



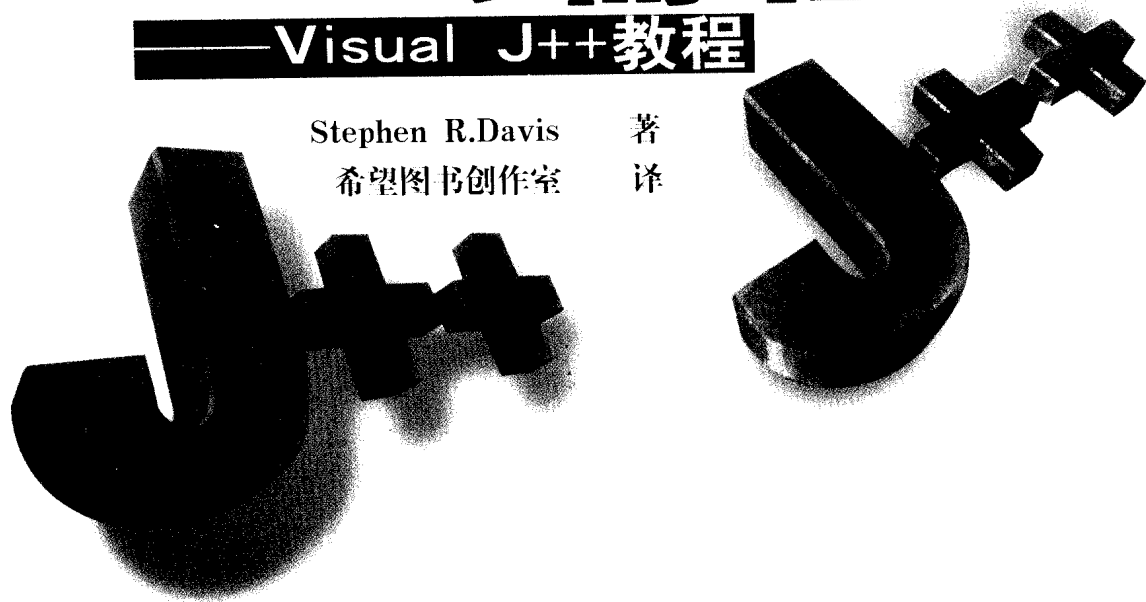
Teach Yourself Java Programming
the Quick and Easy Way with Microsoft Visual J++

学习 J++ Java 编程

——Visual J++教程

Stephen R.Davis
希望图书创作室

著
译



2JA

科学出版社
龍門書局

Microsoft Press

内 容 简 介

本书是 Java 语言的入门书籍。它全面地介绍了 Java 语言特点、基本概念,并通过大量的例子深入浅出地帮助你学习编写 Java 程序。

本书突出的特点是作者深入浅出、循序渐进的论述风格和构思精巧的例子。全书分为两大部分,共十七章;第一部分介绍如何编写 Java 应用程序,其中重点介绍了 Java 的类、对象引用、类的继承、程序包、例外处理和文件 I/O 等内容;第二部分介绍如何编写 Java 小应用程序,其中重点介绍了用 AppletWizard 快速创建小应用程序、事件的处理、多线程、动画技术、抽象窗口工具等内容。

本书适合于计算机用户、大专院校师生,以及所有对 Java 编程语言感兴趣的读者阅读参考。

欲购本书或欲得到有关本书技术支持的读者,请与北京海淀 8721 信箱书刊部联系(邮编:100080)。电话:010-62562329,010-62541992,或传真至 010-62561057。

版 权 声 明

本书英文版名为《Learn Java Now》,由 Microsoft Press 出版,版权归 Microsoft Press 所有。本书中文版由 Microsoft 授权出版。未经出版者书面许可,本书的任何部分不得以任何手段复制或传播。

