

Visual J++ 教程

● 刘甲耀 严桂兰 刘波 编著



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
URL: <http://www.phei.com.cn>

Visual J++ 教程

刘甲耀 严桂兰 刘 波 编著

電子工業出版社

Publishing House of Electronics Industry

内 容 提 要

本书阐述 Visual J++ 面向对象程序设计方法,包括以下内容:三个基本概念(Java 程序设计、Visual J++ 程序设计、面向对象程序设计);Java 语言基础;对象的操作;选择、循环程序结构与数组;Java 类的创建与应用程序;Java Applet 基础;图形绘制与动画制作;图形用户界面设计;图形化窗口设计;修饰符、访问控制与类设计;软件包、软接口与异常处理;Java 的输入/输出;COM、ActiveX 与 JDBC;网络通信与声像播放;程序的可视化动态调试;JavaScript 简介;VBScript 简介;用 Scripting 集成 Applet 和 ActiveX 控制。本书可作大专院校计算机及相关专业的研究生、大学本科生教材,并可供各行各业从事计算机应用,特别是从事网上编程人员使用。

书 名: **Visual J++ 教程**

编 者: 刘甲耀 严桂兰 刘 波

责任编辑: 施玉新

印 刷 者: 北京天竺颖华印刷厂

出版发行: 电子工业出版社出版、发行

北京市海淀区万寿路 173 信箱 邮编 100036 发行部电话 68214070

经 销: 各地新华书店经销

开 本: 787 × 1092 1/16 印张: 31.25 字数 800 千字

版 次: 1998 年 8 月第 1 版 1998 年 8 月第 1 次印刷

书 号: ISBN 7-5053-4785-3
TP · 2321

定 价: 38.00 元

凡购买电子工业出版社的图书,如有缺页、倒页、脱页者、本社发行部负责调换

版权所有·翻印必究

前 言

Java 是一种基于网络的通用的面向对象编程语言,适用于编写各种各样的软件。Java 语言与 C 和 C++ 相似,但省略了一些难以理解和不安全的语义特征。Java 简单易学,适用于各种平台和操作系统,编译后的代码能在国际互联网(Internet)上传送,并确保用户安全运行。

Visual J++ (简称 VJ++) 是开发、编辑、测试与调试 Java applet (小应用程序)和 application(应用程序)的微软的集成开发环境(IDE),具有 Java 开发和利用的强壮性、普通库的重用性和 Java 代码的维护性,并具有微软新技术、部件对象模型(COM)与 Java applet 和 application 结合的内部能力,因而使用 Visual J++ 更容易学习 Java。

为适应当前 Internet 的迅猛发展及当前各行各业学习 Java 语言的需要,特别是大专院校为研究生和本科生而开设面向对象程序设计课程的需要,我们特编写《Visual J++ 教程》。

本书的特点是:

1. 适应当前国际通信网络及计算机迅猛发展,为培养人材,及时提供最新的由浅入深的教材。
2. 开发工具与语言相结合,采用最新版本(Visual J++)。
3. 取材广泛,由浅入深,重点、难点分明,易学、易掌握。
4. 方法与示例并举,所有示例均在微机上通过,并提供教学习题软盘。

美国黎曼兄弟投资银行刘涌博士、加州 Berkeley 大学刘浩博士、花旗银行总部张聪硕士、深圳先科集成电路设计公司谷惠英总工程师均参加本书部分编写与程序调试工作。全书由童隆正老师审校。

