



21世纪高职高专规划教材·计算机系列

Java应用 开发教程

宋 波 编著

2JA-43



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>

109

21 世纪高职高专规划教材·计算机系列

TP312.54-43
586

Java 应用开发教程

宋 波 编著

電子工業出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书内容分为四部分,Java 语言应用基础、MySQL 数据库应用基础、Java 数据库程序设计、Java 服务器程序设计。本书构建了完整的浏览器/Web 服务器/数据库三层应用结构模式,将面向对象程序设计技术、动态交互式 Web 网页开发技术和 Web 数据库开发技术融为一体。

本书在内容、例题、习题的选择和组织上力求体现先进性、系统性和实用性,可作为高职高专计算机专业 and 高等学校非计算机专业教材,也可以作为 Internet 应用开发人员的参考书。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,侵权必究。

图书在版编目(CIP)数据

Java 应用开发教程/宋波编著. —北京:电子工业出版社,2002.2

21 世纪高职高专规划教材·计算机系列

ISBN 7-5053-7295-5

I .J… II .宋… 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(2001)第 088112 号

责任编辑:童占梅 章海涛

印 刷:北京李史山胶印厂

出版发行:电子工业出版社 <http://www.phei.com.cn>

北京市海淀区万寿路 173 信箱 邮编 100036

经 销:各地新华书店

开 本:787×1092 1/16 印张:16.5 字数:422.4 千字

版 次:2002 年 2 月第 1 版 2002 年 2 月第 1 次印刷

印 数:5 000 册 定价:23.00 元

凡购买电子工业出版社的图书,如有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系。

联系电话:(010)68279077

前 言

一、本书的定位

Java 语言是目前推广速度最快的程序设计语言,它采用面向对象程序设计技术,功能强大且简单易学,特别适用于 Internet 程序设计,目前已经成为应用广泛的 Web 程序设计语言。MySQL 是一个适合于中小型应用、优秀的网络数据库系统。MySQL 具有小巧灵活、简单易学、快速安全等特点,特别适用于网站建设。用 MySQL 构建的数据库可以在 Internet 上进行自由地访问。

目前,单纯编写 Java 语言与 MySQL 数据库的书籍较多,但是将两者有机地结合起来又适用于 Java 应用开发的书籍较少。两者所应用的软件都可以从 Internet 上免费下载,其实验环境的构建在单机和网络环境下都可以实现,且具有软硬件环境投资少、经济实用、构建简单等特点,对各类高等院校的教学与实验都非常适用。

本书在编写上充分体现了简单易用的特点,步骤清晰,内容丰富,并带有许多插图帮助读者理解基本内容,同时对内容的编排和例题的选择作了严格的控制,确保一定的深度和广度。书中的每个例题都配有执行结果插图,并对源程序进行了详细说明,在每章后面附有思考和练习题。学习本书的读者应该对计算机操作有一定的认识,有一门计算机高级语言基础的读者学习本书会感到得心应手。

二、本书的特色

本书在知识体系的构造上,以简明实用、便于学习、反映计算机技术最新发展和应用为特色,突出实用性与应用性。

本书的主要特色是引入了浏览器/服务器体系结构的概念,以面向对象的 Java 语言作为程序开发语言,以简单易学、功能全面、具备网络数据库服务器功能的 MySQL 数据库作为后台数据库,以 JDBC 技术为数据库连接接口,以 Java Servlet 作为 Web 服务器程序设计的 API,构成了一个完整、典型的浏览器/Web 服务器/数据库的三层应用结构模式,从而使读者通过本书的学习,将具有开发基于 Web 的小型应用系统的能力。

另外,本书所使用的所有计算机软件都可以通过 Internet 免费下载,即使读者的计算机没有与局域网或 Internet 相连接,也可以在一台独立的计算机上完成本书所有源代码的编译、执行操作。

三、内容编排

本书内容包括四个部分,第 I 部分 Java 语言应用基础,包括第 1~7 章。第 1~3 章介绍 Java 语言的基本特性及基本语法,包括 Java 语言概述、数据类型、运算符与表达式以及流程控制语句。第 4~6 章介绍 Java 语言基本的面向对象程序设计技术,包括 Java 类与对象、继承与多态、异常处理以及基本输入与输出。通过这 3 章的学习,读者可以对面向对象技术和 Java 语言的面向对象程序设计有较为扎实的了解和掌握。要掌握好 Java 语言并利用它解决实际