学习 Java 编程

——Visual J++ 教程

Stephen R. Davis 著

希望图书创作室 译

燕卫华 校

责任编辑:汪亚文

科学出版社 出版
龙 门 书 局

北京东黄城根北街 16 号

邮政编码:100717

北京兰空印刷厂印刷

新华书店北京发行所发行 各地新华书店经售

*

1998 年 4 月第 一 版 开本:787×1092 1/16

1998 年 4 月第一次印刷 印张:18 1/4

印数:1~5000 字数:410 000

ISBN 7-03-006456-9/TP·905

定价:25.00 元

感谢我的书迷妻子 Jenny 和儿子 Kinsey,
没有他们的支持和耐心,我不可能完成本书。

译者前言

Visual J++ 是 Microsoft 推出的最新 Java 集成开发工具,它与 Visual C++、Visual SourceSafe 等共用一个开发环境。也就是,如果你曾经是 Visual C++ 的用户,那么,Visual J++ 对你就并不陌生。而且,考虑到 Java 与 C++ 的诸多相似性,在 Visual J++ 中编写 Java 程序,会让你感觉到从 C++ 向 Java 的过渡是多么的平稳。

使用过 Visual J++ 正版软件的用户一定都知道在该软件中包含一个在线英文教程 Learn Java Now。Microsoft 在发行该教程英文电子版的同时,也按打印方式出版了该书。本书就是该教程的中文版,翻译为《学习 Java 编程——Visual J++ 教程》。Visual C++ 的用户一定知道希望公司与科学出版社联合出版的《Visual C++ 4.0 教程》,这是 Visual C++ 的在线教程的中文版。如果《学习 Java 编程》是伴随你走进 Visual C++ 的伙伴,那么,如果你想学习 Visual J++,请选择《学习 Java 编程》。如果说《Visual C++ 4.0 教程》偏重于介绍完成一些典型应用的实现方法和步骤,那么,《学习 Java 编程》的风格则是 Java 基本语法、概念和原理与实现方法和步骤并重。所以,不管你是 Java 语言的初学者,还是有一定经验的 Java 程序员,《学习 Java 编程》都将使你受益。

本书由希望图书创作室策划,并交克立兹软件研究所翻译。由于译者水平有限,错误在所难免,恳请读者指正。

鸣 谢

本书的封面署名只有我的名字是大大的误导。编辑、技术审查员以及印制人员对书的出版是必不可少的,本书也不例外。首先我要感谢编辑 Ina Chang 和 Jim Fuchs。我也要感谢 Ted Chiang、Sara Tulli 和 Stephen Horne 耐心的技术投入。没有我的代理人 Claudette Moore 和组稿编辑 Eric Stroo 的前期工作,我的书不可能出版。最后,我要感谢在第十六章中负责制作漂亮图表的图表艺术家 Michael Vitor。

我的其它书的读者知道,如果没有家里的宠物在窗口频繁出现,我是无法写作的。我悲伤地通知大家,经常到我家前院来的兔子 Beavis 和 Butthead 已经去了大牧场,小猫 Marley 去年冬天死于白血病。不过,更令人高兴的消息是,另一只猫 Bob 去年冬天生了 5 只健康、快乐和充满好奇心的小猫。另外,原来 Davis 动物家族的成员狗 Trude 和 Scooter 仍旧很健壮,虽然 Scooter 已经太老,在没人观赏的时候,已无兴趣追逐汽车了。最后,我要介绍院子家族中的最新成员,公寓猪 Penny(还好 Penny 的公寓管理员愚蠢之极,居然没有发现在公寓中养猪的明智,因而在很短的时间内就申请领养了 Penny——在管理员带着屠夫回来之前)。

尽管我和编辑们都尽了很大努力,但若书中仍然存在一些错误,我将承担全部责任。

引言

万维网(World Wide Web)如暴风雨般席卷了全世界。电视广告末尾经常出现广告商的主页地址。甚至像CNN公司的夜间商业报道这样正式的节目中,也会邀请观众在Web上加入他们,仿佛全美的家庭访问Web就像给祖母打电话那样容易。

