



21世纪 高等职业教育通用教材

JAVA语言程序设计

● 欧阳桂秀 主编

上海交通大学出版社

21 世纪高等职业教育通用教材

Java 语言程序设计

主 编 欧阳桂秀
副主编 石 彦 杰



上海 交 通 大 学 出 版 社

内容简介

本书由浅入深地讲述了 Java 语言的特点、基础知识、流程控制语句、数组、类、图形用户界面设计、异常处理、输入输出流、Java Applet、图形处理、多线程机制,以及网络程序等内容。每章给出了大量经过调试运行的实例,通俗易懂,便于初学者入门。每章后附有习题,便于初学者进一步理解本章内容。

本书可作为高职、高专院校计算机专业的教材,也可供有关读者自学。

图书在版编目(CIP)数据

Java 语言程序设计 / 欧阳桂秀主编. —上海: 上海交通大学出版社, 2004
21 世纪高等职业教育通用教材
ISBN 7-313-03545-4

I. J... II. 欧... 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 (2003) 第 121953 号

Java 语言程序设计

欧阳桂秀 主编

上海交通大学出版社出版发行

(上海市番禺路 877 号 邮政编码 200030)

电话: 64071208 出版人: 张天蔚

上海书刊印刷有限公司印刷 全国新华书店经销

开本: 787mm × 1092mm 1/16 印张: 13.25 字数: 319 千字

2004 年 2 月第 1 版 2004 年 2 月第 1 次印刷

印数: 1—2050

ISBN 7-313-03545-4/TP · 588 定价: 22.00 元

版权所有 侵权必究

21 世纪高等职业教育通用教材

编 审 委 员 会

主 任 名 单

(以姓氏笔划为序)

编审委员会顾问

叶春生 詹平华

编审委员会名誉主任

李 进 李宗尧

编审委员会主任

闵光太 潘立本

编审委员会常务副主任

东鲁红

编审委员会副主任

孔宪思 王俊堂 王继东 白玉江

冯拾松 匡亦珍 朱懿心 吴惠荣

李 光 陈 礼 赵祥大 洪申我

饶文涛 秦士嘉 黄 斌 董 刚

薛志信

序

发展高等职业教育,是实施科教兴国战略、贯彻《高等教育法》与《职业教育法》、实现《中国教育改革与发展纲要》及其《实施意见》所确定的目标和任务的重要环节;也是建立健全职业教育体系、调整高等教育结构的重要举措。

近年来,年轻的高等职业教育以自己鲜明的特色,独树一帜,打破了高等教育界传统大学一统天下的局面,在适应现代社会人才的多样化需求、实施高等教育大众化等方面,做出了重大贡献。从而在世界范围内日益受到重视,得到迅速发展。

我国改革开放不久,从1980年开始,在一些经济发展较快的中心城市就先后开办了一批职业大学。1985年,中共中央、国务院在关于教育体制改革的决定中提出,要建立从初级到高级的职业教育体系,并与普通教育相沟通。1996年《中华人民共和国职业教育法》的颁布,从法律上规定了高等职业教育的地位和作用。目前,我国高等职业教育的发展与改革正面临着很好的形势和机遇:职业大学、高等专科学校和成人高校正在积极发展专科层次的高等职业教育;部分民办高校也在试办高等职业教育;一些本科院校也建立了高等职业技术学院,为发展本科层次的高等职业教育进行探索。国家学位委员会1997年会议决定,设立工程硕士、医疗专业硕士、教育专业硕士等学位,并指出,上述学位与工程学硕士、医学科学硕士、教育学硕士等学位是不同类型的同一层次。这就为培养更高层次的一线岗位人才开了先河。

高等职业教育本身具有鲜明的职业特征,这就要求我们在改革课程体系的基础上,认真研究和改革课程教学内容及教学方法,努力加强教材建设。但迄今为止,符合职业特点和需求的教材却还不多。由泰州职业技术学院、上海第二工业大学、金陵职业大学、扬州职业大学、彭城职业大学、沙洲职业工学院、上海交通高等职业技术学院、上海交通大学技术学院、上海汽车工业总公司职工大学、立信会计高等专科学校、江阴职工大学、江南学院、常州技术师范学院、苏州职业大学、锡山职业教育中心、上海商业职业技术学院、潍坊学院、上海工程技术大学等百余所院校长期从事高等职业教育、有丰富教学经验的资深教师共同编写的《21世纪高等职业教育通用教材》,将由上海交通大学出版社等陆续向读者朋友推出,这是一件值得庆贺的大好事,在此,我们表示衷心的祝贺。并向参加编写的全体教师表示敬意。

