



万水 Windows 软件精通指南系列

Visual J++

精通指南

the comprehensive Guide to Visual J++

[美] Daniel Joshi, Ramesh Chandank 著

郑晓惠 孔敬 张宇 李季 译

辛再甫 审校

Java, Visual J++ 和 ActiveX 技术的主要参考书

- ◆运用 Java 语言集成 ActiveX 技术的方法
- ◆ Visual J++ 可执行工程示例贯穿全书
- ◆编程提示和技巧前所未有



中国水利水电出版社

tp312
QX/1

万水 Windows 软件精通指南系列

Visual J⁺⁺ 精通指南

[美] D. 乔希 R. 坎达克 著
郑晓惠 孔敬 张宇 李季 译
辛再甫 审校

中国水利水电出版社

356/25

内 容 简 介

《Visual J++ 精通指南》基本上分为三部分,第一部分介绍 Java 程序语言和 Visual J++ 环境,第二部分详细介绍了 Java 程序设计技巧及其功能,最后一部分介绍了 ActiveX 技术,以及如何把 ActiveX 技术和 Java 语言用于 Visual J++ 中。Java 是一种面向对象的编程语言,以其特有的网上优势风靡于全世界。Visual J++ 是 Microsoft 为 Java 开发的一个集成开发环境(IDE)。与其他 Java 的 IDE 相比, Visual J++ 把 Java 的一些增强功能和 ActiveX 技术结合在一起,集成了 ActiveX 和 Java 的优点,且使用 Wizard、虚拟开发环境、集成调试环境和在线服务之后,Java 开发变得非常方便。本书通过大量实例来详解 Visual J++ 的用法,使读者对它有个清晰的认识。这是国内讲解 Visual J++ 的第一本书。

本书的目的是使读者全面了解 Java 语言和 Visual J++ 的使用。相信读者通过本书能很快熟悉 Visual J++ 并成为 Java 编程专家。

Original English Language edition published by Ventana Communications Group, Inc., P. O. BOX 13964, RTP, North Carolina 27709 - 3964. TEL: 919/544 - 9404, FAX: 919/544 - 9472. Copyright(c) 1997 by Ventana Communications Group. All rights reserved.

本书英文版权为 VENTANA 公司所有, VENTANA 公司已将中文版独家版权授予北京万水电子信息有限公司。未经许可,不得以任何形式和手段复制或抄袭本书内容。

图书在版编目(CIP)数据

Visual J++ 精通指南/(美)乔希(Joshi, D.), (美)坎达克(Chandak, R.)著;郑晓惠等译. —北京:中国水利水电出版社, 1997.5

(万水 Windows 软件精通指南系列)

ISBN 7-80124-394-3

I. V… II. ①乔… ②坎… ③郑… III. 程序语言, Visual J++ - 指南 IV. TP312

中国版本图书馆 CIP 数据核字(97)第 04377 号

书 名	Visual J++ 精通指南
作 者	Daniel Joshi Ramesh Chandak 著, 郑晓惠 孔敬 张宇 李季译, 辛再甫 审校
出版、发行	中国水利水电出版社(北京市三里河路 6 号 100044) 北京万水信息有限公司(北京三里河一区 39 栋 100045)
排 版	北京万水电子信息有限公司
印 刷	北京牛山世兴印刷厂
规 格	787×1092 毫米 16 开本 29.25 印张 655 千字
版 次	1997 年 5 月第一版 1997 年 5 月第一次印刷
印 数	0001—5000 册
定 价	49.50 元

出版者的话

近年来,中国水利水电出版社及其与美国万国集团的合资公司北京万水电子信息有限公司,一直致力于电脑图书的出版。1994年,我社购买了美国 MIS 出版公司的 5 本 Windows 自学教程的中文版版权;1995年,我社又与美国万国集团联袂购买了美国 Peachpit 出版公司的 7 本电脑书(其中有 6 本是 Macintosh 苹果电脑参考书)的版权;最近,我们又已经组织数套电脑丛书,如《中文版软件详解丛书》、《万水流行软件用户伴侣丛书》、《万水电脑彩色与平面设计丛书》、《万水计算机普及与提高系列》、《万水网络与数据库丛书》等。这些图书有的已经出版,有的将在近期内陆续推出。已经出版的图书,在出版界和电脑用户中产生了良好的影响。

目前,中国水利水电出版社与北京万水电子信息有限公司筹划出版一套《万水 Windows 软件精通指南系列》丛书。本系列将涵盖当今最为流行的 Windows 软件的编程、运行与维护,力求使读者能够通过本系列达到精通该软件的目的。其共同特点是,每本书都由熟悉软件的专家编写与翻译,详细介绍 Windows 软件的基础知识与高级技巧。书中插图丰富,直观简明,一目了然,因此可以说,本套丛书具有很强的针对性、实用性和可操作性,最适合国人阅读参考。

《Visual J++精通指南》是这套丛书中的一种,本书通过大量实例来详细介绍 Visual J++ 的用法,使读者对它有个清晰的认识。全书共分为三个部分:第一部分介绍 Java 程序语言和 Visual J++ 环境,第二部分详细介绍了 Java 程序设计技巧及其功能,第三部分介绍了 ActiveX 技术,以及如何把 ActiveX 技术和 Java 语言用于 Visual J++ 中。

