



“九五”国家重点电子出版物规划项目
计算机知识普及系列

2001
编程宝典

丛书

5

自由编程世界奇葩

Java 2.0

网络·多线程

编程实例教程

北京希望电子出版社 总策划
李荣贵 佟来生 邓必孟 裴 龙 编写



本光盘内容包括：
本版书中的部分实例程序源文件



北京希望电子出版社
Beijing Hope Electronic Press
www.bhp.com.cn



“九五”国家重点电子出版物规划项目
计算机知识普及系列

2001
编程宝典

丛书

5

00107542

TP312JA

87

自由编程世界奇葩

Java 2.0

网络·多线程

编程实例教程

北京希望电子出版社 总策划
李荣贵 佟来生 邓必孟 裴 龙 编写



本光盘内容包括：
本版书中的部分实例程序源文件



北京希望电子出版社
Beijing Hope Electronic Press
www.bhp.com.cn

内 容 简 介

Java 是 Sun 公司推出的新一代面向对象程序设计语言, 特别适合于 Internet 应用程序开发。它的跨平台可移植性优势使得 Java 语言正如火如荼, 风靡计算机世界。而 IBM 公司为 Java 开发的 VisualAge for Java 3.0, 则以企业的需要为己任, 努力实现企业业务的网络化。该开发工具可视化程度较高, 编程也极为方便。

本盘配套书分为两篇, 共十三章。第一篇为“Java 语言初步”, 集中介绍了面向对象编程的基本概念和 Java 语言的语法规则。主要内容包括: Java 概述、Java 程序设计基础、Java 面向对象程序设计、数组与字符串、Java 的输入与输出流。第二篇“IBM VisualAge for Java 3.0 程序设计”, 详细讲解了 IBM VisualAge for Java 3.0 开发工具的使用以及 Applet 和 Java Application 应用程序的编程。主要内容包括: IBM VisualAge for Java IDE 介绍、在 IDE 中开发应用程序、运行和调试程序、异常处理、Java 网络编程、多线程程序设计、图形用户界面构件、图形和图像。本书还重点介绍了实用性强的网络编程、多线程程序设计和可视化编程基础——图形用户界面 (GUI) 设计技术。

本盘配套书内容详实、深入浅出、注重理论与实际的应用, 是 Java 网络开发人员重要的自学指导书, 同时也是高等院校相关专业师生教学、自学参考用书和社会相关领域推荐教材。

本光盘内容包括本版书中的部分实例程序源文件和本版电子书。

系 列 盘 书: 2001 编程宝典丛书 (5)

盘 书 名: 网络编程新冲浪 Java 2.0 网络·多线程编程实例教程

总 策 划: 北京希望电子出版社

文 本 著 者: 李荣贵 佟来生 邓必孟 裴龙 编写

C D 制 作 者: 希望多媒体开发中心

C D 测 试 者: 希望多媒体测试部

责 任 编 辑: 马红华 高菊香

出版、发行者: 北京希望电子出版社

地 址: 北京中关村大街 26 号, 100080

网址: www.bhp.com.cn

E-mail: lwm@hope.com.cn

电话: 010-62562329, 62541992, 62637101, 62637102, 62633308, 62633309

(发行和技术支持)

010-62613322-215 (门市) 010-62531267 (编辑部)

经 销: 各地新华书店、软件连锁店

排 版: 希望图书输出中心

C D 生 产 者: 北京中新联光盘有限责任公司

文 本 印 刷 者: 北京双青印刷厂

开本 / 规格: 787 毫米, 1000 毫米, 1/16 开本, 32 印张, 600 千字

版次 / 印次: 2001 年 1

印 数: 0001—50

本 版 号: ISBN 7-9

定 价: 39.00 元

说明: 凡我社光盘配套图书若有缺页、倒页、脱页、自然破损, 本社负责调换。

前 言

Java 是 Sun 公司推出的新一代面向对象程序设计语言，特别适合于 Internet 应用程序开发，它的跨平台可移植性优势使得 Java 语言正如火如荼，风靡计算机世界。

