

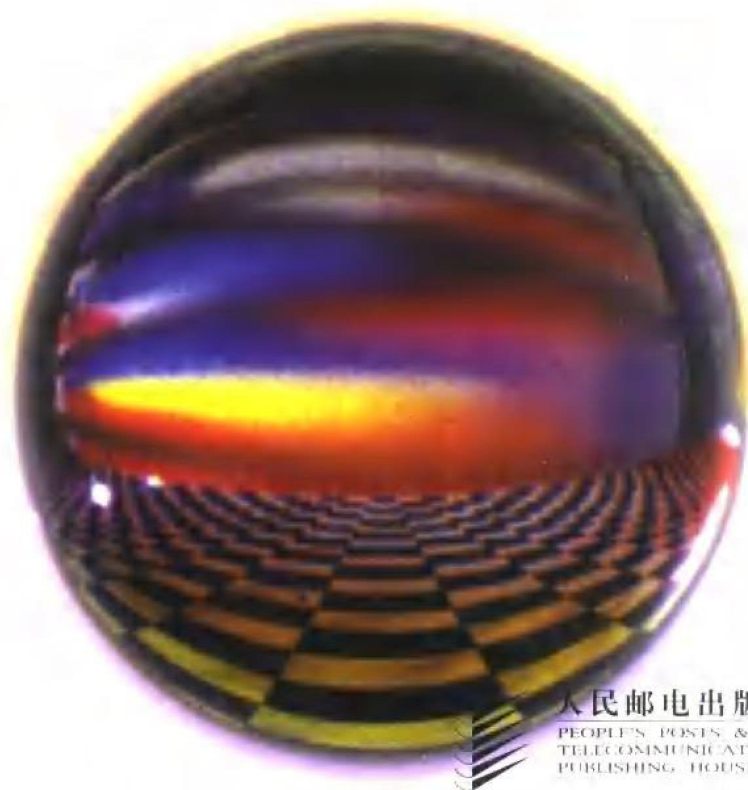
计算机编程指南丛书

Visual J++ 1.1

编程指南

宜晨主编

晶辰工作室 李青元 张福浩 朱孝明 等 编著



人民邮电出版社
PEOPLE'S POSTS &
TELECOMMUNICATIONS
PUBLISHING HOUSE

内 容 简 介

Visual J++是目前非常优秀、十分流行的软件开发工具之一,本书所介绍的Visual J++ 1.1是其最新版本。全书共分为14章,内容包括:Visual J++的有关基本知识,面向对象的编程基础,Java类库的构成与基本类库,Applet编程,AWT的使用,资源编辑器,网络编程,Visual J++中的调试方法,ActiveX及Visual J++在电子地图中的应用等。它将Visual J++ 1.1在开发、编译、调试Java应用程序及集成COM(部件对象模型)等方面的优秀特性淋漓尽致地展现在读者面前。本书对于Visual J++ 1.1各种功能的介绍力求通俗实用,书中还提供了大量具体实例以及相应的屏幕图以帮助读者理解。因此,当你边读边操作而通览完本书时,将会能像高手那样娴熟地运用Visual J++ 1.1来编制应用程序。

本书结构合理,文笔流畅,图文并茂,十分适合于个人自学或者进行培训之用,它也可作为大专院校计算机课程的参考书。

计算机编程指南丛书

Visual J++ 1.1 编程指南

◆ 主 编 宜 晨

编 著 晶辰工作室 李青元 张福浩 朱孝明 等
责任编辑 刘君胜

◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街14号
北京密云春雷印刷厂印刷
新华书店总店北京发行所经销