编写、出版这样的丛书对我们来说尚属首次,错漏、失当之处在所难免。我们诚恳地欢迎您对本书提出修改意见,以便我们再版时进行修正;我们也热诚地欢迎您就本丛书提出选题建议或自荐、推荐作者,以使其尽可能地臻于完善;如对本丛书其他各册感兴趣,也请随时同我们联络。联络地址为:北京万水电子信息有限公司,北京市三里河一区 39 栋,邮政编码:100045;电话:(010) 6852.8689, 6853.3313;传真:(010) 6853.3313; E-Mail: mchannel@public3.bta.net.cn。

中国水利水电出版社
北京万水电子信息有限公司
1997 年 5 月

译者序

Visual J++ 是 Microsoft 为 Java 开发的一个集成开发环境 (IDE), 与其他 30 多个 Java IDE 比较, Visual J++ 包括了其他 Java IDE 的所有功能, 并允许用户在 Java 程序中调用 ActiveX 控制。除了在 Visual J++ 中可生成 Java applet、应用程序和 appletcation 之外, 还附加了一些工具来链结 ActiveX 控制而不影响其成为独立的 Java 开发环境, 这些工具包括 Developer Studio、Microsoft Wizard 和一个极有效的集成调试器。

为了使广大的中国用户能迅速了解 Visual J++ 的特点、用法, 学会使用 Visual J++ 来开发 Java 程序, 特别是在 Java 程序中学会调用先进的 ActiveX 控制, 成为 Java 编程专家, 我们精心选择了这本 *the Comprehensive Guide to Visual J++* 奉献给大家。

这本书深入浅出, 运用大量的编程实例, 并通过一步步操作指导, 使读者轻松方便地学会在 Visual J++ 环境中编写 Java 程序。

这本书有以下特点:

1. 内容全面。书中详述了 Visual J++ 的特征, 并通过大量的实例来引导读者全面掌握它的用法, 使读者对 Java 和 ActiveX 有一个更切实的了解。
2. 适应面广。适合于所有 Java 开发者和 ActiveX 开发者。如果您具有 C 或 C++ 编程经验, 或者使用过 Visual C++, 希望把程序移置到 Java 或 Visual J++ 中, 本书也作了清晰的介绍。
3. 指导性强。本书花了大量篇幅来探讨编程技巧, 使读者能在短时间内学习很多编程经验。

译者于清华园

1997 年 5 月

前 言

《Visual J++精通指南》阐述了 Java 的技术要点,消除了人们对 Java 的神秘感,使读者对 Java 语言的局限性有了明确的认识。在过去一年中,Java 吸引了众多知名的 Internet 开发者的关注。在那段时间,Java 也成为大众媒介的焦点,也正因此,很多人对 Java 的了解只局限于传媒报道中所讲的一切。

大众所关注的内容歪曲了 Java 的本来面目。因此,一些人断言 Java 能做一切事情,而另一些人则持反对意见。人们对于发展现状也持两种态度:ActiveX 集团认为 ActiveX 技术更好,而 Java 集团宣称 Java 更胜一筹。这两种技术 Java 和 ActiveX 在不同方面都有其优缺点,这就更显出 Microsoft 的 Visual J++ 的强大,Visual J++ 给 Java 开发者提供了一个开发 Java 程序的虚拟环境,在此环境下开发的程序与使用 JavaSoft 的 JDK 编制的程序完全兼容,Visual J++ 也可以让人们同时使用 Microsoft 的 ActiveX 技术来工作。ActiveX 是一项在 Microsoft 的推动下不断变化的技术,其中包含 OLE(Object Linking and Embedding)功能,但它主要是集中在网络应用。简而言之,Visual J++ 是用来连接 ActiveX 和 Java 的第一座桥梁。因此,Visual J++ 成为一种独特的产品,可以使用户充分利用这两种技术。

毫无疑问,只要 Internet 还在运行,Java 和 ActiveX 就会存在,最起码,使用了 Visual J++,这两种技术就可共存。我们希望开发者在使用这两种技术时能达到同样效果。

谁需要这本书

《Visual J++精通指南》给出了 Visual J++ 的全面综述并配有一些技巧。这本书不要求读者有高级的 Java 程序设计经验。

如果您是以下几类读者中的一种,就会感到本书适合于您:

- 您是一位使用 JDK(Java Development Kit)技术的 Java 开发者,但您想学习 Microsoft 的 Visual J++ 来提高效率。
- 您是一位 ActiveX 开发者,希望了解 Microsoft 并希望运用 Java 和 Visual J++ 来给您一个更好的 ActiveX 解决方案。
- 您具有一些比较老的程序语言开发经验,并想学习如何用 Java 来编程。
- 您具有 C 或 C++ 编程经验,或者使用过 Visual C++,希望移置程序到 Java 或 Visual J++ 程序上。

在过去一些介绍 Java 的书籍中,主要的缺点是只着重于应用 JDK 技术和文本编辑器来作为生成实用的 Java 小应用程序(applet)和一些应用程序的主要手段,而《Visual J++精通指南》简化了生成实用的 Java applet 和应用程序的过程,因此它教会读者如何把 Visual J++ 作为您的主要开发环境。总之,《Visual J++精通指南》中花费了许多精力来总结应用技巧,使读者成为一名熟练的 Java 开发者。同时,教给读者有关 Microsoft 的第一手经验。