Java 的一大应用就是创建 Java 小程序 applet，将它们作 WWW（万维网）网页的一部分在 Internet 或企业网络上发布，并下载到几乎所有种类的计算机上运行，利用 applet 中的网络界面与用户进行友好交互并在后台快速计算。这一创新的应用彻底改变了传统的漂亮但仅限于静态的 Web 页面，从而真正让网页“互动”起来，同时，它也带动了当今流行的动态网页设计热潮。

IBM 公司为 Java 开发的 VisualAge For Java，则以企业的需要为己任，努力实现企业业务的网络化。其本身是基于对象的，支持增量编译与调试，并把自动版本控制下放到方法一级。该开发工具可视化程度较高，编程极为方便，用户不必编写任何代码就可以创建非常复杂的行为并将其添加到应用程序中。而且，再次令人鼓舞的是，VisualAge 2.0 以上版本也开始支持 SQL, DB2, Oracle 等大型数据库的设计开发。

在阅读本书时，如果读者曾到 Wide World Web 做过网上冲浪，了解一些 HTML 或拥有自己的网页，并学过一门语言如 Pascal, Basic, C/C++ 等，那么学用 Java 更是得心应手，因为 Java 语言规范比 C++ 更简捷更安全，它抛弃了 C 语言中的指针和内存管理，增加了“垃圾回收”机制，系统自动回收内存资源，大大减少了用户在这方面的烦琐工作。

本书强调实用，没有过多的花俏修辞，每章各节都附有大量的应用实例，因此希望读者在学用时应反复上机实践，仔细剖析典型实例，如果读者能坚持不懈地作到反复实践，相信应用水平会日臻成熟。

由于时间仓促，加上对新软件的使用经验尚且不足，因此，书中难免存在一些不妥之处，望广大读者原谅，并提出宝贵意见，以使我们再版时不断使其日臻完善。

本书由李荣贵、佟来生、邓必孟、裴龙等编写，另外，黄静、王崇东、张全、陈坤、陈易、汪方怡、黄臻、赵涛等同志参与了本书部分章节的写作、插图和录入工作。导向科技校对中心李长咏、杨志松、谢军、王小波、魏敏、毛勇、何小敏、方晴、王小苔等参与了本书的校对工作。

读者在使用本书的过程中如有其它问题、意见或建议可以访问 <http://www.dx-kj.com> 或与 dxkj@dx-kj.com, dxkj@21cn.com 与我们联系。

作者

2000 年 9 月

目 录

第一篇 Java 语言初步

第 1 章 概述	3
1.1 Java 的历史	3
1.1.1 什么是 Java	3
1.1.2 Java 带来的影响	4
1.1.3 Java 语言的应用前景	5
1.2 Java 语言的特点	5
1.2.1 Java 是简单的	5
1.2.2 Java 是面向对象的	5
1.2.3 Java 是静态类型的	6
1.2.4 Java 是编译型的	6
1.2.5 Java 是体系结构中立的	6
1.2.6 Java 是健全的	6
1.2.7 Java 是小巧的	7
1.2.8 Java 是多线程的	7
1.2.9 Java 是可收集无用的存储单元的	7
1.2.10 Java 是快速的	7
1.2.11 Java 是安全的	7
1.3 几种 Java 语言的开发工具简介	7
1.3.1 Java 如何进行布局管理	8
1.3.2 Java Workshop 2.0	8
1.3.3 JBuilder 1.0	9
1.3.4 VisualAge for Java Enterprise Version 1.0	10
1.3.5 Visual Cafe for Java Database Development Edition 2.0	11
1.4 对 Java 语言的常见误解	12
1.4.1 Java 是 HTML 的扩充	12
1.4.2 Java 是一种很容易学会的编程语言	12
1.4.3 Java 是一个简单的编程环境	12
1.4.4 Java 将成为所有平台的统一 编程语言	12
1.4.5 Java 是解释型的, 它对于特殊平台	

