



计算机技术丛书

Visual J++ 6.0

实用编程技术

何 斌 王运坚 编著

人民邮电出版社

内 容 提 要

Microsoft Visual J++ 6.0 是 Microsoft 公司 1998 年推出的一种功能强大、使用灵活的最新 Java 语言开发工具。本书从一个程序开发人员的角度出发,主要介绍了如何使用 Visual J++ 6.0 来开发 Java 小程序。全书共分 14 章,内容包括 Java 语言概述、Java 基础知识、安装 Visual J++ 6.0、Visual J++ 6.0 集成开发环境、Visual J++ 6.0 初步、Visual J++ 6.0 用户界面设计、Visual J++ 6.0 绘图、Visual J++ 6.0 多媒体程序设计、Visual J++ 6.0 多线程程序设计、Visual J++ 6.0 动画设计、Visual J++ 6.0 异常处理、JavaScript 简介、VBScript 简介和脚本语言和小程序等等。

本书内容丰富,叙述详细,实用性强。无论读者是一个刚刚涉及 Visual J++ 6.0 编程的初学者,还是一个精通 Visual J++ 6.0 程序设计的高手,都可以在本书中找到自己需要的内容。

本书适合所有的计算机用户和从事编程工作的技术人员阅读参考。

计算机技术丛书

Visual J++ 6.0 实用编程技术

◆ 编 著 何 斌 王运坚

责任编辑 靳文娟

◆ 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:32.5

字数:813 千字

1999 年 10 月第 1 版

印数:1-5 000 册

1999 年 10 月北京第 1 次印刷

ISBN 7-115-08123-9/TP·1330

定价:48.00 元

前 言

在计算机技术飞速发展的今天，随着 Internet 的普及，人们的生活和工作将越来越离不开信息网络的支持。Java 是 Sun 公司开发的一种激动人心的新型编程语言。利用它可以开发出适合在 Internet 上运行的程序，使以前只有静态图像和文本的网页变成栩栩如生的动态网页。

自从 Sun 推出了 Java 语言以来，Java 的影响力越来越大，发展也十分迅速。目前已经出现了许多种 Java 语言的开发环境，在众多的 Java 语言开发工具中，Visual J++ 无疑成为目前非常受欢迎的一种。Microsoft Visual J++ 6.0 是 Microsoft 公司 1998 年推出的最新版本。它不但继承了 Visual J++ 早期版本的所有优点，如集成化开发环境、强大的调试能力等，还进行了一系列的改进，如提供了一个全新的 Java Windows 基本类库（Windows Foundation Classes, WFC）和完善的超文本语言（HTML）支持等等。

本书从一个程序开发人员的角度出发，对如何使用 Visual J++ 6.0 开发小程序进行了详细的介绍，并专门编写了很多示例来帮助读者。无论读者是一个刚刚涉及 Visual J++ 6.0 编程的初学者，还是一个精通 Visual J++ 6.0 程序设计的高手，都可以在本书中找到自己需要的内容。

参加本书编写工作的除了封面署名作者外，还有程凡、钱力鹏、赵海宇、许庆丰、程卫峰、吴频、张建强、姚伟、黄诚昆、刘宝华、阎义洲、钟兆鹏、杜英歌、李文超、何淑宇、罗裕辉、郑涛、洗文霞、李仪、刘志宇、桂大为、马立、王柱、鲁巍、李心亮、张雪敏、甘强、吴军等等。

由于计算机技术的飞速发展，新技术、新概念层出不穷，加上时间紧迫，因此书中的错误和纰漏在所难免，希望广大读者给予批评指正。

作者

1999 年 7 月

丛 书 前 言

世界上发达国家普遍重视发展以计算机和通信为核心的信息技术、信息产业和信息技术的应用，一些经济发达国家信息产业发展迅速。

当前，我国处于国民经济高速发展时期。与此相伴随，必将有信息技术、信息产业和信息技术应用的高速发展。各行各业将面临信息技术应用研究与发展的大课题以及信息化技术改造的大任务、大工程。