本书内容

Visual J++ 基本上分为三部分,第一部分介绍 Java 程序语言和 Visual J++ 环境,第二部分详细介绍了程序设计技巧和 Java 的功能,最后一部分介绍了 ActiveX 技术,并介绍了如何把 ActiveX 和 Java 用于 Visual J++ 中。

下面就各章所包括的内容作一介绍:

第一章,“介绍 Java 和 Visual J++”:让读者对 Visual J++ 和 Java 有了一个大概了解。也把 Java 工程作为一个整体来论述,并着重介绍了 Java 的优点,也打破了一些对 Java 的神话认识。

第二章,“Java 语言结构”:这一章详细介绍了 Java 程序语言的各个概念,主要数据类型和控制流结构。在这章中所举的例子都在 Visual J++ 环境中编译并试运行过。

第三章,“面向对象编程介绍”:这一章全面介绍了 Java 中的面向对象编程过程和 Visual J++ 中面向对象的工具。从一开始,本章假定读者对对象这一概念一无所知。介绍了生成实际环境中基于对象的 Java applet 和应用程序所应具备的知识。

第四章,“Java applet 介绍”:这章概括了所有 Java applet 模型以及一些有用的 applet 编程技巧。

第五章,“用 applet 和 AppletWizard 工作”:这一章详细介绍了 Visual J++ 中的 AppletWizard,同时也详述了如何把参数传递到 Java applet 中或把自变量传递到 Java 应用程序中。最后在本章的结尾介绍了调试,并给出一个临时设有错误的实例。用这些临时设有错误的例子,一步步查看程序员的逻辑思维,并运用 Visual J++ 来改掉这些错误。

第六章,“Java 中的多线程”:这一章讨论了在现代应用程序中有关多线程应用的所有要点,然后在一个真实环境中来展示如何充分利用 Visual J++ 的所有功能并利用各种 wizard (向导)来创建多线程 Java applet 和应用程序。

第七章,“利用 AWT 设计用户界面”:在这章中,详细介绍了所有的组件,并说明了这些组件在 Java 资源类库中生成用户界面的作用。

第八章,“利用资源编辑器设计用户界面”:第八章的内容是对第七章内容的补充,在第七章中讨论了如何设计用户界面。第八章中主要讲述如何用 Visual J++ 中的几种资源编辑器来自动生成用户界面。

第九章,“事件处理”:在这章中,可以了解 Java 如何在应用程序中接收和处理事件,同时也包括执行 Java 程序的一些最有用事件处理编程技巧。

第十章,“Visual J++ 中的调试方法”:这章中重点讨论了如何在 Visual J++ 环境中调试 Java applet 和应用程序。注意,这章中用到了 Visual J++ 中的所有调试工具。

第十一章,“利用流处理输入输出”:这章综合介绍了流在计算机中如何工作,理解使用流的意义,以及实际环境中流的处理过程。本章还包括几个指导,让读者能在一个指定文件中练习读写各种数据类型。

第十二章,“网络编程”:对 Java 联网工作进行整体介绍,详细讨论了一些快速联网方法以及低级分组编程技巧,并引入 TCP/IP 的概念。同时也包括了其他相关网络的主题,如

applet 间对话以及 applet 与用户浏览器和网络间的对话。

第十三章,“Java 数据库连接(JDBC)”：这章介绍了 JDBC、API 的概念以及 JDBC 的优缺点。同时用一个 Visual J++ 工程实例来说明如何在前台运用 Java, 又如何用 JDBC 来把它连接到后台的一个 SQL 数据库中。

第十四章,“高级 Java 计算”：这章讲了一些 Java 中不常用到,但很复杂的问题。主要是详细论述字节码、高级异常处理和原生方法的执行过程方面的内容。

第十五章,“在 Java 中使用 ActiveX”：这章是对 ActiveX 的全面介绍,以及它与 Java 的关系、Visual J++ 在何处介入。它从基本知识入手,引证实例并介绍了开发工具。

硬件与软件要求

Visual J++ 可以用于目前支持 Windows 95 和 Windows NT 4.0 的任意系统之上,注意,这本书基于 Visual J++ 的完整版考虑,对于任何共享版都不适用。

在线更新

正如我们所知,Internet 在日益更新,我们尽最大努力来赶上潮流,事实上当这本书拿去印刷的时候,新的站点又开始上网(并且在不断增加)。Ventana 公司已经想好了如何来应付这种情况,以保证这本书的内容不断更新:读者可以通过 Ventana 的 WWW 地址 <http://www.vmedia.com/updates.html> 来查看 *The Comprehensive Guide to Visual J++ Online Updates*。只要访问该站点,就可以发现有关《Visual J++ 精通指南》的更新资料。

结束语

许多人只是盲目热衷于使用 Java,只有那些想真正了解 Java 的人才可能从这本书里得到益处,他们将了解有关 Java 的所有优缺点。

Visual J++ 是对 Java 的一场革命。随着软件巨头 Microsoft 置身于 Java 语言中,推出了强有力的开发环境 Visual J++, 这就掀起了 Java 的又一次改革,并由此向前不断发展。