在本书编写过程中,得到国立华侨大学电脑系 94 级学生、研究生、系微机室杨毅华等多方人员的大力支持和帮助,在此,一一表示衷心感谢。

作 者

目 录

第 1 章 三个基本概念	(1)
1.1 Java 程序设计的基本概念	(1)
1.1.1 Java 的含义	(1)
1.1.2 Visual J++ 的含义	(2)
1.1.3 在 HTML 页上嵌入 Java 小应用程序	(2)
1.1.4 Java Applets 与 Java Applications 的区别	(4)
1.1.5 Java 语言的特点	(4)
1.1.6 Visual J++ IDE	(6)
1.2 Visual J++ 程序设计的基本概念	(7)
1.2.1 Visual J++ 的安装	(7)
1.2.2 Visual J++ 的图像漫游	(9)
1.2.3 Visual J++ 的启动与项目工作空间窗口	(9)
1.2.4 使用 VJ++ 编制 VJ++ Application	(12)
1.2.5 使用 VJ++ 编制 VJ++ Applet	(14)
1.2.6 使用 VJ++ 编制包含 Applet 的 HTML 文件	(16)
1.3 面向对象程序设计的基本概念	(17)
1.3.1 对象与类	(17)
1.3.2 属性与方法	(18)
1.3.3 继承、软接口与软件包	(23)
1.3.4 子类的创建	(27)
第 2 章 Java 语言基础	(31)
2.1 简单语句与注解	(31)
2.1.1 简单语句	(31)
2.1.2 注解	(31)
2.2 变量与数据类型	(32)
2.2.1 概述	(32)
2.2.2 变量说明	(32)
2.2.3 变量的标识符	(33)
2.2.4 变量类型	(34)
2.2.5 变量赋值	(35)
2.2.6 注意事项	(35)
2.3 文字常量	(36)
2.3.1 数字文字	(36)
2.3.2 布尔文字	(36)
2.3.3 字符文字	(36)
2.3.4 字符串文字	(36)

2.4 表达式与运算符	(37)
2.4.1 算术运算	(38)
2.4.2 多重赋值与组合赋值	(41)
2.4.3 增 1 与减 1 运算	(42)
2.4.4 关系运算	(43)
2.4.5 逻辑运算	(43)
2.4.6 按位运算	(44)
2.4.7 运算符优先级	(44)
2.5 字符串运算	(46)
2.5.1 用法	(46)
2.5.2 注意	(47)
第 3 章 对象的操作	(48)
3.0 创建新对象	(48)
3.1.1 new 的使用	(48)
3.1.2 new 的作用	(50)
3.1.3 内存管理	(50)
3.2 类变量和数据成员的访问与设置	(50)
3.2.1 数据成员的表示	(50)
3.2.2 数据成员的赋值	(51)
3.2.3 类变量的定义与访问	(52)
3.3 方法的调用与对象的引用	(53)
3.3.1 方法的调用	(53)
3.3.2 对象的引用	(55)
3.4 对象与基本类型的强制转换	(56)
3.4.1 强制转换基本类型	(56)
3.4.2 强制转换对象	(57)
3.4.3 基本类型与对象的相互转换	(58)
3.5 有关对象	(58)
3.5.1 比较对象	(58)
3.5.2 确定对象的类	(60)
3.6 Java 类库	(60)
第 4 章 选择、循环程序结构与数组	(62)
4.1 块语句	(62)
4.1.1 用法	(62)
4.1.2 应用示例	(62)
4.2 选择程序结构	(63)
4.2.1 单、双分支选择结构	(63)
4.2.2 多分支选择结构	(68)
4.3 面向问题循环结构	(72)
4.3.1 用法	(72)
4.3.2 注意事项	(73)

