

VISUAL J++

入门与提高



Fig. 42.

...the earth, and
all other bodies, that
solid as a stone...

入门与提高

系列丛书

徐国标 郭新明 等 编
西南交通大学出版社

Visual J++ 入门与提高

徐国标 郭新明 等编
黄武 欧阳

西南交通大学出版社

内 容 提 要

Java 是一种简单的、面向对象的、分布式的、健壮的、安全的、与硬件平台无关的、可移植的、高性能的、多线程的以及动态解释执行型的程序设计语言。Java 的许多内容都是为网络服务的。本书详细介绍了 Java 方面的基础知识、Java 的安装、Java 与面向对象技术、Java 的编译与调试等内容。并列举了许多编程实例以帮助读者有效地掌握 Java，早日步入 Java 编程世界的神秘殿堂。

本书编写通俗易懂，深浅恰当，实例多。既可作为广大计算机爱好者的工具书，也可作为大专院校在校生或各类培训班的教科书。

本书无反盗版标识不得销
售。违者必究，举报有奖。举
报电话：(028)7600560 7600564

Visual J++ 入门与提高

——入门与提高系列丛书

徐国标 郭新明 等编

责任编辑 王 旻

*

西南交通大学出版社出版发行

(成都二环路北一段 610031)

郫县印刷厂印刷

*

开本：787×1092 1/16 印张：27.75

字数：683 千字 印数：1—4000 册

1998 年 6 月第 1 版 1998 年 6 月第 1 次印刷

ISBN 7-81057-182-6/T·272

定价：40.00 元

目 录

第一章 Visual J++简介	(1)
§ 1.1 Java 的起源	(1)
§ 1.2 什么是 Java	(1)
§ 1.3 Java 语言的特点	(2)
§ 1.4 为什么要学 Java 语言	(5)
§ 1.5 Visual J++的新特征	(6)
§ 1.6 Java 的应用发展展望	(11)
第二章 安装运行 Visual J++	(12)
§ 2.1 Visual J++ 1.1 安装需求	(12)
§ 2.2 Visual J++ 1.1 安装步骤	(12)
§ 2.3 运行 Visual J++ 1.1	(17)
§ 2.4 退出 Visual J++	(18)
§ 2.5 删除 Visual J++	(19)
第三章 使用 Developer Studio 开发环境	(21)
§ 3.1 认识 Developer Studio	(21)
§ 3.2 Visual J++ Developer Studio 菜单命令	(23)
§ 3.3 项目工作区	(55)
§ 3.4 Java 文本编辑	(57)
第四章 Java 与面向对象技术 (OOP)	(61)
§ 4.1 Java 的类	(61)
§ 4.2 创建对象	(61)
§ 4.3 Java 对象的使用	(62)
§ 4.4 Java 对象的释放	(62)
§ 4.5 Java 类的创建	(62)
§ 4.6 Java 类成员函数	(65)
§ 4.7 Java 类的继承和查找	(67)
§ 4.8 最终类与抽象类	(69)
§ 4.9 Java 接口与包	(69)
§ 4.10 抽象的 Java 程序框架	(71)

第五章 Java 语言基础	(73)
§ 5.1 Java 数据类型	(73)
§ 5.2 Java 表达式和运算符	(78)
§ 5.3 Java 语句	(81)
§ 5.4 Java 数组和字符串	(91)
第六章 开始编程	(94)
§ 6.1 Java Applet Wizard	(94)
§ 6.2 Applet Wizard 应用程序生成步骤	(94)
§ 6.3 检查生成文件	(100)
§ 6.4 测试 test 程序	(105)
§ 6.5 理解 HTML	(107)
§ 6.6 理解 test. java 源程序	(109)
§ 6.7 添加代码	(112)
第七章 编译与调试.....	(113)
§ 7.1 Java 编译菜单	(113)
§ 7.2 Java 编译器 JVC	(119)
§ 7.3 Java 的调试功能	(124)
第八章 使用 Visual J++ 资源模板	(140)
§ 8.1 Visual J++ 支持的 Windows GUI 部件	(140)
§ 8.2 Visual J++ 资源模板	(141)
§ 8.3 创建对话框	(143)
§ 8.4 对话框编辑使用	(145)
§ 8.5 创建 Java 程序菜单	(149)
§ 8.6 用 Resource Wizard 生成源代码	(153)
§ 8.7 使用资源代码的 Java 程序	(163)
第九章 Visual J++ 编程进阶	(170)
§ 9.1 Java 的图形功能	(170)
§ 9.2 Java 的图像显示功能	(186)
§ 9.3 Java 动画设计	(192)
§ 9.4 Java 的声音播放功能	(201)
§ 9.5 Java 多线程程序设计	(203)
第十章 Java 编程实例	(208)
§ 10.1 Java 动画标语制作	(208)
§ 10.2 用 Java 画任意圆弧	(215)