的重要应用太慢了	12
1.4.6 所有的 Java 程序都在 Web 页中运行 ...	12
1.4.7 Java 消除了 CGI 编程的需要	13
1.4.8 Java 将彻底改变客户/服务器计算 ...	13
1.4.9 使用 Java, 用户可以用 500 美元 的 Internet 设备来代替计算机	13
1.4.10 Java 将允许放弃基于部件的 计算模式	13
1.5 简单的 Java 程序	13
第 2 章 Java 程序设计基础	16
2.1 定义常量	16
2.1.1 常量	16
2.1.2 整型常量	16
2.1.3 实型常量	16
2.1.4 字符型常量	17
2.1.5 字符串型常量	17
2.2 定义变量	17
2.2.1 变量	17
2.2.2 整型变量	18
2.2.3 实型变量	18
2.2.4 字符型变量	19
2.2.5 布尔型变量	19
2.2.6 各类数值型数据间的混合运算	20
2.3 运算符和表达式	21
2.3.1 算术运算符	21
2.3.2 关系运算符	23
2.3.3 布尔逻辑运算符	24
2.3.4 位运算符	25
2.3.5 赋值运算符 =	28
2.3.6 条件运算符	29
2.3.7 表达式	29
2.4 Java 流程控制	29

2.4.1 分支语句	30	5.1.1 read()方法	78
2.4.2 循环语句	35	5.1.2 available()方法	78
第3章 Java 面向对象程序设计	40	5.1.3 close()方法	78
3.1 面向对象的基本概念	40	5.1.4 可标记流	78
3.1.1 对象	40	5.1.5 skip()方法	78
3.1.2 消息	41	5.2 OutputStream 类	79
3.1.3 类	42	5.2.1 write()方法	79
3.2 对象及对象的使用	42	5.2.2 flush()方法	79
3.2.1 创建对象	42	5.2.3 close()方法	79
3.2.2 引用对象	43	5.3 字节数组的输入输出 (I/O)	79
3.2.3 撤消对象	44	5.3.1 ByteArrayInputStream 类	79
3.3 类的定义及实现	45	5.3.2 ByteArrayOutputStream 类	80
3.3.1 类说明	45	5.3.3 StringBufferInputStream 类	80
3.3.2 类体	47	5.4 文件输入输出 (I/O)	80
3.4 方法的定义和实现	52	5.4.1 File 类	80
3.4.1 方法的说明	52	5.4.2 FileDescriptor 类	81
3.4.2 方法体	55	5.4.3 FileInputStream 类	81
3.4.3 构造子	57	5.4.4 FileOutputStream 类	81
3.4.4 析构子	60	5.5 过滤 I/O	84
3.5 重用技术	60	5.5.1 FilterInputStream 类	84
3.5.1 继承	60	5.5.2 FilterOutputStream 类	84
3.5.2 界面	65	5.5.3 缓冲 I/O	84
第4章 数组与字符串	68	5.5.4 数据 I/O	85
4.1 数组	68	5.5.5 管道 I/O	86
4.1.1 数组的声明及初始化	68	5.6 随机访问文件	86
4.1.2 数组的引用	69	5.6.1 创建随机访问文件	86
4.1.3 多维数组	70	5.6.2 访问信息	86
4.2 字符串	71	5.6.3 增加信息	87
4.2.1 不可变字符串	71	5.6.4 追加信息例子	87
4.2.2 可变字符串	75	5.7 StreamTokenizer 类	87
第5章 Java 的输入与输出流	77	5.8 URL 输入流	88
5.1 InputStream 类	78		

第二篇 IBM VisualAge For Java 3.0 程序设计

第6章 VisualAge for Java IDE 介绍 ...	93	6.2 建立你的第一个 Java 小程序	94
6.1 安装 IBM VisualAge for Java	93	6.2.1 创建 Java 小程序	94
6.1.1 运行安装之前	93	6.2.2 测试 Java 小程序	95
6.1.2 安装步骤	93	6.3 VisualAge for Java IDE 概述	96

