

21 世纪
高职高专精品教材

◎ 林桂花 主编 杨文艳 林跃进 副主编

Java 编程技术基础

Java BianCheng JiShu JiChu



经济管理出版社

ECONOMY & MANAGEMENT PUBLISHING HOUSE

Java 编程技术基础

主 编：林桂花

副主编：杨文艳

林跃进

经济管理出版社

图书在版编目(CIP)数据

Java 编程技术基础/林桂花主编. —北京: 经济管理出版社, 2006

ISBN 7-80207-502-5

I. J... II. 林... 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(2006)第 006636 号

出版发行: 经济管理出版社

北京市海淀区北蜂窝 8 号中雅大厦 11 层

电话:(010)51915602

邮编: 100038

印刷: 北京银祥印刷厂

经销: 新华书店

策划编辑: 王光艳

责任编辑: 申桂萍

技术编辑: 杨 玲

责任校对: 平 实

787mm×1092mm/16

19 印张 402 千字

2006 年 2 月第 1 版

2006 年 2 月第 1 次印刷

印数: 1 — 6000 册

定价: 35.00 元

书号: ISBN 7-80207-502-5 / F·472

·版权所有 翻印必究·

凡购本社图书,如有印装错误,由本社读者服务部

负责调换。联系地址:北京阜外月坛北小街 2 号

电话:(010)68022974

邮编:100836

前 言

互联网的普及和应用改变了我们的生活、工作和休闲的方式，同时，也彻底改变了传统的编程方式。想一想，在 Windows 环境下编写的程序能够不做修改就直接拿到 UNIX 系统上运行吗？显然不可以，因为程序必须转换成为计算机硬件的机器指令来执行，专门为某种计算机硬件和操作系统编写的程序是不能直接放到另外的计算机系统平台上执行的，至少要做移植工作。

而互联网上的计算机是多种多样的，即异构的，计算机的类型和操作系统都存在差异，如 SUN 工作站的硬件是 SPARC 体系，软件是 UNIX 中的 Solaris 操作系统，而 PC 机的硬件是 INTEL 体系，操作系统是 Windows 或者是 Linux，因此，一般的编程语言如 COBOL、FORTRAN、C、C++ 等编写的程序由于不能跨平台，基本上只能适用于特定的系统；互联网的应用要求程序代码可以通过网络在各种计算机上运行，这就迫切需要一种跨平台的编程语言，使得用它编写的程序能够在网络中的各种计算机上正常运行，Java 语言就是在这种需求下产生的。目前，Java 语言已经成为最流行的网络编程语言之一，许多学校和培训机构都开设了 Java 课程，各个层次的 IT 人员争相学习 Java 技术，Java 语言正逐步成为世界上程序员应用最多的编程语言。

本教材针对高职高专学生的特点，从实用的角度出发，简要介绍了 Java 编程技术基础知识，包括 Java 语言发展状况、Java 语言基础、Java 面向对象特性等；同时侧重编程能力培养，结合 Applet 编程、GUI 编程、多线程、流式 I/O 以及网络编程、数据库编程等知识，推出了许多经典实例，便于学生理解并灵活运用所学编程基础知识。

对于学有余力的同学，在学习本教材的同时，可以配合 Sun 的 Java 认证程序员考试（SCJP，即 275 考试）相关内容同步学习；本课程结束后，建议进一步学习 JSP、Servlet 和 J2EE 的相关知识，进一步了解 Java 编程的系统架构及其实用技术，以符合软件企业对 Java 编程工作岗位的要求。



本书是大连职业技术学院多位专业教师教学实践的总结，由林桂花、杨文艳精心设计了课程体系，组织了丰富的教学内容，杨文艳执手完成了全书内容的书写，邓少华对软件环境进行了配置并完成了后三章程序的编写调试，使本书顺利成稿。