为了适应信息技术应用大众化的趋势，提高应用水平，我们组织编写、出版了这套“计算机技术丛书”。这套丛书以实用化、系列化、大众化为特点，介绍实用计算机技术。

这套丛书采取开放式选题框架，即选题面向我国不断发展着的计算机技术应用的实际需要和国际上的实用新技术，选题不断增添又保持前后有序。

这套丛书中的著作还拟配合出版软件版本，用软盘形式向读者提供著作中介绍的软件，以使读者方便地使用软件。

我们希望广大读者为这套丛书的出版多提意见和建议。

目 录

第一章 Java 语言概述	1
1.1 Java 语言的历史和发展前景	1
1.1.1 Java 语言的起源.....	1
1.1.2 Java 语言的现状.....	2
1.1.3 Java 语言的未来.....	2
1.2 Java 语言的特点	3
1.2.1 简单性.....	3
1.2.2 面向对象	3
1.2.3 分布式.....	4
1.2.4 解释型.....	4
1.2.5 健壮性.....	4
1.2.6 安全性.....	4
1.2.7 平台无关性.....	5
1.2.8 可移植性	5
1.2.9 高性能.....	5
1.2.10 多线程.....	5
1.2.11 动态性	6
第二章 Java 基础知识	7
2.1 Java 的数据类型	7
2.1.1 Java 中的关键字.....	7
2.1.2 Java 是一种强类型的语言	7
2.1.3 Java 基本数据类型	8
2.1.4 常量	10
2.1.5 变量	12
2.1.6 类型自动转换和强制转换	14
2.2 Java 的字符串和数组	15
2.2.1 字符串.....	16
2.2.2 数组	16
2.3 Java 的表达式和运算符	18
2.3.1 算术运算符.....	19
2.3.2 位运算符	19
2.3.3 关系运算符.....	20
2.3.4 逻辑运算符.....	21
2.3.5 “?”运算符	21
2.3.6 运算符的优先级	22

2.4 Java 的流程控制	22
2.4.1 分支流程	23
2.4.2 循环流程	26
2.4.3 跳转语句	27
2.5 面向对象程序设计和类	28
2.5.1 面向对象的概念	28
2.5.2 类	30
2.5.3 继承	35
2.5.4 Java 的软件包	39
第三章 安装 Visual J++ 6.0	41
3.1 系统要求	41
3.2 安装 Visual J++ 6.0	41
3.2.1 从 Visual Studio 6.0 中安装 Visual J++ 6.0	41
3.2.2 安装 MSDN	49
3.3 Visual J++ 6.0 的新特征	54
第四章 Visual J++ 6.0 集成开发环境	56
4.1 Visual J++ 简介	56
4.2 Visual J++ 6.0 窗口界面	57
4.2.1 新建项目对话框	57
4.2.2 菜单栏	59
4.2.3 工具条	86
4.3 定制环境	84
4.3.1 “Environment”设置	95
4.3.2 “Text Editor”设置	99
4.3.3 “Debugger”设置	104
4.3.4 “Analyzer”设置	105
4.3.5 “Data Tools”设置	106
4.3.6 “Form Designers”设置	108
4.3.7 “HTML”设置	109
4.3.8 “Projects”设置	110
4.3.9 “Security”设置	112
第五章 Visual J++ 6.0 初步	113
5.1 创建第一个 Java 小程序	113
5.2 分析用模板创建的小程序	118
5.3 HTML 语言简介	123
5.3.1 HTML 文档页面设计	123
5.3.2 HTML 的文字效果	127

5.3.3 APPLET 标记	132
5.4 分析 Page1.htm 文件	133
第六章 Visual J++ 6.0 用户界面设计	135
6.1 文本框	135
6.1.1 文本框控件方法	135
6.1.2 使用文本框控件	136
6.2 文本区域	142
6.2.1 文本区域控件方法	142
6.2.2 使用文本区域控件	143
6.3 布局设计	146
6.3.1 BorderLayout 布局	146
6.3.2 CardLayout 布局	148
6.3.3 FlowLayout 布局	149
6.3.4 GridLayout 布局	149
6.3.5 GridBagLayout 布局	150
6.4 命令按钮	153
6.4.1 命令按钮控件方法	153
6.4.2 使用命令按钮控件	154
6.5 Java 事件	156
6.5.1 Visual J++ 1.1 事件	156
6.5.2 Visual J++ 6.0 动作事件	159
6.6 面板	162
6.6.1 使用面板	162
6.6.2 创建自定义面板类	166
6.7 复选框和单选按钮	174
6.7.1 复选框控件	174
6.7.2 Visual J++ 6.0 选项事件	178
6.7.3 单选按钮控件	183
6.8 滚动条	189
6.8.1 滚动条控件简介	189
6.8.2 Visual J++ 6.0 调整事件	194
6.9 下拉列表框	201
6.10 列表框	207
6.11 窗口和菜单设计	213
6.11.1 添加窗口	213
6.11.2 添加菜单	219
6.11.3 菜单事件处理	228
6.12 对话框	235
6.12.1 对话框成员方法	235