网页飞速发展,每个页面都向公众惊呼“看着我!”这种观众参与的竞争使Web页面的设计者们越来越精于如何设计他们的页面。竞争越激烈,设计出的产品就越好,就像人们希望的那样。但是,用一种静态的HTML(超文本描述语言)页只能做到这些。程序设计者们立刻开始探索是否能将Web页面设计成为交互式的页面,也就是使这些页更聪明。

猛一看,这似乎不成问题。人们可以用他们喜欢的语言(比如C、C++、Visual Basic,或任意一种)编写程序,编译程序,并且将可执行文件存储在带有HTML页面的服务器上。只要对HTML语言进行小小扩充,就可以允许Web浏览器自动从服务器上装载这个程序,并在客户计算机执行。这个程序能够打开窗口,弹出对话框,唱歌,跳舞等等。瞧!多么聪明的web页面。

然而,这个计划存在几个问题。首先,传统的编程语言不是为在Web上执行而设计的。它们甚至不具备完成最常用的网络操作所需要的特性。程序员不得不编写自己的函数来实现最基本的网络操作,例如从服务器上装载一个图像文件,并在客户机上显示。用传统的非Web语言编写的Web程序很快就显得很零散。此外,由程序员各自实现基本网络操作毫无疑问会对同一问题引出许多不同的解决方法。当程序员们试图阅读并理解其他人的程序时,将发现这很复杂。

另外,传统的程序会生成相当大的可执行程序。当然,“大”是相对的。当你从硬盘驱动器上执行一个500KB的程序时,没有什么值得兴奋的。这种程序在你的手指离开鼠标时就加载并且准备执行了。然而,即使用一个28.8kbps的调制解调器,通过电话线来载入这样的程序也需要超过2分钟的时间,这还是在没有因线路忙而引起的通信延迟的情况下。

其次就是安全问题。难道你真的想让你的浏览器不受自己控制地在网络上执行传统的程序吗?一旦某个Visual Basic程序被下载到客户机并开始执行后,它就会为所欲为。在传统的环境下,这并不是个问题。如果我写了一个清除硬盘的程序,并执行它,(猜猜怎么样?)它将我的硬盘清除了,除了我自己,我能责怪谁呢?(也许我可以怪那个编译器的制造者?)另一方面,如果我偶尔看了一眼某个家伙的主页,那个主页下载了一个清除硬盘的程序到我的计算机上,这下我可是有理说不清了。

在计算机上执行清除硬盘这个操作之前,浏览器可能提示说:“即将下载具有潜在危险的程序,该程序会清除您的计算机,浏览器不对发生的事负责,继续吗?”这时,你该怎么办?面对这种威胁,几乎没人敢执行这样一个聪明的网上程序。

最后,不同的运行环境也是个问题。与Web相连的是各种不同型号的计算机,我们创建的这些可执行程序支持哪种计算机呢?当然,毫无疑问,与网络相连的大部分家用计算机是PC机,但还有许多苹果机、UNIX机等等。当然我们可以设计某种方案,使浏览器能够区分主机类

型与服务器。当客户机是 PC 机时,传送这个可执行程序,当客户机是苹果机时,传送另外一个程序。即使是这样,还远远不能满足用户需求。在 PC 机中,人们使用 Microsoft Windows 3.1、Microsoft Windows 95、Microsoft Windows NT 以及 OS/2 等作为操作系统,谁知道将来又会出现什么操作系统取代以上这些呢?为了支持所有可能的客户机环境,这种多种可执行程序的解决方法,使 Web 程序员增加了沉重的负担。

0.1 Java 是什么

Java 语言被设计成可以解决以上所有的问题的编程语言。它包含对 Web 的内嵌支持,把程序员从必须以特殊的方法解决常见的网络问题的负担中解脱出来。它生成非常小的可执行程序以便更快地在可能很慢的通信线路上传输。Java 加强了安全性——Java 程序不能访问任何客户机没有特别允许访问的东西。Java 是与机器无关的。同样的程序可以在 PC 机、苹果机,甚至 UNIX 机上执行(只要该机器上有支持 Java 的浏览器)。