愿本书能引导你顺利开始 Java 编程！

编者

2005 年 12 月

目 录

第1章 Java 编程入门 / 1

- 1.1 Java 语言的发展 / 1
- 1.2 Java 工作原理及其特点 / 3
- 1.3 编写及运行 Java 程序 / 5
- 1.4 Java 程序基本规范 / 10
- 1.5 实训：建立 Java 开发环境 / 16
- 1.6 补充内容：JCreator Pro 的使用 / 18
- 习题一 / 22

第2章 Java 语言基础 / 23

- 2.1 Java 语言的标识符和关键词 / 23
- 2.2 Java 语言的数据类型 / 24
- 2.3 运算符与表达式 / 28
- 2.4 Java 程序的流程控制 / 34
- 2.5 Java 语言中的数组 / 41
- 2.6 字符串的处理 / 44
- 2.7 实训：结构化程序的设计 / 47
- 习题二 / 48

第3章 Java2 面向对象技术 / 53

- 3.1 面向对象技术基础 / 53
- 3.2 类的定义 / 55
- 3.3 对象 / 62
- 3.4 static 关键字 / 65
- 3.5 包与访问权限 / 67
- 3.6 类的继承 / 72
- 3.7 抽象类和接口 / 76
- 3.8 实训：面向对象程序设计 / 81
- 习题三 / 82



第4章 Java Applet / 87

- 4.1 Applet 的含义及编程实现 / 87
- 4.2 Applet 的类层次及生命周期 / 88
- 4.3 Java 图形程序设计 / 92
- 4.4 Applet 多媒体文件操作 / 95
- 4.5 实训: Applet 程序的设计与发布 / 98
- 习题四 / 101

第5章 Java CUI 编程 / 103

- 5.1 用 AWT 编写 GUI 程序 / 103
- 5.2 AWT 布局管理 / 105
- 5.3 AWT 事件处理 / 111
- 5.4 常用的 AWT 组件类 / 116
- 5.5 用 Swing 编写 GUI 程序 / 125
- 5.6 实训: GUI 程序设计 / 132
- 习题五 / 134

第6章 Java 多线程编程 / 135

- 6.1 线程简介 / 135
- 6.2 多线程的两种实现方法 / 137
- 6.3 线程的同步与死锁 / 142
- 6.4 实训: Java 多线程编程 / 146
- 习题六 / 146

第7章 Java 语言异常处理 / 149

- 7.1 什么是异常 / 149
- 7.2 异常的处理 / 152
- 7.3 自定义异常及其使用 / 157
- 7.4 实训: Java 异常处理程序的编写 / 159
- 习题七 / 159

第8章 Java 流式 I/O / 163

- 8.1 文件的输入输出 / 163
- 8.2 节点流类 / 166
- 8.3 过滤流类 / 171
- 8.4 对象的串行化(Serialization) / 176
- 8.5 输入输出流类的继承关系 / 178
- 8.6 实训: Java 流式 I/O 编程 / 180
- 习题八 / 181



第 9 章	Java 网络编程 / 183
9.1	网络编程基础知识 / 183
9.2	基于 URL 的网络编程 / 186
9.3	基于 Socket 的网络编程 / 193
9.4	实训: Java 实现网络通信 / 202
	习题九 / 205
第 10 章	JDBC 编程技术 / 207
10.1	JDBC 概述 / 207
10.2	JDBC 编程 / 209
10.3	实训: Java 访问数据库程序的设计 / 214
	习题十 / 220
第 11 章	用 RMI 技术实现网络聊天室 / 221
11.1	网络聊天室应用背景及 RMI 介绍 / 221
11.2	聊天室总体设计 / 223
11.3	业务流程分析 / 224
11.4	网络聊天室的程序实现 / 225
11.5	网络聊天室程序的部署 / 230
11.6	网络聊天室全部参考程序 / 231
第 12 章	J2EE 技术与网上购物系统实例 / 253
12.1	企业级应用 (J2EE) 概述 / 253
12.2	J2EE 的体系结构 / 254
12.3	J2EE 应用组件 / 256
12.4	网上购物系统的实现 / 257
12.5	网上购物系统的完整参考代码 / 265
附录一	j2sdk1.4 中的部分常用包索引 / 281
附录二	java.lang 包中的接口、类和异常索引 / 283
附录三	课后习题答案 / 287