6.12.2 对话框使用示例	236
第七章 Visual J++ 6.0 绘图	246
7.1 与图形有关的几个基本概念	246
7.1.1 图形坐标系统简介	246
7.1.2 paint()、repaint()和 update()方法	246
7.2 颜色模型	247
7.2.1 Color 类中的颜色常数	247
7.2.2 Color 类中的常用方法	248
7.2.3 Color 类的构造方法	248
7.2.4 Color 类使用示例	248
7.3 字体	252
7.3.1 Font 类	252
7.3.2 FontMetrics 类	253
7.3.3 获取系统可用字体	254
7.3.4 设置字体	254
7.3.5 输出字符串	255
7.3.6 字体应用示例	255
7.4 Graphics 类中常用的绘图方法	260
7.4.1 输出字符	260
7.4.2 画线	262
7.4.3 绘制矩形	264
7.4.4 绘制椭圆	269
7.4.5 绘制椭圆弧	270
7.4.6 绘制多边形	271
7.4.7 复制图形	273
7.4.8 清除图形	275
7.4.9 异或模式	276
7.5 绘制橡皮筋形体实例	278
7.5.1 鼠标事件和鼠标移动事件	278
7.5.2 绘制橡皮筋形体实例	284
第八章 Visual J++ 6.0 多媒体程序设计	292
8.1 图像装载与显示	292
8.1.1 Image 类	292
8.1.2 URL 类	293
8.1.3 图像的装载	295
8.1.4 图像的显示	295
8.2 声音文件的播放	298

第九章 Visual J++ 6.0 多线程程序设计	302
9.1 线程的基本概念	302
9.1.1 什么是多线程	302
9.1.2 线程的有关概念	303
9.2 线程的创建	305
9.2.1 继承 Thread 类	305
9.2.2 实现 Runnable 接口	310
9.3 Thread 类简介	314
9.3.1 构造方法	314
9.3.2 成员变量	315
9.3.3 成员方法	315
9.4 线程的优先级和调度	318
9.5 线程的交互作用	321
9.5.1 问题的产生	321
9.5.2 对象的锁定	326
9.5.3 同步化方法	330
9.5.4 嵌套同步化问题	332
9.6 线程间的通信	333
9.6.1 使用管道类进行通信	333
9.6.2 使用标志变量进行通信	339
9.6.3 使用 wait() 和 notify() 方法进行通信	343
第十章 Visual J++ 6.0 动画设计	350
10.1 动画制作的基本技术	350
10.1.1 基本框架	350
10.1.2 保持恒定的帧速度	353
10.1.3 绘制每一帧图片	353
10.2 消除闪烁现象	360
10.2.1 限制重画区域	360
10.2.2 重载 update() 方法	360
10.2.3 双缓冲技术	361
10.3 Visual J++ 6.0 动画设计实例	367
第十一章 Visual J++ 6.0 异常处理	390
11.1 Visual J++ 6.0 中异常处理	390
11.2 Visual J++ 6.0 中的异常类	391
11.2.1 包 java.lang 中定义的异常	392
11.2.2 包 java.util 中定义的异常	393
11.2.3 包 java.io 中定义的异常	394
11.2.4 包 java.net 中定义的异常	394