6.3.1 IDE 简介.....	96	7.2.5 版本化程序元素.....	141
6.3.2 重要概念.....	97	7.2.6 创建开放修订版.....	142
6.4 浏览 VisualAge for Java IDE.....	102	7.2.7 从工作区删除程序元素.....	143
6.4.1 工作台窗口.....	102	7.2.8 复制或移动程序元素.....	143
6.4.2 浏览器.....	104	7.2.9 重命名程序元素.....	143
6.4.3 VisualAge for Java IDE 中的符号 与快捷键.....	109	7.2.10 比较两个程序元素.....	144
6.4.4 定制 VisualAge for Java IDE.....	112	7.2.11 比较程序元素的修订版.....	145
6.4.5 IDE 失效或工作区毁坏.....	116	7.2.12 合并类或接口的修订版.....	146
6.4.6 Applet 浏览器.....	116	7.3 在代码片段编辑测试窗中组织 用户思路.....	146
6.4.7 查找程序元素.....	117	7.3.1 代码片段编辑测试窗.....	146
6.4.8 在 IDE 中打印.....	119	7.3.2 实验代码片段.....	147
6.4.9 访问上下文敏感 API 帮助.....	120	7.4 管理资源库.....	150
6.4.10 使用快速启动窗口.....	120	7.4.1 备份资源库.....	150
6.4.11 生成类的 HTML 文档.....	121	7.4.2 从资源库中删除程序元素.....	150
第 7 章 在 IDE 中开发应用程序.....	124	7.4.3 恢复程序元素.....	151
7.1 创建程序和程序元素.....	124	7.4.4 压缩资源库.....	151
7.1.1 创建项目.....	124	第 8 章 运行和调试程序.....	152
7.1.2 创建包.....	125	8.1 运行程序.....	152
7.1.3 Java 小程序与 Java 应用程序.....	125	8.1.1 在 IDE 中运行 Applet.....	152
7.1.4 生成类.....	126	8.1.2 在 IDE 中运行应用程序.....	152
7.1.5 生成方法存根.....	127	8.1.3 对 Applet 作运行时更改.....	153
7.1.6 创建接口.....	127	8.2 调试程序.....	153
7.1.7 创建方法.....	127	8.2.1 在开发周期中用集成调试器调试.....	153
7.1.8 创建字段.....	127	8.2.2 手工启动集成调试器.....	154
7.1.9 生成字段存取方法.....	128	8.2.3 挂起、恢复和终止线程.....	154
7.1.10 编写与格式化源代码.....	128	8.2.4 在源代码中设置断点.....	155
7.1.11 从资源库向工作区添加项目和包.....	134	8.2.5 配置并设置断点条件.....	157
7.1.12 从资源库向工作区添加类和方法.....	135	8.2.6 在外部类中设置断点.....	160
7.1.13 更换工作区的修订版.....	136	8.2.7 选择调试器要捕获的异常.....	161
7.1.14 动态装入外部类.....	136	8.2.8 清除并且禁止断点.....	163
7.1.15 从文件系统导入文件.....	137	8.2.9 检查并修改变量值.....	163
7.1.16 从另一个资源库中导入.....	137	8.2.10 单步跟踪方法.....	164
7.1.17 在项目中包含资源文件.....	137	8.2.11 调试过程中修改代码.....	165
7.2 修改程序元素.....	140	8.2.12 在集成调试器中求值表达式.....	166
7.2.1 保存更改的代码.....	140	8.2.13 单步执行时监视一个表达式的值.....	168
7.2.2 存储工作区.....	140	8.2.14 生成类跟踪.....	168
7.2.3 编译代码.....	141	8.2.15 设置调试器选项.....	169
7.2.4 查找和修正问题.....	141	8.3 导出代码.....	169

8.3.1 导出代码	169
8.3.2 导出字节码	170
8.3.3 导出资源文件	171
8.3.4 调试导出	171
8.3.5 导出到另一个资源库	171
第 9 章 异常处理	174
9.1 异常的概念	174
9.1.1 采用新的出错处理机制	174
9.1.2 异常	175
9.2 异常处理	176
9.2.1 try/catch 块	176
9.2.2 finally 块	179
9.2.3 try/catch/finally 组合使用	181
9.3 异常类的层次	182
9.3.1 运行异常	183
9.3.2 非运行异常	183
9.4 激发异常	184
9.5 创建自己的异常	186
9.5.1 定义一个新的异常类	186
9.5.2 创建程序中的异常	187
第 10 章 Java 网络编程	194
10.1 网络基础	194
10.1.1 计算机网络的概念、特点及功能	194
10.1.2 网络协议简介	195
10.1.3 IP 地址	196
10.2 用 InetAddress 类进行 Internet 寻址	197
10.2.1 java.net 包	197
10.2.2 用 InetAddress 类进行 Internet 寻址	198
10.3 用 URL 类访问网络资源	201
10.3.1 什么是 URL	201
10.3.2 URL 类的创建与地址查询	202
10.3.3 用 URL 类访问资源	203
10.3.4 URLConnection 类	204
10.3.5 AppletContext 类	205
10.4 Java 的 Socket 通讯机制	206
10.4.1 Socket 类	206
10.4.2 有连接通讯方式的客户 / 服务器 应用工作模式	207
10.5 Java 的数据报通讯机制	212