◆ 开本:787×1092 1/16

印张:23.75

字数:590千字

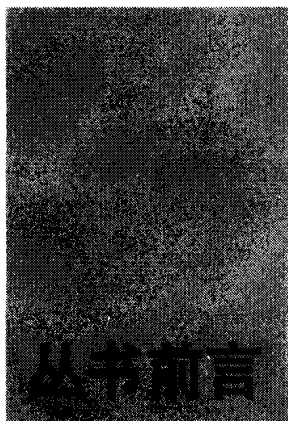
1998年9月第1版

印数:1-5 000册

1998年9月北京第1次印刷

ISBN 7-115-07233-7/TP·743

定价:35.00元



近十多年来，计算机技术发生了难以置信的巨大变化。无论是在功能、速度、可靠性及易用性方面，与十多年前相比都不可同日而语。今天，计算机和通信技术相结合构成了四通八达的计算机网络，其触角深入到了世界上的每一个角落，深入到了普通百姓家中；因特网已经变得十分火爆，人们坐在家中就可以通过因特网给远方的亲朋好友发电子邮件，还可以坐在家中通过因特网或局域网办公或进行商业贸易，甚至在网上购物。尤其近年来，计算机硬件和软件都不断更新换代，新技术、新产品层出不穷，即使是业界人士也不敢丝毫怠慢，必须随时留心世界上所发生的每一变化。

编程语言作为计算机时代的有机组成部分，同样发生了巨大的变化。从Basic到今天的Visual Basic，从dBase到Visual FoxPro，还有与C语言渊源深厚、目前如日中天的Java(Visual J++)，它们都变得更加强大并且更加容易使用。在如此巨大的变化之下，相当多的业界人士和计算机爱好者对优秀的编程语言书籍都更加渴望和期盼。这是因为不管系统软件多么强大，应用软件多么丰富，都不可能完全适合各行各业具体而细致的需求，更何况需求的变化总是领先于应用软件的出现。应用程序开发者的使命就是选择和运用最合适自己的程序设计语言，不断地编写出新的软件来满足人们对计算机的要求。

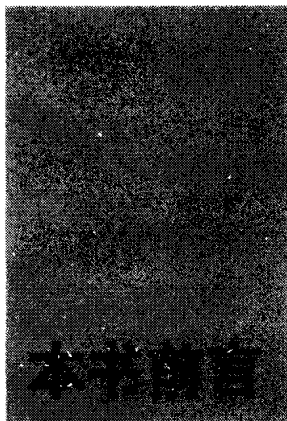
然而，编程语言的书籍却越来越厚，致使读者要花费很多时间才能掌握其所介绍的内容。我们编写这套书的出发点，是要让读者花最少的时间掌握最多最新的内容。我们把每一本书的篇幅都控制在300页至500页之间，这样，既不会单薄到仅仅能够探讨一些入门概念，也不会庞大到需要“十年磨一剑”的毅力去钻研。我们将通过一些具体实例，简明扼要、深入浅出地介绍相应语言中最基本、

最实用、最新颖、最关键的特征，使读者只需花费一、两周时间，就能真正掌握一种编程工具，解决工作中出现的实际问题。

我们这套书所涉及的编程语言也是经过精心选择的，它们都是最新的版本，都具有前所未有的强大功能及鲜明的特色。Visual Basic是最容易学习的语言；PowerBuilder是构造Client/Server数据库应用的优秀语言；Visual FoxPro在我国是市场基础非常雄厚的桌面数据库语言；而Visual J++，则是如日中天的Java语言的Microsoft版本。作为“一次学习到处编程”的技术，伴随着Internet/Intranet的发展，其前途更是不可限量。

总而言之，如果您想要学习现代编程技术，希望找到能显著改善程序设计效率的工具，那么，我们的这套书将不会使您失望，每一位读者都可以从中迅速而有效地获取必要的信息和技术。

晶辰工作室
1998年4月



本书的内容主要包括Visual J++的有关基本知识、面向对象的编程基础、Java类库的构成与基本类库、Applet编程、AWT的使用、资源编辑器、网络编程、Visual J++中的程序调试方法、ActiveX及Visual J++在电子地图中的应用等，将Visual J++ 1.1在开发、编译、调试Java应用程序及集成COM(部件对象模型)等方面的优秀特性详细介绍给读者。笔者在编著过程中力图让读者一书在手，便可对Visual J++ 1.1 运用自如。书中结合大量的实例讲解Visual J++ 1.1的设计、编程、调试方法，帮助读者深刻理解Visual J++ 1.1版的基本概念和技术，为读者以后自己使用Visual J++ 1.1版编程打下良好的基础。