高职教育的教材面广量大,花色品种甚多,是一项浩繁而艰巨的工程,除了高职院校和出版社的继续努力外,还要靠国家教育部和省(市)教委加强领导,并设立高等职业教育教材基金,以资助教材编写工作,促进高职教育的发展和改革。高职教育以培养一线人才岗位与岗位群能力为中心,理论教学与实践训练并重,二者密切结合。我们在这方面的改革实践还不充分。在肯定现已编写的高职教材所取得的成绩的同时,有关学校和教师要结合各校的实际情况和实训计划,加以灵活运用,并随着教学改革的深入,进行必要的充实、修改,使之日臻完善。

阳春三月,莺歌燕舞,百花齐放,愿我国高等职业教育及其教材建设如春天里的花园,群芳争妍,为我国的经济建设和社会发展作出应有的贡献!

叶春生

前 言

当前,随着计算机技术的飞速发展,软件产业也在发生着翻天覆地的变化。面对目前流行的多种开发语言,如何选择一种较好的语言才能使我们以不变应万变、面对强大的网络传输信息而应付自如,这是许多程序员都感觉棘手的事情。Java 语言的出现为我们提供了一种最佳的选择。

Java 语言是美国 SUN 公司推出的一门优秀的面向对象的程序设计语言,它不仅具有当前流行计算机开发语言所具有的所有特点,而且还具有自己的特色和风格。它集面向对象的编程技术、稳定性、安全性于一体,特别是它与平台的无关性及多线程和强大的网络支持能力,更增大了它的应用范围、增强了它的生命力,成为 Internet 的主流开发语言之一。学好、用好 Java 语言,就等于掌握了计算机应用和网络开发的一件利器。学习 Java 语言是学习语言工具的最佳选择。

作为一本高职高专教材,学以致用是编写本书的指导思想。本书针对高职高专的特点,在讲解理论的同时列举了大量的实例,以增强学生的理解、促进应用。为了加强学生实践能力的培养,每章都有一定量的思考题和练习题,以考核学生掌握每章的程度。

本书由欧阳桂秀任主编,石彦杰任副主编。第 1、7 章由湖北鄖阳师范高等专科学校曹大有编写;第 2, 3, 4, 6 章由武汉职业技术学院欧阳桂秀编写;第 5, 9, 11 章由河北工业职业技术学院石彦杰编写;第 8, 10 章由武汉职业技术学院陈晓红编写。

编者感谢所有关心、支持编写这本书的领导、同事和各位朋友。

由于编者水平有限,书中难免存在一些缺点和错误,殷切希望专家和读者批评指正。

编 者

2003 年 9 月

目 录

1	JAVA 语言概述	1
1.1	Java 语言的发展	1
1.2	Java 语言的特点	2
1.3	Java 语言的开发环境	3
1.4	Java 语言程序	7
2	数据类型、运算符与表达式	12
2.1	数据类型	12
2.2	标识符和关键字	12
2.3	基本数据类型	13
2.4	常量和变量	14
2.5	各类数值型数据间的混合运算	16
2.6	运算符	17
2.7	表达式	20
2.8	注释	20
2.9	Math 类	21
3	控制语句	23
3.1	赋值语句	23
3.2	分支语句	23
3.3	循环语句	26
3.4	跳转语句	29
4	数组	32
4.1	一维数组	32
4.2	二维数组	34
4.3	String 类	36
5	面向对象程序设计	42
5.1	面向对象程序设计的基本特点	42
5.2	类	45
5.3	对象	55
5.4	类的继承	57
5.5	多态与重载	60

5.6 接口	61
5.7 包	65
6 图形用户界面设计	70
6.1 图形用户界面概述	70
6.2 Java 的事件处理机制	71
6.3 AWT 组件库	72
7 异常处理和输入输出	121
7.1 异常的处理	121
7.2 InputStream 与 OutputStream 类	126
7.3 文件输入输出处理	129
8 JAVA APPLET	136
8.1 Java Applet 简介	136
8.2 Java Applet 的编写与执行	137
8.3 Java Applet 类的继承关系	141
8.4 Java Applet 的生命周期和主要方法	142
8.5 Java 小应用程序(Applet)和 Java 独立程序(Application)的区别	143
8.6 Applet 的事件处理	145
8.7 Applet 中的图形、声音文件	148
8.8 JavaApplet 中字体属性的设置	150
9 JAVA 绘图	154
9.1 Graphics 类	154
9.2 Graphics2D 类	166
9.3 Color 类	171
9.4 Font 类	175
9.5 图像的处理	176
10 JAVA 高级编程	179
10.1 Java 多线程机制	179
10.2 Java 与网络	183
11 JDBC 简介	192
11.1 JDBC 连接数据库的方法	192
11.2 JDBC-ODBC Bridge	193
参考文献	199