11.3 声明异常.....	395
11.4 抛出异常.....	396
11.5 捕获异常.....	402
11.6 finally 子句.....	403
第十二章 JavaScript 简介.....	406
12.1 JavaScript 基础.....	406
12.1.1 JavaScript 基本概念.....	406
12.1.2 JavaScript 数据类型.....	408
12.1.3 JavaScript 变量.....	409
12.1.4 JavaScript 常量.....	410
12.1.5 JavaScript 数组.....	410
12.1.6 JavaScript 表达式.....	412
12.1.7 JavaScript 运算符.....	413
12.2 JavaScript 的语句.....	414
12.2.1 分支语句.....	414
12.2.2 循环语句.....	415
12.2.3 对象操作语句.....	417
12.3 JavaScript 函数.....	422
12.3.1 自定义函数.....	422
12.3.2 JavaScript 的内部函数.....	423
12.4 JavaScript 事件.....	425
12.5 JavaScript 对象.....	429
12.5.1 Navigator 对象.....	429
12.5.2 window 对象.....	429
12.5.3 URL 对象.....	430
12.5.4 frame 对象.....	431
12.5.5 document 对象.....	431
12.5.6 anchors 对象.....	432
12.5.7 form 对象.....	432
12.5.8 history 对象.....	433
12.5.9 link 对象.....	433
12.5.10 Math 对象.....	434
12.5.11 Date 对象.....	435
12.5.12 string 对象.....	436
第十三章 VBScript 简介.....	437
13.1 VBScript 基础.....	437
13.1.1 VBScript 基本概念.....	437
13.1.2 数据类型.....	437

13.1.3 变量	441
13.1.4 常量	442
13.1.5 数组	442
13.1.6 表达式	445
13.1.7 运算符	445
13.2 VBScript 语句	447
13.2.1 分支语句	447
13.2.2 循环语句	448
13.3 VBScript 过程和内部函数	451
13.3.1 过程	451
13.3.2 内部函数	453
13.4 VBScript 事件	459
13.5 VBScript 对象	463
第十四章 脚本语言和小程序	464
14.1 通过 VBScript 访问 Applet.....	464
14.2 通过 JavaScript 访问 Applet.....	479
附录 A HTML 语言快速参考	484
A.1 普通属性和事件.....	484
A.2 文档基础标记.....	485
A.3 物理文本类型标记.....	486
A.4 内容文本类型标记.....	487
A.5 文档空间标记.....	488
A.6 表格标记.....	489
A.7 列表标记.....	491
A.8 链接标记.....	491
A.9 图像标记.....	492
A.10 表单标记.....	493
A.11 框架标记.....	496
A.12 多媒体标记.....	497
附录 B Java 软件包快速参考	500
B.1 java.lang 软件包	500
B.2 java.util 软件包	502
B.3 java.io 软件包	503
B.4 java.net 软件包	504
B.5 java.awt 软件包	505
B.6 java.awt.image 软件包.....	506
B.7 java.applet 软件包	507

第一章 Java 语言概述

1.1 Java 语言的历史和发展前景

什么是 Java? Java 原来指的是印度尼西亚一个盛产咖啡的岛屿,即我国古代所指的爪哇国。但是,在当前计算机行业中,Java 指的是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、与平台无关的、安全的解释型编程语言。Java 语言是随着 Internet 的 WWW (World Wide Web, 万维网) 出现而走红的。但一开始 Java 并不是为了 WWW 而设计的,在经过几番波折后,Java 终于成为一种流行的编程语言。

1.1.1 Java 语言的起源

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,他们都是非常杰出的程序员。在规划产品的时候,他们首先考虑到这个产品必须具有平台无关性,因而十分自然地把扩展 C++ 编译器作为最初的考虑。但很快他们便意识到这个产品还必须有高度的简洁性和安全性,而 C++ 在这些方面显然是无法胜任的。于是他们又决定自行开发一种语言,取名为“Oak”。这就是 Java 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 的子公司来开发使用 Oak 语言的家电产品。一开始,他们的确开发出一些技术上非常成功的家电软件,但非常不幸的是,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 年解体, Oak 语言也几乎陷入了绝境。在这关键的时刻, Sun 公司的创始人之一 Bill Joy 开始加入该项目,他独具慧眼地发现正在蒸蒸日上的 Internet 上的 WWW 似乎还缺少些什么。

在 Sun 开发 Oak 语言的同时, Internet 上的 WWW 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 开发研究的, TimBerners-Lee 设计的实际上是超文本标记语言 (Hyper Text Markup Language, HTML),它在问世之初并未引起世人的注意。直到 1993 年 7 月,位于伊立诺斯大学的 NCSA 推出了 Mosaic1.0 版本。作为一种设计新颖、使用方便的 WWW 浏览器, Mosaic 在 1994 年引起世人广泛的注意,直到这时, WWW 页面才渐渐为人所知。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 (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。但是,当时 WWW 页面虽然很丰富,有悦耳的声音、优美的图片,图文并茂,但是它的致