此外,Java 本身是一种简单、功能强大的语言。它是面向对象的语言。它的语法鼓励程序员编写模块化的、可维护的程序。对于初学者来说,Java 很像另一种面向对象的,C++ 语言。毫无疑问功能强大的 C++ 语言的确对 Java 语言有很大的影响,但是,Java 在某些基本思路上不同于 C++ 语言。

C++ 的主要目标之一是与 C 语言向下兼容,而 C 语言根本不是面向对象的语言。当 C++ 决定通过支持遗留的代码(指 C 语言)来确保它被接受时,就已付出了高昂的代价。C++ 语言非常复杂,在它绞尽脑汁地与一种几乎 30 年以前设计的语言相互兼容时,就陷入了某种语法困境。

Java 的设计者们更喜欢简明决定放弃与 C 语言的向下兼容。这就允许 Java 语言采用了更简单、更一致的样式,某些方面类似于 Smalltalk 语言(如果你并不熟悉 C++ 或 Smalltalk 语言,别担心,本书并不要求)。无论有无 Web,Java 语言都是一种用途广泛、博采众长的好语言。

0.2 本书为谁而写

本书并不是关于制作万维网主页的。假设你已经知道了怎样访问和使用 Web。所有这些表明你已经知道哪个是你的浏览器的工作端。我没有假设你已经彻底了解了 HTML。(我所教给你的只是使你的小应用程序安装在 HTML 页面上并执行它所需最少的知识。

我会教你使用 Microsoft Visual J++ 开发环境进行 Java 编程。我假设你已经熟悉编程概念,但是我并没有假设你对某种特定的语言很熟悉。我欢迎 C、C++ 以及 Visual Basic 的程序员学习本书,即使是汇编程序员也可以。

0.3 本书的约定

我认为采取一种标准编码风格会使程序更易于理解,而且更易于写程序。在我的编程风格中,大量地采用了 Microsoft 的匈牙利命名法则,如表 0-1 所示:

表 0-1 程序代码约定

约定	举例
类名以大写字母开头。	<code>public class Student;</code>
对象名以小写字母开头,只要有可能就使用类名或类名的一部分。	<code>Student student;</code>
原始类型,以指明其类型的一个字母开头(唯一例外的是使用 I 和 J 作为循环计数)。	<code>Boolean bFlag;</code> <code>byte yField;</code> <code>char cOneChar;</code> <code>int nValue;</code> <code>long lValue;</code> <code>float fVlaue;</code> <code>double dValue;</code>
数据成员前加 <code>m_</code> 。	<code>int m_nAnIntMember;</code>
函数以大写字母开头,前面不加类型符号。	<code>public void Afunction();</code> <code>public int SomeOtherFunction();</code>
对象名为多个单词的组合,每个单词开头都用大写。	<code>char cOneChat;</code>
括号用于每个控制结构之后,即使只有包含一条句,也要用括号括起来。括号符合控制结构(这两者 Java 都不要求用)。	<code>if (nA > nB)</code> <code>{</code> <code> nA = nB;</code> <code>}</code>
成员函数放于数据成员前(只是为了查找的方便)。	不适用
类(静态)成员放在目标(非静态)成员前(也只是为了查找的方便)。	不适用

C/C++:许多读者很熟悉 C 和 C++ 语言。虽然 Java 是另外一种不同的语言,但它们的相似性是很明显的。当我把 Java 与 C、C++ 语言比较、对照时,你可以看到图标 C/C++。不过,只有在对理解 Java 有帮助时我才这样做。