4.3.3 应用示例·····	(73)
4.4 前判断与后判断循环结构·····	(77)
4.4.1 前判断循环结构·····	(77)
4.4.2 后判断循环结构·····	(78)
4.4.3 应用示例·····	(78)
4.5 break、continue 与带标号的循环·····	(80)
4.5.1 跳出循环(break)·····	(80)
4.5.2 继续循环(continue)·····	(81)
4.5.3 带标号的循环·····	(82)
4.5.4 应用示例·····	(83)
4.6 数组·····	(88)
4.6.1 定义数组变量·····	(88)
4.6.2 创建数组对象·····	(89)
4.6.3 访问数组元素·····	(89)
4.6.4 改变数组元素·····	(94)
4.6.5 多维数组·····	(95)
第5章 Java 类的创建与应用程序·····	(98)
5.1 类的创建·····	(98)
5.1.1 类的定义·····	(98)
5.1.2 数据成员和类变量的创建·····	(99)
5.1.3 方法的创建·····	(101)
5.2 Java 应用程序的创建与命令行参数·····	(108)
5.2.1 Java 应用程序的创建·····	(108)
5.2.2 Java 应用程序与命令行参数·····	(108)
5.3 方法重载·····	(111)
5.3.1 用法·····	(111)
5.3.2 应用示例·····	(111)
5.4 构造方法·····	(114)
5.4.1 作用·····	(114)
5.4.2 基本构造方法·····	(114)
5.4.3 调用其它的构造方法·····	(115)
5.4.4 重载构造方法·····	(117)
5.5 方法的抑制(Overriding methods)·····	(118)
5.5.1 创建抑制现有方法的方法·····	(119)
5.5.2 调用原先的方法·····	(121)
5.5.3 构造方法的抑制·····	(123)
5.5 结束方法(Finalize methods)·····	(124)
5.6.1 用法·····	(124)
5.6.2 注意·····	(124)
第6章 Java Applet 基础·····	(125)
6.1 Applet 中的 ActiveX 与 Applet 的创建·····	(125)

6.1.1	Applet 与 Application 的差异	(125)
6.1.2	Applet 中的 ActiveX	(126)
6.1.3	Applet 的创建	(126)
6.1.4	主要的 Applet 组织	(127)
6.1.5	VJ++ Java Applet Wizard 的使用	(128)
6.2	在 Web 页上包含 Applet	(134)
6.2.1	<APPLET> 标记	(134)
6.2.2	制作适用于 Web 页的 Java Applets	(136)
6.3	<APPLET> 标记的进一步介绍	(136)
6.3.1	ALIGN 标记	(136)
6.3.2	HSPACE 和 VSPACE 标记	(138)
6.3.3	CODE 和 CODEBASE 标记	(138)
6.4	传递参数给 Applets	(139)
6.4.1	用法	(139)
6.4.2	注意	(140)
第 7 章	图形绘制与动画制作	(144)
7.1	图形类与图形坐标系统	(144)
7.1.1	图形类	(144)
7.1.2	图形坐标系统	(144)
7.2	图形的绘制与填充	(145)
7.2.1	画线	(145)
7.2.2	画矩形	(145)
7.2.3	画多边形	(149)
7.2.4	画椭圆	(152)
7.2.5	画弧	(153)
7.2.6	一个简单的绘图示例	(157)
7.2.7	图形的拷贝与清除	(158)
7.3	文本与字体	(161)
7.3.1	字体对象的创建	(161)
7.3.2	字符与字符串的描写	(161)
7.3.3	找出有关字体的信息	(163)
7.4	颜色	(165)
7.4.1	颜色对象的使用	(165)
7.4.2	当前颜色的测试与设置	(166)
7.4.3	单色示例	(167)
7.5	VJ++ 动画的创建	(168)
7.5.1	绘图与重画	(168)
7.5.2	Applet 的开始与停止	(169)
7.5.3	start() 与 stop() 的组合使用	(169)
7.6	动画的多线程控制	(171)
7.6.1	多线程的含义与作用	(171)
7.6.2	多线程 Applets 的编写	(172)