本书从不同的深度和广度介绍了Visual J++ 1.1版所具有的新功能、新特性。使读者能掌握最新开发技术，紧密跟踪技术发展的时代潮流。在内容编排方式上，注意承前启后、循序渐进、系统全面而又轻重错落有致，有所侧重而不平铺直叙。

本书是为那些想要尽快掌握Visual J++ 1.1技术和开发自己的应用程序的读者而编写的，不同的用户可以结合自己的需要阅读。为此，笔者有如下一些建议：

① 入门用户：也许您是一位Windows应用程序的开发者或即将成为Windows应用程序的开发者，正在寻找一种得心应手的开发平台，掌握最新技术，并且希望持续得到技术升级和支持，以便最大限度地提高工作效率。您可以从头到尾仔细阅读本书，并且运行每一章所给出的例子，它可以将您带入Visual J++ 1.1版技术的世界。

② 初级用户：也许您使用过Visual J++ 1.0版，但对于Visual J++ 1.1版的新功能、新特性尚不甚了解，本书前8章的一部分和后6章的内容可以帮助您跟上Visual J++ 1.1版的脚步。

③ 高级用户：也许您正在急于寻找网络编程方法、Visual J++中的程序调试方法，或者正在制作电子地图，那么您在后6章可以找到所需要的技术内容。此外，前8章中对基本技术有系统详细的总结，因而也方便您随时查阅。

阅读本书以后，读者将对Visual J++ 1.1版应用程序设计以及如何用它建立自己的应用程序有一个比较全面的理解。这些知识或许将对读者以后的工作和娱乐有所帮助。

为了方便读者对本书的阅读，现将书中的一些格式范例列表说明如下。

约定示例	含 义
[expressionlist]	在语法说明中，方括号 ([]) 内部的项是可选的。
{While Until}	在语法说明中，花括号 ({}) 和竖线 () 表示可在两个或者多个选项中选择1个。除非所有的选项又包含在方括号 ([]) 中，否则必须选择1个选项。
Alt+F1	键名之间的加号 (+) 表示键组合。例如，Alt+F1 表示在按 Alt 同时按 F1 键。
Down Arrow	按照键上的箭头方向使用单个方向键。(Left、Right、Up 或 Down)。“箭头键”是这些键的统称。
Backspace, Home	按照各自的名字使用其它定位键。

注解约定包括技巧、注意、说明、提示等，它们的格式如下：

技巧：
✎ 特别介绍一些程序开发中的小技巧 and 扩展知识。

注意：
⚠ 特别提示读者一些注意事项。

说明：
📖 专业技术名词术语解释或说明。

提示：
💡 提示如何得到更详细的信息。

本书是集体努力的结晶：图书框架和整体结构由晶辰工作室设计；第一、二、三章由李青元、张福浩、朱孝明、宜晨编写；第四章由张长富、宫士友、牛文华编写；第五章由王永清、单国栋、蔡润清编写；第六章由郑莉萍、席承兵、许辉编写；第七章由伏圣信、朱元秋、肖辉编写；第八章由李晓惠、姜兵、石海峰编写；第九章由侯英、汪国红、孙蔚敏编写；第十章由朱学英、尹京堂、刘为国编写；第十一章由尹九阳、李力凡、姚兴国编写；第十二章由冯戈力、王学军、刘为东编写；第十三章由郝岗、张乐兵编写；第十四章由闵永明、张福浩、李青元编写；最后由宜晨、李青元负责全书统稿和审校。此外，黄文刚、童晓民、朱孝明、李匀等人参与了本书的资料收集整理、书稿审校；赵婷婷、章之惠、刘敏芳等人参与了本书的录入编排。