致 谢

我想应该首先感谢 Neweleen Trebnik。她对事业的热情和献身精神给我以巨大的鼓舞。此外,我还要感谢 Michelle Nichols,他为我开展工作提供了很大方便。最后,我要感谢项目编辑, Jennifer Rowe。

感谢 Rodney Runolfson,我的午餐伙伴,在那些富有创造力的日子里我们一起讨论面向对象编程技术,正如我在第三章中所提到的那样。

最后还要感谢我的父亲,他获得的四个学位和三十年教龄(现在投身于 Internet 的教学)时刻提醒我自己还有很长的路要走。

关 于 作 者

Daniel Joshi 是《幸福》杂志评出的 500 家公司共同拥有的一位开发者,并且是 Joshi Group (<http://www.joshigroup.com>)的一名管理者, Joshi Group 是 Microsoft 产品供应公司。作为一名 Java IDE 专家,他也是著作 *Teach Yourself Cafe in 21 Days* 的主要作者。他还与人合著了其他几本有关 Java 和 Web 程序的畅销书。Daniel 在 19 岁时就出版了他的第一本书,这本《Visual J++ 精通指南》是他不到 20 岁时写的第四本书。

目 录

出版者的话

译者序

前言

致谢

关于作者

第一章 介绍 Java 和 Visual J⁺⁺ (1)

- 1.1 什么是 Java (1)
 - 1.1.1 Java 的来历 (1)
 - 1.1.2 Java 的未来 (2)
- 1.2 现实环境中的 Java (2)
 - 1.2.1 小应用程序、应用程序和混合型程序 (2)
 - 1.2.2 Java applet 的功能和局限 (3)
 - 1.2.3 Java 在网上的例子 (3)
- 1.3 了解 Java (5)
 - 1.3.1 Java 基于 C 和 C⁺⁺ (7)
 - 1.3.2 Java 是多线程操作系统 (7)
 - 1.3.3 Java 的平台独立性 (9)
 - 1.3.4 Java 安全性 (10)
 - 1.3.5 Java 是动态的 (10)
- 1.4 Visual J⁺⁺; Java IDE (10)
- 1.5 理解 Visual J⁺⁺ (11)
 - 1.5.1 Developer Studio (12)
 - 1.5.2 Developer Studio wizard (12)
 - 1.5.3 集成的调试器 (13)
- 1.6 用 Visual J⁺⁺ 创建第一个 Java 程序 (14)
 - 1.6.1 编写程序 (14)
 - 1.6.2 编译程序 (17)
 - 1.6.3 执行程序 (18)
- 1.7 结束语 (19)

第二章 Java 语言结构 (21)

- 2.1 关键字 (21)

- 2.2 常量 (22)
 - 2.2.1 数字常量 (22)
 - 2.2.2 非数字常量 (22)
- 2.3 基本数据类型 (23)
 - 2.3.1 整型 (23)
 - 2.3.2 指导:声明数据类型 (24)
 - 2.3.3 布尔逻辑 (27)
 - 2.3.4 指导:声明布尔变量 (27)
 - 2.3.5 字符 (28)
 - 2.3.6 指导:声明字符数据类型 (28)
 - 2.3.7 浮点 (29)
 - 2.3.8 转换 (30)
 - 2.3.9 指导:数值间的转换 (30)
- 2.4 运算符 (32)
 - 2.4.1 算术运算符 (32)
 - 2.4.2 指导:算术运算符 (33)
 - 2.4.3 赋值运算符 (34)
 - 2.4.4 比较运算符 (34)
 - 2.4.5 指导:比较运算符 (35)
 - 2.4.6 逻辑运算符 (36)
 - 2.4.7 位运算符 (36)
 - 2.4.8 指导:位运算符 (38)
 - 2.4.9 指导:移位操作符 (41)
- 2.5 运算符的优先级 (42)
 - 2.5.1 代数中的运算符优先级 (43)
 - 2.5.2 Java 中的运算优先级 (43)
- 2.6 控制流 (44)
 - 2.6.1 图解控制流 (44)
 - 2.6.2 if 条件句 (44)
 - 2.6.3 指导:if 条件句 (49)
 - 2.6.4 ? 三元运算符 (50)
 - 2.6.5 for 循环 (51)
 - 2.6.6 指导:for 循环 (54)
 - 2.6.7 while 和 do while 循环 (55)
 - 2.6.8 指导:while 循环 (56)

2.6.9 switch 语句	(58)
2.6.10 指导:switch 语句	(61)
2.6.11 循环中断	(63)
2.6.12 指导;break 语句	(65)
2.7 结束语	(67)

第三章 面向对象编程介绍 (68)