1 Java 语言概述

几乎每个使用计算机的人，都知道 Internet 是当今的时尚，至于 Java 语言在 Internet 开发上的重要地位，也许很多读者并不知道。Java 最初的用途并不是在 Internet 上，甚至连 Java 这个名字也不是原名。

1.1 Java 语言的发展

Java 的产生大约在 1990 年左右。有趣的是，当时 Java 的设计者 James Gosling（詹姆士·高斯灵）正率领着 Sun 的一个专家小组用 C++ 开发用于家电产品的操作系统和应用软件，发现这个编程语言的“大哥大”并不能很好的胜任，于是开发者就以 C++ 为基础开发设计了一种新的程序设计语言，命名为 Oak。Oak 先被用于家用电器的控制系统中。可以通过手持设备控制家庭中的电视、录像机等电器，后来又被用于 VOD 视频点播控制系统中。慢慢地 Oak 系统成熟了起来，不过却发现 Oak 这个名字已经被注册了，于是 Oak 的开发者 Sun 公司就起了 Java 这个名字。关于 Java 这个名字还有一个流传很广的故事。这个故事说，Java 是几个技术人员在一起品尝 Java 岛（印尼的一个地名）出产的咖啡时想起来的，Java 这个地区的咖啡是非常有名的，这些技术人员也许希望 Java 程序设计语言同样出名吧。

到 1992 年秋天，开发小组开发出一种具有小型可视界面的手持遥控设备，这个设备被称为 *7。*7 中有一个叫 Duke 的动画人物，这个动画人物引导人们操作图形界面，后来 Duke 成为 Java 的吉祥物。

1994 年中期，Internet 的成功引起了 Sun 公司的注意。Java 小组注意到了网页的一个根本问题：它们都是单调无味的。无论一个网页上有多少嵌入的图形、声音和图像，网络浏览器无法和这个网页交互。虽然开发者可以通过 HTML 将数据从网页上发送到一个基于服务器的程序，但这也是他所能做的全部。网页需要交互性，正是这个灵感促使 Java 小组致力于使 Java 语言向 Internet 和万维网发展。随着 Internet 的普及，Sun 公司发现 Java 非常适合于 Internet 编程，就提出了小应用程序 Applet 的概念，这种程序是可以嵌入到网页中运行的程序。

1994 年秋天，Sun 公司发行了 WebRunner 和 HotJava，它们都是用 Java 开发的网页浏览器。网络先锋 Netscape 公司注意到了这种新技术，他们宣布他们的浏览器 Netscape Navigator 将支持 Java。随着 Java 的不断完善，Sun 公司在 1996 年前后，正式发布了 Java 语言的正式版本，以其强大的功能迅速成为网络时代的首选编程语言，被称为网络时代的 C++，在国外非常受宠。终于，Java 语言也像 Java 地区出产的咖啡一样在全世界出名了。但遗憾的是，家用电器市场却是未被 Java 占领的极少数领域之一。

从今天的标准来看，最早的 Java 开发工具是非常原始的，比如 Sun 公司的 Java Developers Kit (JDK) 1.0。这类开发环境不仅没有交互功能，而且调试功能也非常有限。至 1996 年初，几个公司都发行了它们自己的 Java 交互开发环境。1996 年中期，Microsoft 公司发行了 Java 交互开发环境：Microsoft Visual J++ 1.0。

Microsoft Visual J++ 1.0 完全符合 Sun 公司的 Java 语言规范。它是一个开发 Java 小程序和 Java 独立程序的编程平台，包括 Java 编译器、编辑器、调试器和大量在线文件。

近年来，随着网络的蓬勃发展，Java 语言受到越来越多的网络程序设计人员的青睐。人们称其为 21 世纪的语言，这恐怕是 Java 的开发者们所始料不及的。

实际上，Java 不仅十分适用于网络开发，用它编写一般的应用程序也丝毫不比其他语言逊色。你可以在 Java 的教科书中找到很多描述 Java 特点的形容词，如面向对象、平台无关性、分布式、健壮性、安全性、可扩展性、多线程等等。对于初学者来说，要在还没有学会 Java 前明白这些特点的含义比较困难。只是希望大家记住：Java 是真正跨平台、纯粹的面向对象、适合单机和 Internet 开发的编程语言。适合单机和 Internet 开发前面已经作过介绍。跨平台是指它可以在不同操作系统上运行。而面向对象则是目前程序设计的主流思想，因为 Java 是新型语言，所以能够成为纯粹的面向对象的语言，像 C++ 这种“历史悠久”的语言因为要与以前的版本兼容，所以也就不不得不包含很多非面向对象的内容，从而使得程序设计和维护都比较困难。