§ 10.3	用 Java 实现音频播放功能	(218)
§ 10.4	Java 后台线程示例	(224)
§ 10.5	Java 制作条形图	(231)
§ 10.6	用 Java 制作闪烁字体	(238)
§ 10.7	旋转不停的头像	(239)
§ 10.8	用 Java 制作万年历	(244)
§ 10.9	屏幕布局示例	(257)
§ 10.10	绘制时钟	(259)
§ 10.11	温度转换例程	(262)
§ 10.12	颜色抖动图案显示	(265)
§ 10.13	点线绘制例程	(271)
§ 10.14	客户数据库管理例程	(276)
§ 10.15	图形绘制例程	(291)
§ 10.16	统计图显示例程	(301)
§ 10.17	图像操作例程	(311)
§ 10.18	跳动的盒子	(318)
§ 10.19	盯着鼠标的眼睛	(320)
§ 10.20	上下浮动的字符	(326)
§ 10.21	控制面板布局示例	(328)
§ 10.22	简单的数据库应用实例	(333)
§ 10.23	滚动显示图片	(353)
§ 10.24	控制滚动图片	(356)
§ 10.25	滚动的文本标题	(360)
§ 10.26	显示复杂数学图形	(366)
§ 10.27	排序演示实例	(367)
§ 10.28	电子数据表的制作	(371)
第十一章 Visual J++ 基本类库参考		(391)
§ 11.1	Applet 类	(391)
§ 11.2	BorderLayout 类	(392)
§ 11.3	Button 类	(393)
§ 11.4	Canvas 类	(394)
§ 11.5	CardLayout 类	(394)
§ 11.6	Checkbox 类	(395)
§ 11.7	CheckboxGroup 类	(396)
§ 11.8	Choice 类	(396)
§ 11.9	Color 类	(397)
§ 11.10	Component 类	(399)
§ 11.11	Container 类	(404)

§ 11.12	Dialog 类	(405)
§ 11.13	Dimension 类	(406)
§ 11.14	Event 类	(407)
§ 11.15	FileDialog 类	(411)
§ 11.16	FlowLayout 类	(412)
§ 11.17	Font 类	(413)
§ 11.18	FontMetrics 类	(415)
§ 11.19	Frame 类	(416)
§ 11.20	Graphics 类	(418)
§ 11.21	Image 类	(420)
§ 11.22	Insets 类	(421)
§ 11.23	Label 类	(422)
§ 11.24	List 类	(423)
§ 11.25	Menu 类	(424)
§ 11.26	MenuBar 类	(425)
§ 11.27	MenuComponent 类	(426)
§ 11.28	MenuItem 类	(427)
§ 11.29	Panel 类	(428)
§ 11.30	Point 类	(428)
§ 11.31	Rectangle 类	(429)
§ 11.32	Scrollbar 类	(431)
§ 11.33	TextArea 类	(432)
§ 11.34	TextComponent 类	(433)
§ 11.35	TextField 类	(434)
§ 11.36	Toolkit 类	(435)
§ 11.37	Window 类	(437)

第一章 Visual J++ 简介

§ 1.1 Java 的起源

Java 语言的产生,要追溯到 1991 年。当时, Sun 微系统公司为了开发在家用消费类电器上的并互式操作能力,专门开发了 Java 语言(当时称之为 Oak(橡树)语言),以满足家用电器产品标准化、安全可靠和使用简单的设计要求。虽然这种技术在当时是有创新的,但在那时商业上的应用并不成功,所以没有引起太多的重视。

到 1994 年底,随着 Internet 的迅速发展,WWW 在 Netscape 和 Mosaic 软件的支撑下迅速兴旺起来,并显示出极大的发展力,Internet 应用程序的开发,蕴育着无限的商机。在这种情形下, Sun 公司感到市场需要开发 WWW 上的“中性”浏览器,即不依赖于任何平台,并且具有实时交互能力和高度的安全性、可靠性。这和当初 Java 的设计初衷不谋而合。于是 Sun 用 Java 语言开发了其新一代浏览器产品 HotJava,并取得了巨大的成功,推动了 Java 语言的发展,使之成为当今 Internet 上最受欢迎的编程语言。

目前,世界最大的 Internet 浏览器制造商 Netscape 的 Navigator, Microsoft 的 Internet Explorer 均支持 Java。其它的著名大公司,如 IBM, Novell, Oracle, SGI 和 Borland 也都相继购买了 Java 的许可,Java 语言的地位得到了广泛的肯定,成为 21 世纪的热门编程语言。

§ 1.2 什么是 Java

在 Sun 的 Java 语言技术白皮书上,Java 是这样定义的。

Java 是一个简单的,面向对象的、分布式的、健壮的、安全的与硬件平台无关的、可移植的、高性能的、多线程以及动态解释执行型的程序设计语言。

Sun 公司副总裁,Java 语言的设计师詹姆斯·高斯林先生的一番解释也许可以更详细地道出 Java 语言的奥妙之处:

- Java 并不是专门为 Internet 而开发的语言。应该说 Java 是一种像 C++ 一样通用的程序设计语言,人们可以用它设计开发出各种各样的应用软件。

- Java 被许多人看作为多线程,具有无用单元收集回收功能的 C++ 改进版本。即你能用 C++ 写程序,你就能用 Java 编写程序;反言之,如果你能用 Java 写程序,你也能用 C++ 写程序。