0.4 本书的各部分

Java 可生成两种类型的程序。独立的、与 Web 无关的程序,称为 Java 应用程序,简称 app。编制 app 程序时,Java 仅仅是一种简单的面向对象的语言。Java 真正“闪光”的地方是:使用 Java 生成作为 Web 页面的一部分执行的程序。这些程序称为小应用程序(applet)。Visual J++ 的 AppletWizard 能轻而易举地生成功能强大的小应用程序。

当然,每个人都想直接学习编写小应用程序。但是跑之前一定要先学会走。所以本书分为两部分。

第一部分,“过滤 Java:编写 Java 应用程序”,介绍 Java 语言的基础,包括变量类型、控制结构、Java 的类、构造函数、继承、面向对象编程、异常以及文件 I/O。

第二部分,“快速生成 Java 小应用程序:使用 AppletWizard”,介绍怎样使用 Visual J++ AppletWizard 生成高质量的小应用程序。其内容包括处理事件、从 HTML 页面中读取参数、多线程、动画及声音。这一部分的最后一章说明怎样使用 AWT(抽象窗口工具箱),这是一组 Java 的类的集合,可以用来生成诸如对话框和菜单的复杂且与平台无关的界面。

本书最后有两个附录。附录 A 为安装指令,可以帮助你安装运行。附录 B 包含 Java 语言的创始者——Sun Microsystems 公司编制的几个有趣的程序。

0.5 其它信息源

Java 最好的信息源当然来自于 Web。表 0-2 列出的几个 Web 站点是有用的学习起点。

表 0-2 Web 上的 Java 信息

站点	资源
//www.javasoft.com	该站点由 Java 语言的创始者创办,包含着正宗的 Java 信息。
//www.microsoft.com/visualj/	该站点由 Visual J++ 的开发者创办,是 Visual J++ 最好的专用信息源
//www.gamelan.com	Gamelan 是 Java 信息的交换站,它基本上维持着与其它一序列 Java 站点的链接

目 录

第一部分 过滤 JAVA:编写 JAVA 的应用程序

第一章 第一个 JAVA 程序	(2)
1.1 HELLO, WORLD	(2)
1.2 这个程序是怎样运行的	(3)
1.3 VISUAL J++的作用	(3)
1.4 即时编译	(4)
1.5 小结	(5)
第二章 我可以编写一个简单的语句吗	(6)
2.1 JAVA 表达式	(6)
2.2 注释	(6)
2.3 声明变量	(7)
2.4 平滑的运算符	(9)
2.5 流程控制	(13)
2.6 例程: APP1_2	(17)
2.7 小结	(19)
第三章 JAVA 类简介	(20)
3.1 定义一个类	(20)
3.2 类是什么	(22)
3.3 对象	(25)
3.4 例程: 一个简单的银行帐户	(26)
3.5 小结	(29)
第四章 继续介绍 JAVA 的类	(30)
4.1 重载方法	(30)
4.2 THIS 是什么	(32)
4.3 类的成员	(33)
4.4 访问控制 (ACCESS CONTROL)	(38)
4.5 小结	(40)
第五章 对象引用	(44)
5.1 引用的作用	(44)
5.2 什么时候对象拥有对象	(46)
5.3 使用引用	(46)
5.4 清除丢失的对象: 垃圾回收	(51)

5.5	使用数组.....	(53)
5.6	小结.....	(56)
第六章	正确初始化一个对象	(57)
6.1	问题.....	(57)
6.2	构造函数和 JAVA 的解决方案	(58)
6.3	关于静态数据.....	(64)
6.4	小结.....	(68)
第七章	扩展当前的类:继承.....	(69)
7.1	什么是继承.....	(69)
7.2	继承举例.....	(71)
7.3	重设父类方法.....	(72)
7.4	实现的细节.....	(78)
7.5	接口.....	(82)
7.6	最后的银行帐户例子.....	(84)
7.7	小结.....	(88)
第八章	真空包装的 JAVA:程序包	(89)
8.1	建立程序包.....	(89)
8.2	JAVA 的库程序包.....	(92)
8.3	小结.....	(97)
第九章	使用例外处理错误	(98)
9.1	为什么使用例外.....	(98)
9.2	抛出例外.....	(99)
9.3	捕获例外	(100)
9.4	例外的好处	(102)
9.5	创建你自己的例外类	(103)
9.6	处理多个例外	(105)
9.7	FINNALLY	(108)
9.8	小结	(109)
第十章	文件 I/O	(110)
10.1	使用标准的 I/O 对象	(110)
10.2	创建你自己的文件 I/O 对象	(113)
10.3	更复杂的文件 I/O	(114)
10.4	小结.....	(116)