7.6.3 多线程应用示例	(173)
7.7 动画闪烁的解决办法	(175)
7.7.1 产生闪烁原因与解决途径	(176)
7.7.2 解决闪烁问题的具体办法	(176)
第8章 图形用户界面设计	(187)
8.1 资源编辑器与 AWT 概述	(187)
8.1.1 资源编辑器的使用	(187)
8.1.2 AWT 元件与 AWT 类	(188)
8.2 简单对话资源的创建	(190)
8.2.1 Java Resource Wizard 的用法	(191)
8.2.2 使用 Java Resource Wizard 生成的类	(191)
8.3 基本的用户界面部件	(193)
8.3.1 标签(Labels)	(193)
8.3.2 按钮(Bullons)	(195)
8.3.3 文本域(Text Fields)	(197)
8.3.4 检查框(Checkboxes)	(200)
8.3.5 无线电按钮(Radio Buttons)	(201)
8.3.6 选择菜单(Choice Menus)	(201)
8.4 面板与布局	(203)
8.4.1 布局管理器	(203)
8.4.2 FlowLayout 类	(204)
8.4.3 网格与网格包布局	(204)
8.4.4 边界布局(Bord Layouts)	(206)
8.4.5 卡片布局(Card Layouts)	(207)
8.4.6 使用资源编辑器创建对话框的布局	(209)
8.4.7 镶边(Insets)	(210)
8.5 面板与部件的嵌套	(211)
8.5.1 嵌套面板	(211)
8.5.2 事件与嵌套式面板	(212)
8.6 其它用户界面部件	(213)
8.6.1 文本区域(Text Areas)	(213)
8.6.2 滚动列表框(Scrolling Lists)	(214)
8.6.3 滚动条与滑块	(214)
8.6.4 画板(Canvases)	(216)
第9章 图形化窗口设计	(217)
9.1 窗口、菜单与对话框	(217)
9.1.1 画面	(217)
9.1.2 菜单	(219)
9.1.3 对话框	(225)
9.1.4 文件对话框	(228)
9.2 用户界面动作与事件的处理	(229)

9.2.1	点击鼠标	(229)
9.2.2	移动鼠标	(231)
9.3	键盘事件	(235)
9.3.1	KeyDown 和 KeyUp 方法	(236)
9.3.2	缺省键	(236)
9.3.3	输入、显示和移动字符的示例	(237)
9.3.4	修饰符键的测试	(238)
9.4	AWT 事件处理器	(239)
9.5	用户界面动作与事件的处理	(241)
9.5.1	菜单动作	(243)
9.5.2	窗口事件	(244)
9.5.3	其他用户界面事件	(244)
第 10 章	修饰符、访问控制与类设计	(246)
10.1	修饰符概述	(246)
10.1.1	Java 修饰符的种类	(246)
10.1.2	用法	(246)
10.1.3	注意	(246)
10.2	控制访问方法与变量	(247)
10.2.1	访问控制的重要性	(247)
10.2.2	四个 P 保护	(247)
10.2.3	方法保护与继承	(252)
10.2.4	实例变量保护与访问符方法	(253)
10.3	类变量与方法	(255)
10.3.1	用法	(255)
10.3.2	访问方式	(255)
10.3.3	应用实例	(255)
10.4	类、方法与变量的结束	(257)
10.4.1	结束类(Finalizing Classes)	(257)
10.4.2	结束变量(Finalized Variables)	(257)
10.4.3	结束方法(Finalized methods)	(258)
10.5	抽象类与方法	(258)
10.5.1	抽象类	(258)
10.5.2	抽象方法	(259)
第 11 章	软件包、软接口与异常处理	(264)
11.1	大小型程序设计 with 软件包	(264)
11.1.1	大型程序设计与小型程序设计	(264)
11.1.2	软件包的含义	(264)
11.2	软件包的使用	(265)
11.2.1	全软件包与类名	(265)
11.2.2	import 命令	(265)
11.2.3	名字冲突	(266)

