特性之一。

无论一个对象方法的目的是什么, 最终都表现为对对象数据的操作, 包括把对象的状态告知外界以及改变对象的状态等。只要有足够的方法, 就没有必要去直接操作对象的属性, 对象与外界打交道全部通过其自身的方法来实现。封装性有效地把对象的属性隐藏在对象的内部, 以免属性被不正当存取。同时, 对于外界而言, 不需要了解对象的方法是如何工作的, 只需要知道对象有哪些方法以及如何调用这些方法。就好比一辆汽车要停车, 你并不需要知道变速箱、液压系统、刹车系统等机构的细节如何运作, 只要踩下离合器和刹车踏板即可。因此, 封装使得程序的设计与实现有效地分开, 降低了方法之间的相互依赖, 提高了对象之间的独立程度, 提升了程序的可维护性。

继承是指在一个已有类的基础上, 通过添加新的数据和方法, 构成一个新类的过程。继承是一种机制, 它使得一个新类不仅拥有自己独特的数据和方法, 而且具有被继承类的数据和方法, 但却不用写重复的代码, 实现了代码复用, 方便了功能扩充。由继承得到的类称为子类, 被继承的类叫做父类或超类。例如, 我们已经创建了一个“手机类”, 它有通

话、发短信功能,我们可以继承此类,创建一个子类“新手机类”,增加拍照功能,这样,新创建的子类不但有拍照功能,而且也有它的父类所具有的通话、发短信功能。通常,父类中只定义框架性内容,而子类则在父类的基础上更加细化。

多态,这个词是从希腊文而来,意思是“许多状态”。在同一个类中可有许多同名的方法,但其参数数量与类型不同,这些同名不同参数的方法之间构成多态。

作为面向对象的设计语言,Java 很好地实现了类的封装、继承、多态。这将在以后的内容中逐步展开学习。

三、Java 运行机制

Java 程序的开发必须经过编写、编译、运行三个步骤。编写是指在 Java 开发环境中进行程序代码的输入,最终形成后缀名为 .java 的 Java 源文件,它是文本文件。编译是指使用 Java 编译器对源文件进行错误排查的过程,编译后将生成后缀名为 .class 的字节码文件,这不像 C 语言那样最终生成可执行文件。运行是指使用 Java 解释器将字节码文件翻译成机器代码,执行并显示结果。这一过程如图 1-1。

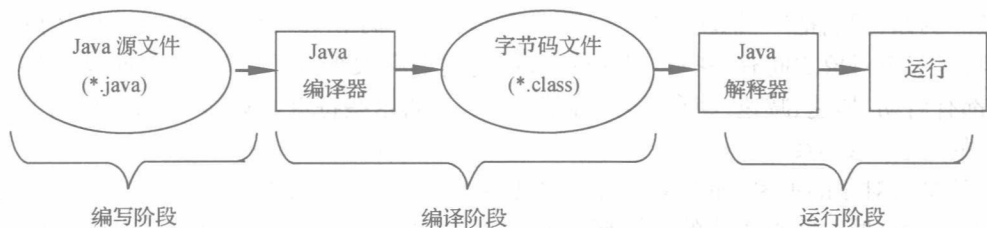


图 1-1 Java 程序运行流程

字节码文件是由 Java 编译器编译后生成的二进制目标代码文件,它是一种和任何具体机器环境及操作系统环境无关的中间代码,必须由 Java 解释器专门解释执行。因此 Java 是一种在编译基础上进行解释运行的语言。Java 解释器负责将字节码文件翻译成具体硬件环境和操作系统平台下的机器代码,以便执行。因此 Java 程序不能直接运行在现有的操作系统平台上,它必须运行在被称为 Java 虚拟机的软件平台之上。

Java 虚拟机(JVM)是运行 Java 程序的软件环境,Java 解释器就是 Java 虚拟机的一部分。在运行 Java 程序时,首先会启动 JVM。

JVM 把 Java 字节码程序和具体的硬件平台以及操作系统环境分隔开来,隐藏了这种不同软硬件平台的具体差别。这样,只要在不同的计算机系统上安装了针对于该系统的特定 JVM,就相当于获得了一个通用平台,Java 程序就可以运行,而不用考虑当前具体的软硬件平台环境,也不用考虑字节码文件是在何种平台上生成的。JVM 实现了真正的二进制代码级的跨平台移植,是 Java 实现平台无关的基础。Java 的这种运行机制可以通过图 1-2 说明。