命弱点是所有的 WWW 页面都是静态的,给人一种呆板的感觉。因此,急需一种机制来实现具有动感的 WWW 页面。嵌入一种语言则是一种最简单的方案,该语言必须简洁、安全,并且与平台无关。

Joy 抓住了这个机遇,将 Oak 语言重新定位在 WWW 浏览器上。在继续完善 Oak 语言的同时,还用 Oak 语言开发出一个真正的 Internet 上的 WWW 浏览器——WebRunner,即后来的 HotJava 的前身。1995 年 1 月, Oak 语言被重新以一个响亮的名称命名——Java。

1995 年春季, Sun 公司公布了 Java 完整的技术规范,立即得到了包括 Netscape 公司在内的各大 WWW 厂商的广泛支持。不久,许多著名的大公司,如 IBM、DEC、HP、Adobe、Novell、Oracle、SGI 和 Borland 等公司都购买了 Java 的使用许可,Java 语言的地位真正地得到了巩固。Microsoft 一开始对 Java 语言持观望态度,但在反复研究之后,也购买了 Java 的使用许可,并很快推出了自己的集成化开发工具——Visual J++1.0,从此 Java 语言有了非常方便的开发工具。

1.1.2 Java 语言的现状

Java 语言从孕育到现在的如日中天,只不过短短几年的时间。可以说,Java 还是一种十分年轻的程序设计语言,它还存在着许多缺点。例如,由于 Java 是解释运行的,用它开发的应用程序运行速度相对使用 C++ 开发的应用程序来说慢了不少。而且,除了安全性和与平台无关优点以外,Java 比起 C++ 根本没有什么优势。同时为了安全性,Java 语言也失去了不少灵活性。

但是,Java 的运行速度的低下可以从它与操作平台无关中得到补偿。经常提到的“Write once, Run anywhere”(一次编写,到处可用)正是它的与操作平台无关优点的体现。而且,Java 语言的安全性也是十分重要的,只有绝对安全的网页,用户才敢放心地访问。

目前,Java 语言已经风靡全球。据 Forester Research 公司对《幸福》杂志评比出的世界 1000 家大型企业进行的调查,目前在它们中间已有 62% 正在使用 Java 从事开发工作;有 42% 已将 Java 纳入自己企业一年以内的战略开发计划。这项调查表明,Java 已经为大家接受。

计算机自从 50 年代诞生以来,经历了几个发展阶段:终端/主机计算机、PC 计算机、客户机/服务器,而现在已经发展到一个新的阶段,即 Java 计算机时代。这个时代的特点就是以 Java 为代表的网络计算机。这是第一次真正地独立于平台的计算机方案,它能充分发挥出 Internet 的作用。Java 是实现“一个世界,一个网络”构想的关键,这样一个透明的、全球连接的和信息交互的网络,可将所有最新的计算机技术、电话、出版、媒体和娱乐集成一体。过去妨碍这一构想成为现实的是计算机平台的不能兼容问题,而 Java 语言却可以成为允许各类系统相互兼容和共享应用环境的桥梁。这就使各类软件可以“一次编写,到处可用”,这样,相同的软件可在不同器件上运行,无论是 PC 机、苹果机、UNIX 计算机还是机顶盒、PDA(个人数据助理)、移动电话,乃至智能元器件,无一例外。Java 的这一贡献将使全人类受益,也给人们的生活方式带来极大的改变。

1.1.3 Java 语言的未来

自从 1997 年 Sun 公司经国际标准化组织 ISO 认可作为 Java 的国际标准提供者以来,Sun 公司加大了推行 Java 应用的力度。Java 在 WWW 中的地位得到认可后,Sun 又反过来扩展

Java 在家电领域的应用。Java 有了向更广阔的领域扩展的趋势。Sun 公司高层人士声称：“Java 的潜力远远超过作为编程语言带来的好处。事实上，Java 是一种新的计算模式的使能技术。”由此看来，Java 的前途是非常光明的。

1.2 Java 语言的特点

Java 语言的白皮书是这样描述 Java 的：Java 是一个简单的、面向对象的、分布的、解释的、健壮的、安全的、独立于平台的、可移植的、高性能的、多线程的以及动态的程序设计语言。