问题,还需要熟练掌握 Java 应用程序接口。第 7 章介绍 Web 服务器、Java Applet 小程序以及 Java 工具类。

第Ⅱ部分 MySQL 数据库应用基础,包括第 8~10 章。第 8 章介绍数据库技术的基本概念,数据库系统的组成及作用,数据库系统结构,MySQL 体系结构、主要特点以及安装、管理。第 9 章介绍 MySQL 数据目录管理,创建与删除 MySQL 数据库等。第 10 章介绍结构化查询语言 SQL 在 MySQL 数据库中的应用。通过本部分的学习,读者将对 MySQL 数据库的使用有一个比较全面的了解和掌握,从而为 Java 数据库程序设计打下一个良好的基础。

第Ⅲ部分 Java 数据库程序设计,包括第 11~12 章。第 11 章介绍 JDBC 体系结构、驱动程序的类型、MySQL 数据库的 JDBC 驱动程序以及 JDBC 的主要特点。第 12 章介绍 JDBC API 中主要的类与接口的功能与用途,并对它们所提供的方法做了概括性的介绍,以实例说明了怎样应用这些类与接口进行 Java 数据库程序设计。

第Ⅳ部分 Java 服务器程序设计,包括第 13~15 章。第 13 章介绍 Servlet 的基本概念及 Servlet API,并以两个简单的实例介绍 Servlet 的两种程序设计类型。第 14 章介绍 Java 服务器程序设计,包括 Servlet 生命周期、Servlet 与 HTML 表单、JDBC 在 Servlet 中的应用、在 Servlet 中实现多表联合查询,最后以一个实例介绍如何实现 Servlet 会话跟踪。第 15 章介绍 Java RMI 程序设计,包括 RMI 技术的基本概念,RMI 程序设计的具体步骤,RMI 程序设计中使用的类和接口的概括性介绍,最后以一个实例演示如何综合运用 Java Servlet、JDBC 和 RMI 三种技术实现一个分布式数据库设计。

四、使用指南

本书按照 5 千字/学时的编写原则安排授课学时分配,参考学时数为 72。其中,理论教学的参考学时数为 46,上机实验的参考学时数为 26,每学时 50 分钟。课程结束之后还可以安排 1~2 周的课程设计或实训时间。建议授课学时数分配如下:第 1 章 2 学时,第 2 章 2 学时,第 3 章 2 学时,第 4 章 4 学时,第 5 章 2 学时,第 6 章 4 学时,第 7 章 4 学时,第 8 章 4 学时,第 9 章 2 学时,第 10 章 4 学时,第 11 章 2 学时,第 12 章 4 学时,第 13 章 2 学时,第 14 章 4 学时,第 15 章 4 学时。课程学时数较少的学校,可以用 42 个授课学时只讲第 1~14 章,上机实验的参考学时数为 22 学时,或用 36 个学时讲 1~12 章,上机实验的参考学时数为 18 学时。

对于选择本书作为教材的学校,或者购买本书的读者,可以访问电子工业出版社的网站(<http://www.phei.com.cn>),本书中所有例题的源代码、插图及相关应用软件安装画面的电子文档均可以从那里下载。

五、软件应用环境

- (1) 操作系统:Microsoft Windows 98/2000 Professional
- (2) Java 编译、运行环境:Java 2 SDK1.3.1
- (3) Web 服务器:Apache 1.3.6
- (4) Java Servlet 引擎:Apache Jserv 1.1
- (5) 网络数据库服务器系统:MySQL 3.23.28
- (6) JDBC 驱动程序:mm.mysql-2.0.4-bin

特别感谢东北大学信息科学与工程学院副院长、博士生导师于戈教授为本书的出版撰写了序言,沈阳工业大学信息科学与工程学院许占文教授也对本书的出版提出了许多宝贵意见,

电子工业出版社的童占梅老师对本书出版给予了热情支持。由于作者水平有限,尽管竭力而为之,仍难免有欠妥之处,敬请广大读者和专家批评指正。

宋 波

2001 年 11 月于沈阳

第 I 部分 Java 语言应用基础

第 1 章 Java 语言概述

第 2 章 数据类型、运算符与表达式

第 3 章 流程控制语句

第 4 章 Java 类与对象

第 5 章 继承与多态

第 6 章 异常处理、基本输入与输出

第 7 章 Web 服务器与 Java 工具类

第 1 章 Java 语言概述

Java 是当今流行的新兴网络编程语言，它的面向对象、跨平台、分布式应用等特点给程序设计人员带来了一种崭新的计算概念，使 WWW 从最初的单纯提供静态信息，发展到现在的提供各种各样的动态服务，从而使 Internet 发生了巨大的变化。