第二部分 快速生成 JAVA 小应用程序:使用 APPLETWIZARD

第十一章	你的第一个小应用程序:初级 JAVA	(118)
11.1	小应用程序.....	(118)

11.2	它们是怎样工作的.....	(121)
11.3	小结.....	(125)
第十二章	VISUAL J++ APPLETWIZARD	(126)
12.1	APPLETWIZARD 入门	(126)
12.2	编写“HELLO, WORLD”小应用程序	(129)
12.3	小结.....	(134)
第十三章	处理事件.....	(135)
13.1	事件.....	(135)
13.2	鼠标事件.....	(136)
13.3	键盘事件:SCREENTYPE APPLET	(149)
13.4	小结.....	(155)
第十四章	多线程.....	(156)
14.1	STATICMARQUEE 小应用程序	(156)
14.2	一个 PANNING MARQUEE	(161)
14.3	锁定资源.....	(168)
14.4	小结.....	(170)
第十五章	动画技术.....	(171)
15.1	闪烁的 STOCK 小应用程序.....	(171)
15.2	什么引起了闪烁.....	(178)
15.3	通过避免屏幕清除来减少闪烁.....	(179)
15.4	通过双缓冲技术避免闪烁.....	(186)
15.5	小结.....	(193)
第十六章	运动的图象.....	(194)
16.1	ANIMATION 小应用程序	(194)
16.2	增加音频.....	(208)
16.3	小结.....	(209)
第十七章	ABSTACT WINDOW TOOLKIT(抽象窗口工具箱)	(210)
17.1	利用 ABSTACT WINDOW TOOLKIT	(210)
17.2	利用简单组件.....	(211)
17.3	布局策略(LAYOUT POLICY)	(215)
17.4	FRAME(框架)类.....	(221)
17.5	自动地建立一个容器:RESOURCEWIZARD	(226)
17.6	小结.....	(237)
附录 A	安装和其它要素.....	(238)
A.1	安装 VISUAL J++	(238)
A.2	安装 INTERNET EXPLORER	(240)
A.3	创建你的第一个应用程序	(240)
A.4	在 VISUAL J++ 下使用 SUN JDK	(241)

附录 B 选出的 SUN JDK APPLETs	(242)
B.1 CARDTEST	(242)
B.2 NERVOUSTEXT	(247)
B.3 MOUSETRACK 和 MOUSERUN	(250)
B.4 DITHERTEST	(256)
B.5 结论	(265)

第一部分 过滤 Java:编写 Java 的应用程序

- 第一章 第一个 Java 程序
- 第二章 可以编写一个简单的语句吗?
- 第三章 Java 类简介
- 第四章 继续介绍 Java 的类
- 第五章 对象引用
- 第六章 正确初始化一个对象
- 第七章 扩展当前的类:继承
- 第八章 真空包装的 Java:程序包
- 第九章 使用例外处理错误
- 第十章 文件 I/O

第一章 第一个 Java 程序

Java 语言同 C++ 语言一样,属于面向对象的编程语言。与 C++ 语言一样,Java 功能性的编程基础是无差错的。不同于 C++ 语言的是,Java 不是嫁接了面向对象特征的功能性语言,这就使得 Java 语言更简单,语法上更一致。另外,Java 还是一种具有安全性的语言。

