



普通高等教育“十一五”国家级规划教材
高职高专计算机类专业规划教材

Java 程序设计

马世霞 主编

Java CHENGXU SHEJI

机械工业出版社
CHINA MACHINE PRESS



高职高专计算机类专业规划教材

1	计算机专业英语 (赠电子课件)	桑利君
2	C语言程序设计 (赠电子课件)	李素萍
3	C语言案例教程 (赠电子课件)	赵 睿
4	计算机网络基础 (赠电子课件)	顾可民
5	操作系统 (赠电子课件)	桑利君
6	数据结构 (赠电子课件)	王宇川
7	计算机实用软件教程 (赠电子课件)	石素卿
8	Visual Basic 6.0程序设计 (赠电子课件)	温丹丽
9	网页设计与制作实用教程 (赠电子课件)	马绍惠
10	 Java程序设计 (赠电子课件)	马世霞
11	局域网组建与维护 (赠电子课件)	何曙辉
12	动态网页设计 (ASP) (赠电子课件)	庞英智 齐志
13	动态网页设计 (JSP) (赠电子课件)	郭伟业
14	 微软网络操作系统 (赠电子课件)	鞠光明
15	 数据库系统原理与应用 (SQL Server 2000) (赠教学资源包)	刘志成
16	面向对象程序设计——Visual Basic 6.0 (赠电子课件)	朱丽敏
17	网络工程与综合布线技术 (赠电子课件)	王继水 朱葛俊
18	 现代网页设计及网站建设技术 (赠电子课件)	黄建华
19	计算机组装与维护 (赠电子课件)	陶 洪
20	计算机组装与维护实训 (赠电子课件)	和 枫
21	 Visual Basic.NET 程序设计 (赠电子课件)	田 更
22	 计算机网络实训案例教程 (赠电子课件)	孙桂芝
23	 基于C#的ASP.NET程序设计 (赠教学资源包)	翁健红
24	数据库原理与应用 (Access 2003) (赠电子课件)	彭湘凯 曾光辉
25	SQL Server 2000数据库程序设计 (赠电子课件)	庞英智
26	Visual Basic.NET 程序设计 (赠电子课件)	白德淳
27	局域网组建与维护 (赠电子课件)	梁建华
28	现代办公自动化 (赠电子课件)	马永涛
29	计算机基础 (赠电子课件)	王宇川

注：书名后加  的为普通高等教育“十一五”国家级规划教材

- ISBN 978-7-111-22904-9
- 策划：王玉鑫
- 封面设计/电脑制作：马精明

编辑热线：(010)68354423

地址：北京市百万庄大街22号 邮政编码：100037
 联系电话：(010)68326294 网址：http://www.cmpedu.com(机工教材网)
 (010)68993821 E-mail: cmp@cmpedu.com
 网址：http://www.cmpbook.com(机工门户网)
 E-mail: cmp@cmpbook.com

定价：27.00元

ISBN978-7-111-22904-9



9 787111 229049 >

普通高等教育“十一五”国家级规划教材
高职高专计算机类专业规划教材

Java 程序设计

主 编	马世霞		
副主编	郭祖华	茹庆云	
参 编	程跃华	王 端	刘 丹
	葛洪芳	王 蕾	



机械工业出版社

本书是以 Java 2 技术为背景的 Java 应用开发技术基础教程。按照由浅入深, 通俗易懂的原则介绍了 Java 编程语言, 让读者迅速上手。

本书共分为 13 章, 分别介绍了 Java 概论, Java 基本语法, 面向对象编程, 异常处理, Applet 程序设计, 图形用户界面, 输入与输出, 多线程, 图形、动画与多媒体, 访问数据库, 网络编程, 游戏及实验。

本书强调基本概念、技术和方法的阐述, 注重理论联系实际。书中列举许多实例, 每章都有练习题, 利于读者提高解决实际问题的能力。

本书以高职高专学生为主要对象, 可以作为高职高专计算机类的教材及职业培训教材, 也可做为其他专业的选学教材。

本书配有电子课件、习题解答、源程序代码、案例等相关辅助教学资料, 凡使用本书作为教材的教师可登录机械工业出版社教材服务网 www.compedu.com 下载。咨询邮箱: cmpgaozhi@sina.com。咨询电话: 010-88379375。

图书在版编目(CIP)数据

Java 程序设计/马世霞主编. —北京: 机械工业出版社, 2008. 1

普通高等教育“十一五”国家级规划教材. 高职高专计算机类专业规划教材

ISBN 978 - 7 - 111 - 22904 - 9

I. J… II. 马… 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 182403 号

机械工业出版社 (北京市百万庄大街 22 号 邮政编码 100037)

责任编辑: 王玉鑫 责任校对: 陈立辉

封面设计: 马精明 责任印制: 邓 博

北京京丰印刷厂印刷

2008 年 1 月第 1 版 · 第 1 次印刷

184mm × 260mm · 18.5 印张 · 454 千字

0 001 — 4 000 册

标准书号: ISBN 978 - 7 - 111 - 22904 - 9

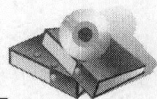
定价: 27.00 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换
销售服务热线电话: (010) 68326294