- Java 的许多内容都是为网络应用服务的。

- Java 可以实现网络上安全地漫游。对 Java 而言,有许多机制可使用户能够安全地漫

游 Internet。而这些机制对用户而言是透明的，它们具有抵制病毒和网络黑客 Hacker 的侵袭能力。

谈到为什么取名是 Java 而不是 Oak 或者 D++，其实带有许多的偶然性，当初设计家用电器类专用语言时，Gosling 首先看手改写 C++ 的编译器，但后来他感到 C++ 无法满足需要，准备重新开发一个新的语言。据说 Gosling 从窗口外的一颗老橡树得到灵感，当时创建的工作目录取名为 Oak（橡树），后来才发现 Oak 已经是 Sun 公司另一个语言的注册商标，才重新取名 Java。Java 是印度尼西亚一个盛产咖啡的岛屿，也即我们汉语文字中的“爪哇”一词，这里设计者指的是爪哇岛上的咖啡名字。提及咖啡，总是不禁令人感到那扑面而来的清香和优雅宜人的环境。设计者起这样一个名字，也许想在如今这样一个充满紧张和竞争的时代营造一种轻松愉快的气氛，至于是否为作者当时盯到了桌上的咖啡而灵机一动，那就不得而知了。

从此，Java 以独特的一杯热气腾腾的咖啡标志分不开了。以致商人也纷纷推出“网络咖啡屋”来招徕顾客。围绕着 Java，Sun 公司也相继推出了关于“咖啡豆（Java Bean）”和“纯”咖啡的一系列产品政策，可以这么说，Java 开创了 Internet 一个新的里程碑。

§ 1.3 Java 语言的特点

正如上一节所述，Java 语言最初以 C++ 为基础而设计的，因此它在形式上和 C，C++ 极为相似。如果用户已经学习过一门 C 或 C++ 语言，那么可以很快学会 Java。但是与你以前学习的 C/C++ 不同，Java 语言更具其优秀的特点，主要表现为：

一、简单性和熟悉性

简单性是 Java 语言最初设计的基本出发点。Java 语言摒弃了 C++ 中某些不合理的因素和某些不确定的特征，相比而言，Java 语言比 C++ 要小得多，程序员要编写可靠的应用程序就会大大减轻负担。此外语言的基本概念也很容易掌握，程序员不需花长时间的培训就能很快进入角色。

Java 语言删除了 C/C++ 中很少使用和容易出错的语句。如 goto 语句和指针的使用等。同时也不支持 C++ 的操作符重载和多重继承等特性，也不支持 C 语言的预处理器、头文件和 struct 和 Union C 联合，语句和 typedef 以及 #define 语句。

无用存储单元收集功能是 Java 区别于 C/C++ 的另一重要特征，它把程序员从复杂的内存管理分配和释放中解放出来，从而也避免程序出现指针指向非法区域，空指针赋值和内存未释放等的错误。

二、面向对象的特性

Java 语言是一种全新的面向对象程序设计语言 (oop)，它既不是从某一种语言上派生而来，也不是从任何一种语言升级而得到的。Java 的面向对象特性意味着程序员首先要考虑的是对象中的数据和定义在数据上的操作，而不仅仅是过程的如何定义，这一点对习惯于用 C 语言编写过程化程序的程序员而言，使用 Java 语言时要考虑改变问题的思考方式。

Java 语言也并非全部面向对象，像数字和其它简单类型则是非面向对象的，这样做的目的是为了保持高性能。

Java 语言提供了简单的类机制和直观的动态界面模型。在对象中封装了状态量和方法。子类可以继承或重载父类的方法。但 Java 不持 C++ 语言的模板机制, Java 中所有的类都是 Object 类的子类, Java 是单一继承性类分级结构, 即一次每一个类只能继承另一个类, 不允许多态继承, 从而避免了不必要的复杂。同时 Java 也摒弃了 C/C++ 中的某些非面向对象特性, 如全局变量的使用, 调用外部函数和定义 struct 结构变量等。

Java 语言之所以设计成为面向对象的, 乃是因为面向对象技术在软件工程中占有重要的地位, 它大大减少了软件开发和维护的费用, 使程序开发大型软件更为简便。作为新一代的编程语言, Java 的纯面向对象特点, 使其得到广泛应用铺开了道路。

三、健壮性

Java 语言的类型法则比 C 和 C++ 更为严格。Java 要求显式地声明方法, 从而保证了编译器在编译时即可发现方法调用上的错误隐患, 进一步保证程序的可靠性。

在内存使用方面, Java 不支持指针的分配和使用, 从而只能使用数组模型, 这就杜绝了程序任意访问内存而可能产生的隐患。

Java 的自动垃圾回收机制, 即系统自动对无用的内存单元进行回收, 就不会像其它语言那样由于动态内存分配使用不当而造成系统崩溃。

Java 采用了集成的面向对象的异常处理模型, 管理所有不正常的程序结束, 并继续执行未出错的部分。程序错误或可能导致错误的调用, 如打开一个不存在的文件等, 在 Java 语言中通过使用 throw/catch 语句, 程序员可以把错误处理代码放在同一个地方, 这样既可以简化错误处理任务和恢复, 同时也增强了程序的可读性。

四、平台无关性

作为 Internet 编程的首选语言, Java 支持异构网络平台上的应用程序开发。即应用程序能够在各种软件平台和操作系统上运行。

Java 语言经编译生成的与体系结构无关的字节码 (byte-code) 巧妙地解决了这个问题。Java 的字节代码所表示的含义由 Java 虚拟机 (VM) 格式严格定义。对每一个运行的平台, 需由该平台的 Java 虚拟机解释执行该 Java 字节码。这种结构中立的的原则, 非常适合 Internet 多平台特性。因此, 只需一次编译 Java 源程序, 其产生的字节码就可以在任何时候的任意平台上运行。