1.1 Hellow,World

通常使用一种新语言编写的第一个程序就是将“Hellow,world”输出到屏幕上。我们用 Java 语言编写这样一个程序,如下所示:

```
public class App1_1
{
    public static void main(String args[])
    {
        System.out.println("Hellow,world");
    }
}
```

要创建这个程序,从菜单 File 中选择 New 命令,在 New 对话框中,选择 Text File(文本文件)然后单击 OK。这样就打开了一个空白窗口供你输入上面程序代码(注意:你可能想先创建一个子目录来存放你的第一个程序)。现在,正确输入上面的程序代码,在你编译并执行完后,我会解释这些语句的意思。

输入完毕后,将它保存文件 App1_1.java 中。

注解:文件名与类名的匹配很重要,扩展名应为 .java。

要编译程序,在 Build 菜单中选择 Compile。在编译文件之前,Microsoft Visual J++ 需要一个工程(工程是组成一个程序的所有文件的容器)。因为你没有工程,所以 Visual J++ 会问你,你是否想建立一个缺省的工程? 选择 Yes,Visual J++ 将自动在当前目录下生成一个称作 App1_1 的任务,并且将文件 App1_1 包括进工程中。(如果你选择 No,Visual J++ 将拒绝往下执行)。

要执行该程序,在 Build 菜单中选择 Execute。

第一次执行该程序时,Visual J++ 不知道 app1_1 是一个应用程序还是一个小应用程序。应用程序(简称 app)被设计成一个与传统程序一样在主机上执行的程序,所以 Java 应用程序执行起来与其它程序一样。另一种是小应用程序(applet),它被设计成作为 HTML(超文本标记语言)Web 页面的一部分来执行的程序。本书的第一部分的创建的程序都是应用程序,我们将在第二部分介绍小应用程序(applet)。

选择 Run Class As Application 选项按钮,然后单击 OK。你的窗口将类似与图 1-1。



图 1.1 第一次执行你的程序,你必须告诉 Visual J++ 该程序是一个应用程序。

然后 Visual J++ 打开输出窗口,显示所希望的“Hello,world”消息

注解:本书中的这个图中所显示的版本号与你执行应用程序时显示的可能不一样。

1.2 这个程序是怎样运行的

让我们回头来确切地了解,这个程序是怎样运行的。程序的第一个语句定义我们的初始化类,我们将在第 3 章中详细地讨论类。现在我们只是说每个程序必须有一个类,它必须标记为 public,它的名字必须与 Java 源文件的名字相同。

接下来的语句定义该程序的第一个(也是唯一一个)函数。

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

执行程序总是从 main 函数开始,它必须被如上定义。开闭括号包括了程序代码。在这个程序中只有一个 Java 语句:

```
System.out.println("Hello,world");
```

System 是一个指向程序所在执行系统的对象,out 是缺省的输出对象。

C/C++:在 C 语言中与 out 等价的是 stdout,C++ 语言的输入输出流等价于 cout。

println 函数只是将它的参数 String 打印到输出设备上,然后换行。

1.3 Visual J++ 的作用

“Hello,world”的程序看上去相当直接,其中 Visual J++ 所起到的作用是很有趣的。

当你使用 Visual J++ 编译程序时,它并非生成你通常看到的那种全是 80x86 机器指令组成的程序,而是由 Visual J++ 编译器输出所谓的 Java 字节码,Java 字节码是为某种虚拟的 Java 机器(这种机器实际并不存在)而编写的指令代码。要执行程序,必须要有一个 Java 字节码解释器来执行字节码。也就是说,无论你使用那种计算机和操作系统都需要模拟 Java 虚拟机(简称 JVM)。不过,一般情况下,你并不注意到这一点,因为当你在 Visual J++ 集成开发环境(简称 IDE)中运行应用程序时,Visual J++ 会自动做这个工作。

当你在 Web 上执行小应用程序是,是由浏览器来实现 Java 字节码的解释的(Microsoft In-

ternet Explorer V3.0 或以上版本、Netscape Navigator V2.0 或以上版本、以及其它的一些软件在 PC 机上实现了 Java 字节码解释器)。如图 1-2 所示。