本章主要是对 Java 语言进行概括性的介绍，包括它的历史、与 Internet 的关系以及它的几个重要特点，以此来了解 Java 语言所具备的基本特征；同时也将介绍 Java 程序设计所需要的软件环境，Java 程序设计与执行的流程以及 Java 2 SDK 常用工具。

1.1 Java 语言简史

Java 诞生于 1990 年 12 月，是由美国 Sun Microsystems 公司的 Patrick Naughton 和 James Gosling 领导的一组工程师设计的一种小型计算机语言，当时主要是为了在电视、烤面包箱等家用消费类电子产品上进行交互式操作。这就要求这种语言能够产生简短、高效的执行代码，用它编制的程序必须能够运行在不同种类的处理器的上。同时，为了减少使用和维护的成本，这种语言必须是简单和面向对象的。这个项目在 Sun 公司中被称为 Green（绿色项目），而这种语言被称为 Oak，后改称为 Java。

在 1995 年 5 月召开的 SunWorld'95 大会上，Sun 公司发布了用 Java 语言开发的浏览器 HotJava，显示了 Java 在 Internet 上的强大功能。世界上著名的软硬件大公司如 IBM, Borland, Symantec 等都注册了 Java 使用许可证。Netscape 公司从 1996 年发布的 Netscape 2.0 浏览器开始，提供对 Java 语言的支持。即使是微软公司，也注册并宣布支持 Java 语言，IE 浏览器从 4.0 开始就支持 Java 语言，同时 Windows 本身也配套提供了 Java 虚拟机。

Java 与 C 和 C++ 有着直接的联系。Java 的语法是从 C 继承来的，它的对象模型改编自 C++。Java 和 C, C++ 的这种联系是非常重要的，原因是许多编程人员都非常熟悉 C/C++ 语法，这就使得 C/C++ 编程人员可以容易地学习 Java。反之，Java 编程人员也可以轻松地学习 C/C++。尽管 Java 受 C++ 的影响，但是它并不是 C++ 的增强版，它与 C++ 既不向上兼容，也不向下兼容。Java 并不是取代 C++，它是为解决特定的一系列问题设计，而 C++ 是为解决另外的一系列问题设计，两者将并存很长一段时间，读者在今后的学习中会逐渐体会到这一点。

Java 语言目前有 4 个主要的版本：Java 1.0（1995 年）、Java 1.1（1997 年）、带有 SDK 1.2（Software Development Kit）的 Java 2（1998 年）和带有 SDK 1.3 的 Java 2（2000 年）。

除了 Java 开发工具包外，还可以使用各种商业开发工具，比较流行的如 Visual Cafe for Java, Borland JBuilder 等。本书中的程序是在 Java 2 SDK 1.3.1 环境下测试通过的，如果使用 SDK 1.3.1 以外的工具来创建 Java 程序，则需要确保开发工具能够支持 Java 2。

1.2 Java 语言与 Internet

传统的国际互联网，尤其在 WWW (World Wide Web) 的应用上，人们所能接触的信息是静态的，虽然通过图像、声音等多媒体的表现方法，使得传递信息的方法更为友好和丰富，但是在使用者的交互方面却无法找出适当的解决方法。

这个问题的最初解决方法是利用服务器端的程序设计，也就是最初发展出来的 CGI (Common Gateway Interface) 程序设计。网页设计者利用 C 或者 Perl 语言编写所需要的 CGI 程序，编译成 HTTP (HyperText Transfer Protocol) 服务器端可执行的程序。而客户端的浏览器，将客户的输入数据转换为参数传入 HTTP 服务器，HTTP 服务器再将参数传递给对应的 CGI 程序，最后将 CGI 程序的输出再传送给客户，其执行过程如图 1.1 所示。