当然 Java 的解释执行方式速度会变慢, 目前正在研制可以将 Java 字节码直接转变为机器码的编译器, 即编译型 Java。

目前 Java 虚拟机, 在 Windows95, NT 和各种 Unix 版本下, 都有其实现, 因此用 Java 开发的程序均可以在多种平台上使用。也正是这种平台无关性, 才使 Java 能够拥有其广阔的应用前景。

五、分布交互性

Java 是面向网络的语言, 使用提供的类库可以方便地支持 Internet 的 TCP/IP 协议, 并可通过资源统一定位器 URL 来访问 Internet 上的各种对象。使用 Java 打开远程文件和打开本地文件没有什么区别。因此, Java 具有分布式编程的特点。同时支持 TCP/IP 协议的 Socket 类可以提供可靠的流式网络连接。因此也可以用 Java 来创建客户/服务器应用程序。

Java 语言的多线程和同步机制, 可以使用户的应用程序在运行时可以同时并行执行多

个例程，所以在进行网络接收和发送数据的同时，程序还可以与用户进行交互操作。

六、安全性

Internet 上的安全性问题日益突出，其网上用户数据的安全性是 Internet 进一步发展和正常使用的根本保障。如果没有安全方面的保证，当用户以网络上下载程序执行时要冒巨大的风险。Java 自身设计的安全机制可以有效地防止病毒程序的滋生蔓延和对本地/远程文件造成破坏。

Java 禁止程序任意访问内存，这通过不支持指针，对内存的访问必须通过对象实例化变量来实现。这样有效地避免了恶意的程序员使用修改程序指针的方法来访问对象私有成员和破坏程序的目的。

Java 的另外一个有效的安全措施是字节代码验证功能，Java 虚拟机对网络上下载的代码在执行前检查字节代码是否违反了语言的限制，从而防止网络病毒绕过 Java 编译器而产生的病毒代码。

另外运行时关于字节代码从网上何处来的信息，可用来判断字节代码来自防火墙的内部还是外部，并进一步确定这些 bytecode 来做什么事。在未来的 Java 版本中将会有公共密钥加密机制。所有这些机制的使用，构成了使用 Java 的安全的环境。

七、多线程运行能力

Java 的多线程处理提供了并行执行多个任务的能力，并且为客户端提供高度的 GUI 交互能力。众所周知，使用 C/C++ 来开发多线程应用程序时非常麻烦，例如首先必须保证多个例程可以并发执行。如果有公用临界资源的共享，则必须把使用该资源的线程序列化，保证一次只有一个线程存取该临界资源。另外，程序员还必须负责线程的锁定和解锁。这种手工编程方式既繁琐，又易出错，造成资源使用死锁 (dead-lock)。

Java 具有复杂的同步机制，可以大大简化多线程应用程序的开发。负责线程的启动、运行、终止和检查线程的状态都可以用 Java 提供的线程类 (thread class) 来完成。

八、动态性

在很大程度上，Java 语言比 C/C++ 更具动态特性，以适应变化的环境。而用 C/C++ 开发环境开发的动态性问题在代码生成时就已经发生。打个比方说，你买了某公司的专用类库文件来开发自己的应用，当该公司推出新的类库版本，修改了类后，你也必须得重新编译它们来更新版本。否则如果用户的工作环境中两者不配套就会发生兼容性问题。而 Java 完全避免了这类问题。在使用 Java 编写的类库中，用户可以随意地将新方法和成员函数添加到类库中而不会对客户端造成影响。此外，用户还可以动态地从网络上任何一个地方安装代码模块，以动态地适应不断变化的环境。

虽然 Java 编译器在编译时的静态检查十分严格，但 Java 字节码与运行时 Java VM 的连接阶段是动态的。其中类是根据需要时进行连接的。因此新的代码模块可以根据需要，甚至可以通过网络从源库中获取。Java 的这种动态特性，使得它可以在 Internet 的分布环境中动态地维护应用程序和支持类库间的一致性，故不必像 C/C++ 的类库那样升级后必须重新编译其应用程序。

九、高性能机制

Java 语言由于具有上述特点从而达到了其优越的性能。即 Java 虚拟机以全速运行而不

需要检查其运行时的环境。垃圾自动回收程序以一种低优先级的后台线程方式运行,来确保应用程序需要时有足够可用的内存。

Java 还可以通过即时编译器(JustInTime)来将 Java 语言的字节代码编译成本地机器内码。这种即时性具有创新意义,因为它提供了本地代码性能的等级,同时又不会损害应用程序的可移植性。因此 C/C++ 的仅有一处胜于 Java 的优点,即运行速度快的特点也岌岌可危了。

图 1-1 示出了 Java 应用周期图,清楚地说明了 Java 语言源代码通过 Java 编译器,生成字节代码(bytecode)通过网络传输到 Java 虚机器解释或编译执行的过程图。