10.5.1 数据报简介	212
10.5.2 数据报方式的通迅过程	213
10.5.3 数据报的创建	213
10.6 网络安全措施	219
10.6.1 网络安全与 SecurityManager 类	219
10.6.2 覆盖 SecurityManager 类中的 检验方法	220
10.6.3 安装新的 SecurityManager 类对象	222
第 11 章 多线程程序设计	224
11.1 线程相关概念	224
11.2 Java 线程的创建	225
11.2.1 继承 Thread 父类	225
11.2.2 实现 Runnable 接口	231
11.3 线程状态与线程控制	234
11.4 线程组	237
11.5 线程优先级和线程调度	238
11.6 线程同步和死锁	247
第 12 章 图形用户界面构件	255
12.1 抽象窗口工具集 AWT 简介	255
12.2 GUI 标准构件的使用方法	258
12.2.1 Component 类提供的方法	258
12.2.2 添加构件到容器中	260
12.2.3 改变构件的外观行为	260
12.3 事件处理机制	261
12.3.1 键盘事件	264
12.3.2 鼠标事件	265
12.3.3 动作(Action)事件	266
12.3.4 其他事件	267
12.4 创建用户界面	268
12.4.1 按钮(Button)	268
12.4.2 检查盒(Checkbox)	271
12.4.3 选择框(Choice)	274
12.4.4 标签(Label)	277
12.4.5 列表(List)	280
12.4.6 滚动条(Scrollbar)	283
12.4.7 文本输入域(TextField&TextArea)	285
12.4.8 框架(Frame)	292
12.4.9 对话框(Dialog)	292
12.5 布局控制	297

12.5.1 流控制(FlowLayout).....	297	第 13 章 图形和图像.....	322
12.5.2 边界控制(BorderLayout).....	298	13.1 应用 components 类作图.....	322
12.5.3 栅格控制(GridLayout).....	300	13.2 应用 Graphics 类.....	323
12.5.4 卡片控制(CardLayout).....	302	13.2.1 颜色.....	324
12.5.5 栅格包控制(GridBagLayout).....	307	13.2.2 作图方法.....	325
12.6 创建和使用菜单.....	312	13.2.3 设置大小.....	332
12.6.1 菜单(Menu)简介.....	312	13.3 处理图像.....	332
12.6.2 菜单的创建步骤.....	312	13.3.1 应用 MediaTracker.....	335
12.6.3 菜单的事件处理.....	313	13.3.2 创建脱屏图像板面.....	338
12.6.4 菜单实例.....	314	13.3.3 producer 和 consumer.....	341

第一篇 Java 语言初步

第 1 章 概述

第 2 章 Java 程序设计基础

第 3 章 Java 面向对象程序设计

第 4 章 数组与字符串

第 5 章 Java 的输入与输出流

原书空白页

第 1 章 概 述

主要内容

- Java 的历史
- Java 语言的特点
- 几种 Java 语言的开发工具简介
- 对 Java 语言的常见误解
- 简单的 Java 程序

Java 是 Sun 公司推出的新一代面向对象程序设计语言，特别适合于 Internet 应用程序开发。与其它高级语言相比，它的平台无关性和可移植性是其无与伦比的一大优势，因此，“连 Internet，用 Java 编程”，已成为技术人员的一种时尚。Java 作为软件开发的一种革命性的技术，其地位已被确立。当今是以网络为中心的计算机时代，不支持 HTML 和 Java，应用程序的应用范围将局限于同构的环境（相同的硬件平台）。intranet 正在成为企业信息系统最佳的解决方案，而其中 Java 将发挥不可替代的作用。intranet 的目的是把 Internet 用于企业内部的信息系统，它的优点表现在：价廉、使用方便和易于管理。用户不管使用何种类型的机器和操作平台，界面是统一的 Internet 浏览器，而数据库、Web 页面、应用程序(用 Java 编的 applet)则存储在 WWW 服务器上，无论是开发人员，还是管理人员，亦或是用户都可以受益于该解决方案。