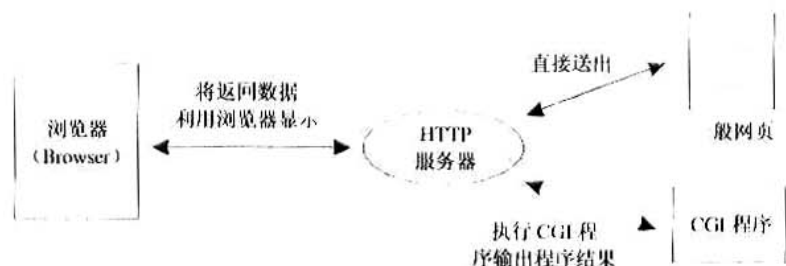


图 1.1 CGI 程序执行流程

但是上述方法有一个明显的缺点，就是这些程序都是由服务器执行完毕后，再将结果传送给客户。当客户数量增加时，服务器的负担将大大加重，因此服务器能够同时响应的客户并不多。对于需要提供复杂服务的应用系统而言，这样的做法显然效率低下。另外，使用 CGI 时，数据都是显式地以字符串参数的形式传送到 HTTP 服务器，没有经过任何编码与加密，传送的数据内容随时可能被他人窃取，不适用于需要保密的商务网络等用途。

要解决上面所提到的问题，就必须在国际互联网应用上有一个能够有效降低服务器负载，提供友好的客户界面，并能够对传送数据加以保密的系统。

Java 语言跨平台的特点使客户可以从网络上下载 Java 程序后直接运行，也就是说，Java 程序可以不加修改就能够在多种平台上运行。Java 编译器生产的字节码 (Byte Code) 是独立于平台的。客户端的浏览器从服务器上下载的就是这种字节码，然后由浏览器上的 Java 解释器解释执行。

现在几乎所有的浏览器，如 Netscape Navigator, Microsoft Internet Explorer 都支持 Java，它们可以下载包含在 Web 网页中的 Java 程序，在本地 Web 客户端系统上运行它。

另外，在数据保密方面，由于采用 Client/Server 体系结构，Java 应用程序可以根据自己的需求发展所需的加密与解密方法，不需要受到传统 CGI 体系结构的限制。

Java 程序在浏览器上的下载及执行过程如图 1.2 所示。

- (1) Java 编译器将 Java 源程序编译成字节码；
- (2) 客户机的浏览器与服务器进行连接，要求下载 Java 字节码文件；
- (3) 服务器将字节码文件通过网络传送给客户机；
- (4) 客户机上的 Java 解释器解释执行字节码；

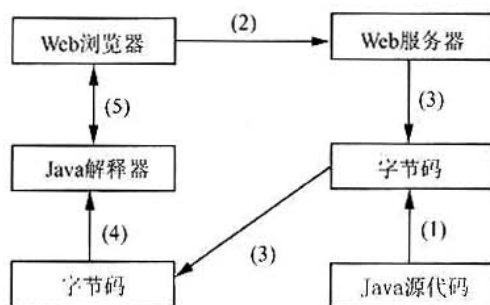


图 1.2 Java 程序的下载及执行过程

(5) 在浏览器上显示并与用户交互。

1.3 Java 语言的特点

Java 是一种定位于网络编程的计算机语言，Java 所具有的特点也是围绕着这一中心展开并为之服务的。另外，Java 语言也充分利用了当代软件技术的最新成果，如面向对象、跨平台、多线程等。限于篇幅，本节将针对 Java 语言的几个重要特点来加以说明。

1. 简单易学

Java 语言去除了 C++ 中不容易理解和掌握的部分，如指针操作、结构体类型、运算符重载、虚拟基础类等，降低了学习难度。而且 Java 的基本语法与 C 语言基本一致，因此对学过 C 语言的人来说感觉很容易入门。另一方面，学习 Java 语言对计算机的硬件环境要求很低。用 Java 语言编制的程序，可以在内存容量很小的计算机上独立运行。Java 语言基本的解释器及类支持模块仅需要 40 KB 的内存，即使再加入基本的标准库以及对线程的支持，也只需要增加 175 KB 的内存。

2. 平台无关性

目前 Sun 公司已经推出了针对 Microsoft Windows 95/98/NT/2000, Linux, Apple Macintosh 及 Solaris 等操作系统平台下的 Java 运行系统。只要安装了针对不同操作系统平台的 Java 运行系统，Java 程序就可以在任意的处理器上运行。Java 运行系统中的 Java 解释器能够生成与操作系统平台体系结构无关的字节码指令，这些字节码指令对应于 Java 虚拟机中的表示。Java 解释器得到字节码后，对它进行转换，使之能够在不同平台上运行。