3.1 Visual J++ 工程工作区和工程	(68)
3.1.1 工程工作区窗口	(69)
3.1.2 工程工作区文件	(71)
3.1.3 组织工程工作区	(72)
3.1.4 指导:生成一个工程工作区	(72)
3.2 面向对象编程	(75)
3.2.1 通常所用的编程方法	(76)
3.2.2 用对象来编程	(76)
3.2.3 对象的概念	(76)
3.2.4 Java 中用类实现对象	(76)
3.2.5 类的实例化	(78)
3.2.6 类变量	(79)
3.2.7 类的方法	(79)
3.2.8 方法的重载	(80)
3.2.9 指导:生成方法并重载	(80)
3.3 复杂数据类型	(88)
3.3.1 字符串	(89)
3.3.2 数组	(89)
3.3.3 作用域	(90)
3.3.4 指导:用范围来隐藏变量	(91)
3.4 其他的运算符优先级	(94)
3.5 继承	(94)
3.5.1 方法覆盖	(98)
3.5.2 指导:方法覆盖	(98)
3.6 对象转换	(103)
3.7 修饰字	(104)
3.7.1 访问修饰字	(104)
3.7.2 类修饰字	(106)
3.8 接口	(107)
3.9 结束语	(109)

第四章 Java applet 介绍 (110)

4.1 用 applet 工作	(110)
-----------------------	-------

4.1.1 applet 的优点	(110)
4.1.2 applet 的缺点	(111)
4.1.3 Java applet 基础	(111)
4.1.4 生命周期(Life-cycle)方法	(111)
4.1.5 Java 类和包的输入	(113)
4.2 指导:创建第一个 applet	(114)
4.3 HTML 与 applet	(122)
4.3.1 提供替用的文本	(123)
4.3.2 指定 applet 的位置	(125)
4.3.3 指定 applet 的显示方式	(125)
4.4 增强 Java applet 的图形处理	(126)
4.4.1 paint()方法	(126)
4.4.2 Graphics 类	(126)
4.4.3 图形坐标系统	(127)
4.4.4 字体	(127)
4.4.5 颜色	(129)
4.5 指导:创建第二个 applet	(130)
4.5.1 在 Netscape Navigator 中观看 applet	(134)
4.5.2 在 JDK 的 AppletViewer 中观看 applet	(135)
4.6 结束语	(135)

第五章 用 applet 和 AppletWizard 工作 (137)

5.1 java.util 包	(137)
5.2 介绍 AppletWizard	(139)
5.2.1 启动 AppletWizard	(139)
5.2.2 第一步	(139)
5.2.3 第二步	(141)
5.2.4 第三步	(142)
5.2.5 第四步	(142)
5.2.6 第五步	(143)
5.2.7 New Project Information 对话框	(144)
5.2.8 用 AppletWizard 创建一个 applet	(144)
5.3 传递信息到 Java 程序	(151)
5.3.1 传递自变量(参数)到 Java 应用程序	(151)

5.3.2 指导:在 Java applet 中传递自变量	(153)	7.2 AWT 组件	(209)
5.3.3 指导:在 Java applet 中传递参数	(159)	7.2.1 按钮(Button)	(209)
5.4 Java appletcation	(169)	7.2.2 画板(Canvas)	(210)
5.4.1 两种用法如何区别	(169)	7.2.3 复选框(Checkbox)	(211)
5.4.2 指导:用 AppletWizard 创建一个 appletcation	(169)	7.2.4 复选框组(CheckboxGroup) ...	(212)
5.5 如何在 Visual J++ 中进行调试	(175)	7.2.5 组合框(Choice, Combo Box) ...	(213)
5.6 结束语	(183)	7.2.6 标签(Label)	(215)
第六章 Java 中的多线程	(184)	7.2.7 列表(List)	(216)
6.1 多线程操作系统的评述	(184)	7.2.8 滚动条(ScrollBar)	(218)
6.2 指导:创建第一个多线程 Java 程序	(185)	7.2.9 文本组件(TextComponent) ...	(219)
6.3 线程	(187)	7.2.10 单行文本编辑域(TextField)	(220)
6.4 多线程 Java 程序设计技术	(188)	7.2.11 多行文本编辑域(TextArea)	(221)
6.4.1 创建 Thread 类的子类	(189)	7.3 AWT 容器	(223)
6.4.2 指导:创建 Thread 子类	(190)	7.3.1 框架(Frame)	(223)
6.4.3 实现 Runnable 接口	(195)	7.3.2 面板(Panel)	(226)
6.5 线程的生命周期	(196)	7.3.3 对话框(Dialog)	(226)
6.5.1 spawned 状态	(196)	7.3.4 文件对话框(FileDialog)	(226)
6.5.2 running 状态	(196)	7.3.5 菜单(Menu)	(228)
6.5.3 blocked 状态	(197)	7.4 AWT 布局管理器	(234)
6.5.4 dead 状态	(199)	7.4.1 BorderLayout 布局管理器	(235)
6.6 有关多线程的几个问题	(199)	7.4.2 CardLayout 布局管理器	(236)
6.6.1 线程优先级	(199)	7.4.3 FlowLayout 布局管理器	(238)
6.6.2 指导:设置线程的优先级	(200)	7.4.4 GridLayout 布局管理器	(239)
6.6.3 线程同步	(204)	7.4.5 无布局管理器	(241)
6.6.4 线程组	(205)	7.5 结束语	(242)
6.6.5 守护线程	(205)	第八章 利用资源编辑器设计用户界面	(244)
6.7 结束语	(206)	8.1 介绍资源编辑器	(244)
第七章 利用 AWT 设计用户界面	(207)	8.1.1 Graphic Resource Editor(图形资源编辑器)	(244)
7.1 AWT 软件包的使用	(207)	8.1.2 Dialog Resource Editor(对话框资源编辑器)	(245)
7.1.1 介绍 AWT 组件	(207)	8.1.3 Menu Resource Editor(菜单资源编辑器)	(247)
7.1.2 介绍 AWT 容器	(208)	8.2 介绍 Java Resource Wizard	(247)
7.1.3 介绍 AWT 布局管理器	(208)	8.2.1 第一步	(248)
7.1.4 添加组件	(209)	8.2.2 第二步	(248)