晶辰工作室
1998年4月



第一章	Java 与 Visual J++	1
1.1	Java 概述	1
1.1.1	Java 的起源	2
1.1.2	Java 的现状	3
1.1.3	Java 的未来	3
1.2	Java 的语言特点	4
1.2.1	出自于 C++	4
1.2.2	简单性	4
1.2.3	面向对象	5
1.2.4	分布式	5
1.2.5	支持多线程	5
1.2.6	解释型	6
1.2.7	健壮性	6
1.2.8	安全性	6
1.2.9	平台无关性	7
1.2.10	统一的类库体系	7
1.3	关于 Visual J++	8
1.3.1	Visual J++ 在 Java 家族中的地位	8
1.3.2	集成化的开发环境	8
1.3.3	智能化的开发向导	9
1.3.4	断点设置、跟踪与单步执行功能	10
1.3.5	方便实用的资源编辑器	10
1.3.6	无所不在的联机帮助	11
1.3.7	支持 DCOM 和 Active X	12
1.3.8	关于 100% 纯 Java 问题	12
1.4	应经常访问的网址	13
1.5	从这里开始	14

1.5.1 安装 Visual J++	14
1.5.2 用 Visual J++ 开始一个工程	14
1.5.3 工作区	17
1.5.4 工作区文件	18
1.5.5 工作区的组织	18
1.5.6 不建工程直接调试 Java 程序	21
1.6 结束语	24
第二章 数据类型、变量、表达式	25
2.1 Java 的符号集	25
2.2 标识符与关键字	26
2.3 常量	27
2.3.1 数值常量	27
2.3.2 字符常量	27
2.3.3 布尔型常量	28
2.4 运算符和分隔符	28
2.5 基本数据类型与变量的声明	29
2.5.1 整型	29
2.5.2 声明整型类型实例	30
2.6 浮点型及其声明	30
2.7 字符型及其声明	31
2.8 声明数组	32
2.9 运算符及其优先级	32
2.10 布尔运算符	35
2.11 位运算符	36
2.12 数组运算符	40
2.13 运算符的优先级	41
2.14 类型转换	41
2.15 结束语	42
第三章 流程控制	43
3.1 概述	43
3.2 if 条件语句	44
3.2.1 简单条件语句	44
3.2.2 复合条件语句	45
3.3 循环语句	46
3.3.1 while 语句和 do while 语句	47
3.3.2 for 循环	48
3.4 switch 语句	49
3.5 转移语句	51
3.6 返回语句	53
3.7 结束语	54
第四章 面向对象的编程基础	55
4.1 面向对象编程的基础	55

4.1.1 过程式编程与面向对象编程.....	55
4.1.2 面向对象的基本概念.....	56
4.1.3 什么是对象.....	56
4.1.4 什么是类.....	57
4.2 Java 语言中的对象和类.....	58
4.2.1 Java 对象和类的层次.....	58
4.2.2 类声明.....	58
4.2.3 实例变量.....	59
4.2.4 类的实例化.....	59
4.3 类方法.....	61
4.3.1 方法的声明.....	61
4.3.2 方法重载.....	62
4.4 类的继承.....	68
4.4.1 类的继承.....	68
4.4.2 this 关键字.....	69
4.4.3 super 关键字.....	71
4.4.4 方法覆盖.....	72
4.5 对象转换.....	73
4.6 对象的可访问性.....	74
4.6.1 变量和方法的可访问性.....	74
4.6.2 类修饰符.....	74
4.7 接口.....	75
4.8 包.....	77
4.8.1 包声明.....	77
4.8.2 加载其他包.....	78
4.9 一个综合实例.....	78
4.10 结束语.....	81
第五章 Java 类库构成与基本类库.....	83
5.1 Java 类库的整体结构.....	83
5.1.1 Java 类库.....	83
5.1.2 浏览器类库.....	85
5.2 语言基础类库.....	86
5.2.1 对象类.....	87
5.2.2 包含数据类型的类.....	87
5.2.3 数学类.....	90
5.2.4 字符串类.....	91
5.2.5 系统类.....	95
5.3 输入输出类库及使用.....	97
5.3.1 输入输出类库概述.....	97
5.3.2 输入流类和输出流类.....	98
5.3.3 文件访问.....	100
5.3.4 字节数组的输入输出.....	105
5.3.5 随机存取文件类.....	107