1.1 Java 的历史

1.1.1 什么是 Java

Java 是一种由 Sun 公司开发而形成的新一代编程语言，使用它可在各式各样的不同种机器、不同种操作平台的网络环境中开发软件。不论你使用的是哪一种 WWW 浏览器、哪一种计算机或哪一种操作系统，只要 WWW 浏览器（如 IE 5.0 或 Netscape 3.0）注明了“支持 Java”，你就可以看到生动活泼的主页。Java 正在逐步成为 Internet 应用的主要开发语言，它彻底改变了应用软件的开发模式，带来了自 PC 以来的又一次技术革命，为迅速发展的信息界增添了新的活力。

Sun 公司的 Java 语言开发小组成立于 1991 年，其目的是开拓消费类电子产品市场，例如，交互式电视、烤面包箱等。Sun 公司内部人员把这个项目称为 Green，那时 World Wide Web 还在图纸上呢。该小组的领导人是 James Gosling。他是一位非常杰出的程序员，出生于 1957 年，于 1984 年加盟 Sun Microsystem 公司，之前在一家 IBM 研究机构工作。他是 Sun News 窗口系统的总设计师，也是第一个用 C 语言实现的 EMACS 的文本编辑器 COSMACS 的开发者。

1991 年，SUN Microsystem 公司的 Jame Gosling，Bill Joe 等人，为在电视、控制烤面

包箱等家用消费电子产品上进行交互式操作而开发了一个名为 Oak 的软件(即一种橡树的名字),但当时并没引起人们的注意,到了 1994 年,WWW 已如火如荼地发展起来。Gosling 意识到 WWW 需要一个中性的浏览器,它不依赖于任何硬件和软件平台,它应是一种实时性较高、安全可靠、有交互功能的浏览器。于是 Gosling 决定用 Java 开发一个新的 Web 浏览器。

这项工作由 aughton 和 Jonathan Payne 负责,到 1994 年秋天,完成了 WebRunner 的开发工作。WebRunner 是 HotJava 的前身,这个原型系统展示了 Java 可能带来的广阔市场前景。WebRunner 改名为 HotJava,并于 1995 年 5 月 23 日发表后,在产业界引起了巨大的轰动,Java 的地位也随之而得到肯定。又经过一年的试用与改进,Java 1.0 版终于在 1996 年年初正式发表。

1.1.2 Java 带来的影响

Java 虽出现的时间不长,但已被业界所接受,IBM, Apple, DEC, Adobe, Silicon Graphics, HP, Oracle, Toshiba, Netscape 和 Microsoft 等大公司已经购买了 Java 的许可证。Microsoft 还在其 Web 浏览器 Explorer 3.0 版中增加了对 Java 的支持。

另外,众多的软件开发商也开发了许多支持 Java 的软件产品。如: Borland 公司的基于 Java 的快速应用程序开发环境 Latte; Metrowerks 公司和 Natural Intelligence 公司分别开发的 Macintosh 的 Java 开发工具; Sun 公司的 Java 开发环境 Java Workshop; Microsoft 也开发出系列 Java 产品。数据库厂商如 Illustra, Sybase, Versant, Oracle 都在开发支持 HTML 和 Java 的 CGI(Common Gateway Interface)。在以网络为中心的计算机时代,不支持 HTML 和 Java,就意味着应用程序的应用范围只能局限于同质的环境。

intranet 正在成为企业信息系统最佳的解决方案,它的优点表现在:便宜、易于使用和管理。用户不管使用何种类型的机器和操作系统,界面是统一的 Web 浏览器,而数据库、Web 页面、应用程序(用 Java 编的 applet)则存储在 WWW 服务器上。开发人员只需维护一个软件版本,管理人员省去了为用户安装、升级、培训之苦,用户则只需一个操作系统,一个 Internet 浏览器足矣。