8.2.3 确认对话框	(249)
8.3 指导:开发用户界面	(250)
8.3.1 开发过程	(251)
8.3.2 创建工程	(251)
8.3.3 创建资源	(252)
8.3.4 将资源翻译为 Java 代码	(258)
8.3.5 给工程添加文件	(258)
8.3.6 完成 UI 应用程序的源代码 ..	(259)
8.3.7 编译并执行工程	(261)
8.4 指导:给应用程序添加菜单	(262)
8.4.1 创建资源	(262)
8.4.2 将资源翻译为 Java 代码	(268)
8.4.3 给工程添加文件	(271)
8.4.4 完成菜单的应用程序源代码	(272)
8.4.5 编译并执行工程	(273)
8.5 指导:开发 Java applet 的用户界面	(274)
8.5.1 利用 AppletWizard	(274)
8.5.2 创建资源	(276)
8.5.3 将资源翻译为 Java 代码	(279)
8.5.4 给工程添加文件	(281)
8.5.5 完成 UI 的 applet 源代码	(282)
8.5.6 编译并执行工程	(285)
8.6 结束语	(286)

第九章 事件处理..... (287)

9.1 Java 怎样处理事件	(287)
9.2 AWT 怎样处理事件	(287)
9.3 事件处理技术	(289)
9.3.1 handleEvent()技术	(289)
9.3.2 指导:覆盖 handleEvent()方法	(291)
9.3.3 事件类型	(295)
9.3.4 支持方法技术	(296)
9.4 结束语	(322)

第十章 Visual J++ 中的调试方法

10.1 Visual J++ 中的调试对话方法	(323)
--------------------------------	-------

10.1.1 开始调试对话	(323)
10.1.2 断点	(325)
10.1.3 观察变量和表达式信息	(326)
10.1.4 Watch 窗口	(327)
10.1.5 Variables 窗口	(328)
10.1.6 Call Stack 窗口	(329)
10.1.7 Disassembly 窗口	(330)
10.1.8 Exceptions 窗口	(331)
10.1.9 Thread 窗口	(332)
10.1.10 关闭调试对话	(332)
10.2 结束语	(332)

第十一章 用流来处理输入/输出 ... (335)

11.1 理解“流”的概念	(335)
11.2 InputStream 类	(337)
11.2.1 InputStream 类的方法	(337)
11.2.2 ByteArrayInputStream 类	(338)
11.2.3 FileInputStream 类	(339)
11.2.4 SequenceInputStream 类	(340)
11.2.5 StringBufferInputStream 类 ..	(340)
11.2.6 FilterInputStream 类	(341)
11.2.7 PipedInputStream 类	(342)
11.3 OutputStream 类	(342)
11.3.1 OutputStream 类的方法	(343)
11.3.2 ByteArrayOutputStream 类 ..	(343)
11.3.3 FileOutputStream 类	(344)
11.3.4 FilterOutputStream 类	(344)
11.3.5 PipedOutputStream 类	(346)
11.4 File 类	(346)
11.4.1 GetInfo 方法	(347)
11.4.2 GetStatus 方法	(347)
11.5 RandomAccessFile 类	(347)
11.5.1 read()方法	(348)
11.5.2 write ()方法	(348)
11.5.3 独立方法:seek()、getFilePointer()和 length()	(348)
11.6 FileDescriptor 类	(349)
11.7 StreamTokenizer 类	(349)
11.8 java.io 异常事件	(349)
11.8.1 指导:文件输入输出	(349)