购书热线电话: (010) 88379639 88379641 88379643

编辑热线电话: (010) 68354423

封面无防伪标均为盗版



3.7.7 数学类	78	6.3.5 列表与选择事件	128
3.8 本章小结	81	6.3.6 复选框与选择事件	128
3.9 知识测试	81	6.3.7 单选按钮组与选择事件	130
第4章 异常处理	84	6.3.8 滚动条与调整事件	132
4.1 异常处理的概念	84	6.3.9 鼠标、键盘事件	135
4.2 异常类	85	6.4 布局管理器	138
4.2.1 异常类的层次结构	85	6.5 本章小结	145
4.2.2 Exception 类及其子类	86	6.6 知识测试	145
4.2.3 Error 类及其子类	87	第7章 输入与输出	147
4.3 异常处理	87	7.1 I/O 流概述	147
4.3.1 异常的产生	87	7.2 字节流	148
4.3.2 抛出异常	88	7.2.1 InputStream 类和 OutputStream 类	149
4.3.3 捕获异常	92	7.2.2 文件流 FileInputStream/ FileOutputStream	150
4.4 创建自己的异常类	94	7.2.3 标准流	156
4.5 本章小结	97	7.2.4 过滤字节流	157
4.6 知识测试	98	7.2.5 数据流	158
第5章 Applet 程序设计	100	7.3 字符流	158
5.1 Applet 的生命周期和 Applet 的 方法	100	7.3.1 Reader 类和 Writer 类	158
5.1.1 Applet 类的层次关系	100	7.3.2 InputStreamReader 类和 OutputStreamWriter 类	159
5.1.2 Applet 的创建	101	7.3.3 BufferedReader 类和 BufferedWriter 类	160
5.1.3 Applet 的生命周期	101	7.4 本章小结	163
5.1.4 Applet 类的显示方法	103	7.5 知识测试	164
5.2 Applet 标记	104	第8章 多线程	165
5.3 Applet 通信	105	8.1 线程概述	165
5.3.1 同一页 Applet 之间的通信	105	8.1.1 进程与线程	165
5.3.2 Applet 与浏览器之间的通信	108	8.1.2 Thread 类	166
5.4 Applet 程序示例	109	8.1.3 线程的生命周期	168
5.5 本章小结	111	8.2 线程的实现	168
5.6 知识测试	112	8.3 线程的同步	171
第6章 图形用户界面	113	8.4 本章小结	177
6.1 AWT 简介	113	8.5 知识测试	178
6.1.1 AWT 图形用户界面设计	113	第9章 图形、动画与多媒体	180
6.1.2 AWT 类	114	9.1 文本和字体	180
6.2 事件处理概述	115	9.1.1 文本输出	180
6.3 AWT 控件	118	9.1.2 字体设置	182
6.3.1 标签	119	9.2 图形	185
6.3.2 按钮与动作事件	119		
6.3.3 文本框与文本区	121		
6.3.4 下拉列表与选择事件	126		



9.2.1 绘制直线	185	11.1.1 Java 与网络编程	225
9.2.2 矩形	187	11.1.2 处理主机名称及 IP 地址的 InetAddress 类	226
9.2.3 椭圆	189	11.2 传输层协议 TCP 和 UDP	227
9.2.4 圆弧	190	11.3 Java 与统一资源定位符 (URL)	228
9.2.5 多边形	192	11.3.1 URL 基础知识	228
9.3 颜色	195	11.3.2 在 Java 中实现 URL	228
9.3.1 Color 类	195	11.4 Java 与 Socket 编程	234
9.3.2 Graphics 类与 Color 类	195	11.4.1 Socket 原理	234
9.4 图像	197	11.4.2 Java 与 Socket 实现	237
9.4.1 图像的显示	198	11.4.3 Java 与 UDP 上 Socket 的 实现	244
9.4.2 图像的缩放显示	200	11.5 本章小结	249
9.5 声音的播放	202	11.6 知识测试	249
9.6 动画的生成	205	第 12 章 游戏	250
9.7 本章小结	213	12.1 吃豆游戏介绍	250
9.8 知识测试	213	12.2 吃豆游戏实现	251
第 10 章 访问数据库	215	第 13 章 实验	273
10.1 概述	215	实验一 熟悉 Java 编程环境和 Java 程序 结构	273
10.1.1 JDBC 简介	215	实验二 Applet 程序设计	274
10.1.2 JDBC 的用途	217	实验三 Java 基本语法	275
10.2 JDBC 应用程序接口	217	实验四 面向对象基础	276
10.2.1 数据库连接	217	实验五 异常处理	277
10.2.2 向数据库发送 SQL 语句	218	实验六 图形用户界面	279
10.2.3 数据结果集	218	实验七 输入输出	280
10.2.4 关闭数据库连接	218	实验八 多线程	281
10.3 配置 ODBC 数据源	219	实验九 图形、动画与多媒体	282
10.3.1 建立数据库	219	实验十 数据库	284
10.3.2 建立数据源	220	实验十一 网络编程	285
10.4 数据库编程实例	222	参考文献	287
10.5 本章小结	223		
10.6 知识测试	224		
第 11 章 网络编程	225		
11.1 网络编程的基本概念	225		

第 1 章 Java 概论

学习目标

- 1. 掌握 Java 的安装、配置方法, Java 工作原理。
- 2. 理解 Java 基本概念、特点。
- 3. 了解 Java 的发展简史。

重点

- 1. 理解 Java 虚拟机的概念。
- 2. 熟练掌握 Java 运行环境设置和开发工具的使用。

难点

- 1. 两类 Java 程序编写、调试、运行的区别。
- 2. JDK 工具包的使用。

Java 是由 Sun 公司开发的新一代面向对象程序设计语言, 具有安全、跨平台、可移植、分布式应用等显著特点, 因此得到了广泛的使用。同时, 由于 Java 与 Internet 的紧密结合, 它已经成为当今主要的网络应用开发工具。

1.1 Java 产生的背景

1990 年, 美国 Sun 公司启动了一个叫 Green 的项目, 其最早的目的是为家用消费电子产品开发一个分布式代码系统, 这样我们可以把信息发给电冰箱、烤面包机、微波炉、电视机等家用电器, 与它们进行信息交流, 并对它们进行控制。