1.2 Java 语言的特点

虽然 Java 是以 C++ 为模型生成的，但它的开发者们赋予它许多新的特点，尽管 Java 保留了许多 C++ 的风格，实际上它是一种新的编程语言，具体来说有如下特点：

(1) 简单性。这是 Java 最重要的特点。Java 语言的设计者虽然是基于 C++ 来设计 Java 的，但是却删除了 C++ 中几乎不用或很难用好的一些面向对象的特征，如：Java 不支持 struct、union 和 pointer 数据类型；不支持 typedef 和 #define；不支持运算符的重载和多重继承性。Java 内部有一个自动分配和释放内存的系统，使程序员从繁琐的事务中解放出来，在编程中少犯错误。

(2) 解释性，可移植性和平台无关性。这三个词实际上表达的是同一个意思。Java 编译后生成的是字节代码，而非特定计算机的本机指令，执行字节代码时还需经 Java 虚拟机解释。这样做有两个优点：其一，只要有虚拟机的支持，“编译”后的 Java 程序可以运行在任何平台上；其二，与传统的“编译—连接—测试—出错—调试”模式相比，可以提高编程效率。这是因为在编译字节代码时，进行类型和其他所有可能的检查，从而降低了出错的可能性，而其中的某些检查在真正的编译性语言中是无法做到的。

(3) 面向对象性。面向对象是一项很强的设计功能，因为它促成明确的界面定义，并允许开发者建立可重复使用的“软件 IC”。简单地说，面向对象是一种以对象及其界面为重心的程序设计方法。Java 的面向对象机制实质上就是 C++，并包括 Objective C 的一些延伸，以提供更强有力的解决方案。

(4) 分布性。Java 具有把分布在网络上的对象当作本地对象处理的能力。在 RMI(remote method invocation, 远过程引用) 标准的支持下，调用位于美国的对象和调用本地的对象差不多同样容易。必要时它还可以轻而易举地把类从远端机器装入本机。

(5) 健壮性。Java 在编译和运行时都进行严密的检查，以杜绝类型匹配错误和其他潜在问题。Java 取消了一些容易引发问题的特征（如指针），而且能自动回收全局内存堆的废弃单元，从而避免了内存流失现象的出现。

(6) 高性能。虽然字节码性能已相当不错,不过有些情况下还是要求程序有更高的执行性能。字节码可以在运行时由应用程序的特定 CPU 翻译成机器码,这对于习惯使用一般编译器与动态装载器(loader)的设计者而言,有点类似将最终的机器码生成器放到动态装载器之内。字节码格式在设计上就考虑到机器码的生成,产生的程序代码有一定的合理性:它会自动配置暂存器,而编译器在生成字节码时会执行一些最优化程序。

(7) 动态性。这里谈论的不是 Java 的个性,而是指 Java 是动态连接这一事实。新类在需要时才被装入,并且还可以通过网络进行。过时的连接也能被动态解除。

(8) 安全性。Java 的安全机制分为多级,包括 Java 语言本身的安全性设计以及严格的编译检查、运行检查和网络接口级的安全检查。在语言方面,Java 摒弃了指针类型和数据类型的隐式转换,从而对内存访问进行了严格的限制。在编译期间,Java 编译器并不分配内存,而是推迟到运行时由解释器决定。这样,编程人员就无法通过指针来非法访问内存。在运行期间,Java 的运行环境提供了四级安全保障机制:字节码校验器、类装载器、运行时内存布局、文件访问限制。在网络接口级,用户可按自己的需要来设置网络访问权限。另外,Java 的未来版将采用公开密钥法以及其他加密技术,来核实从网络上传输过来的代码的源主机及该代码的完整性。

(9) 多线程性。这指的是多个线程并发执行的能力。Java 在最基本的层次上支持多线程,它提供的原语可避免线程间的相互干扰。由于有线程同步方法的支持,所以多线程使用方便,且在任何地方的行为方式都是相同的(即使是使用本机代码的地方)。