5.3.6 流令牌类	108
5.4 工具类库及应用	110
5.4.1 Utility 类库的结构	110
5.4.2 日期和时间类	111
5.5.3 向量类及应用	113
5.5.4 哈希类表及使用	118
5.5.5 位集类	121
5.5.6 字符串令牌类	122
5.6 结束语	124
第六章 Applet 编程	125
6.1 Applet 类库	125
6.2 Applet 类的继承层次	126
6.3 Applet 类的构造方法、变量和方法	126
6.4 Applet 的结构	127
6.5 创建第一个 applet	129
6.6 HTML 与 Applet	136
6.6.1 HTML	136
6.6.2 J++ 与 HTML 标记	138
6.7 Applet 的图形处理	140
6.7.1 Paint() 方法与 Graphics 类	140
6.7.2 字体与颜色	141
6.8 一个 applet 实例	143
6.9 AppletWizard	145
6.9.1 用 AppletWizard 创建一个新工程	145
6.10 Applet 的消息传递	157
6.10.1 Applet 鼠标消息的传递	157
6.10.2 Applet 的参数传递	159
6.11 Java Appletcation	166
6.12 结束语	171
第七章 AWT 的使用	173
7.1 AWT 类库的层次	173
7.2 AWT 组件	174
7.2.1 按钮	174
7.2.2 画布	175
7.2.3 复选框	175
7.2.4 复选框组	176
7.2.5 单选框	176
7.2.6 标签	177
7.2.7 列表	178
7.2.8 滚动条	179
7.2.9 文本组件	180
7.2.10 单行文本编辑域	181
7.2.11 多行文本编辑域	182

7.3 AWT 容器	182
7.3.1 框架	183
7.3.2 面板	185
7.3.3 对话框	185
7.3.4 文件对话框	186
7.3.5 菜单	186
7.4 布局管理器	188
7.4.1 边界布局管理器类	188
7.4.2 卡片布局管理器类	189
7.4.3 流式布局管理器类	190
7.4.4 网格布局管理器类	191
7.4.5 无布局管理器	191
7.5 AWT 应用实例	192
7.6 结束语	194
第八章 资源编辑器	195
8.1 向系统加入对话框	195
8.1.1 对话框资源编辑器	195
8.1.2 Java 资源向导	198
8.1.3 创建对话框的实例	199
8.2 给应用添加菜单	206
8.2.1 创建资源	206
8.2.2 将资源转换为 Java 代码	209
8.2.3 把文件添加到工程	210
8.2.4 修改应用程序源代码	210
8.2.5 编译执行代码	211
8.3 用资源编辑器开发 Applet 界面——对话框	212
8.4 向 applet 中添加菜单	216
8.5 结束语	219
第九章 多线程	221
9.1 有关多线程的几个概念	221
9.1.1 线程与进程	221
9.1.2 线程的执行与并发性	221
9.1.3 优先级	222
9.1.4 调度	222
9.2 不用编写任何代码而创建一个多线程	222
9.3 Thread 类	223
9.3.1 Thread 类构造方法	224
9.3.2 Thread 类的方法	224
9.4 线程的生命周期	225
9.4.1 新生状态	226
9.4.2 运行状态	226
9.4.3 堵塞状态	226
9.4.4 死亡状态	227

9.5 实现多线程	227
9.5.1 创建 Thread 类的子类的方法实现多线程	227
9.5.2 实现 runnable 接口	231
9.6 设置线程优先级实例	234
9.7 了解多线程的几个新概念	236
9.7.1 线程组	236
9.7.2 线程同步	237
9.7.3 守护线程	237
9.8 结束语	237
第十章 JDBC	239
10.1 JDBC 的工作机制	239
10.2 JDBC 的优点及缺点	241
10.3 Java Database Wizard 的应用	241
10.4 结束语	257
第十一章 网络编程	259
11.1 通信协议	259
11.2 通信在 Java 编程中的应用	261
11.2.1 与浏览器通信	261
11.2.2 用 Internet 协议通信	262
11.2.3 Applet 之间的通信	262
11.2.4 与服务器的通信	264
11.3 网络通信实例——myNet 工程	265
11.4 java.net 类库中的常用类	270
11.5 结束语	271
第十二章 Visual J++ 中的调试方法与异常	273
12.1 Visual J++ 中的调试方法	273
12.1.1 为调试做准备	274
12.1.2 断点	275
12.1.3 观察变量和表达式信息	276
12.1.4 Watch 窗口	277
12.1.5 Variables 窗口	277
12.1.6 Call Stack 窗口	277
12.1.7 Disassembly 窗口	278
12.1.8 Exceptions 窗口	278
12.1.9 Thread 窗口	279
12.1.10 关闭调试对话	279
12.2 异常	279
12.2.1 消除软件错误	280
12.2.2 错误处理和异常	280
12.2.3 引发异常	280
12.2.4 声明异常	281
12.2.5 声明与截获	281