11.2.4	CLASSPATH 变量和类的定位	(266)
11.3	用户软件包的创建	(267)
11.3.1	选取软件包名	(267)
11.3.2	创建目录结构	(267)
11.3.3	使用 package 在软件包中添加自己的类	(268)
11.3.4	创建具有 Visual J++ 用户接口的软件包	(268)
11.3.5	软件包与类保护	(269)
11.4	软接口的定义	(271)
11.4.1	单一继承问题	(271)
11.4.2	抽象设计与具体实现	(272)
11.4.3	软接口与类	(272)
11.5	软接口的实现与使用	(273)
11.5.1	implements 关键字	(273)
11.5.2	多重软接口的实现	(274)
11.5.3	软接口的其它使用	(275)
11.6	软接口的创建	(275)
11.6.1	新软接口的创建	(276)
11.6.2	软接口内的方法	(276)
11.6.1	软接口的扩展	(277)
11.7	异常处理	(282)
11.7.1	try...catch 结构	(282)
11.7.2	finally 语句	(285)
11.7.3	使用 throw 创建异常	(289)
第 12 章	Java 的输入/输出	(292)
12.1	系统类的标准输入/输出	(292)
12.1.1	用法	(292)
12.1.2	应用示例	(293)
12.2	输出数据流类的使用	(295)
12.2.1	类 OutputStream	(295)
12.2.2	类 ByteArrayOutputStream	(296)
12.2.3	类 FileOutputStream	(299)
12.2.4	类 PipedOutputStream	(299)
12.2.5	类 FilterOutputStream	(299)
12.2.6	类 BufferedOutputStream	(300)
12.2.7	类 DataOutputStream	(300)
12.2.8	类 PrintStream	(301)
12.3	输入数据流类的使用	(303)
12.3.1	类 InputStream	(303)
12.3.2	类 ByteArrayInputStream	(305)
12.3.3	类 FileInputStream	(306)
12.3.4	类 PipedInputStream	(307)
12.3.5	类 SequenceInputStream	(308)

12.3.6	类 StringBufferInputStream	(310)
12.3.7	类 FilterInputStream	(311)
12.3.8	类 BufferedInputStream	(311)
12.3.9	类 DataInputStream	(311)
12.3.10	类 LineNumberInputStream	(316)
12.3.11	类 PushbackInputStream	(318)
12.4	其它相关类的使用	(318)
12.4.1	类 File	(318)
12.4.2	类 FilenameFilter	(319)
12.4.3	类 RandomAccessFile	(319)
12.4.4	类 FileDescriptor 和 StreamTokenizer	(321)
12.5	封套类数据类型的使用	(322)
12.5.1	用法	(322)
12.5.2	应用示例	(322)
第 13 章	COM, ActiveX 与 JDBC	(325)
13.1	部件对象模型(COM)	(325)
13.1.1	从 Visual J++ 调用 COM 对象	(325)
13.1.2	Type Library Wizard 的运行	(326)
13.1.3	Type Library Wizard 的作用	(327)
13.1.4	在 Java 中使用 COM 对象	(328)
13.1.5	按照 COM 对象表示 Visual J++ Applet	(329)
13.2	ActiveX 控制	(329)
13.2.1	受托和未受托的 Applet	(330)
13.2.2	创建带符号的 .CAB 文件	(331)
13.2.3	确保用户机上 JavaSupport 是最新的	(331)
13.3	数据库访问	(331)
13.3.1	数据访问对象(DAO)	(332)
13.3.2	远程数据对象(RDO)	(332)
13.3.3	Java 数据库联通性(JDBC)	(333)
第 14 章	网络通信与声像播放	(334)
14.1	Java 与网络连接概述	(334)
14.2	在 Applet 内创建连接	(334)
14.2.1	方法	(334)
14.2.2	应用实例	(335)
14.3	打开 Web 连接	(341)
14.3.1	openStream()	(341)
14.3.2	URLConnection 类	(345)
14.3.3	Socket 类	(345)
14.4	Applet 连接的其它问题	(345)
14.4.1	showStatus()方法	(346)
14.4.2	Applet 信息	(346)