第 1 章 Java 编程入门

本章主要内容：本章介绍 Java 语言的产生和发展历程，阐述 Java 语言的特点及 Java 跨平台性的实现原理，介绍 Java 作为一种面向对象编程语言的基本编程思路，学习 Java 程序的两种方式：Java application 和 Java applet，掌握 Java 程序的结构和编程规范，学习阅读 Java API 文档。最后通过实践学会安装 Java 开发环境和 API 文档，学会编写小的 Java 程序并生成相应的帮助文档。

1.1 Java 语言的发展

1.1.1 Java 的产生

1991 年，美国 Sun Microsystems 公司的 Green 项目研究小组为了能够在消费电子产品上开发应用程序，积极寻找合适的编程语言。消费电子产品种类繁多，包括 PDA、家电、手机等等，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，存在着平台不同的问题。当时最流行的编程语言是 C 语言和 C++ 语言，但是，对于消费电子产品而言，C++ 语言过于复杂和庞大，并不适用，安全性也并不令人满意。于是，研究小组就着手设计和开发了一种语言——Oak。Oak 是一种精巧而且安全的语言，但是最初在商业上并未获得成功。到了 1995 年，随着互联网的蓬勃发展，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互联网的需要，于是改进了该语言的设计，并给该语言取名为 Java（此名称源于印尼一个盛产咖啡的小岛——爪哇岛）。

Java 语言产生于 C++ 语言之后，是完全的面向对象编程语言，充分吸取了 C++ 语言的优点，采用了程序员所熟悉的 C 语言和 C++ 语言的许多语法，同时又去掉了 C



语言中指针、内存申请和释放等影响程序健壮性的部分，可以说 Java 语言是站在 C++ 语言这个“巨人的肩膀上”前进的。

1.1.2 Java 的发展

Sun 公司于 1997 年 2 月发布了 Java 开发工具包 1.1 版 (JDK1.1)，以全面支持使用 Java 语言进行开发，随后的几年时间里，Sun 公司不断对 Java 语言进行改进，并在其新的 JDK 版本中发布。

由于 JDK1.2 版与之前版本的差别较大，Sun 公司将 JDK1.2 版及其以后版本命名为 Java 语言第二版 (Java 2)，在 2005 年之前的两三年时间里，Java 2 的 1.4 版 (j2sdk1.4) 被 IT 界广泛使用。

2004 年 9 月 30 日，J2SE1.5 发布，对 J2SE1.4 做了许多技术和性能上的改进，为了表示这个版本的重要性，J2SE1.5 更名为 J2SE5.0；2005 年 6 月，JavaOne 大会召开，Sun 公司公开 Java SE 6。此时 Sun 公司公布，Java 的各种版本已经更名，取消了其中的数字“2”（如 JavaSE 6 而不是 J2SE 6）。尽管 JavaSE 6 已经发布，但目前 J2SE1.4 仍在广泛使用。

现在所说的 Java 通常有两种含义：Java 是一种编程语言，也是一种平台。

作为一种语言，Java 是一种跨平台开发语言，能开发出跨平台的应用对象和应用程序。例如：Oracle 8i 的安装程序就是用 Java 开发的。Java 语言具有跨平台、面向对象、多线程、分布式、解释执行、安全、高性能、简单等特点。

作为一种平台，Java 平台包括两部分内容：Java 虚拟机和 Java API。

1.1.3 Java 技术的应用领域

Java 作为一种跨平台开发语言和平台，得到了绝大多数 IT 产业公司、开发人员和产品用户的广泛认可。包括 IBM、Microsoft、Apple、Oracle 在内的许多大公司购买了 Java 的许可证，Java 的开发工具亦日渐丰富，如 Borland 公司的 JBuilder、Sun 公司的 JavaWorkShop、NetBeans、IBM 的 VisualAge、微软的 Visual J++、Xerox 公司的 JCreator、FreeJava 等等，许多数据库厂商如 SyBase、Oracle、DB2 等也都开发了相应的数据库驱动程序以支持 Java 编程。