我们可以设想未来的计算方式,每个 HomePage 的实质是一个多媒体应用程序,这些程序用 Java 来开发。Java 应用程序运行在异质的机器、异质的操作系统之上,甚至于电冰箱、烤面包箱、防盗电子设备之中,用 Internet 把所有的电子设备连接起来,通过 TCP/IP 进行信息的交流。Java 应用程序之间既可以交换消息,也可以交换程序(一个 Java 的小应用程序 applet)。或许有一天,我们可以在 Netscape 浏览器里查看电冰箱的温度,向烤面包箱发一个电子邮件。

Java 的出现是计算机信息交换的一个重要里程碑。在单机时代,程序进程之间靠共享存储进行量交换;在网络时代,运行在不同宿主机上的程序按网络协议进行无格式的消息(二进制字符流)交换,消息的语义由交换程序双方维护;在 Java 时代,运行在网络上的程序进程交换的是小应用程序(applet)。小应用程序是什么?它是一个对象,由一组属性和方法构成,是一个可执行的实体;不仅有数据的状态,而且有定义在数据上的操作。未来可能进行代理(Agent)交换,代理有一定的智能性,那便是信息交换的更高级阶段。

随着 Internet 的崛起,Java 的诞生,巨型、臃肿的应用软件开始向小型化发展,由

众多“生活”在 Internet 上的小应用程序 (applet) 相互协作完成信息的处理与传递。Java 会加速应用软件的小型化、网络化的趋势。随着 Internet 的发展, 软件必然是面向“大众”, 而不是“少数人”的奢侈品, 薄利多销将是网络应用软件的重要特点。分布式对象技术保证了用多少, 下载多少。

Java 连同 Internet, WWW 正在改变应用软件的开发和使用方式, 一切都要围绕着网络, 围绕着与平台无关。很多人认为, Word, Excel 等传统的处理工具都必然走向萎缩, 因为它们是单机时代的产物。信息的价值在于使用和共享, Internet 和 Web 是信息的使用和共享最快捷、最便宜的方式, Word 将演化成为 Web 上的写作工具, Excel 则将演化成为 Web 上的电子表格工具。

1.1.3 Java 语言的应用前景

Java 语言有着广泛的应用前景, 从以下几个方面来考虑其应用范围:

- 所有面向对象的应用开发, 包括面向对象的事件描述、处理、综合等。
- 计算过程的可视化、可操作化软件的开发。
- 动态画面的设计, 包括图形图像的调用。
- 交互操作的设计(选择交互、定向交互、控制流程等)。
- Internet 的系统管理功能模块的设计, 包括 Web 页面的动态设计、管理和交互操作设计。
- intranet (企业内部网) 上的软件开发(直接面向企业内部用户的软件)。
- 与各类数据库连接查询的 SQL 语句实现。
- 其它应用类型的程序。

1.2 Java 语言的特点

1.2.1 Java 是简单的

Java 与 C++ 极为相似, 但却简单得多。高级编程语言的所有特性中, 不是绝对需要的都已删去了。例如, Java 中没有运算符重载、标题文件、预处理、指针运算、结构、联合、多维数组、模板及隐式类型变换等。如果你知道一点 C、C++ 或 Pascal, 你很快就会驾驭 Java。下面是一个简单的 Java Hello World 程序举例:

```
public class HelloInternet {  
    public static void main(String argv[]) {  
        System.out.println("Hello Internet!");  
    }  
}
```

1.2.2 Java 是面向对象的

Java 是一种面向对象的编程语言, 除了简单的类型, 如数字和布尔算子之外, Java 中的大部分都是对象。

正如任何面向对象的语言一样, Java 代码也按类组织。每个类定义一组规定对象行为的方法。一个类可以继承另一个类的行为。在类的根层次上, 通常是类对象。

Java 支持单继承类层次结构。这就是说, 每个类一次只能继承一个别的类。有些语言允许多继承性, 但这可能造成混乱, 使语言不必要复杂化。例如, 难以想像, 一个对象会继承两个完全不同类的行为。

Java 还支持摘要类的接口。这就允许编程人员先定义接口的方法, 而不急于马上确定方法的实现。一个类可以实现多个接口, 从而具有真正多继承性的许多优点。一个对象也可实现任意数量的接口。Java 接口与 IDL 接口极其相似。很容易建立 IDLJ 编译器。这就是说, Java 可用于 CORBA 对象系统来建立分布式对象系统。鉴于在许多计算机系统中都采用了 IDLJ 接口和 CORBA 对象系统, 这种兼容性更为重要。