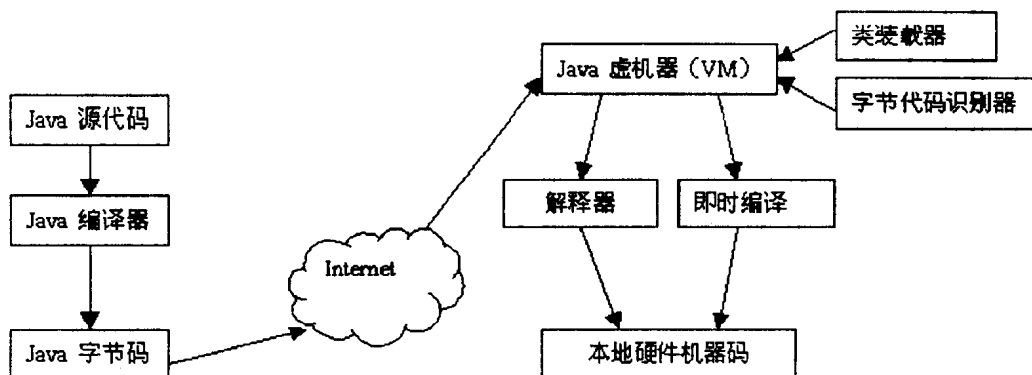


图 1-1 Java 应用周期图

十、丰富的类库

Java 语言提供了大量的类库。其语言包支持字符串处理、多线程、异常处理等。实用程序包支持 Hash 表、stack、时间、日期等处理。I/O 包支持文件系统、外设、网络等 I/O 操作; AWT 包提供创建用户图形界面的元素类; 网络包支持 TCP/IP 协议的 HTTP, FTP 等 Internet 访问。

§ 1.4 为什么要学 Java 语言

目前有几百种语言可供程序开发人员用来编程以解决某些特定领域的问题。语言的覆盖范围广,有的是全解释性的语言,像 Unix 的 shell 语言、awk、TCL、Perl 语言等,所有这些都像 C/C++ 一样只是用于纯粹的编程。Shell 和 TCL 语言是一种解释性高级语言,它是在系统一级处理文件和过程,而不是数据结构,故适合于作原型快速开发,具有一定的移植性,但性能上要差一些,执行速度比本地机器代码和解释性字节码慢得多,C/C++ 可以在机器代码级编译,用于大型程序设计、性能很高,其缺陷是对分布式多平台应用和多线程的使用等方面难以实现。

Java 语言对上述语言的长处进行了综合。除了编程相当简单,高度可移植性和结构中立交外,对无很大计算量的应用,Java 其分布计算和多线程设计能力,成为 Internet 网络编程语言的主流。

表 1-1 列出了 Java 语言与各种代表性语言设计功能上的比较,以此说明 Java 的优势

所在。

表 1-1 Java 语言与其它流行语言的比较

	Java	C	C++	SmallTalk	Perl	Shells
简单性	好	中	差	好	中	中
面向对象	好	差	中	好	差	差
稳定性	好	差	差	好	中	好
安全性	好	差	差	中	中	中
解释执行	好	差	差	好	好	好
动态性	好	差	差	好	差	中
可移植性	好	中	中	中	中	中
结构中立	好	差	差	中	中	中
多线程	好	差	差	差	差	差
异常处理	好	差	中	好	差	差
性能	高	高	高	中	低	低

§ 1.5 Visual J++ 的新特征

Microsoft Visual J++ 1.1 是由 Microsoft 公司于 1997 年开发出的最新集成可视化 Java 开发环境, 由于它集成于 Microsoft Developer Studio 用户界面中, 故取名为 Visual J++, 类似于用户使用过的 Visual C++ 版本。Visual J++ 编程可以说是专门为熟悉 Visual C++ 的用户提供的一种便捷的开发手段。

Visual J++ 并不是 Java 语言的对立版本, 相反是在 Sun Microsystem 公司授权许可下的 Java 语言设计和开发规范, Microsoft 很明确表示 Visual J++ 技术完全支持它们, 同时提供了新的功能支持。具体包括以下几个方面:

1. 集成开发环境支持

早期的 Java 编译器是基于命令行方式的。但这并不影响用于 Internet Web 页面的 Java 应用程序开发。直到程序员使用了 Java 大约一年以后, 基于图形化的 GUI 集成开发工具才开始出现在市场上。

Visual J++ 集成开发环境是一个完整的开发平台, 可以用来创建各种 Java 应用程序和 Applet。如图 1-2 所示。

2. 内嵌的 Java 虚拟机

在 Unix 和早期的 OS 平台上, Java 虚拟机是嵌入在 Netscape, Mosaic 等浏览器中的。而 Microsoft 以操作系统部件的方式给出了 Java 虚拟机, 从而使 Java 应用程序不仅只局限于浏览器内运行, 而可以在其它应用程序上执行, 用户也可以在自己的应用程序中运行 Java 应用程序, 从而使 Java 的应用开发如同 VB, VC++ 一样。