当时最流行的编程语言是 C 和 C++ 语言, Sun 公司的研究人员考虑是否可以采用 C++ 来编写程序, 但是研究表明, 由于家电价格较低, 而 C++ 过于复杂和庞大, 安全性又差, 运行语言所需的内存和处理芯片的花费, 已超过家电本身的成本, 为了解决此问题, 最后基于 C++ 开发了一种新的语言 Oak (橡树), Oak 是一种用于网络的精巧而安全的语言。但后来发现 Oak 已是 Sun 公司另一个语言的注册商标, 才改名为 Java。Java 的取名也有一段趣闻, 有一天, 几位 Java 成员组的成员正在讨论给这个新的语言取什么名字, 当时他们正在咖啡馆喝着 Java 咖啡 (太平洋岛屿爪哇盛产的一种味道非常美妙的咖啡), 有一个人灵机一动说就叫 Java 怎样, 希望 Java 语言就像端到编程人员面前的热腾腾的咖啡一样, 让人感觉很舒服, 建议得到了其他人的赞赏。于是, Java 这个名字就这样传开了。1995 年 5 月, Sun 公司对外正式发布了 Java 语言。

1.2 Java 语言特点

Java 自诞生之日起, 就受到了全世界的关注。这是因为它自身众多的突出的优点造成



的, 具体来说有以下主要特点。

1. 简单易学

Java 的简单首先体现在它具有精简的系统, 并且对硬件的要求不高。它的基本解释器仅有 40KB, 加上标准类库和线程的支持, 也只需要 215KB。其次 Java 语言采用了 C 语言中的大部分语法, 熟悉 C 语言的程序员会发现 Java 语言在语法上与 C 语言极其相似。另外 Java 采用内存自动回收机制, 程序员不必费心管理内存, 使程序设计更加简单, 同时大大减少了出错的可能。

2. 面向对象

Java 是完全面向对象的程序设计语言, 所有的 Java 程序均是类, Java 语言的设计集中于类、对象及其接口, 它提供了简单的类机制以及动态的接口模型。对象中封装了它的成员变量以及相应的方法, 实现了模块化和信息隐藏; 而类则提供了一类对象的原型, 并且通过继承机制, 使子类的对象可以使用其父类所提供的方法, 实现了代码的重用, 缩短了软件开发时间, 降低了开发成本。

3. 平台无关性

Java 是平台无关的语言, 用 Java 写的应用程序不用修改就可在不同的软硬件平台上运行。Java 主要靠 Java 虚拟机 (JVM) 实现平台无关性。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 的字节码。同时在 JVM 上, 有一个 Java 解释器用来解释 Java 字节码。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。

4. 可移植性

Java 程序的平台无关性使其可以很方便地被移植到网络的不同平台上, 可以在任何安装了 Java 解释器和运行环境的计算机系统上运行。而且, Java 编译器是由 Java 语言实现的, Java 运行系统由标准 C 语言实现, 这使 Java 系统本身也具有可移植性。

5. 分布性

Java 在网络程序设计上极为优异, 是因为 Java 为程序开发提供了专门针对互联网应用的类库, 这使得程序员使用网络上的资源和使用本地资料一样容易。

6. 安全性

用于网络分布环境下的 Java 必须要防止病毒的入侵, 而 Java 程序的三级代码安全检查机制就可防止非法代码的入侵, 阻止对内存的越权访问, 避免病毒的侵害。Java 程序在执行前, 要经过三次检查, 分别是 Java 本身的代码检查, 字节码的检查, Java 解释器的检查, 如图 1-1 所示。

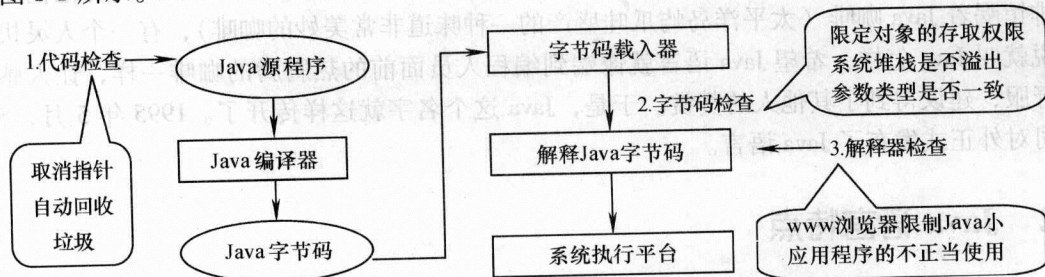


图 1-1 Java 的安全措施



7. 支持多线程

Java 内置了多线程功能，可使用户程序并行执行。Java 提供的同步机制保证了对共享数据的正确操作。使用多线程，程序设计者可以分别用不同的线程完成特定的行为，而不需要采用全局的事件循环机制，这样就很容易实现网络上的实时交互行为，减少用户等待时间。

1.3 Java 平台和虚拟机

平台是支持程序运行的软硬件环境，Java 平台是指在 Windows、Linux 等系统平台支持下的一种 Java 程序开发平台，主要由 Java 虚拟机（JVM，Java Virtual Machine）和 Java 应用程序接口（Java API）两部分组成。Java 虚拟机易于移植到不同硬件的平台上，是 Java 平台的基础；Java 应用程序接口由大量已做好的 Java 组件（组件是一种类）构成，这个接口提供了丰富的 Java 资源，使 Java 程序开发的效率比其他语言大大提高。