12.2.6 使用 try 语句	281
12.2.7 截获异常	282
12.2.8 嵌套的异常处理	285
12.2.9 NestedExceptionTest 分析	289
12.3 结束语	290
第十三章 ActiveX	291
13.1 什么是 ActiveX?	292
13.1.1 ActiveX Internet 模型	292
13.1.2 ActiveX 控件	292
13.1.3 ActiveX 原稿	293
13.1.4 HTML 中的 ActiveX 对象	293
13.2 ActiveX 和 OLE	294
13.3 ActiveX 与 Java	294
13.4 ActiveX 和 COM	295
13.4.1 为什么要有 COM	295
13.4.2 为什么要使用 ActiveX COM 对象	296
13.5 ActiveX COM 对象及 Java	297
13.5.1 在 Java applet 中使用 ActiveX COM 对象	297
13.5.2 用 Java 创建 ActiveX COM 对象	297
13.6 结束语	298
第十四章 Visual J++ 在电子地图中的应用	299
14.1 浏览器启动类	300
14.2 窗口布局类	304
14.2.1 用对话框进行布局	305
14.2.2 将 Java 资源文件转化为 Java 源程序	306
14.2.3 修改由 Java Applet Wizard 生成的 Java 源程序	307
14.3 菜单类	314
14.3.1 使用资源编辑器编辑菜单资源文件	314
14.3.2 使用 Java Resource Wizard 生成 Java 源文件	316
14.4 地图窗口类	320
14.4.1 地图窗口类的任务及工作原理	320
14.4.2 地图窗口类的变量	321
14.4.3 地图窗口类操作方法的功能定义	321
14.4.4 地图窗口类的方法实现	322
14.5 地图模型大类	331
14.5.1 地图头类	332
14.5.2 多边形类	332
14.5.3 多边形弧段索引类	333
14.5.4 线状要素类	333
14.5.5 弧段类	334
14.5.6 弧段点类	334
14.5.7 注记类	334
14.5.8 地图模型类概述	335

14.5.9 地图模型类的读地图头方法.....	336
14.5.10 地图模型类的读多边形方法.....	338
14.5.11 地图模型类的多边形绘图方法.....	341
14.5.12 DOS/Windows 环境二进制文件读取中数据高低位反向问题及处理	344
14.5.13 实例程序执行效果.....	345
14.6 结束语	348
附 录.....	349
附录 1 实例用到的试验数据文件	349
附录 2 DialogLayout 类的源程序	351
附录 3 MapModel 类中未列出方法的源程序清单	355
附录 4 EagleEyeWin 类的源程序清单	361
附录 5 TableWin 类的源程序清单	362
附录 6 TableMenu 类的源程序清单.....	364



Java与Visual J++

有人说，“以前，作为计算机软件人员，不能不会C++，今后不能不会Java”。这话可能有点极端，但它确实道出了Java语言在软件界中越来越重要的地位，Microsoft公司开发的Visual J++是Java这个大家族中的佼佼者之一。本章简要地介绍Java与Visual J++的来历与主要特点。谈到Visual J++必然要谈到Java，顺便将提及Java怎样由Oak在Internet的春风化雨下演变为Java的，并着重比较Java与C++的异同。然后介绍Visual J++在Java这个大家族中的地位及特点，还讨论了100%纯Java问题，介绍了有关网点的网址。最后介绍怎样使用Visual J++开始编制一个简单的Java程序，以及Visual ++的工程、工作区等重要概念。