Java 技术广泛应用于各个领域，按其应用不同，Sun 公司将 Java 分为三个版本：

■ Java SE (即以前的 J2SE)

Java 平台标准版 (Standard Edition)，主要是为台式机和 workstation 提供一个开发和运行的平台。我们在学习 Java 的过程中，主要采用 J2SE 来进行开发。

■ Java EE (即以前的 J2EE)

Java 平台企业版 (Enterprise Edition)，建立在 JavaSE 的基础上，它是 JSP (Java



Server Page)、Servlets、EJB、JTS (Java Transaction Service)、Java mail 以及 JMS (Java Message Service)等多项技术的混合体,主要用于开发分布式、服务器端多层结构的应用系统,如电子商务网站等。

■ Java ME (即以前的 J2ME)

Java 平台小型版 (Micro Edition),主要是面向消费电子产品,为消费电子产品提供一个 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。

本书主要介绍 Java SE 技术,它是进行 Java 编程的基础,最后几个章节简单介绍了 Java EE 技术。建议在学完 Java SE 后继续进行 Java EE 或 Java ME 的学习,以达到软件企业 Java 程序员工作岗位的要求。

1.2 Java 工作原理及其特点

Java 语言之所以得到广泛应用,是由于它具有跨平台、面向对象、安全、多线程、简单易用等许多优越的特性。

1.2.1 跨平台特性

这是 Java 语言最大的优势,与常见的编译型高级语言不同,Java 语言编译器产生的二进制代码是一种与具体机器指令无关的指令集合,只要有 Java 运行时环境存在,编译后的代码便可以在任何平台上运行,Java 运行时环境被称为 Java 虚拟机 (Java Virtual Machine, JVM)。Java 编译器产生的代码 (字节码) 由 Java 虚拟机解释执行,如图 1.2.1 所示。

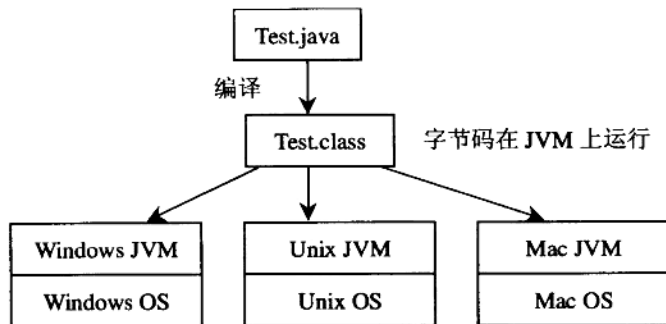


图 1.2.1 JVM 与跨平台性



由此, Java 实现了跨平台性, 即“一次编写, 处处运行 (Write once, Run anywhere)”。

1.2.2 面向对象特性

和所有新一代的程序设计语言一样, Java 也采用了面向对象技术并更加彻底, 所有的 Java 程序和 Applet 程序均是对象。封装性实现了模块化和信息隐藏, 继承性实现了代码的复用。Java 提供了功能强大的类库 (或称为应用程序编程接口 API), 可以帮助编程人员很容易地实现一些复杂的功能。面向对象技术使得应用程序的开发变得简单, 节省代码。

1.2.3 无用内存 (垃圾) 自动回收

程序的执行过程中, 部分内存在使用过后就处于废弃状态, 如果不及时进行无用内存的回收, 就会导致内存占用过多 (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的时候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往往由于程序员的疏忽而致使内存无法回收, 同时也增加了程序员的负担。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。