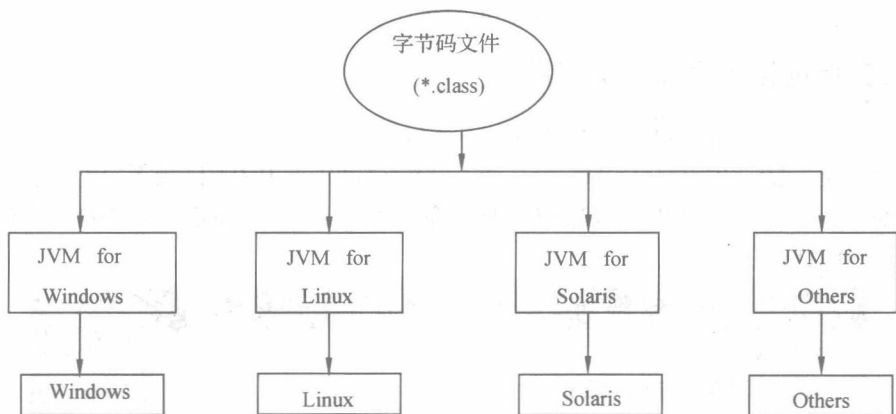


图 1-2 JVM 工作方式

Java 语言这种“一次编写,到处运行(write once,run anywhere)”的方式,有效地解决了目前大多数高级程序设计语言需要针对不同系统来编译产生不同机器代码的问题,即硬件环境和操作平台的异构问题,大大降低了程序开发、维护和管理开销。

需要注意的是,Java 程序通过 JVM 实现了跨平台特性,但 JVM 不是跨平台的。也就是说,不同操作系统之上的 JVM 是不同的,Windows 平台之上的 JVM 不能用在 Linux 上面,反之亦然。

●●● 任务二 安装 Java 环境

【了解任务】了解 Java 运行需要的环境,下载 Java SE 开发工具并进行安装,掌握 path、classpath 环境变量的设置。

一、获取开发工具

Sun 公司不仅提供了丰富的 Java 语言和运行环境,而且还免费提供了 Java 软件开发工具集(Java Development Kits,简称 JDK)。到目前为止,先后发布了多个主要的 JDK 的版本。JDK 包括 Java 的编译器、解释器、调试器等开发工具以及 Java API 类库。编程人员和最终用户可以利用这些工具来开发 Java 程序。其调试工具主要包括 Java 语言编译器 javac.exe,用于编译 Java 源程序;Java 字节码解释器 java.exe,用于解释运行 Java 程序,显示程序运行结果;小应用程序浏览工具 appletviewer.exe,用于测试并运行 Java 小程序。

为了建立起 Java 的运行环境,首先要通过网络下载 JDK。可从以下任一网站下载:Java 官方网站(中文,http://www.java.com/zh_CN/;英文,http://java.sun.com/)、Sun 公司网站(中文,http://cn.sun.com/;英文,http://www.sun.com/)、Java 中文站(http://www.java-cn.com/,一个 Java 技术专题网站),建议同时下载其 Java Documentation,这是 Java 帮助文档。

本书只涉及 Java SE,所使用的版本是 J2SE5.0。下载的文件是 jdk-1_5_0_07-win-dows-i586-p.exe 和 jdk-1_5_0-doc.zip,版本号中的“07”,代表 J2SE5.0 的第 7 次更

新。读者也可以直接下载最新的版本 Java SE6。

二、安装过程和设置环境变量

第一步,运行下载的 jdk-1_5_0_07-windows-i586-p.exe 文件,开始进行 Java 环境的正式安装。默认安装于 C:\Program Files\Java\jdk1.5.0_07 目录下,用户可以更改这个默认安装目录。考虑到将来调试程序时的命令方式,安装路径不应有空格,建议安装在 c:\jdk1.5.0_07 目录下。