3. 面向对象

Java 是面向对象的计算机语言。面向对象技术的核心是以更接近于人类思维的方式建立计算机逻辑模型，它利用类和对象机制将数据及其操作封装在一起，并通过统一的接口与外界交互，使反映现实世界实体的各个类在程序中能够独立及继承。这种方法有利于提高程序的可维护性和可重用性，大大提高了开发效率和程序的可管理性，使得面向过程语言难于操纵的大规模软件，可以很方便地创建、使用和维护。编写 Java 程序的过程就是设计、实现类，定义类的属性、行为的过程。

4. 友好的 Web 使用接口

Java 提供了完整的 AWT (Abstract Window Tools) 接口开发工具, 可以让用户比较轻松地设计窗口对象、配置窗口版面、处理窗口事件等。此外, Java 2 中所新增的 Swing 接口对象, 使用户接口的开发更为简单而且美观。

5. 支持动态链接技术

在 Windows 操作系统中, 如果多个应用程序同时使用相同的函数, Windows 将会把这个函数加载到内存中, 这些应用程序可以共同使用内存中的这个函数。每一个应用程序不需要在内存中复制相同的函数, 这就是动态链接的概念。动态链接的最大优点是节省内存的使用。

Java 语言支持动态链接技术。Java 语言的动态链接发生在程序执行时, 当某一个类的内容被更新时, 不会影响与此相关的其他类。也就是说, 用户可以很容易地对一个类添加新的方法和新的实例变量, 可以简单、直观地查询运行时间类型信息, 而不会对运行该程序的客户造成任何干扰。

6. 安全稳定

网络应用程序对安全可靠提出了较高的要求。也就是说, 客户通过网络获取并在本地运行的应用程序必须是可信赖的, 不会充当病毒的传播者而攻击客户本地的资源。Java 语言不支持 C++ 中的指针, 一切对内存的访问都必须通过对象的实例变量来实现, 这样就保证了客户获取的应用程序是安全的。同时, 网络应用程序也应该是稳定的, 不会轻易产生死机等错误。Java 提供了内存垃圾自动收集机制来进行内存的自动管理, 保证了 Java 程序运行的可靠性。

7. 支持多线程

多线程机制使应用程序能够并行执行, 而且同步机制保证了对共享数据的正确操作。通过使用多线程, 可以分别用不同的线程完成特定的行为, 而不需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。

拥有多线程的程序可以充分利用系统资源, 改善 CPU 的闲置状态, 缩短 CPU 闲置时间, 对同一时间运行多个应用程序提供了最佳处理方式。Java 语言支持多线程, 因此 Java 语言设计的交互式应用程序有较快的执行速度。

1.4 编译和运行 Java 程序

为了鼓励程序设计人员用 Java 语言开发软件, Java 语言的发明者 Sun Microsystems 公司提供了免费的软件开发工具包 Java 2 SDK 1.3.1 (截止本书出版时的最新软件版本), 其中包括编译器、虚拟机、调试器等许多有用的开发工具。Windows 版 Java 2 SDK 1.3.1 可以从 Sun 公司的站点 <http://java.sun.com/products/jdk/1.3/> 上获得, 它是免费的。本书所有的例题、习题均是在 Java 2 SDK 1.3.1 环境下编译运行的。

1. 安装和配置 Java 开发工具包

从 Sun 公司站点下载 Windows 版的 Java 2 SDK 1.3.1 软件包, 然后进行安装, 它将自动安装到 C:\JDK1.3.1 下。安装好之后, 还必须修改环境变量 PATH 和 CLASSPATH, 以便将其提供的开发工具所在的路径包含进去。在 Windows 95/98 操作系统下, 可以按以下方式修改 autoexec.bat 文件中的环境变量 PATH:

```
PATH = %PATH%; C:\JDK1.3.1\bin
```

```
SET CLASSPATH=.; C:\jdk1.3.1\lib\tools.jar; C:\jdk1.3.1\lib\dt.jar
```

在 Windows 2000 操作系统下, 可以在控制面板 (Control Panel) → 系统控制台 → 高级 → 环境变量中进行上述完全相同的修改。

重新启动 Windows 操作系统, 在 MS-DOS 命令窗口中输入以下命令:

```
C:\> path
```

如果环境变量 PATH 已经被修改, 应该看到如下显示:

```
PATH=C:\WINDOWS; C:\WINDOWS\COMMAND; C:\JDK1.3.1\bin
```