1.2.4 健壮且安全

Java 在编译及运行程序时, 都要进行严格的检查。字节码的执行需要经过三个步骤: 首先, 由类装载器 (class loader) 负责把类文件 (.class 文件) 加载到 Java 虚拟机中, 在此过程需要检验该类文件是否符合类文件规范; 其次, 字节码校验器 (byte-code verifier) 检查该类文件的代码中是否存在着某些非法操作, 例如 Applet 程序中写本机文件系统的操作; 最后, 如果字节码校验器检验通过, 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。另外, 垃圾回收机制也增加了 Java 的健壮性。

作为网络编程语言, Java 必须提供足够的安全保障, 并且要防止病毒的侵袭。Java 在运行应用程序时, 严格检查其访问数据的权限, 比如不允许网络上的应用程序修改本地的数据。下载到用户计算机中的字节代码在其被执行前要经过核实, 字节代码被核实后由 Java 解释器来执行, 解释器通过阻止对内存的直接访问来进一步提高 Java 的安全性。同时 Java 极高的健壮性也增强了 Java 的安全性。



1.2.5 多线程支持

Java 内置对多线程进行支持。多线程是指一个应用程序同时存在两个或两个以上的执行线索（线程），由于处理器在同一时刻只能执行一个线程，通过处理器在不同的线程之间快速地切换，使多个线程上的任务同时执行。其他语言如 C++ 没有内置的多线程机制，因此，必须调用操作系统的多线程功能进行多线程编程，应用 Java 进行多线程编程更加方便有效。

1.2.6 简单易用

Java 的简单首先体现在精简的系统上，力图用最小的系统实现足够多的功能；对硬件的要求不高，在小型的计算机上便可以很好地运行；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；Java 采用的是相对简单的面向对象技术，去掉了运算符重载、多继承的复杂概念，而采用了单一继承、类强制转换、多线程、引用（非指针）等方式；垃圾回收机制也使得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。

1.3 编写及运行 Java 程序

Java 程序分为 Java application（应用程序）和 Java applet（小应用程序）两种。Java 应用程序是完整的程序，可以独立运行；而 Java 小程序则是嵌入 HTML 编写的 Web 页面中运行的程序，不能独立运行，由 Web 浏览器内包含的 JVM 来解释执行，即必须通过网页调用执行。

虽然 Java 应用程序和 Java 小程序都有各自的使用场合和使用方法，但是他们的开发过程都是一样的，从编写到运行通常要经过三步：

■ 编辑源文件

使用某种文本编辑器（如记事本）编辑好源程序，并保存到以 .java 为扩展名的文件中。

■ 编译源文件，生成字节码文件

使用 Java 编译器工具（javac.exe）编译源文件，得到扩展名为 .class 的字节码文件以便在 JVM 中运行。

■ 运行编译好的字节码文件

对于 Java application，可以通过 java.exe 工具来启动 JVM 以执行字节码文件。

对于 Java applet，则需要嵌入网页（.html 文件），通过网页运行时被调用，在浏



览器（如 IE）中启动 JVM 执行字节码文件。

下面通过两个简单的程序实例，来介绍 Java 程序的基本结构及开发过程。

1.3.1 Java application 实例

【例题 1.3.1】 HelloWorld 应用程序

HelloWorld.java

// 这是我们的第一个 Java application

```
public class HelloWorld{    // 程序员给这个类取名为 HelloWorld
```

```
    public static void main(String args[]){ // main 是类的主方法
```

```
        System.out.println("Hello World! "); // 在命令行打印出字符串 "Hello World! "
```

```
    }
```

```
}
```

在 Java 中，程序都是以类的方式组织的，Java 源文件都保存在以 java 为后缀的 .java 文件当中。每个可运行的程序都是一个类文件，或者称之为字节码文件，保存在 .class 文件中。而作为一个 Java application，类中必须包含主方法，程序的执行是从 main 方法开始的，方法头的格式是固定不变的，即：

```
public static void main(String args[])
```