1.3 Java 语言的开发环境

在使用任何语言之前,都必须安装有关的开发软件。Java 的开发环境分为基于 MS-DOS 或 UNIX 的命令行开发工具和基于 Windows 的集成开发工具。本节将分别给予介绍,其中重点介绍基于命令行的开发工具 SDK。

1. Java SDK 的安装和设置

SDK 被称为系统开发工具包(System Developing Kit)。Java 的 SDK 不止一个,建议选用 Sun 开发的软件。本书选用的是 J2SE SDK (Java 2 标准版 SDK, Java 2 Standard Edition SDK),这是 Java 标准版的 SDK1.3,也常称为 JDK1.3(Java 开发工具包,Java Developing Kit)。可以像安装其他软件那样安装 JDK,使用默认文件夹 c:\jdk1.3,本书中使用的文件夹是 d:\jdk1.3。安装完后,浏览一下文件夹中的内容,其中的 bin 文件夹中存放的是 Java 的可以执行文件,建议大家在 autoexec.bat 文件中加入以下两行:

```
path= d:\jdk1.3\bin;  
set classpath=.
```

JDK 中包括如下开发工具:

javac: 即 Java 编译器,用于将 Java 源程序编译成字节码。用法: javac[-option] 文件名,详细说明如表 1.1 所示。

表 1.1 javac 中的选项与含义

选 项	含 义	选 项	含 义
-classpath <path>	类文件查找路径	-nowarn	关闭警告信息
-d <directory>	类文件输出目录	-O	优化编译代码
-g	生成 jdb 使用的调试信息	-verbose	启用详尽模式

java: 即 Java 的解释器, 用于执行 Java 的字节码文件, 这种解释执行不需要 WWW 浏览器的参与。用法: java [-option] class, 这里 class 是指类名, 不能包括扩展名 “.class”。因为 java 将参数中的点号视为文件分隔符, 若输入: java shemp.class, java 就会按路径 shemp/class.class 寻找文件。命令中的选项如表 1.2 所示。

表 1.2 java 命令中的选项与含义

选 项	含 义	选 项	含 义
-help	打开帮助信息	-oss<number>	设置每个线程的栈容量
-version	打开版本号	-ms<number>	初始内存量
-v 或 -verbose	启用详尽模式	-mx<number>	最大堆尺寸
-debug	允许远程调试	-class<目录表>	类文件搜索目录
-noasyncgc	禁止异步回收废弃单元	-prof	输出 profile 中数据
-verbosegc	回收废弃单元时给提示信息	-verify	读入类时用字节代码检验
-cs	检验源文件的修改时间	-noverify	不检验类
-ss<number>	设置进程的 C 栈容量		

javadoc: Java 的文档生成器, 用于为 Java 类文件和类库生成 HTML 文档。文档中的信息来自类定义及源程序中特定格式的“文档注释”(即以 “/*” 开始, 以 “*/” 结束)。用法: javadoc [option] [classpath], 选项如表 1.3 所示。

表 1.3 javadoc 中的选项与含义

选 项	含 义	选 项	含 义
-version	含版本域	-doctype<type>	指定生成的文档类型
-author	含作者域	-noindex	不生成域和方法的序号
-sourcepath<目录表>	类文件搜索路径	-notree	不生成类层次
-classpath<目录表>	作用同上		

javah: 头文件生成器, 它能为 Java 程序中的本机代码生成 C 头文件。用法: javah [option] classname, 选项如表 1.4 所示。

表 1.4 javah 中的选项与含义

选 项	含 义	选 项	含 义
-o outfile	输出存入文件 outfile	-stubs	生成票根文件
-d directory	指定保存文件的目录	-verbose	设置输出为详尽模式
-td directory	设置临时文件的目录	-classpath path	指定类文件路径

javap: Java 的反汇编程序, 它可将 Java 类文件反汇编成可读形式。用法: javap [option] class, 选项如表 1.5 所示。

表 1.5 javap 中的选项与含义

选 项	含 义	选 项	含 义
-l	输出行号和局部变量表	-verify	对类进行检验
-p	输出私有和隐藏方法	-v	输出详尽信息
-c	输出反汇编代码	-h	生成放入 C 头文件中的信息
-classpath	类文件搜索路径	-version	输出 javap 版本号

Appletviewer: 它用来运行能够嵌入到 HTML 文件中的 applet 程序,且无需浏览器的支持。其用法为:

appletviewer [-debug] classname

其中, -debug 选项将在调试器 jdb 中启动 appletviewer。

jdb: Java 调试器,它类似于 UNIX 的 dbx 和 gdb,是基本的命令行调试器,它有自己的虚拟机拷贝。

2. 基于 Windows 下的开发工具

Java 基于图形方式下的可视化开发工具主要有: Sun 公司提供的 Java WorkShop 和 Microsoft 公司提供的 Visual J++。Java WorkShop 既是 Java 程序的开发工具,同时又是一个 WWW 浏览器。它的主要工具有:

Portfolio Manager (公文管理器): 可以创建一个 project, 从一个 Portfolio Manager 向另一个 Portfolio Manager 拷贝 project, 并组织相关的 project 文件。

Project Manager: 可阅读和编辑一个 project 的属性。

Source Editor (源文件编辑器): 阅读和编辑 project 源文件。它也提供一些功能,可以实现一些常用的调试功能,如设置断点、单步执行源代码。

Build Manage: 编译并连接 (build) 所有 project 文件,或只编译并连接新修改过的一部分 project。

Source Browser: 浏览组成 project 的类、方法和源文件的文档。

Debugger: 可加载并改正阻碍程序正确运行的错误。用它可以启动、运行、单步执行和停止一个线程。也可以设置断点、条件表达式、捕获并忽略异常,以及检查程序中的包和类。

Project Tester: 运行 project。如果 project 是一个 Applet 或远程 Applet, Project Tester 会为它生成一个 HTML 文件,并在 Web 浏览器中运行这个文件;如果 project 是一个 Application,则 Project Tester 在 Web 浏览器之外运行它。

Online Help: 提供 Java WorkShop 的在线帮助信息。

而 Visual J++则是 Microsoft 的 Developer Studio 软件包中的一员。与 JDK 相比, Visual J++会使程序设计工作变得轻松省力。在 JDK 中,程序员要编辑 Java 源文件,并且要用命令行编译 Java 源文件,另外所有的菜单、对话框等基本组件都需要程序员从最基本的做起。而在 Visual J++环境中,所有的东西都被集成到一个环境中,整个程序设计可以说脱离这个环境运行, Visual J++不但包括一个功能全面的工具集,而且排错的在线帮助也十分强大,它的向导甚至能够为用户生成大量的基本代码,而用户只需加入自己需要的功能。Visual J++窗口的几个主要部分为:

菜单栏: 位于窗口的上方,主要菜单有 File、Edit、View、Project、Build、Windows 和

Help。菜单的多少会随着任务的不同而变化。

工具栏：位于菜单栏的下方，它可以根据需要确定显示哪些常用的工具图标。初次进入 Visual J++ 时，在主窗口中将显示出四个辅助窗口：Project、Explorer、Toolbox、Properties 和 Task List。本书中的程序都是在 JDK1.3 中完成的。

3. Java 应用程序的开发周期和 Java 虚拟机

Java 应用程序的开始周期包括编译、下载、解释和执行四个部分。图 1.1 给出了从 Java 源代码开始，经过 Java 编译器到字节码校验器，最后到 Java 解释器的数据流和控制流。其中 Java 字节码装载器和字节码校验器并不限定字节码的来源，代码可能来自本地系统，也可能来自网络上的任意主机。