3. 内置高效的 JVC

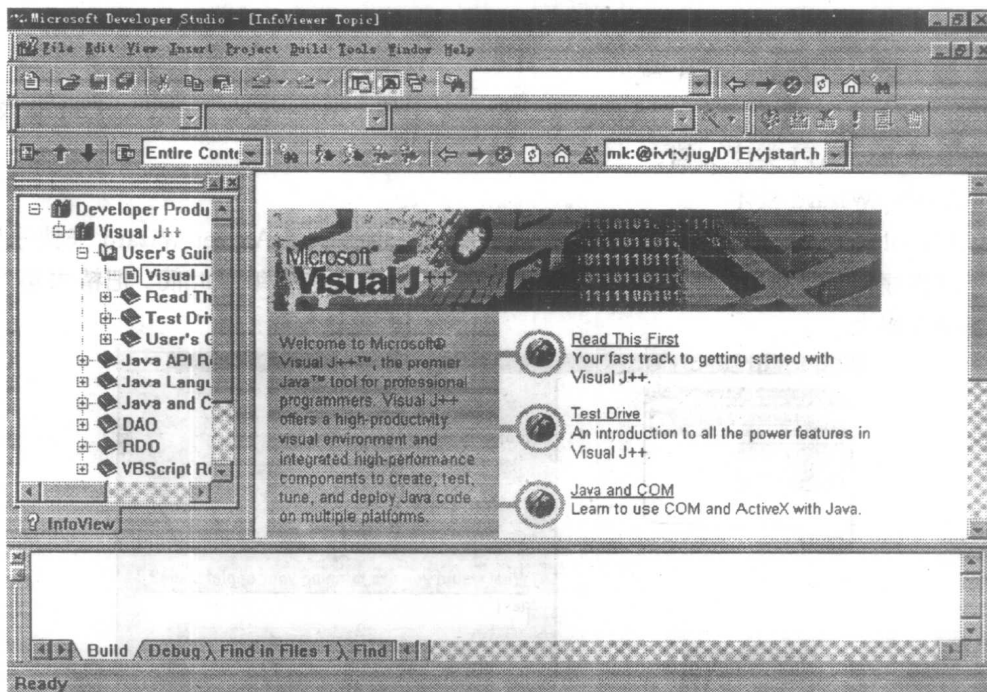


图 1-2 Visual J++ 集成开发环境

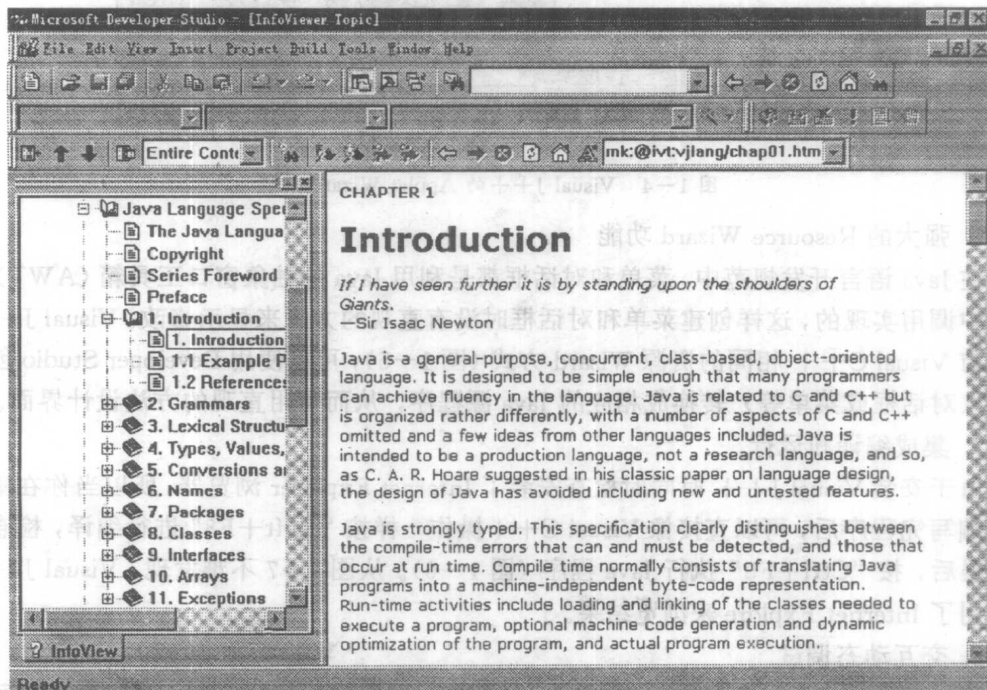


图 1-3 Visual J++ 帮助文档

JVC 是 Visual J++ 自带的 Java 编译程序，类似于 Visual C++ 的 C1, C2 等，但它是目前最快的 Java 编译器之一，每分钟可以编译 100 万行 Java 源代码。编译后的字节码可以

在任何 Java 虚拟机上运行。

4. 强大的在线帮助和源代码

Visual J++ 包含完整的在线联机帮助和参考文档(图 1-3)。其 Books Online 文档中包括详尽的 Java API 类库文档。

5. Applet Wizard 技术

与 Visual C++ 的 AppWizard 一样, Visual J++ 也提供了 Applet Wizard 来快速生成 J++ 应用程序框架(图 1-4)。这种快速成型技术可以大大节省编程时间,把精力集中到问题的解决方案上。