11.8.2 指导: 字节数组输入输出 ...	(353)	14.3.2 访问 Windows 原生方法	(411)
11.8.3 指导: 数据输入输出	(359)	14.3.3 创建原生方法	(412)
11.9 结束语	(363)	14.4 异常事件处理	(414)
第十二章 网络编程	(364)	14.4.1 引发 Java 异常事件	(414)
12.1 联网介绍	(364)	14.4.2 捕捉 Java 异常事件	(415)
12.2 Java 中的网络编程	(367)	14.4.3 不同 Java 类引发的异常事件	(418)
12.2.1 用 Internet 协议通信	(367)	14.4.4 重引发 Java 异常事件	(418)
12.2.2 指导: 使用 URLConnection	(368)	14.5 结束语	(419)
12.2.3 与浏览器通信	(374)	第十五章 在 Java 中使用 ActiveX	(420)
12.2.4 applet 之间的通信	(375)	15.1 什么是 ActiveX	(420)
12.2.5 与服务器通信	(377)	15.1.1 ActiveX Internet 模型	(421)
12.3 结束语	(379)	15.1.2 ActiveX 控制	(422)
第十三章 Java 数据库连接(JDBC)	(380)	15.1.3 ActiveX 原稿	(424)
13.1 开放数据库连接(ODBC)	(380)	15.1.4 HTML 中的 ActiveX 对象 ...	(424)
13.2 ODBC 结构	(381)	15.1.5 从 ActiveX 到 BrandX	(426)
13.3 JDBC 结构	(383)	15.2 ActiveX 和 OLE	(427)
13.3.1 java.sql 包	(384)	15.3 ActiveX 与 Java	(427)
13.3.2 java.sql 接口	(384)	15.4 ActiveX 和 COM	(429)
13.3.3 java.sql 类	(386)	15.4.1 为什么要有 COM	(429)
13.4 JDBC 的实现	(387)	15.4.2 为什么要使用 ActiveX COM 对象	(431)
13.5 使用 JDBC 客户机/服务器运算 ...	(388)	15.5 ActiveX COM 对象及 Java	(432)
13.6 SQL 总结	(390)	15.5.1 在 Java applet 中使用 ActiveX COM 对象	(434)
13.7 结束语	(404)	15.5.2 用 Java 创建 ActiveX COM 对象	(434)
第十四章 高级 Java 计算	(405)	15.6 Visual J++ ActiveX 实例	(436)
14.1 Java 虚拟机	(405)	15.7 网络中实际的 ActiveX	(436)
14.2 理解字节码	(407)	15.8 结束语	(439)
14.2.1 .class 文件	(407)	附录 A	(442)
14.2.2 理解字节码校验	(409)	附录 B	(443)
14.3 原生方法	(410)		
14.3.1 什么是 DLL	(410)		

第一章 介绍 Java 和 Visual J⁺⁺

这一章全面介绍了 Java, 包括 Java 的正式定义, 如何在网上找到 Java 以及 Microsoft Visual J⁺⁺ 如何发展了 Java。本章首先对 Java 进行基本介绍, 然后介绍 Visual J⁺⁺ 及其特性, 最后在 Visual J⁺⁺ 环境中建立了一个 Java 程序实例。

1.1 什么是 Java

Java 是一种面向对象的编程语言。面向对象是指 Java 以一组对象来组织其程序。这实际上不是 Java 的特性, 其他一些编程语言像 SmartTalk 和 C⁺⁺ 在它们的语言中运用了好几年这种面向对象的结构, 但 Java 是一种更容易懂的程序(特别是与 C 或者 C⁺⁺ 相比)。

Java 是在 C⁺⁺ 相当强大之后才初具规模的。在应用程序开发之中 C⁺⁺ 是一种相当普遍的基本程序语言。开发环境由 UNIX 转到 Macintosh 又到 Windows 都可以使用 C⁺⁺。但问题是无论从语言上还是二进制水平来看, C⁺⁺ 在这些平台上不易于互相移植。因为虽然上面提到的每一种操作系统都可使用 C⁺⁺, 但都以不同的方式来使用。这就意味着在不同的平台间转换时, 需要重新写这个程序, 并需在希望运行的环境中选择其编译器来编译。这些过程听起来也很麻烦。但是, Java 没有这些限制; 它与 C⁺⁺ 功能一样强大, 却从设计时就考虑到可以在不同平台间完全移植。

Java 除了具有平台独立性之外, 也很容易使用, 而且很容易学, 特别是与其他一些语言相比如 C⁺⁺、SmartTalk 和 Visual Basic。Java 加强了 C⁺⁺ 的功能, 除去了一些过于复杂的部分。

对于用一些老的, 非基于对象的编程语言(如 Pascal 和 Visual Basic)来编程的人员来说, 其最难于理解的是 Java 的面向对象性。这里的对象并非用于 C, Visual Basic, 以及其他一些网上的描述语言, 它是用一种很复杂的方法来组织程序。然而, 这种方法是未来的趋势, 一旦学会运用一种面向对象的语言, 就会很容易学习其他的面向对象语言。

1.1.1 Java 的来历

1991 年, Sun Microsystem 公司开发了一种程序语言 Oak 语言(得名于这种语言开发者之一 James Gosling 家窗外的橡树), 现在把它叫做 Java, 这种语言移植性很好, 且程序短小, 它很适合于为个人数字助理(Personal Digital Assistants, PDA)和电视等可移动电子产品编写软件。当年的 Oak 语言是一种革命性的语言, 却没有引起电子工业界太多的注意。

直到 1995 年春, Java 才开始流行, 那时, 第一个用 Java 语言编写的 Java 浏览器 HotJava 在公众中传开, 且主要是把 Java 的一种特殊类型应用程序 applet 用于 Internet 之上。applet 体现了 Java 的强大和可移植性。直到 Java 用于 Internet 上的 HotJava 的出现, 才赢得了大

众的认可,并得到工业界一些大的软件和硬件商的推崇。今天,HotJava 不是唯一一个支持 Java applet 的 Internet 浏览器。其他一些大的 Internet 浏览器厂家像 Netscape Navigator(推出 2.0 版本)和 Microsoft Internet Explorer(推出 3.0 版本)也可以使用户看到 Java applet 或其他与 Java 相关的技术。

其他的一些 Java 工具

目前在网上 Java 除了 applet 之外,还有其他一些工具。最引人注目的是 JavaScript 语言(由 Netscape 公司开发),这是一种从 Java 中提取出的面向对象的描述语言。因为 JavaScript 提供了更多的浏览器钩符,就像是 HTML 语言的增强版。