似乎 Java 占全了所有程序设计语言的优点。但是从实事求是的角度，Sun 公司遗漏了 Java 的一个特点，就是 Java 的运行速度比较慢。这往往是解释性语言难以摆脱的阴影。下面我们就结合 Java 与 C++ 的比较，逐一讨论 Java 的上述特点。

1.2.1 简单性

Java 是一种简单的程序设计语言，首先表现在 Java 易学易用。Java 语言中概念非常少，并且这些概念都是广大程序开发人员熟知的。Java 在形式上和 C++ 极为相似，这在一定程度上使得 Java 易学易用。其次，Java 的运行系统十分小，它的基本解释器只有 40KB，加上标准库和线程支持也不过 250KB。

为了保证 Java 的简单性，发明人删除了 C 和 C++ 中许多语言功能。这些功能往往很少使用或极易导致不良的程序设计。例如，Java 就不支持 GOTO 语句，取而代之的是带有标号的 break 语句、continue 语句以及异常处理；Java 不支持头文件，从而取消了 C 的预处理器；为了保证 Java 具有更为统一、纯正的面向对象风格，Java 取消了 C 的 struct 和 union 语句，还有 C++ 的运算符重载和多继承也被排除在外。Java 为了追求语言的简单性的另一惊人之举是把 C 语言程序开发者推崇倍至的指针取消了。由于 Java 取消了 struct，数组和字符串都被当作对象，无需再保留指针机制。Java 会自动处理对象的引用（reference）和递引用（dereference）。另外，自动无用单元的收集也是 Java 的重要特色，它把程序员从复杂的内存管理中解放出来。

1.2.2 面向对象

Java 是一种面向对象（Object-Oriented）的程序设计语言，这意味着程序员着手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。以前的程序开发人员必须改变思考方式。

Java 完全具备面向对象的四大特点：封装、继承、多态和动态。Java 的封装性比 C++ 的封装性要好，Java 无全程变量，没有 main 主函数。在 Java 中绝大部分成员是对象，只有简单数字类型、字符类型、布尔类型例外。Java 提供给用户一系列的类，一个或多个类可以组成包（package）。Java 的类和 C++ 的类一样有层次结构，子类可以继承父类属性和方法。Java 类中的方法均缺省为虚函数，而且 Java 中的类是可以动态加载的。

1.2.3 分布式

Java 支持网络上的应用程序，它是一种分布式（Distributed）的程序设计语言。Java 提供了一个 Java.net 的包，通过该包中的类，可以完成各种层次上的网络连接。例如，URL 类支持 Java 应用程序通过 Internet 打开和访问远程对象。使用 Java 打开远程文件几乎和打开本地文件一样简单。又如，Java 提供了一个 Socket 类，通过该类可以获取可靠的流式网络连接。这样可以大大方便程序开发人员创建分布式的客户机/服务器（Client/Server）应用程序。

1.2.4 解释型

Java 编译产生的是字节代码（byte-code），而不是特定的机器代码，它是一种解释型（interpreted）语言。Java 程序的字节代码必须运行在一个解释器上，因此，Java 是一种解释型语言。但是正是因为 Java 的字节代码是一种与操作系统平台无关的对象文件格式，所以 Java 可以高效地在不同平台之间传输。任何一个 Java 程序都可以在任何一种平台上运行，只要该平台上装有 Java 的虚拟机（Virtual Machine）。Java 的虚拟机其实就是由一个 Java 解释器和运行系统组成的。Java 的虚拟机将 Java 的字节代码翻译成特定的机器代码并执行，它的作用相当于一个高效而又小巧的 CPU。每一种操作系统的 Java 的虚拟机是不同的，这就是 Java 平台无关性的关键所在。那些支持 Java Applet 的 Web 浏览器，例如 Netscape Navigator、Internet Explorer 等当中都嵌入了 Java 虚拟机。

1.2.5 健壮性

Java 是一种强类型语言，这一点和 Pascal 有点相似，比 C++ 语言则更强。Java 要求显式的方法、变量与类型转换声明，这就保证了编译器可以发现方法调用错误，保证程序更加可靠地运行。

Java 的内存管理模式是提高 Java 程序可靠性的一个重要保证。Java 不支持指针操作，这就杜绝了内存的非法访问。Java 的无用单元自动收集防止了内存丢失等动态内存分配所导致的种种问题。另外，Java 的解释器运行时也进行实时检查，可以发现数组和字符串访问的越界问题。