在 MS-DOS 命令提示符下输入下列命令:

```
C:\> java -version
```

如果出现下列提示信息, 就说明已经成功地安装并配置好 Java 2 SDK 1.3.1 了。也就是说, 我们可以编译和运行 Java 程序了。

```
java version "1.3.1"
```

```
Java(TM) 2 Runtime Environment, Standard Edition (build 1.3.1-b24)
```

```
Java HotSpot(TM) Client VM (build 1.3.1-b24, mixed mode)
```

2. 编辑 Java 源程序

Java 程序分为两种类型: Java Application 和 Java Applet。Java Application 必须在所处的操作系统下执行, 而 Java Applet 必须与超文本标记语言 HTML (HyperText Markup Language) 配合使用, 由客户端的 Web 浏览器内部所包含的 Java 解释器来解释运行。

一般计算机高级语言的编程需要经过源程序编辑、目标程序编译生成和可执行程序运行几个步骤。Java 编程也不例外, 可以分为编辑源程序、编译生成字节码和解释运行字节码几个步骤。Java 源程序是以 .java 为后缀的文本文件, 可以使用各种 Java 集成开发环境中的源代码编辑器编写, 也可以使用 Windows 95/98/2000/NT 中的记事本或 MS-DOS 中的 EDIT 软件等编写。

3. 编译生成字节码

计算机高级语言程序从源代码到目标代码的生成过程称为编译。Java 源程序代码经过编译所得到的目标代码称为字节码。字节码是二进制文件, 由 Java 解释器来解释执行字节码。Java 2 SDK 1.3.1 提供的 Java 编译器程序为 javac.exe, Java 解释器程序为 java.exe。

例如, 要编译 Java 源程序 MyClass1_1.java, 就需要在 MS-DOS 窗口中键入如下命令:

```
C:\USER> javac MyClass1_1.java
```

这个命令的功能是调用 Java 2 SDK 1.3.1 软件包中 Java 编译器程序 javac.exe, 检查源代

码文件 `MyClass1_1.java` 中是否有语法错误并生成相应的字节码文件。如果有语法错误, Java 编译器就会在屏幕上输出这些错误所在的源代码的行号和错误信息; 否则, 编译成功并生成字节码文件 `MyClass1_1.class`。

4. 解释与运行字节码

计算机高级语言按照执行模式可以划分为编译型和解释型两种。编译型计算机语言如 C, C++ 等, 生成的字节码经链接后就成为可以直接执行的可执行代码; 解释型计算机语言如 Java, BASIC 等, 其编译后生成的字节码不能直接在操作系统上运行, 需要有一个专门的解释器程序来解释执行。

计算机网络是由基于不同软硬件平台的计算机组成的, 为了使这些计算机都能够顺利地运行编译型应用程序, 必须为各种不同平台开发出不同版本的应用程序, 这对依赖于操作系统的编译型计算机语言来说将是一件非常困难的事情。

Java 是一种解释型语言, 其执行速度与编译型语言相比较慢, 但 Java 语言有一个与平台无关的重要优势。Java 源代码编译生成的字节码不能直接运行在一般的操作系统平台上, 而必须运行在一个称为“Java 虚拟机 (Java Virtual Machine)”的、在操作系统之外的软件平台上。所谓“Java 虚拟机”, 就是负责解释并且执行字节码的程序, 如 Sun 公司提供的 Java 2 SDK 1.3.1 程序。在运行 Java 程序时, 应该先启动这个虚拟机, 然后由它来负责解释执行 Java 的字节码。这样, 利用 Java 虚拟机就可以把 Java 字节码程序跟具体的软硬件平台分隔开来, 只要在不同的计算机上安装针对其特定平台特点的 Java 虚拟机, 就可以把这种不同软硬件平台的具体差别隐藏起来, 使得 Java 字节码程序在不同的计算机上能够面对相同的 Java 虚拟机, 而不必考虑具体的平台差异, 从而实现了真正的二进制代码的跨平台可移植性。Java 语言的运行机制如图 1.3 所示。