14.4.3	Applet 之间的通信	(346)
14.5	图像装载与显示	(347)
14.5.1	图像的装载	(347)
14.5.2	图像的显示	(348)
14.6	声音文件的播放	(354)
14.6.1	基本的声音文件播放功能	(354)
14.6.2	使用 play 方法播放声音文件	(355)
14.6.3	使用 AudioClip 播放声音文件	(356)
14.6.4	使用参数指定声音文件播放	(357)
14.7	动画的播放	(360)
14.7.1	使用 MediaTracker	(360)
14.7.2	使用 Applet Wizard 生成简单的动画程序	(360)
第 15 章	程序的可视化动态调试	(364)
15.1	编译出错的定位与改正	(364)
15.1.1	编译出错与查错方法	(364)
15.1.2	改错方法与实例	(366)
15.2	Microsoft Developer Stdudio 调试器的使用	(370)
15.2.1	用法	(370)
15.2.2	applet 的调试	(370)
第 16 章	JavaScript 简介	(375)
16.1	JavaScript 基础	(375)
16.1.1	JavaScript 基本概念	(375)
16.1.2	数据类型	(377)
16.1.3	变量	(377)
16.1.4	文字	(378)
16.1.5	表达式	(379)
16.1.6	运算符	(379)
16.2	语句	(381)
16.2.1	条件语句	(381)
16.2.2	循环语句	(382)
16.2.3	对象操作语句	(384)
16.3	函数与数组	(387)
16.3.1	函数	(387)
16.3.2	数组	(389)
16.4	事件与 JavaScript 对象	(391)
16.4.1	事件	(391)
16.4.2	JavaScript 对象	(393)
第 17 章	VBScript 简介	(394)
17.1	VBScript 基础	(394)
17.1.1	VBScript 基本概念	(394)
17.1.2	数据类型	(396)

17.1.3 变量	(397)
17.1.4 文字	(398)
17.1.5 表达式	(398)
17.1.6 运算符	(399)
17.2 数组与语句	(400)
17.2.1 数组	(400)
17.2.2 语句	(401)
17.3 过程与内部函数	(405)
17.3.1 过程	(405)
17.3.2 内部函数	(407)
17.4 事件与 VBScript 对象模型	(408)
17.4.1 事件	(408)
17.4.2 VBScript 对象模型	(409)
第 18 章 用 Scripting 集成 Applets 和 ActiveX 控制	(411)
18.1 使用 Applet Wizard 创建 applet	(411)
18.1.1 创建新项目	(411)
18.1.2 Cirl Applet	(412)
18.2 Java Applet 的集成与参数传递	(419)
18.2.1 集成 Java Applet 到 HTML	(419)
18.2.2 向 Java Applet 传递参数	(420)
18.3 Java Applet 控制	(420)
18.3.1 用 VBScript 控制 Java Applet	(420)
18.3.2 用 JavaScript 控制 Java Applet	(423)
18.4 <OBJECT> 标记以及 ActiveX 的集成与控制	(425)
18.4.1 <OBJECT> 标记的使用	(425)
18.4.2 集成 ActiveX 控制到 HTML	(426)
18.4.3 用 Scripting 控制 ActiveX	(430)
附录 A Java 语言参考	(437)
附录 B 类层次图	(443)
附录 C Java 类库	(454)
附录 D Java 与 C 及 C++ 的差异	(461)
附录 E JavaScript 语言参考	(464)
附录 F VBScript 语言参考	(475)
附录 G 本书符号说明	(484)

第 1 章 三个基本概念

Java 是一种基于网络的通用型面向对象编程语言,适用于编写各种各样的软件。本教程讲述 Java 语言,Java 小应用程序(Java applets)与 Java 应用程序(Java applications)。为此,首先得了解 Java 编程、Visual J++ 编程和面向对象编程三个基本概念。本章共讲三个问题:

- Java 程序设计的基本概念
- Visual J++ 程序设计的基本概念
- 面向对象程序设计的基本概念

1.1 Java 程序设计的基本概念

1.1.1 Java 的含义

1. Java 是美国 Sun 公司于 1995 年 5 月公布的一种通用的面向对象编程语言。它是网络时代的一颗明星。Java 语言实际上包含了两层意思:第一,它是 Sun 公司研制的新型简化的基于对象的开放系统语言。它允许软件开发人员将应用分布到万维网(World Wide Web 简称 WWW)或由信息系统厂商所开发的任一前端设备上;第二,它是一种虚拟机(或称 Java 虚拟机),它最终可使所有基于 Java 的应用程序成为普遍存在的程序,即不管硬件或操作系统环境,Java 应用程序到处都可以运行。