Java 破解各机器使用不同的机器语言的策略：就是它定义出自己的一套虚拟机，即 Java 虚拟机。Java 虚拟机的工作原理如图 1-2 所示。

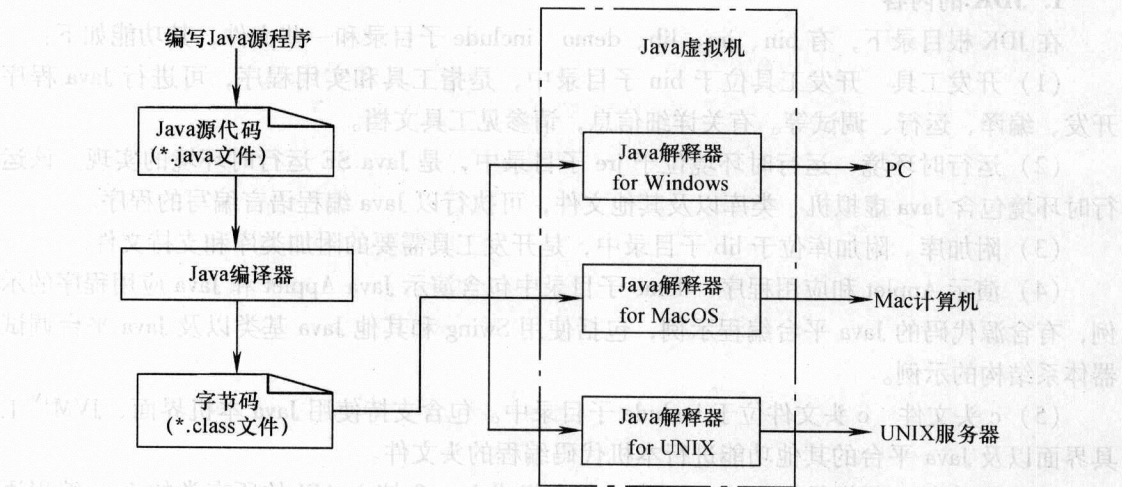


图 1-2 Java 虚拟机的工作原理

为了实现将程序动态地转换成与平台无关的形式，Java 首先通过 Java 编译器将 Java 源程序变换成 Java 字节码（Bytecode），并以“.class”为后缀的文件形式分布式地存放到 Internet 的一些服务器中。然后，以下载的方式动态地加载到 Java 虚拟机中解释执行。这些分布在不同平台上的解释器，即 Java 虚拟机。

编译器：对源代码进行半编译，生成与平台无关的字节码文件。

解释器：分布在网络中不同的操作系统平台上，用于对字节码文件解释执行。

解释就是取出一条指令，执行一条，有点像日常的口译；而编译有点像笔译，全部翻译以后才去执行。经过 Java 解释器的解释，平台就可以执行各种各样的 Java 程序。正因为如此，Java 程序才具有“一次编写，到处运行”的特点，即 Java 具有平台无关性。



1.4 运行环境安装与测试

Java 开发环境的基本要求非常低，只需一个 Java 开发工具包（JDK，Java Development Kit），再加上一个纯文本编辑器即可。为了提高开发效率，可以使用功能强大的文本编辑工具，如记事本、UltraEdit 等。对于熟练的开发人员，为了进一步提高开发效率，还可以使用具有可视化功能的 Java 专用开发工具，如 Jbuilder、J++、NetBeans。本书程序以记事本为编辑工具。

1.4.1 Java 开发包的安装

Sun 公司免费提供了 Java 开发工具包 JDK。JDK 是一种开发环境，用于使用 Java 编程语言生成应用程序、Applet 和组件。不同的操作系统（如 Windows、UNIX/Linux、Mac OS）有相应的 Java 开发包安装程序。读者可以登录 Sun 公司的网站获取 Java 开发包安装程序，本书中使用 Windows 操作系统环境下的 Java 开发包。书中所给的例子程序均在 JDK 1.6.0 环境下调试通过。

1. JDK 的内容

在 JDK 根目录下，有 bin、jre、lib、demo、include 子目录和一些文件，其功能如下：

(1) 开发工具 开发工具位于 bin 子目录中，是指工具和实用程序，可进行 Java 程序开发、编译、运行、调试等。有关详细信息，请参见工具文档。

(2) 运行时环境 运行时环境位于 jre 子目录中，是 Java SE 运行时环境的实现。该运行时环境包含 Java 虚拟机、类库以及其他文件，可执行以 Java 编程语言编写的程序。

(3) 附加库 附加库位于 lib 子目录中，是开发工具需要的附加类库和支持文件。

(4) 演示 Applet 和应用程序 demo 子目录中包含演示 Java Applet 和 Java 应用程序的示例，有含源代码的 Java 平台编程示例，包括使用 Swing 和其他 Java 基类以及 Java 平台调试器体系结构的示例。

(5) c 头文件 c 头文件位于 include 子目录中。包含支持使用 Java 本机界面、JVM™ 工具界面以及 Java 平台的其他功能进行本机代码编程的头文件。

(6) 源代码 源代码位于 src.zip 中。包含组成 Java 2 核心 API 的所有类的 Java 编程语言源文件（即 java.*、javax.* 和某些 org.* 包的源文件，但不包括 com.sun.* 包的源文件）。此源代码仅用于提供信息，以便帮助开发者学习和使用 Java 编程语言。

2. JDK 的基本命令

JDK 包含用于开发和测试以 Java 编程语言编写并在 Java 平台上运行的程序的工具。这些工具被设计为从命令行使用。除了 appletviewer 以外，这些工具不提供图形用户界面。JDK 的基本命令包括 javac、java、jdb、javap、javadoc、appletviewer。

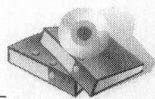
(1) javac Java 编译器，用来将 java 程序编译成字节码。

命令格式：javac [选项] 源程序名

(2) java Java 解释器，执行已经转换成字节码的 java 应用程序。

命令格式：java [选项] 类名 [参数]

(3) jdb Java 调试器，用来调试 java 程序。



启动 jdb 的方法有两种：第一种方法格式与 Java 解释器类似；第二种是把 jdb 附加到一个已运行的 Java 解释器上，该解释器必须是带-debug 项启动的。