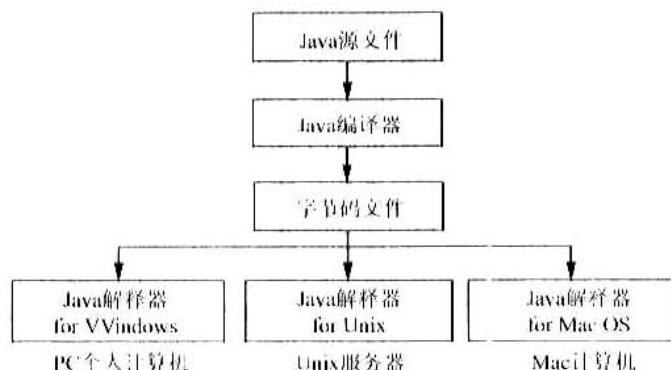


图 1.3 Java 语言的运行机制

当前在 Internet 上使用的主要浏览器, 如 Netscape Navigator, Microsoft Internet Explorer, 在程序内部都嵌入了 Java 虚拟机, 因此, 每当从 HTTP 服务器下载了 Java 的字节码后, 便将字节码的内容转交给虚拟机执行, 然后将程序执行结果显示在网页上。无论是 Sun 公司的 Java 2 SDK 1.3.1 开发工具, 或是这两大网页浏览器, 都有许多不同平台的版本, 因此 Java 的跨平台能力才得以完全的实现。

1.5 Java Application 程序简介

Java Application 是完整的程序，需要独立的 Java 解释器来解释执行。在 Windows 95/98/2000/NT 操作系统下必须进入 MS-DOS 状态下才能执行。

下面举一个实例来说明 Java Application 程序的编译、执行步骤，并对程序结构进行简要的分析，使读者对 Java Application 程序有一个初步的认识。

【例 1.1】编译并运行下列 Java Application 源程序。

```
1: // MyClass1_1.java
2: import java.io.*;
3: public class MyClass1_1
4: {
5:     public static void main(String args[] )
6:     {
7:         String s="Java 语言!";
8:         System.out.println("欢迎学习"+s);
9:     }
10: }
```

在包含 MyApplication.java 的目录中，执行以下命令：

```
javac MyClass1_1.java
```

如果一切工作正常，控制台上应该没有任何输出信息。如果编译成功，Java 编译器将 MyClass1_1.java 源代码翻译成字节码，并存储在一个新的名为 MyClass1_1.class 的类文件中。需要注意的是，Java 语言是区分大小写的，不要混淆大小写字母。不改变当前目录执行 java MyClass1_1 命令，此时在 MS-DOS 命令窗口显示这样的文字：“欢迎学习 Java 语言！”，图 1.4 显示了这一运行结果。

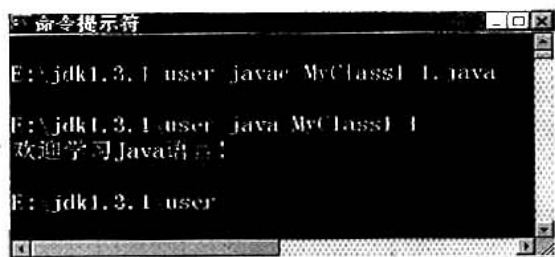


图 1.4 【例 1.1】的运行结果

【程序说明】

(1) Java 源代码中的每一行前面的行号是为方便解释而附加的，在真正的 Java 程序中不应该存在。在 Java 程序中，两斜线 (//) 代表注释行的开始，跟在它后面的所有内容都将被编译器忽略而作为提高程序可读性的注释部分。Java 语言提供了三种方法，对 Java 程序代码进行注释：

① 单行注释：以//开始，到行尾结束。例如：

//第一种方法: 单行注释

(2) 多行注释: 以/*开始, 可以跨越多行, 最后以*/结束。例如:

/*第二种方法:

多行注释*/

(3) doc 注释: 以/**开始, 可以跨越多行, 最后以*/结束, 主要是 Java 2 SDK 1.3.1 为支持 javadoc 而设置的一种注释方法。javadoc 可以由 Java 语言的源程序文件产生出 HTML 的说明文件, 内容包含有 Java 源程序中所用到的 API 说明等。例如:

/**第三种方法:

doc 注释*/

(2) 第 2 行中的 import 语句相当于 C 或 C++语言的#include 描述, 用来指定后面程序中将要用到的类所来自的父类或从哪个类继承而来, *为通配符, 代表多个字符。

(3) 第 3 句中的关键字 class 说明一个类定义的开始。类定义由类头部分(第 3 句)和类体部分(第 4—10 句)组成。类体部分的内容由一对大括号括起来, 在类体内部不能再定义其他的类。任何一个 Java 程序都是由若干个这样的类定义组成的, 就好像任何一个 C 程序都是由若干个函数组成一样。这里只定义了一个类 MyClass1_1。