1.1.2 Java 的未来

Java 语言流行的时间虽然不长,但自从被官方认可之后,已经敞开了大门。去年夏天,Java 推出了一系列应用程序接口(Application Programming Interfaces, API),称为 JDB,使 Java applet 和其应用程序能配合 SQL 数据库使用。

从那以后,就又有了一组新的 API:一组与电力行业有关,使 Java 能在其环境中运行;另一组称为 Jeeves,作为一组运用 Java 的网络服务器 API。

其他一些从 Java 中延伸出的新技术是集成开发环境 IDE(Integrated Development Environment),主要用于 Java。这些都是基于 Windows 的程序,可以让读者建立或编译 Java applet 和应用程序。现代 Java IDE 也允许用对象浏览型工具来查看所编的 Java 程序。

Visual J++ 是 Microsoft 为 Java 做的一个 IDE。Visual J++ 与其他基于 Java 的 IDE 都相似,不同之处在于 Visual J++ 结合了 Java 的一些增强功能和 ActiveX 的技术。ActiveX 是 Microsoft 用于 Internet 上的开发工具箱。它给 Visual J++ 提供了一整套基于 Microsoft OLE(Object Linking and Embedding)的 Internet 技术。在 Visual J++(正式称为 Jakarta)公开化之前,Microsoft 还没有介入 Java 语言的开发领域。

看完这本书之后,就会感到 Visual J++ 集成了两种语言(ActiveX 和 Java)的优点,利用它可以成为单独的一种 Internet 使用工具。Visual J++ 使得学习 Java 更容易:使用 wizard,虚拟开发环境,集成调试环境和在线服务之后,Java 开发变得非常方便。

1.2 现实环境中的 Java

最初,有些人在注意到 Java applet 的严格安全模式时,争辩说尽管 Java 在理论上看似强大,但在现实生活中是不可行的。现在,Java 已经实践了一段时间,并且证明是一种解决开发问题的有效方法。今天,许多公司推崇使用 Java 来进行专门的应用开发。Internet 开发环境也是开发 Java 的非常重要和现实的工具。网络上也用 Java 来解决一些编程问题。让我们再看一下有哪些 Java 类型。

1.2.1 小应用程序、应用程序和混合型程序

Java 语言分为两种不同的应用形式:小应用程序和应用程序。

Java 程序的命名方法

在 Java 中,对所创建的这类软件有一种命名方式。小应用程序(applet)代表可移植且可用于 Internet 上的 Java 软件类型。应用程序(application)代表更传统的 Java 软件,与 C++ 程序的示例相似。混合型应用程序(appletcation)是指 Java 软件在不同的主机环境中可作为不同的类型,或者是小应用程序或者是应用程序。最后说明一点,用户可以使用程序(program)来同时指代这三种类型。

applet 是 Java 程序中特殊的一类。它的独特性在于 HTML 网页可调用它们。applet 在生命期里的各个不同阶段具有不同的行为。在 applet 中这些行为是根据生命周期模型发生的。applet 的设计比传统应用程序更安全,可保证不会对网络产生危害。

Java 应用程序是更传统的 Java 程序。这种程序更接近于其他编程语言编写的程序。Java 应用程序比 Java applet 少了一些安全限制。这些应用程序不能直接在 Internet 上访问。

appletcation 是 applet 和应用程序的混合型。根据它被执行的环境来决定它是 applet 还是应用程序。注意,appletcation 纯粹是一种编程技术,并没有改变以上有关 applet 和应用程序的任何定义。

对大多数人来说,应用程序这个术语很熟悉。所有的编程语言(特别是 C 或 C++)都使用很相似的应用模型。相反,applet 程序在 Java 语言中是比较独特的,也许不太好懂,所以下一步深入到 Java applet 内部和工作过程中来解释。

1.2.2 Java applet 的功能和局限

创建 Java applet 最主要的一个原因是 Java 具有强大的功能。而这种功能强大的语言如果被有破坏意图的用户使用将会对客户环境(Web 浏览器的主机环境)有潜在的危害。为了保护客户机系统,applet 设计为自限制型。另外,大多数 Internet 浏览器对 Java applet 增加了安全性保护以免由 Java applet 引起的有意或无意的损害。但是,所有的这些安全措施都缺乏多样性。

在 Java 世界里,由 Java applet 优缺点引起的有关安全限制的问题一直争论不休。Internet 是一个开放,无限制的环境并带有潜在的不安全性,这就要求 Java 找出一条途径来保护自己,并让每个人都可以使用。因为 Java applet 是一种自包含,自限制,自校验工具,不应被当作“特洛伊木马侵袭”(其中隐藏着毁灭性病毒码)或可感染 applet 程序进而影响客户机环境的工具。

也就是说,为了保护客户环境限定了 applet 的多样化以及前途无量的发展能力。最终,Java 的创建者走向了保守派。大多数 Java 的开发者对这种相当保守的 applet 安全模式表示满意。少数开发者也在抱怨 Java“太过安全”,过于自我限制了。但这部分人属于少数。试想一下,人们愿意拥有一个具有保护性的 applet 还是一群发怒的用户呢?

1.2.3 Java 在网上的例子

在网上很容易找到 Java applet。这些网上的 applet 很多都是共享或免费使用的,用户