(4) javap 反编译，将类文件还原回方法和变量。

命令格式：javap[选项]类名

(5) javadoc 文档生成器，创建 HTML 文件。

命令格式：javadoc[选项]源文件名

(6) appletviewer 小应用程序 Applet 浏览工具，用于测试并运行 Applet。

命令格式：appletviewer[选项]URL

其中 URL 是包含被显示 Applet 的 HTML 文件的统一资源定位符，当 HTML 文件位于本地机上时，只需写出文件名。

3. Java 的安装与配置

在得到 Java 开发包后首先需要进行安装。双击 Java 开发包 (JDK 1.6.0) 安装程序，出现安装界面，随后，安装程序会弹出许可证协议阅读画面如图 1-3 所示，选中接受条款后，单击【下一步(N)】按钮，继续安装，再单击【更改(A)...】按钮，如图 1-4 所示安装路径为 D:\Java\jdk1.6.0，单击【下一步(N)】按钮，弹出如图 1-5、图 1-6 所示画面，就安装成功。

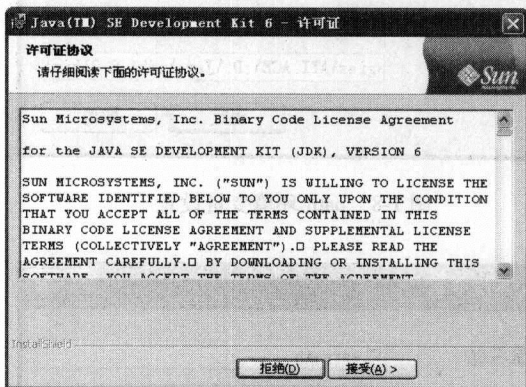


图 1-3 许可证协议阅读画面

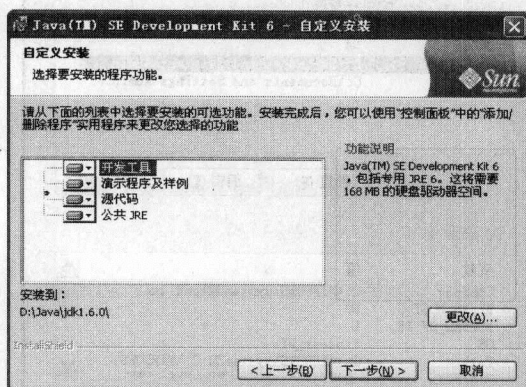


图 1-4 安装内容、安装路径

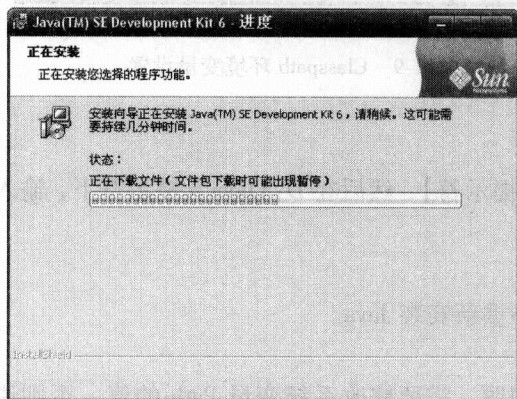


图 1-5 进度显示

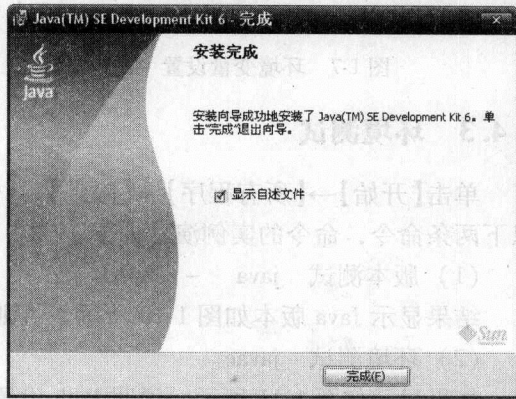


图 1-6 安装完成界面



1.4.2 环境变量设定

设定环境变量为了能够正常使用所安装的开发包，主要包括两个环境变量：Path 和 Classpath。Path 称为路径环境变量，用来指定 Java 开发包中的一些可执行程序（Java.exe、Javac.exe 等）所在的位置；Classpath 称为类路径环境变量。在不同的操作系统上，设定环境变量的方法是不同的。

在 WindowsXP 操作系统下，设置 Path 变量：右击【我的电脑】，在弹出菜单中选择【属性】，弹出【系统属性】对话框，在对话框中选择【高级】选项卡，在该页面单击【环境变量】按钮，进入【环境变量】对话框，如图 1-7 所示。找到变量 Path，双击该行就可以编辑该环境变量的值。在该变量已有的值后，再添加“；D:\Java\jdk1.6.0\bin”（注意：不包括引号，分号“；”不能缺少），单击【确定】按钮进行保存工作，如图 1-8 所示。

设置 Classpath 类路径环境变量：在【系统变量】列表框里，单击【新建】按钮，在【新建系统变量】对话框里，添加 Classpath，并设变量值为“.;D:\Java\jdk1.6.0\jre\lib”（注意：不包括引号，“.;”不能缺少）。如图 1-9 所示。至此，完成环境变量的设定工作。