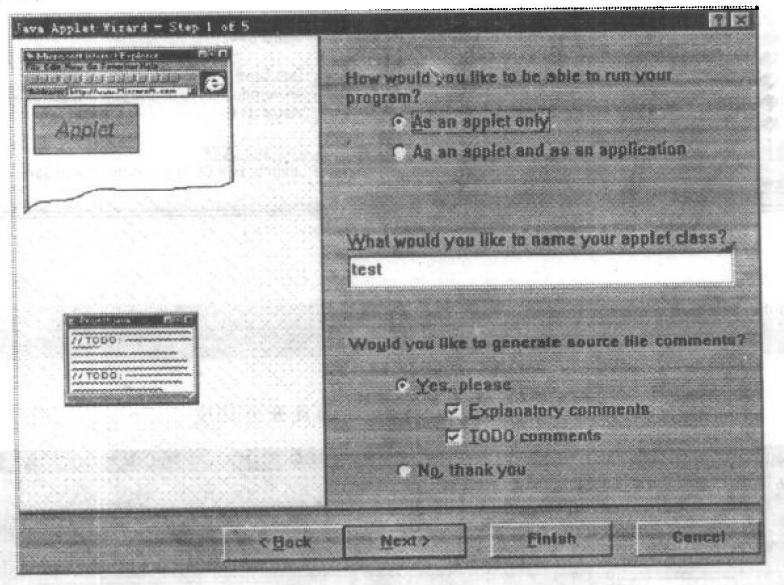


图 1-4 Visual J++ 的 Applet Wizard 功能

6. 强大的 Resource Wizard 功能

在 Java 语言开发规范中,菜单和对话框都是利用 Java 的抽象窗口工具箱(AWT)来源代码中调用实现的,这样创建菜单和对话框时没有直观的方式来显示修改。Visual J++ 提供了与 Visual C++ 相同的资源 Wizard 方式(图 1-5),可以使用 Developer Studio 创建的资源(对话框或菜单等)转换成相应的 Java 源程序,从而可用直观的方法设计界面。

7. 集成编译和运行

由于安装 Visual J++ 时同时检查安装了 Internet Explorer 浏览器,所以当你在集成环境中编写完程序后,可以直接像 Visual C++ 操作一样按“Shift+F8”进行编译,检查无编译错误后,按“Ctrl+F5”执行 Java 程序(图 1-6)。从图 1-7 不难发现, Visual J++ 自动调用了 Internet Explore 来浏览结果。

8. 交互动态调试

Visual J++ 提供了类似 Visual C++ 的动态调试手段(图 1-8),允许在任意可执行程序语句上设置断点和检查数据内容。添加断点和各种调试命令与 Visual C++ 相同。