其中关键字 public 意味着方法可以在类的外部调用，main 方法的参数是一个字符串数组 args，虽然在本程序中没有用到，但是必须明确列出。

■ Java 程序的编辑

Java 程序的编辑可以使用任何一种文本编辑器，例如 UltraEdit、Notepad、Wordpad，然后只要把编辑好的文件存成 .java 文件。当然也可以用一些集成开发环境，例如 Borland 公司的 JBuilder、IBM 公司的 VisualAge 等其他集成开发环境。初学 Java 时，使用 JCreator 进行程序的编辑调试比较简单方便。

图 1.3.1 是用 Notepad 记事本编辑 HelloWorld.java 文件的情况。

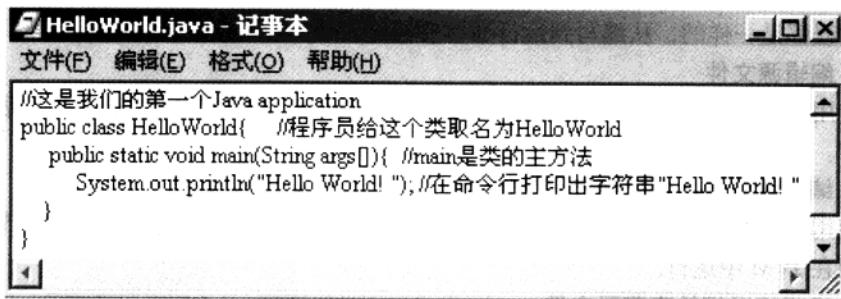


图 1.3.1 在记事本中编辑 Java 源文件



■ Java 程序的编译

Sun 公司为全球的 Java 用户提供了一个免费的 Java 开发包工具包 (Java Develop Kit, 简称 JDK), 其中包括了一系列工具, 如: Java 编译器 `javac.exe`, Java 解释器 `java.exe`, 还有帮助文档生成器 `javadoc.exe` 等。所有这些工具都可以在命令行下运行, 例如我们要编译上述 Java 文件 `HelloWorld.java`, 如果是在 Windows 中进行开发, 就可以在“命令提示符”下进行编译, 在命令行中敲入“`javac HelloWorld.java`”, 如图 1.3.2 所示。

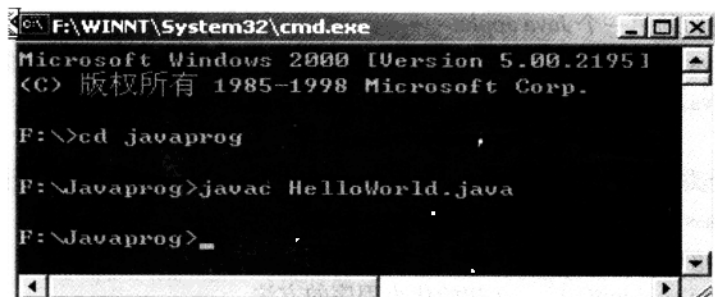


图 1.3.2 在命令提示符下编译 Java 源程序

注意: 如果你对源文件进行了修改, 那么必须重新编译, 生成新的字节码文件。

■ Java application 的执行

当编译结束以后, 在 Java 源文件中的每一个类都会生成相应的 `.class` 文件, 例如上图中就会生成一个 `HelloWorld.class` 文件, 而 Java 程序在执行时调用的是 `.class` 文件。Java application 的执行是在命令行下进行的, 如果是在 Windows 系统中, 就可以在“命令提示符”下敲入“`java HelloWorld`”执行, 该“java”命令会启动 JVM, JVM 读入 `HelloWorld.class` 文件然后执行, 如下图 1.3.3 所示。