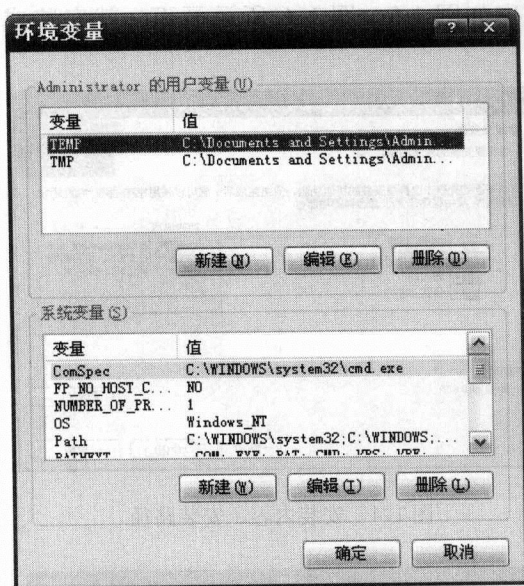


图 1-7 环境变量设置

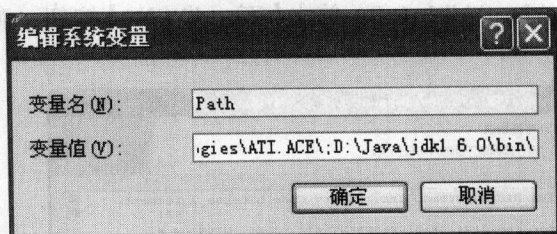


图 1-8 path 环境变量设置

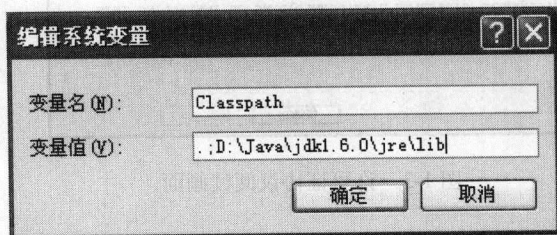


图 1-9 Classpath 环境变量设置

1.4.3 环境测试

单击【开始】→【所有程序】→【附件】→【命令提示符】，然后在 D:\Java > 提示符下，输入以下两条命令，命令的实例演示如下：

(1) 版本测试 `java -version`。

结果显示 Java 版本如图 1-10 所示。否则需要重新安装 Java。

(2) 环境测试 `javac`。

如果显示如图 1-11 所示，说明 Path 设置有问题，需要修改系统变量 Path 的值。否则说明 Path 设置成功。这就说明现在可以编写 Java 程序了。

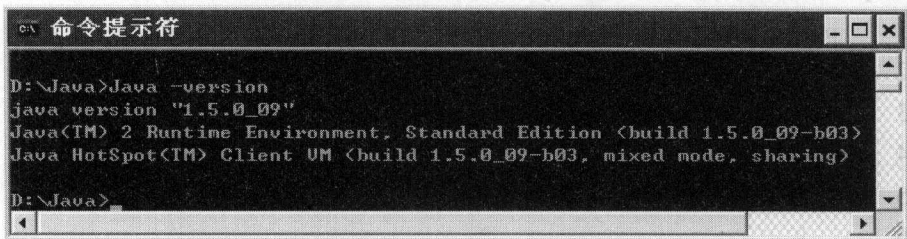


图 1-10 version 命令显示窗口



图 1-11 环境变量设置

1.5 初识两类 Java 程序

按照运行环境的不同，可将普遍使用的 Java 程序分为两种：Java 应用程序（Java Application）和 Java 小程序（Java Applet）。Java 应用程序在本机上由 Java 解释程序来激活 Java 虚拟机，而 Java 小程序则通过浏览器来激活 Java 虚拟机。此外，它们的程序结构也不相同。

1.5.1 Java 应用程序（Java Application）

Java 应用程序即 Java Application 是以类的形式出现的，程序中可以包含一个类，也可以包含多个类。每个应用程序的执行都是从主方法 `main()` 开始，没有 `main()` 方法的应用程序是不能执行的。包含了 `main()` 方法的类叫做主类，程序的主文件名必须与主类名相同。Java 应用程序的执行流程如图 1-12 所示。

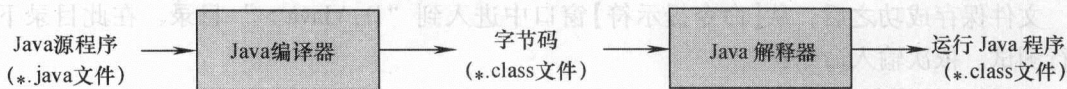


图 1-12 Java 应用程序的执行流程图

一个 Java 应用程序的源代码（*.java）文件经过 Java 编译器编译之后产生 Java 字节码（*.class）文件，再经 Java 解释器解释执行程序，输出结果。

Java 源程序可以在任何一种文本编辑器上编辑，例如：记事本编辑器、UltraEdit 编辑器等，本书程序采用了记事本编辑器。在记事本编辑器中书写 Java 源程序，并将其保存为扩展名为“.java”的文件。

特别说明：本书中所有例题程序源代码之前的序号均是方便了程序分析而另外加的，读者书写源程序时请务必将序号去掉。



【例 1-1】 源程序名为“testHello. java”的程序，在屏幕上输出“Java 欢迎你!”。

```
1 public class testHello
2 {
3     public static void main(String args[] )
4     {
5         System.out.println("Java 欢迎你!");
6     }
7 }
```