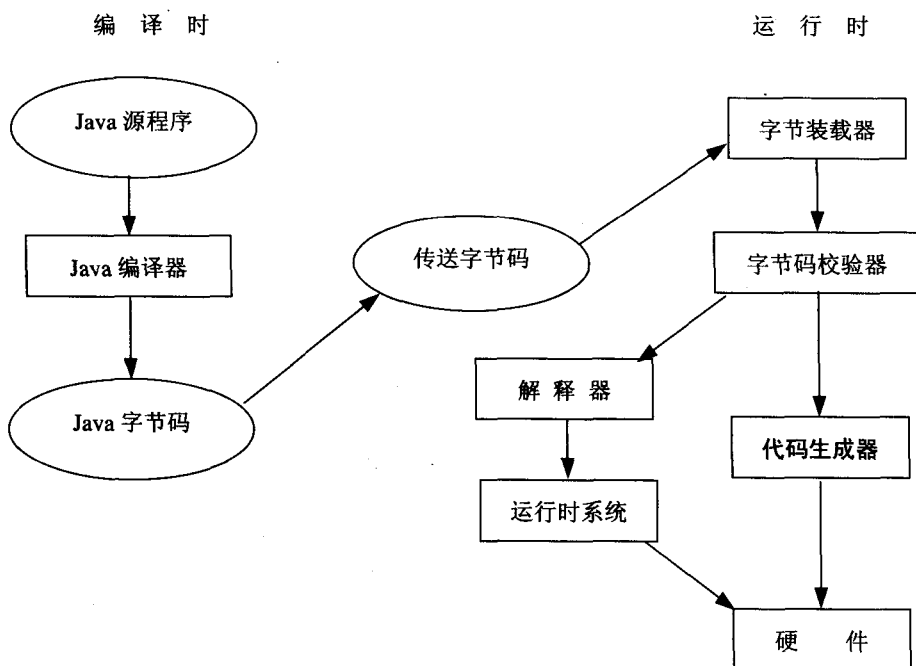


图 1.1 Java 应用程序的开始周期

Java 编译程序将 Java 源程序翻译为 Java 虚拟机的可执行代码（字节码）。这一过程与 C/C++ 不同。当 C 编译器编译生成一个对象的代码时该代码是为在某一特定硬件平台运行而产生的，因此在编译过程中，编译程序通过查找表将所有对符号的引用转换为特定的内存偏移量，以保证程序可自己运行。Java 编译器却不将对变量和方法的引用编译为数值引用，也不确定程序执行过程中的内存布局，而是将这些符号引用信息保留在字节码中，由解释器在运行过程中创建内存布局，然后再通过查表来确定一个方法或符号所在的地址。这样就有效保证了 Java 的可移植性和安全性。

Java 虚拟机的设计目标是提供一个基于抽象规格描述的计算机模型，为解释程序的开发人员提供很好的灵活性，同时也确保了 Java 代码可在符合该规范的任何系统上运行。Java 虚拟机对其实现的某些方面给出了具体的定义，特别是对 Java 可执行代码，即字节码格式给出了明确的规格。这一规格包括操作数的语法和数值，标识符的数值表示方式以及 Java 类文件

中的 Java 对象和常量缓冲池在 Java 虚拟机中的存储映像。这些定义为 Java 虚拟机解释器开发人员提供了所需的信息和开放的开发环境。Java 的设计者希望给开发人员自由地使用 Java。

1.4 Java 语言程序

Java 的程序有两种类型：独立的应用程序和 applet 小程序。应用程序是可以被 java.exe 直接解释执行的独立程序，而 applet 则存在于 Web 页，由 Web 浏览器（或 Sun 的 appletviewer）执行。出于对 Internet 安全上的考虑，applet 被施加了许多限制。禁止 applet 做的事情有：读文件，写文件，更改文件名，检查文件的大小，创建目录，检查文件是否存在，检查文件的尺寸或扩展名，检查文件的时间。

与应用程序相比，applet 程序的基本框架要复杂些。本节的目的就是让读者对 Java 的这两种程序有一个较完整的认识。

在运行程序前，要先在 autoexec.bat 文件里设置 Classpath 路径，否则编译出来的程序可能无法运行。Classpath 路径指出 class 文件所在的路径，设置方法是在 autoexec.bat 文件中加入下面的命令行：

```
path= d:\jdk1.3\bin;  
set classpath= d:\jdk1.3\bin
```

Classpath 路径指向源程序所在的目录。准确地说，Classpath 路径指向的是编译后的 class 文件所在的目录。本书假定源程序和编译后的 class 文件在同一目录 d:\jdk1.3\bin 中。

在 Windows 9x 系统中，设置步骤如下：单击桌面上的“开始”按钮；再单击“运行”，打开“运行”窗口；然后在“打开”文本框中输入“sysedit”命令；最后按“确定”按钮，出现“系统配置编辑程序”窗口。在 autoexec.bat 文件中添加上述命令，存盘后关闭“系统配置编辑程序”窗口。重新启动计算机，使设置生效。

1.4.1 Java Application 程序

1. Java Application 程序简介

每个 Java 应用程序都与面向对象技术有关，为编写程序，本节提供一个应用程序框架，读者先不需了解程序框架的细节，只需知道在什么地方编写代码即可。

现在编写第一个 Java 应用程序，内容是在屏幕上输出：“Hello World!”，其程序框架如下：

```
//c1:HelloWorld.java  
//author:caody  
public class HelloWorld  
{  
    public static void main(String args[])  
    {  
        System.out.println("Hello World! ");  
    }  
}
```

它包括以下三个部分：

① 以“//”开始的注释（comment）部分，注释用于说明程序，以双斜线“//”开始的一行内容均为注释，可以认为它们不是代码的一部分。

② public class 部分，这部分称为主类，原因将在后面的章节中介绍。它以“{”开始，并以“}”结束。而且 public class 后面的内容必须和源程序的文件名一致（包括大小写）。

③ public static void main(String args[])部分，这部分可称为 main()函数或者 main 方法它也必须以“{”开始，以“}”结束。从这部分开始的代码会顺序执行，这里的语句只有下面一句：

```
System.out.println( " Hello World! " );
```