对于 Java 虚拟机来说,同样的 Java 应用程序,完全可在无任何改变和重新编译的情况下的任何平台上运行,其范围可包括:从智能蜂窝电话、膝上型机到基于 Windows 95、NT、OS/2、UNIX 工作站或服务器,甚至运行 MVS 的 AS/400 或 IBMS/390。Java 技术开创了以网络为中心的客户端/服务器企业计算的时代。Java 的发明建立了一个以自由授权为标志的软件世界,Java 可使操作系统像窗格玻璃一样透明。软件、信息和网络技术市场将由质量和服务水平,而不是其占有者专用权来决定。

2. Java 与其它大部分程序设计语言的不同在于:

(1)除了能创建独立的应用程序之外,还能创建 Java 小应用程序,Java 具有和 Java 应用程序一样的基本功能,能在 Java-capable 浏览器内运行;

(2)小应用程序出现在网页(Web page)中,很像图像,但又不同于图像,小应用程序是动态的和交互的;

(3)小应用程序能用来创建动画、图形或者能响应用户输入的区域、游戏或者在同一网页上,文本和图形之间的其它交互作用。

作为一个 Java 开发者,只需创建一次 applet,Java-capable 浏览器执行特定于它们自己的平台的解释。

3. HotJava(Sun 公司的浏览器)是第一个万维网浏览器,它能执行 Java 小应用程序,其它浏览器如 Java Microsoft Internet Explorer 3.0 和 Netscape(导航者)2.0X 也支持 Java applets。

4. 为创建一个小应用程序,必须用 Java 语言编写程序,用 Java 编译器进行编译,并在 HTML Web 页上引用该小应用程序,用制作普通 HTML 和图像文件一样的方式,把结果的 HTML 和 Java 文件放到网址上。因此,当使用一个 Java-capable 浏览器来观察具有嵌入 applet 的页时,该浏览器下载小应用程序到本地系统并执行它,然后,用户可以观察并与小应用程序交互(用户使用非 Java 兼容(non-Java-enabled)浏览器,可以看见文本、静态图形或者什么也没有)。

5. Java 除了用于创建小应用程序外,还能做许多事情。Java 作为一种通用的面向对象编程语言,可以与其它语言(如 C 或 C++)一样,完成同类的任务和解决同类的问题。

1.1.2 Visual J++ 的含义

Visual J++ (简称 VJ++) 是用于开发、编译、测试和调试小应用程序的微软的集成开发环境(Integrated Development Environment, 简写 IDE),它是基于 Microsoft Developer Studio 的,如图 1-1 所示。其具有易使用的健壮性,普通库的重用性,Java 代码的维护性,还具有微软新技术、部件对象模型(Component Object Model, 简称 COM)与 Java 小应用程序和应用程序内部组合能力,详情将在 1.2 节中介绍。