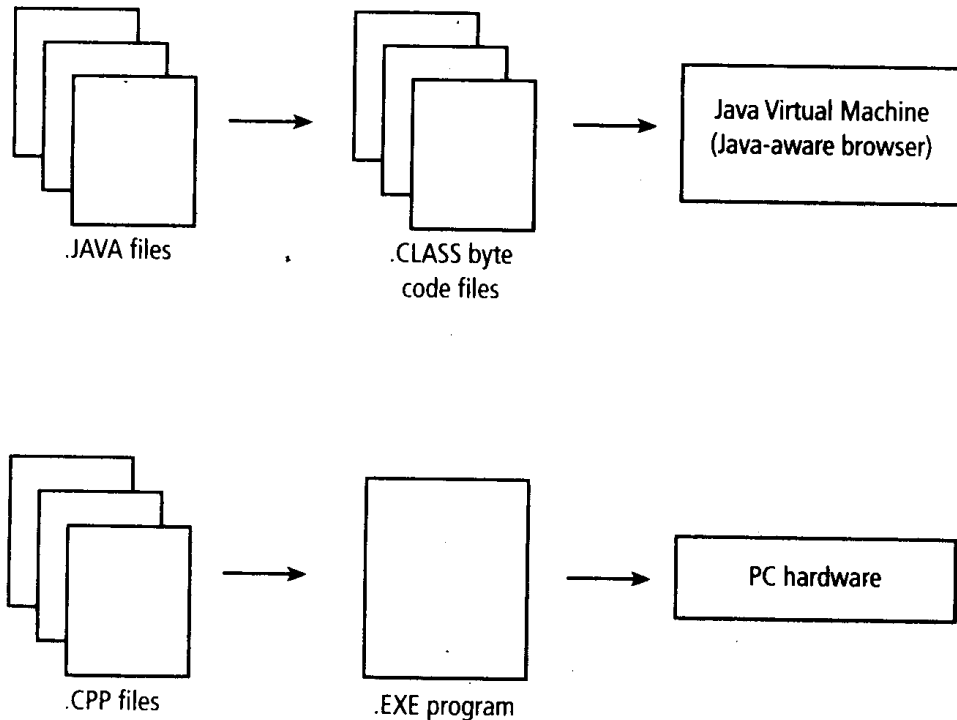


图 1-2 编译 .java 源文件, 产生 .class 输出文件, 该文件包括 Java 字节码指令。
这些指令必须由一个 Java-aware 浏览器下载执行

提供 JVM 有三个目的:

- 机器无关性

只存在一个 JVM(实际上, JVM 根本不存在, 因而说它是虚拟的), 每种不同的 PC 机型都有一个不同的 JVM 仿真器, 这就使得 Java 独立于机型。

- 安全性

经验告诉我们, 软件达到任何合理的安全标准是很困难的。只要程序在 CPU 上以其原始模式执行, 处理系统和浏览器上的任何一个缺陷都会被黑客(寻找机会进入客户机的人)利用。Java 程序不具备这种灵活性, 它在浏览器创建的 Java 虚拟机上执行, 程序执行时, 浏览器一直在场, 监视程序执行, 即保证不会程序不会导致任何伤害。

- 减小小应用程序的大小

JVM 程序比传统的程序小得多。这是因为你的程序调用的内嵌式 Java 库函数是驻留在浏览器中, 而不是链接在程序上的。这就大大地减小了必须下载的 Java 小应用程序的大小, 从而减少了下载的时间。Java 只下载所需要的东西以进一步减少下载时间(一个给定类的字节码, 只有在程序需要时才会被下载)。

1.4 即时编译

在浏览器中解释执行 Java 字节码有其负面影响——即降低了性能。Java 字节码程序执行起来没有通常的原始代码程序快。为解决这个问题, Microsoft 的 Internet Explorer 支持一种即