1. 创建应用程序源文件

1) 打开【附件】中的记事本，在文本编辑界面中键入程序清单。

2) 单击【文件】→【保存】，将文件命名为“testHello. java”，保存到 D:\java > 目录下。

如图 1-13 所示。注意：文件名必须和清单所声明的类名即“testHello”保持一致，而且 Java 是区分大小写的，且扩展名必须是“. java”。

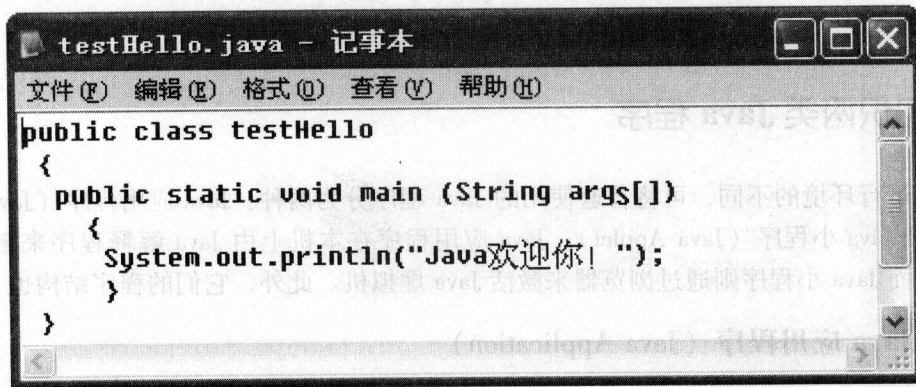


图 1-13 编写 java 代码的记事本界面

2. 编译

文件保存成功之后，从【命令提示符】窗口中进入到“D:\Java >”目录。在此目录下，进行测试。依次输入命令：

1) 输入编译程序 javac 命令：

```
javac testHello. java
```

2) 输入显示文件目录命令：

```
dir
```

这时会发现目录下多了一个“testHello. class”文件，这是 Javac 编译器将源代码编译成字节代码生成类文件的结果。再由 Java 解释，执行“testHello. class”类文件。

3. 运行

输入运行程序命令：

```
java testHello
```

运行结果如图 1-14 所示。

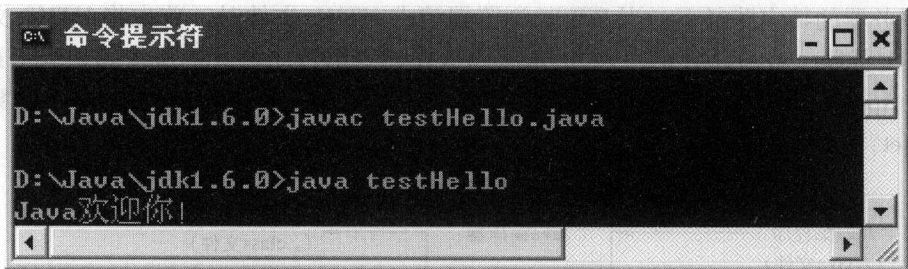
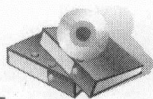


图 1-14 例 1-1 程序运行结果

4. 程序分析

输出的“Java 欢迎你!”, 是 Java 编译器直接执行 Java 应用程序字节代码的结果。

第 1 句: 类的声明, 声明为“testHello”的类 (class)。下面对每个单词进行分析。

public: 说明类的属性为公共类, public 并不是必须的, 可写可不写。

class: 是 Java 类的关键字, 如果想声明一个类, 必须使用关键字 class, 代表以下的内容都是这个类的内容。我们可以将类理解为包含程序逻辑的容器, Java 程序中的任何部分都必须包含在类中。

testHello: 是这个类的名字, 类名必须以字母开头。

第 2、7 句: “{”代表类“testHello”是从这里开始, 最后的结束是在“}”处。

第 3 句: 是 Java 程序的一个特殊方法, 又称 main 方法。同 C 语言中的 main 函数用法一样, 程序从这个方法开始执行。下面对第 3 行中的每个单词进行分析。

public: 访问控制符, 表示 main 方法为公共的。可以被调用, 所以这个 public 不能省。

static: 是将 main 方法声明为静态的, 在这里这个关键字也不能省。

String args[]: 用来接收命令行传入的参数, String 是声明 args[] 可存储字符串数组。本程序中没用到这个参数, 但这个参数不能省略, 否则出错。

第 4、6 句: 大括号也必须是成对出现的。

第 5 句: 是将“Java 欢迎你!”显示到屏幕上。

System. Out. println() 作一解释: System 是 Java 中的 System 类, out 是指 System 类中的一个变量, println 是 out 变量的一个方法。

1.5.2 Java 小程序 (Java Applet)

Java 小程序即 Java Applet, 它不能单独运行, 必须通过 HTML 调入后, 方能执行实现其功能, 它既可以在 Appletviewer 下运行, 也可以在支持 Java 的 Web 浏览器中运行。Applet 程序中必须有一个类是 Applet 类或 JApplet 类的子类, 也是 Applet 的主类。

Applet 程序能跨平台、跨操作系统、跨网络运行, 因此, 它在 Internet 和 WWW 中得到广泛地应用。另外, 由于 Applet 程序代码小, 易于快速下载和发送, 并且它具有不需要修改应用程序就可增加 Web 页新功能的特性, 因此 Applet 程序备受用户青睐。

Java 小程序的执行流程如图 1-15 所示。首先在文本编辑器中编写 Java 源程序 (*.java 文件), 并通过 Java 编译器编译成 Java 字节码 (*.class 文件)。与 Java 应用程序不同的是还得另外编写一个 HTML 文件将该字节码嵌入其中; 再将此字节码文件和 HTML 文件保存



在 Web 服务器的特定路径下。当 WWW 浏览器请求 HTML 文件时，首先将 HTML 文件下载到 WWW 浏览器，若 HTML 中嵌入 Applet，WWW 浏览器再向 WWW 服务器请求下载 HTML 中指定的 Applet 字节码，下载 Applet 字节码后，WWW 浏览器使用内嵌的 Java 解释器解释执行 Applet，显示运行结果。