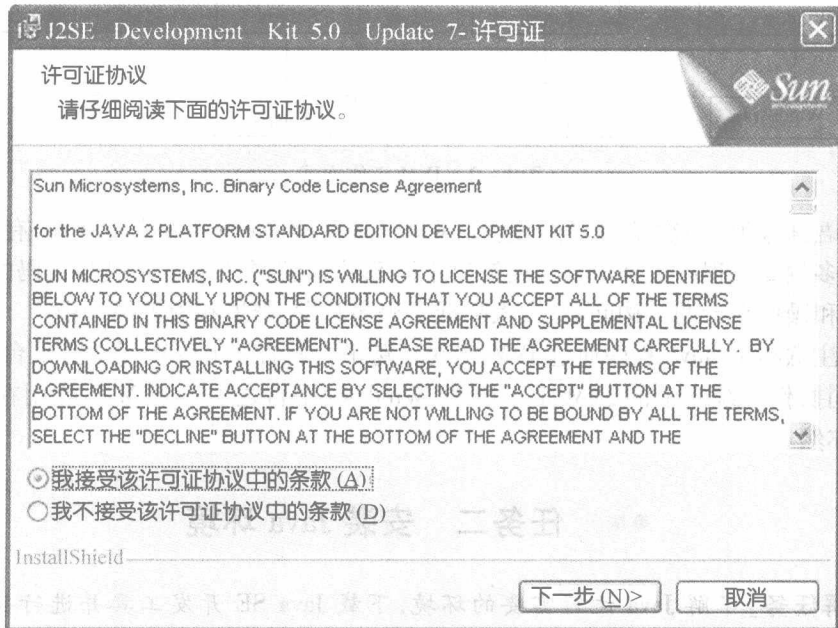


图 1-3 Java 安装过程(一)

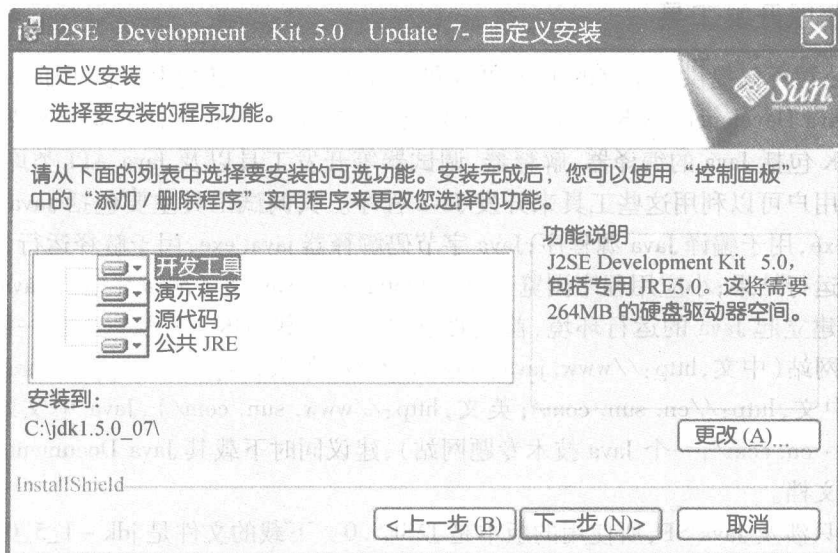


图 1-4 Java 安装过程(二)

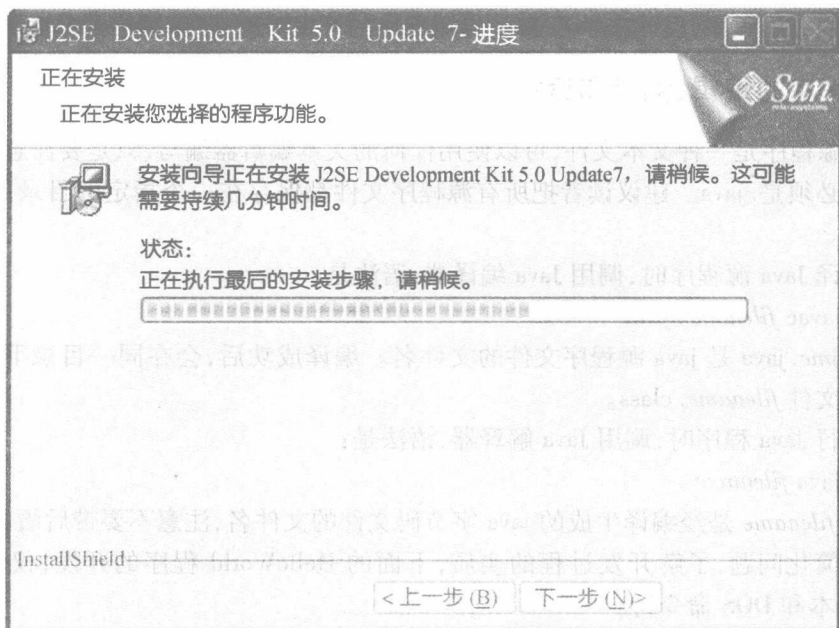


图 1-5 Java 安装过程(三)