需要特别注意的是, Java 源程序的文件名必须与类名完全一致。

(4) 在类体中通常有两种组成成分: 一种是域, 包括变量、常量、对象数组等独立的实体; 另一种是方法, 类似于函数的代码单元块。这两种组成成分称为类的成员。

(5) 类 MyClass1_1 中只有一个类成员即方法 main(), 这是 Java 程序的主函数。第 5 句定义了这个 main 方法的方法头, 第 6—9 句是 main()方法的方法体部分。用来标志方法头的是一对小括号, 在小括号前面并紧靠左括号的是方法名, 小括号里面是该方法使用的形式参数, 这个参数是一个字符串类型的数组, 用来接收用户从命令行输入的内容。在 Java 语言中, 字符串类型虽然不属于基本数据类型但其定义方法与整型、浮点型等基本数据类型相同。方法名前面是用来说明这个方法属性的修饰符, 其具体语法规则将在后续章节中介绍。方法体部分由若干以分号结尾的语句组成, 并由一对大括号定界, 在方法体内部不能再定义其他的方法。

(6) main()方法是所有 Java Application 程序执行的入口点, 任何一个此种类型的 Java 程序有且仅有一个 main 方法, 并且这个 main 方法必须按照本例的格式书写。当执行时, 整个程序将从这个 main 方法的方法体的第一个语句开始执行。

(7) main()方法有两个语句。第 1 个语句定义了一个字符串对象 s, 第 2 个语句通过连接操作符“+”把字符串“欢迎学习”与字符串对象 s 连接起来, 组成一个字符串“欢迎学习 Java 语言!”, 并输出到系统的标准输出设备上。System 是系统内部定义的一个对象; out 是 System 对象中的一个域, 也是一个对象; println 是 out 对象的一个方法, 其作用是向系统的标准输出设备显示其形式参数指定的字符串, 并回车换行。

1.6 Java Applet 程序简介

Java Applet 是另一种非常重要的 Java 程序, 它的源代码编辑与字节码的编译生成步骤与 Java Application 完全相同。它不是可以独立运行的程序, 它的字节码文件必须嵌入到用 HTML 语言编写的文件中, 并由负责解释 HTML 文件的 Web 浏览器充当其解释器, 来解释

执行 Java Applet 的字节码文件。

下面举一个实例来说明 Java Applet 程序的编译、执行步骤，并对程序结构进行简要的分析，使读者对 Java Applet 程序有一个初步的认识。

【例 1.2】编译并运行下列 Java Applet 源程序。

```
1: // MyClass1_2.java
2: import java.awt.Graphics;
3: import java.applet.Applet;
4: public class MyClass1_2 extends Applet
5: {
6:     public void paint(Graphics g)
7:     {
8:         g.drawString("欢迎学习 Java 语言!",10,30);
9:     }
10: }
```

在包含 MyClass1_2.java 的目录中，执行以下命令：

```
javac MyClass1_2.java
```

如果一切工作正常，控制台上应该没有任何输出信息。Javac 程序即 Java 编译器，将源代码翻译成字节码。如果编译成功，它将 MyClass1_2.java 源代码翻译成字节码，并存储在一个新的名为 MyClass1_2.class 的类文件中。

打开编辑器输入名为 MyHtml1_1.html 的文件源代码。

```
1: <HTML>
2: <TITLE>My Java Applet</TITLE>
3: <BODY>
4: <APPLET CODE="MyClass1_2.class" WIDTH=300 HEIGHT=100>
5: </APPLET>
6: </BODY>
7: </HTML>
```

在同一目录下执行如下命令：

```
appletviewer MyHtml1_1.html
```

此时应该弹出一个如图 1.5 所示的窗口，显示这样的文字：“欢迎学习 Java 语言！”。

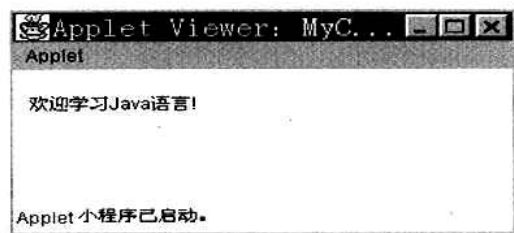


图 1.5 【例 1.2】Java Applet 的运行结果

【程序说明】