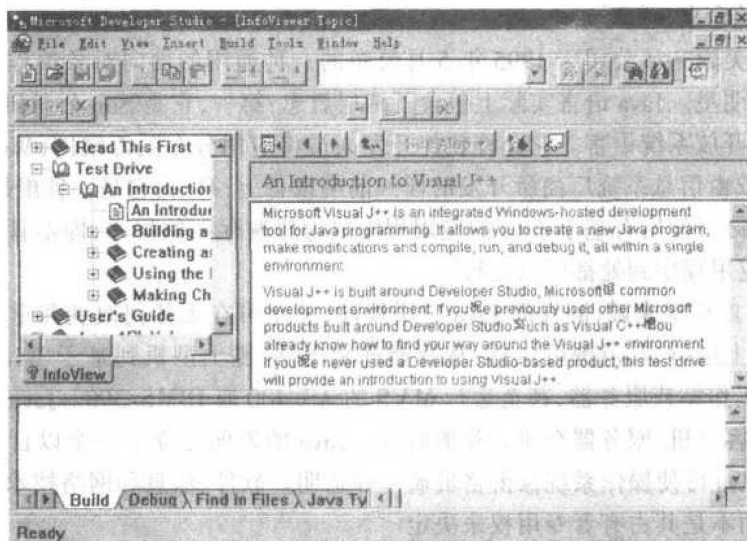


图 1.1 Microsoft Developer studio

1.1.3 在 HTML 页上嵌入 Java 小应用程序

1. 把小应用程序栖身于 HTML 文件

前面,我们介绍 Java、Java 小应用程序、Java 应用程序、开发工具以及 Web 浏览器,这里我们说明如何在 HTML 页上放入小应用程序。

(1) 小应用程序的写法

在 Web 页中包含小应用程序,就是指在 Web 页的 HTML 代码中的小应用程序,为此,要将小应用程序放入一个基本的范围,如例 1-1 所示,这是一个很简单的 HTML 文件。

例 1-1 小应用程序的 HTML 文件

```
<HTML>
<HEAD>
```

```

<TITLE>Hello Java</TITLE>
</HEAD><BODY>
<p>My Java apple says:</p>
<APPLET CODE> "HelloworldApplet.class" WIDTH = 150 HEIGHT = 25>
<APPLET>
</BODY>
</HTML>

```

其中,以 APPLET 开头的行引起启动 Java - capable Web 浏览器装入和使用“HelloworldApplet.class”,所有被编译的 Java 小应用程序.class 文件扩展。

(2) 注意事项

在 HTML 文件中,小应用程序具有<APPLET>标志,这里应当注意几点:

- 如前所述,启动 Java - enabled Web 浏览器,才能显示小应用程序,其它的浏览器将显示各种的结果,而不是显示小应用程序,某些浏览器将什麼也不显示。
- 使用 CODE 属性,表示包含用户的小应用程序类的 Java - enabled Web 浏览器名。
- 使用 WIDTH 和 HEIGHT 属性,表示小应用程序的浏览器尺寸(以像素为单位),浏览器使用这些值。

2. 小应用程序的观察

(1)一旦小应用程序已作为引用包含在 HTML 页上,则需要一个 Java 兼容的网页浏览器来观察它。如果没有一个合适的浏览器,小应用程序观察器应用(Appletviewer Application)就工作,它是 Java 开发工具的一部分。虽然,小应用程序观察器不是 Web 网页浏览器,并且不允许用户来观察整个 Web 页,但可测试小应用程序。

(2)如果使用 HotJava 来观察小应用程序,须保证它不是 alpha 版本,用 VJ++ 开发的小应用程序,不能用 alpha 观察。

(3)要用 Microsoft Internet Explorer 3.0 来观察一个小应用程序文件(见图 1.2)。

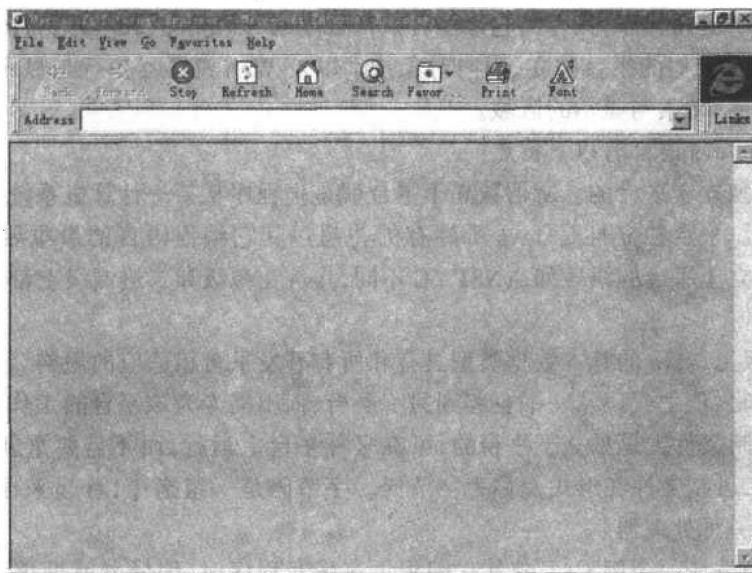


图 1.2 Microsoft Internet Explorer 3.0