1.1 Java概述



什么是Java？Java原本是印度尼西亚一个盛产咖啡的岛屿，即我国古典小说中经常提到的爪哇国，但在现在的计算机界，Java则是指一种计算机编程语言。概括地说，Java是一个简单的、面向对象的、分布的、多线程的、与平台无关的、健壮的、安全的解释型编程语言。Java语言是随着Internet的WWW出现而走红的，但一开始它并不是为WWW设计的，它几经磨难，走过了艰难的成长历程，终于长成现在的参天大树，甚至威胁到C++在编程领域的统治地位。Java的成长经历颇具传奇色，成为业界的一段佳话。

1.1.1 Java的起源

1991年Sun公司成立了一个新的计算机语言开发小组，称之为Green，其目的是开发一种面向家用电器市场的软件产品。该小组最开始只有三个人，领导人James Gosling，成员Naughton和Sheridan，他们都是非常杰出的程序员。在规划这个产品时，他们首先想到这个产品必须具有平台无关性，因而很自然地把扩展C++编译器作为最初的考虑，但很快地他们便意识到这个产品还必须具有高度的简洁性和安全性，而C++在这方面显得无法胜任，于是Gosling决定自行开发一种语言，在为这个语言的名字凝思苦想时，一抬头正好看见窗外的老橡树，“橡树”在英语里叫做“Oak”，这名字好听又好记，而且企盼着自己的产品像窗外的橡树一样生机勃勃，能够在未来的市场上生根、发芽、茁壮成长。这就是Java语言的前身。Oak语言设计初步完成后，Sun公司专门成立了一个名为FirstPerson的子公司，用Oak开发家电产品，他们还确实开发出了一些技术上非常成功的家电软件，非常不幸，由于激烈的市场竞争和其它商业因素，这些在技术上非常成功的产品在市场竞标中却屡遭败绩，未能推向市场。由于毫无业绩，FirstPerson公司于1994年被解体，Oak语言也几乎陷入绝境。Sun公司的决策层急于希望Green小组的开发成果能够有所收益，这时一个关键性的人物Bill Joy，开始介入这个项目，Joy是Sun公司的创始人之一，颇具战略眼光，他注意到Internet上的WWW正喷薄欲出，但似乎还缺少点什么。

在Green小组开发Oak的同时，Internet上的WWW(Wide World Web)的研究也在独立地进行。WWW最初是由欧洲粒子物理实验室(European Particle Physics Lab——CERN)的TimBerners-Lee于1990年开始研究的，TimBerners-Lee设计的实际上是HTML语言，WWW问世之初并未引起太多的重视。1993年7月，位于伊利诺斯大学的NCSA推出了Mosaic 1.0版本。作为一种设计新颖、使用方便的WWW浏览器，Mosaic在1994年引起了世人广泛的注意，直到这时，WWW页面才真正广为人知。WWW是建立在Internet网和客户/服务器模型上，以超文本传输协议(Hyper Text Transfer Protocol——HTTP)、超文本标记语言(Hyper Text Markup Language——HTML)和统一资源定位器(Uniform Resource Location——URL)为基础，能够提供各种Internet服务，并且用户界面一致的信息浏览系统。WWW服务器利用超文本链路来链接各信息片段，这些信息片段可以放在同一主机上，也可以放在不同地理位置的不同主机上，统一资源定位器用来维持Internet上的超文本链路。WWW的出现使得Internet的表现力及魅力陡增，它的影响力已远远超出专业技术人员的应用范畴，而进入到广告、新闻、销售和服务等各个领域，使Internet迅速地走入千家万户。WWW页面虽然很丰富，可以实现声、图、文并茂，但它是静态的。需要通过一种机制来使它具有动感，嵌入一种语言是一种很自然的解决方案，但这种语言必须简练、小巧、与平台无关。Oak正好满足这些要求。

Joy及时地抓住了这个机遇，重新将Oak语言定位于WWW浏览器的应用上。Joy指定Gosling(Joy称他为“世界上最伟大的程序员之一”)继续完善Oak语言，同时指定Naughton用Oak编写一个真正的Internet应用程序，这个应用程序确定为WWW浏览器，后来称之为HotJava。两人果然不负众望，于1994年底完成了各自的任务，1995年1月，Gosling所完成的Oak版本被重新起了一个更响亮的名字——Java。