第二步,需要设置运行环境参数,以便能够在 Windows 的任何目录下面都能编译和运行 Java 程序。对于 Win2000 或 XP 的用户,使用鼠标右击【我的电脑】,依次选择【属性】|【高级】|【环境变量】,打开【环境变量】对话框,在【系统变量】列表框中,单击【新建】按钮,新建环境变量 classpath,其变量值为“.; c:\jdk1.5.0_07\lib”;选择 path 变量,单击【编辑】按钮,在 path 变量的变量值后面加上“.; c:\jdk1.5.0_07\bin”。或者,直接编辑 path 变量,在其变量值后面加上“.; c:\jdk1.5.0_07; c:\jdk1.5.0_07\lib; c:\jdk1.5.0_07\bin”。其中的 c:\jdk1.5.0_07 是实际的 Java 安装目录。

对于使用 Win98 的用户,需要编辑 autoexec.bat 文件,在其最后面添加如下的 2 行语句,保存文件并重新启动系统。

```
set path = % path% ; c:\jdk1.5.0_07\bin
set classpath = . ; c:\jdk1.5.0_07\lib
```

最后,解压缩 jdk-1_5_0-doc.zip 文件,将其释放到 c:\jdk1.5.0_07\doc 目录下。至此,完成了 Java 的安装过程。

●●● 任务三 学习最简单的 Java 程序

【了解任务】通过编写最简单的 HelloWorld 程序,掌握 Java 程序开发的基本过程和 Java 程序的基本结构。

Java 有两类程序,即 Java 应用程序(Java application)和 Java 小程序(Java applet),前者是在命令行中运行的独立的应用程序,它类似于以往用其他高级语言开发的程序;后者

需要嵌入到网页在浏览器中执行(关于 Java 小程序的问题将在以后介绍)。现在先完成一个简单的应用程序,以此来学习 Java 应用程序的开发流程,了解程序的基本结构。

一、程序的编辑、编译和运行

Java 源程序是一种文本文件,可以使用任何的文本编辑器编写,只是要注意存储文件的后缀名必须是 .java。建议读者把所有源程序文件都保存在一个指定的目录下,便于调试和运行。

当编译 Java 源程序时,调用 Java 编译器,语法是:

```
javac filename.java
```

其中 *filename.java* 是 java 源程序文件的文件名。编译成功后,会在同一目录下生成同名的字节码文件 *filename.class*。

当执行 Java 程序时,调用 Java 解释器,语法是:

```
java filename
```

其中 *filename* 是经编译生成的 java 字节码文件的文件名,注意不要带后缀名 .class。

为了简化问题,了解开发过程的实质,下面的 HelloWorld 程序的开发,仅使用 Windows 记事本和 DOS 命令行。

【例 1.1】编写 HelloWorld 应用程序,在屏幕上显示字符串“Hello, World!”。

第一步,编写源程序。启动 Windows 记事本,输入以下程序源代码(必须注意:Java 区分大小写字母):

```
/* HelloWorld.java */
public class HelloWorld {
    public static void main(String args[]) {
        System.out.println("Hello, World!");
    }
}
```

保存文件时,先选择 *.* (所有文件)的文件类型,然后,输入 HelloWorld.java 为文件名;或者直接以带英文双引号“HelloWorld.java”的形式输入文件名。此文件名必须与源代码中 public class 后面的 HelloWorld 同名,且大小写也相同。将文件保存到一个指定的目录,如 C:\test 中。

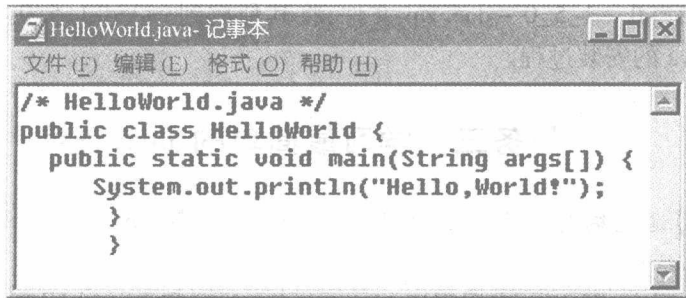


图 1-6 输入 HelloWorld 源程序