异常处理是 Java 保证程序健壮性的另一重要手段。异常是程序错误或可能导致程序错误的非正常状态。通过使用 try/catch/finally 语句，程序员可以把一组错误处理代码放在一个特定的地方，这样便于简化错误处理任务和恢复，同时又加强了程序的可读性。

1.2.6 安全性

随着计算机网络技术的发展，尤其是 Internet 的迅猛发展，计算机的脆弱性越来越突出，计算机安全和信息化安全技术也越来越受到人们的关注。

Java 的安全性是其它任何编程语言所不能比拟的。由于 Java 主要应用于网络应用程序的开发，所以对 Java 的安全性有很高的要求。Sun 对 Java 的安全性做了很大的努力，使 Java 通过自己的安全机制防止了有害程序对用户文件系统的破坏，这样使用户免除了对网络上下载的 Java 应用程序有害的后顾之忧。

Java 的内存分配模式是防止有害代码的最有力的手段。由于 Java 取消了指针操作，程

序就无法访问它不该访问的内存单元。

而且, Java 的运行系统实施字节代码验证, 从而保证了从网络上下载的代码不违反 Java 语言的限制。

1.2.7 平台无关性

Java 的最诱人之处在于它的平台无关性。其它任何编程语言编译产生的代码都不能保证在任何平台上有效地运行。Java 的字节代码是一种与平台无关的代码, 从而保证在任何平台上, 只要拥有 Java 虚拟机, Java 的程序就可以在其上很好地运行。Java 虚拟机是免费的, 目前已经拥有 Windows 95、Windows NT 以及多种 Unix 版本。

1.2.8 可移植性

所谓程序的可移植性, 就是指一个程序不加以任何修改, 就可以从一台计算机移植到任何其它计算机上, 即同一个程序可以应用于各种操作系统上。如果一个程序可以做到这一点, 可以大大减少程序多平台移植的工作量。

Java 平台无关性本身就提供了一种良好的可移植性。Java 的运行系统采用标准的 ANSI C 编写, Java 的编译器采用 Java 来编写, 以保证 Java 本身对硬件平台和操作系统无关, 实现 Java 的可移植性。

1.2.9 高性能

Java 的高性能表现在:

- Java 是一个面向对象的编程语言, 它给用户提供了面向对象的设计技术, 提高了应用系统的可重复使用性和可维护性。
- Java 语言支持多线程, 提高了系统的并行性、交互性和实时性。
- Java 字节代码执行速度虽然比不上机器代码的执行速度, 但是通过编译器翻译成本地机器代码, 其执行速度和通常的 C/C++ 编译产生的机器代码几乎一样快。

1.2.10 多线程

所谓线程, 就是 CPU 使用的基本单位, 它相当于任务内部运行的、有独立程序计数器的程序流。通常, 线程是在一个指定的环境中运行的小进程。一个任务可能有多个线程, 所有这些线程都具有相同的资源和任务共享权。多线程就是指在一个程序中可以同时执行多个线程, 支持多任务。在多线程的环境下, 操作系统同时给 CPU 分配了几个任务或线程。但是计算机的 CPU 不可能在同一个时间内做几件事情, 而是把时间分配到不同的线程, 使每个线程都能够得到一些被执行的机会, 这样每个线程都会有所进展。如果一个线程的工作无法进展, 比如线程所要求的对磁盘的读写权利尚未取得, 或者线程在等待用户的鼠标或键盘输入, 那么它就被挂起, 而将 CPU 让给其它的线程。通常, 线程之间的切换非常迅速, 使用户觉得好像所有的线程都是在同时执行似的。

Java 是一种多线程语言, 它可以同时运行多个线程来处理多个任务。Java 提供了语言内置的多线程控制, 从而大大简化了多线程应用程序的开发工作。Java 程序员可以使用 Java.lang 包提供的 Thread 类来完成多线程应用程序的设计。

1.2.11 动态性

Java 是一个动态的语言，为了实现 Java 语言的动态性，Java 的所有类都被设计成具有运行标识码，从而使得同一类中的各个范例都具有相应的运行类定义。因此，Java 在运行期间可以很容易地确认要使用哪个类，并找到相应的类库。同时，Java 还提供使用类名中所含字符串查找类定义的方法，可以根据字符串将数据类型名计算出来，并将其动态地链接到运行系统中。