免费发布HotJava的决定在Sun公司内部并未遭到太大的反对，这是一种靠用户免费使用来占领市场份额的策略，在早期的Mosaic和后来大红大紫的Netscape都是采用的这种策略。但对于免费发布Java语言规范及其开发环境，Sun公司内部最初缺乏共识，且反对者居多，Gosling本人也不愿意“无偿”奉献自己花费如此心血开发出来的产品。还是Joy力排众议，才

促成了Java和HotJava在Internet上的免费发布。

1995年春季, Sun公司公布了Java的完整技术规范, 立即得到包括Netscape公司在内的各WWW厂商的广泛支持。不久, 许多著名的大公司, 如IBM、Novell、Oracle、SGI和Borland公司都相继购买了Java的使用许可, Java的地位得到了广泛的支持。Microsoft公司一开始对Java无动于衷, Bill Gates在反复仔细研究了Java的技术规范之后, 不得不承认“Java确实是有史以来最伟大的程序设计语言”, 继而, Microsoft也购买了Java的使用许可, 并很快推出了集成化的Java开发环境——Vivual J++, 至此Java在软件界的地位不可动摇。

1.1.2 Java的现状

Java从孕育到现在, 不过才几年的时间, 总的说来, Java还很年轻。用惯了VC++、VB、Borland C++、Delphi的程序员初用Java时会觉得很别扭, 很多在Windows中常用的API函数都没有, 速度也慢得让人大失所望。“什么破玩艺儿, 吹得这么凶, 名不符实”, 初用Java的人大都有这种感觉。这或许是我们对Java的期望值太高了, 不过客观地讲, 除去安全性、跨平台、多线程等优势外, Java与C++相比, 在功能与速度上确实还有不小的差距, 有人说, 这是因为Java是解释型, C++是编译型, 在速度上有数量级的差距这是自然的。为什么不提供编译型的Java呢? 有人说这将失去跨平台的优势, Java在速度上的低下可以从它跨平台的好处中得到补偿, 即“编写一次, 到处运行”(“Write once, Run Anywhere”)。有的用户说, 我就在Windows系列的平台上用, 我不需要跨平台, 但我也还想赶Java的时髦。也许有一天, 会有人提供针对某些流行平台的编译型Java, 因为用户需要这样的产品。据说Borland C++ 5.0就能将Java代码转变为C代码, 执行效率可提高5~10倍。

据调查, 几乎所有的软件公司现在都在学习、研究与使用Java, 由Java所开发出的大型实用系统也已不在少数, 从来没有任何一种计算机语言在这么短的时间内就得到这么广泛的响应, 取得这么大的成功。

1.1.3 Java的未来

1997年Sun公司向国际标准化组织ISO提出申请, 作为Java的国际标准提供者, 经过24个成员国的投票, 中国与美国2票反对, 2票弃权, 20票赞成而获得通过。国人都很纳闷, 怎么偏偏是中国和美国反对。美国是标准的提供者, 美国怎么会反对呢? 这事跟中国有什么关系? 中国又怎么会反对呢? 美国反对是因为美国以Microsoft、Compaq、HP等公司反对Sun作为Java的国际标准提供者, Java既然要想成为国际标准, 就应交给一个中立的学术委员会来制定, 而不能由Sun一家公司说了算, 看来这里面牵涉到巨大的商业利益。中国反对是因为至今还没有哪一个国际标准是全权交给一个公司制定的。尽管中国和美国反对, Sun公司的申请还是获得了通过, 看来大多数成员国对Sun公司为Java语言所做的工作还是予以肯定, Sun公司确实是最有资格作为Java国际标准的提交者。

Sun公司以及IBM、Oracle、Netscape等公司都在大力推进Java的应用。旨在提高Java运行速度的Java芯片也在紧锣密鼓地赶制当中, 预计在1998年下半年批量上市, 有人预计不久的将来, Java的代码的运行速度将和C++一样快。Java在WWW应用中的地位得到肯定以后, Sun又反过来扩展Java在家电领域的应用, Java的应用有向更广阔的领域扩展的趋势。Sun公司的高层人士称“Java的潜力远远超过作为编程语言带来的好处。事实上, Java是一种新计算模式的使能技术”。看来, Java的前程不可限量。