9. 集成 COM 技术

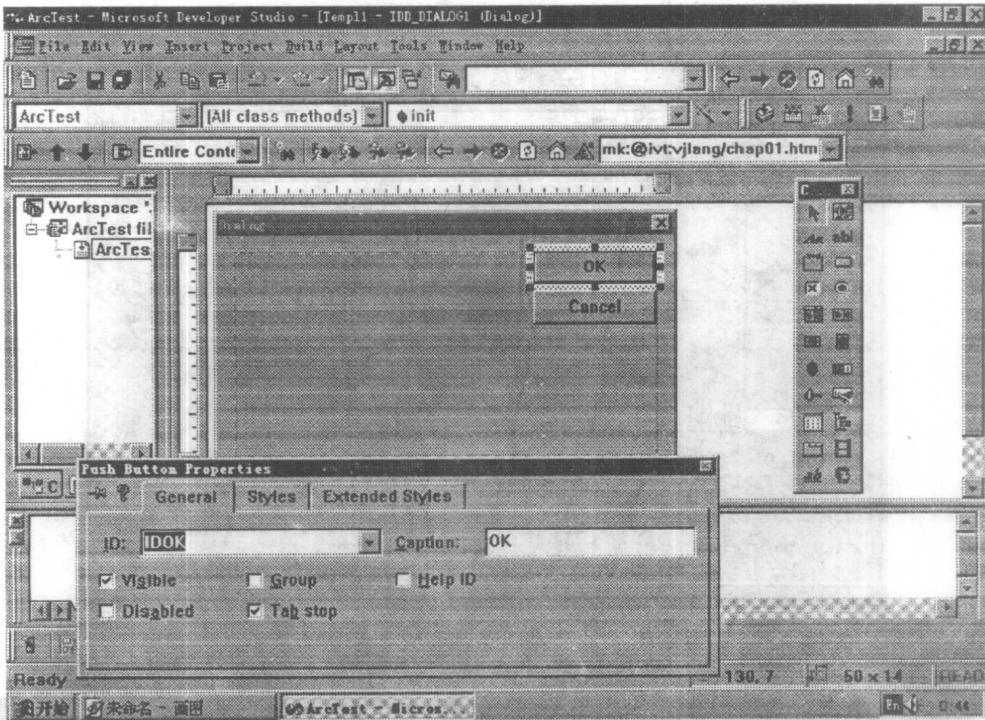


图 1-5 Visual J++ Resource Wizard

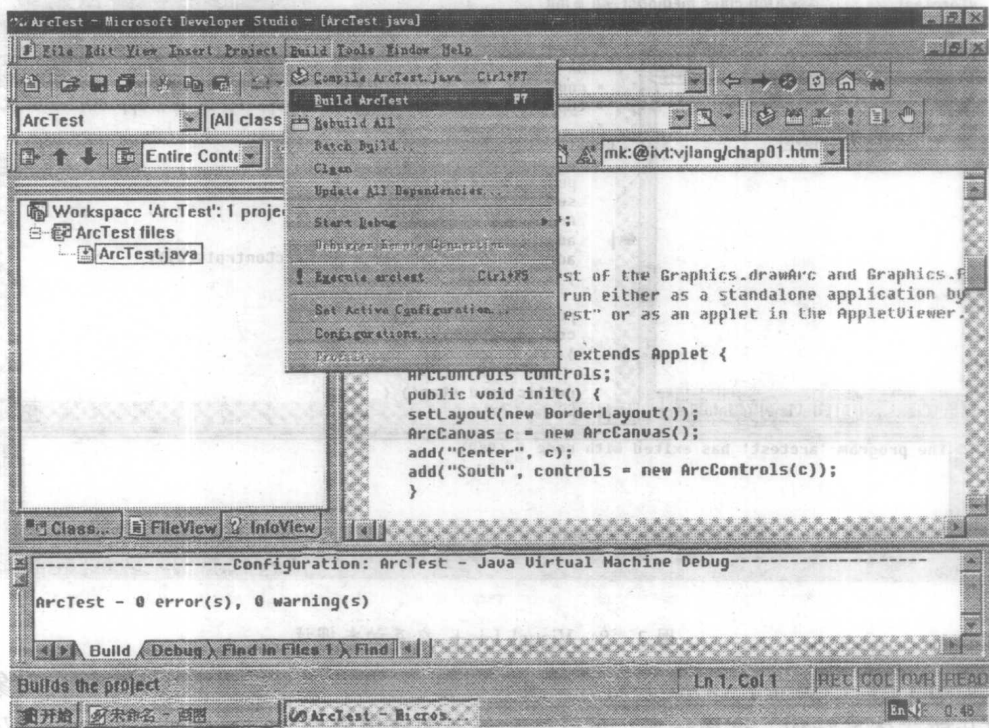


图 1-6 Visual J++ 集成编译运行机制

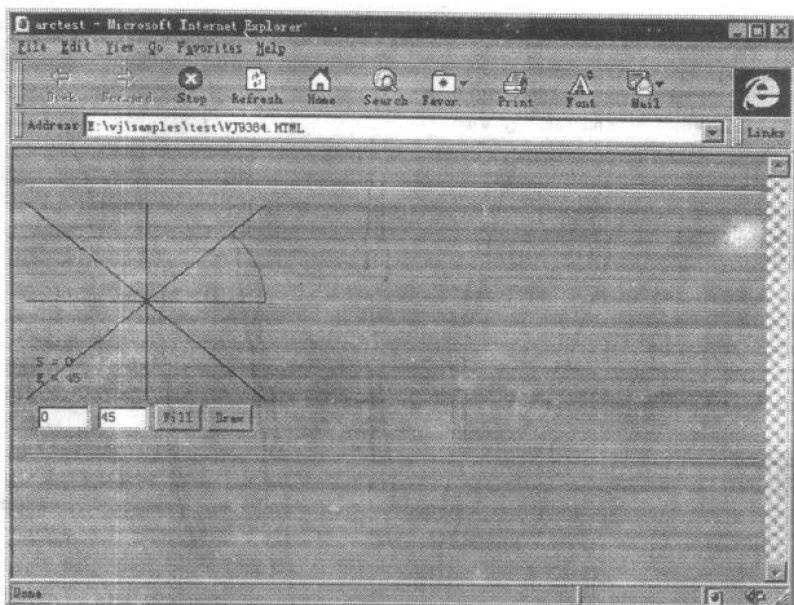


图 1-7 Java 程序运行结果

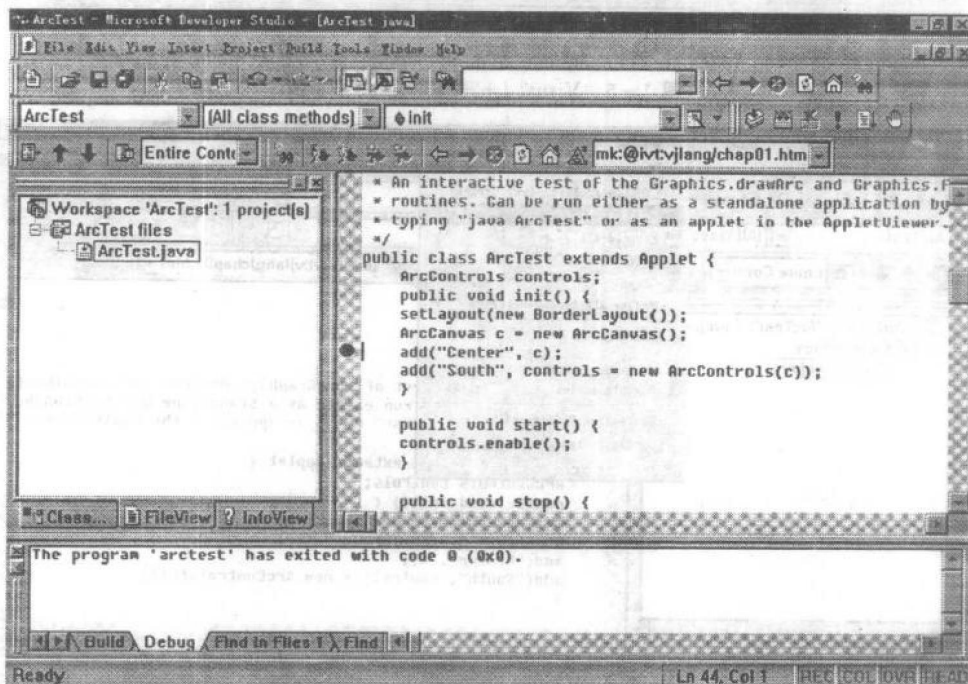


图 1-8 Visual J++ 交互动态调试

Visual J++ 集成了 COM 通用对象模块功能,是用于 C/S 应用模式下的面向对象为基础的开放式体系结构。这样一来,可以组合外部的应用程序,如可以链入 Excel, Word 等应用模块到用户的 Java 应用程序中。

10. 支持 Active X