该语句执行后会在 DOS 窗口下显示：Hello World!，这样的语句被称为输出语句。

2. Java Application 程序的上机步骤

(1) 编辑源程序。设置完 Classpath 路径后，就可以利用任何一种文本编辑器，如 Windows 98 的记事本（NOTEPAD），对上面的源程序进行编辑（注意不能用 word 等软件），然后按规定的文件名进行存盘，如上例的文件名：HelloWorld.java。注意：如果是第一次存盘，应先将“保存类型”改为“所有文件 (*.*)”，“文件名”必须为：HelloWorld.java（一定要有文件后缀名“.java”）。Java 语法规则程序存盘后的文件名必须与 public class 后面的字符串内容完全一致（包括大小写），文件的后缀名应该是 java（注意用记事本程序编写的源程序的后缀名缺省不是 java。如果后缀名不是 java，则编译时会出现不能打开文件的情况）。

图 1.2 所示的就是用记事本程序输入的上述完整程序。



图 1.2 用记事本编辑源程序

在默认情况下，记事本程序只能打开后缀为 txt 的文件，双击后缀为 java 的文件名不能打开源程序，但会出现图 1.3 所示的窗口。

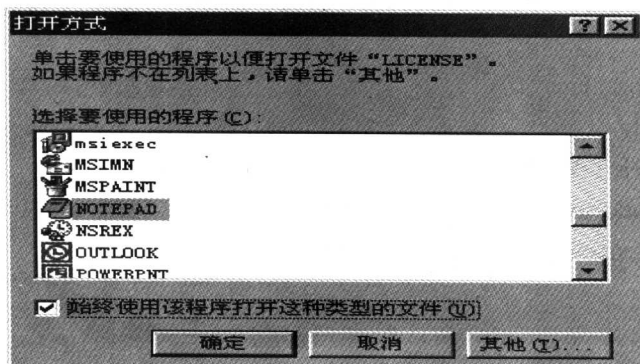


图 1.3 选择打开 Java 程序的关联程序

这时可选择用什么程序打开 Java 源程序。在图 1.3 中找到记事本的图标，并且选中“始终使用该程序打开这种类型的文件”复选框，最后单击“确定”按钮。那么以后只要双击后缀为 java 的源程序文件名，即可自动启动记事本程序并且打开源程序。

(2) 编译。编译 Java 源程序时，首先进入 MS-DOS 方式（在 Windows 9x 桌面上单击“开始”按钮→程序→MS-DOS 方式），然后切换到源程序所在的文件夹（利用 DOS 的 cd 命令），如 d:\jdk1.3\bin，接着使用 Java 编译器（javac.exe）按下述方式对源程序进行编译：

```
d:\jdk1.3\bin >javac HelloWorld.java
```

会生成名为：HelloWorld.class 的类文件。

如果源程序文件没有错误，那么编译的结果没有任何输出，如图 1.4 所示。

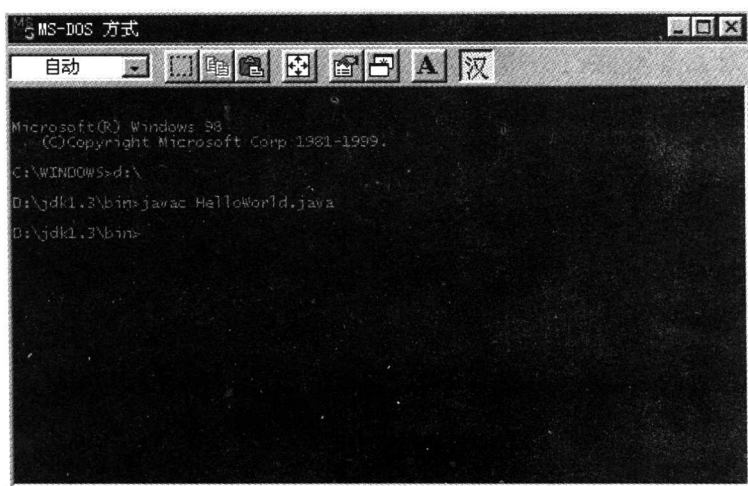


图 1.4 在 MS-DOS 窗口下编译 java 程序

(3) 运行。编译完成后，接着利用下述方法运行：

```
d:\jdk1.3\bin >java HelloWorld
```

就会在屏幕上显示出：HelloWorld!字样来，如图 1.5 所示。



图 1.5 HelloWorld.java 程序运行结果