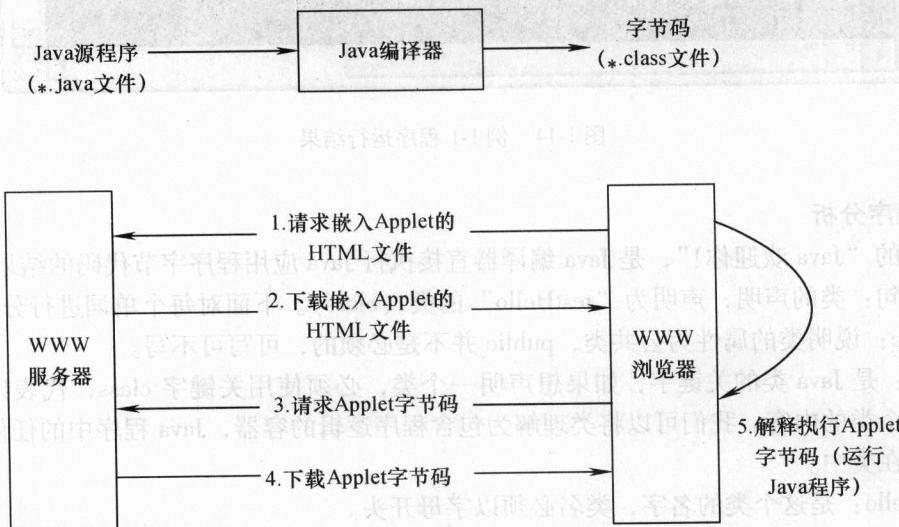


图 1-15 Applet 的执行流程示意图

【例 1-2】 Applet 小程序示例，源程序名为“Hello. java”。显示“你好 Java!”的 Applet 程序。程序清单如下。

1. 创建 Applet 源文件

```
1 import java. applet. * ; //导入 Graphics 类
2 import java. awt. * ; //导入 Applet 类
3 public class Hello extends Applet //Applet 的初始化事件
4 {
5     public void paint( Graphics g)
6     {
7         g. drawString("你好 Java!",10,50); //显示“你好 Java!”
8     }
9 }
```

1) 打开附件中的记事本，在文本编辑界面中键入程序清单。

2) 单击【文件】→【保存】，将文件命名为“Hello. java”，保存到 D:\java > 目录下。

2. 编译 Applet 小程序

Applet 程序 Hello. java 编写完成后，需对其进行编译，自动产生字节码文件 Hello. class，其编译方法与 Java 应用程序相同。

打开 DOS 窗口，输入“javac Hello. java”命令编译生成 Hello. class 文件，如图 1-16 所示。

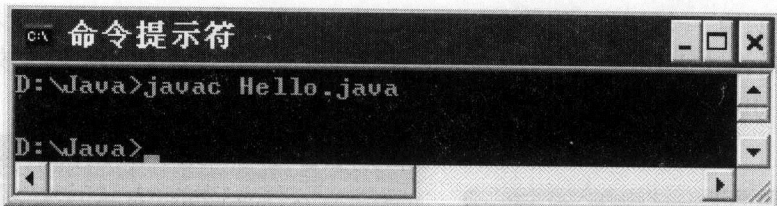
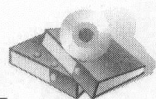


图 1-16 Hello.java 的编辑窗口

3. 编写 HTML 代码

对于 Applet 小程序的字节码程序 Hello.class 必须嵌入到 HTML 代码中, 才可以完成小程序的功能。所以, 还必须为 Hello.class 编写一个 HTML 的代码文件, 将字节码程序嵌入其中。

下面是嵌入 Hello.class 代码的 HTML 程序示例, 文件名为 Hello.html, 如图 1-17 所示。

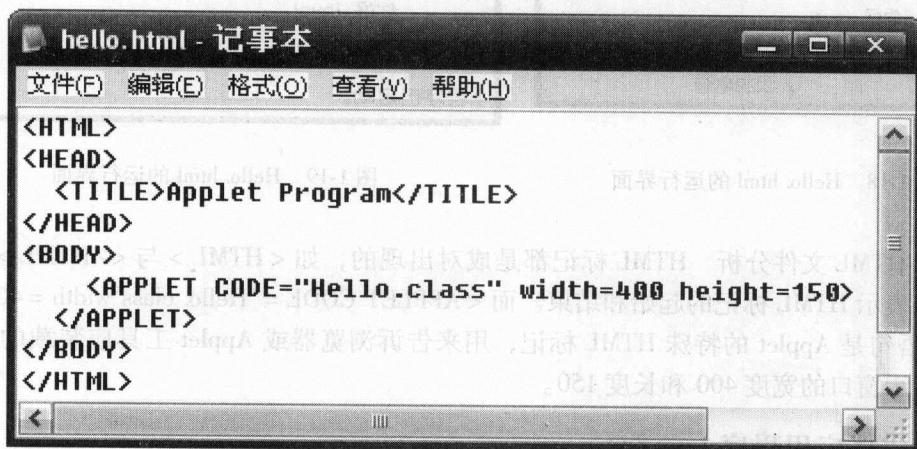


图 1-17 编写 Java 代码的记事本界面

4. 运行 HTML 代码程序

完成 Hello.html 的编写后, Applet 小程序的运行有两种方式: 一种是使用 Internet Explorer 浏览器 (简称 IE) 解释运行它; 另一种是使用 appletviewer 命令运行它。

1) 在浏览器中执行 HTML 程序。双击 Hello.html 文件, 在浏览器中执行小应用程序。运行结果如图 1-18 所示。

2) 打开 DOS 窗口, 输入 “appletviewer Hello.html” 命令, 即使用 Applet 阅读器 (JDK 的 appletviewer) 执行小应用程序, 运行结果如图 1-19 所示。

5. 程序分析

(1) Applet 源文件分析

1) 第 1、2 句是 import 语句, 指出本 Applet 程序所需要的 Graphics 类、Applet 类, 分别在 java.applet.*、java.awt.* 包中。编写 Applet 程序, 通常都要导入 Applet 类。

2) 第 3 句, 由关键字 class 引入类的定义, Hello 为类名, 关键字 extends 说明该类继承 Applet 类, 即定义的 Hello 类是 Applet 的子类, 该类是 public 型。与 Java 应用程序一样, 把一个 public 类存入文件时, 定义类名必须是文件名。编写好的程序存入 Hello.java 文件中。