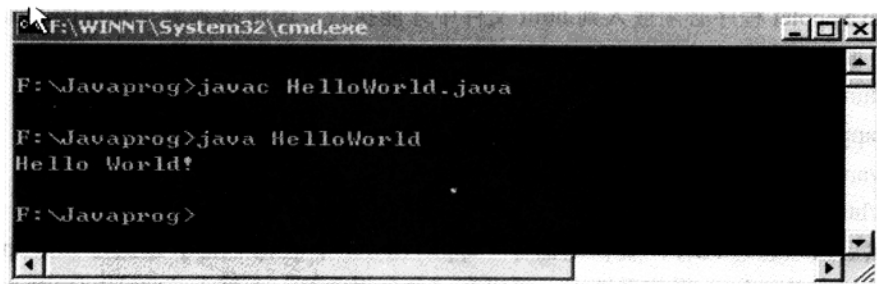


图 1.3.3 编译和运行 HelloWorldApp 程序



1.3.2 Java applet 实例

Java 程序的另一种形式是 Java Applet, Applet 没有 main()方法, 它必须嵌入超文本文件中, 在浏览器中运行。下面这个程序将在浏览器中显示一行字符串。

【例题 1.3.2】 一个简单的 Java applet 小程序

HelloApplet.java

// 这是我们的第一个 Java applet

import java.awt.Graphics; // 在进行显示输出时, 需要用到类 Graphics 的对象;

import java.applet.Applet; //Applet 类是所有的 Java applet 的父类

// 给这个类取名为 HelloApplet, 所有的 applet 程序都是 Applet 类的子类

```
public class HelloApplet extends Applet {  
    String s;  
    public void init() {    // 初始化小程序的方法  
        s=new String("Welcome to Java applet"); // 生成一个字符串对象  
    }  
    public void paint(Graphics g){  
        g.drawString(s,25,25); // 在浏览器中坐标为 (25,25) 的位置显示字符串 s  
    }  
}
```

Applet 程序是从方法 init()开始执行的, 在该方法中完成了对字符串 s 的初始化工作, 而显示功能是在方法 paint()中执行的。paint()方法是类 Applet 的一个成员方法, 其参数是图形对象 Graphics g, 通过调用对象 g 的 drawString()方法就可以显示输出。

■ Java applet 的执行

Java applet 程序的编译方式与 Java application 完全一样, HelloApplet.java 程序经过编译以后就生成了 HelloApplet.class 文件。但其执行方式与 Java application 完全不同, Java applet 程序必须嵌入到 .html 文件中才能执行, 因此必须编写相应的 .html 文件。下面为调用小程序类 HelloApplet.class 的 ***hello.html*** 文件的内容:

```
<html>  
<applet code="HelloApplet.class" height=300 width=300>  
</applet>  
</html>
```

然后可以通过 JDK 所提供的命令 “appletviewer”, 在命令行下面执行 Java applet 程序。如在 Windows 操作系统中, 可以在 “命令提示符” 下敲入 “appletviewer hello.html”, 如图 1.3.4 所示。

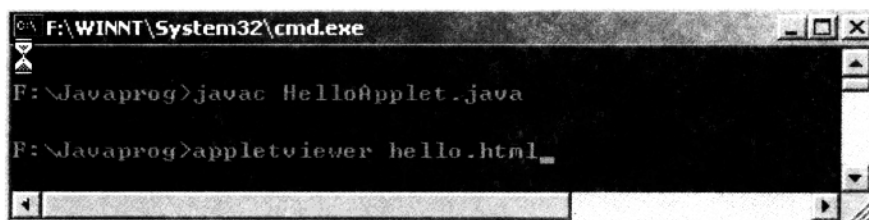


图 1.3.4 在命令窗口下运行小程序

此时，系统会弹出另外一个窗口运行该 Applet 程序，结果如图 1.3.5 所示。

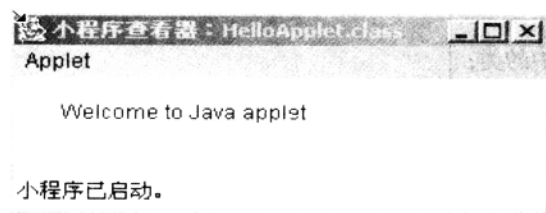


图 1.3.5 小程序运行窗口

Applet 还可以采用另外一种方式运行，那就是直接在浏览器中打开 hello.html 程序，结果如图 1.3.6 所示。



图 1.3.6 IE 中运行的小程序