1.2.3 Java 是静态类型的

在一个 Java 程序中, 必须定义所用对象(数字、字符、数组等)的类型。这有助于编程人员很快发现问题, 因为当程序编译时可以检测类型错误。

不过, Java 系统中的对象也有动态类型。要求某一对象为动态类型往往是可能的, 所以编程人员写的程序可以对不同类型的对象做不同的事。

1.2.4 Java 是编译型的

当运行 Java 程序时, 它首先被编译成字节代码。字节代码非常类似于机器指令, 所以 Java 程序非常高效。然而, 字节代码并不专对一种特定的机器, 所以 Java 程序无需重新编译便可在众多不同的计算机上执行。

Java 源程序被编译成类文件, 它相当于程序的字节代码表现。在一个 Java 类文件中, 所有对方法及实例变量的参照均按名进行, 并在第一次执行代码时加以分辨。这就使得代码更通用, 更不易受修改的影响, 而仍具高效。

1.2.5 Java 是体系结构中立的

Java 语言对每种计算机都一样。比如, 简单的类型都是不变的: 整数总是 32 位, 长整数总是 64 位。令人奇怪的是, 诸如 C 及 C++ 等时髦的编程语言却不是这样。由于这些语言定义如此自由, 使每种编译器及开发环境各有不同, 这使程序的移植成为问题。Java 程序的移植却很容易, 而且不需要进行重新编译。

1.2.6 Java 是健全的

Java 程序不会造成计算机崩溃。Java 系统仔细检测对内存的每次访问, 确认它是合法的, 而且不致引起任何问题。

不过, 即使 Java 程序也可能有错误。如果出现某种出乎意料之事, 程序不会崩溃, 而把该例外抛弃。程序会发现这类例外, 并加以处理。

传统的程序可以访问计算机的全部内存。程序可能(无意识地)修改内存中的任何值, 这就会造成问题。Java 程序只能访问内存中允许它们访问的那些部分, 所以 Java 程序不可能修改不拟修改的值。

1.2.7 Java 是小巧的

由于 Java 的设计是要在小的计算机上运行, 作为一种编程语言来说其系统相对较小。它能有效地在 4MB 以上 RAM 的 PC 机上运行。Java 翻译器只占用几百 KB。这种翻译器对 Java 的平台无关性和可移植性是可靠的。

由于 Java 很小, 它对内存很小的计算机, 如基于 Java 的 PC 机, 以及电视机、烤箱、电话机及家用计算机等都是很理想的。

1.2.8 Java 是多线程的

Java 程序可以执行一个以上的线程。比如, 它可以在一个线程中完成某一耗时的计算, 而其它线程与用户进行交互对话。所以, 用户不必停止工作, 等待 Java 程序完成耗时的计算。

在多线程环境中的编程通常是困难的, 因为许多东西可能同时出现。但是, Java 提供了易于使用的同步特性, 让编程更为方便。

Java 线程通常被映射为实际的操作系统线程, 只要底层操作系统支持这种映射。因此, 用 Java 编写的应用程序可说是“MP 热”。这就是说, 如果它们在多处理器机器上运行, 将更加表现不凡。

1.2.9 Java 是可收集无用的存储单元的

用 C 及 C++ 写软件的编程人员必须仔细跟踪所用的内存块。当一个内存块不再使用时, 他们务必让程序释放它, 从而方可再用。在大的项目中, 这可能很困难, 并往往成为出错和内存不足的根源。

在 Java 的情况下, 编程人员不必为内存管理操心。Java 系统有一个叫做“无用单元收集器”的内置程序, 它扫描内存, 并自动释放那些不再使用的内存块。

1.2.10 Java 是快速的

Java 比典型的脚本语言大为有效, 但它比 C 慢 20 倍。这对大多数应用是可接受的。不久的将来, 代码生成器就可供利用了, 这将使得 Java 程序几乎近于用 C 或 C++ 编写的程序那么快。

1.2.11 Java 是安全的

Java 程序没有指针, 因为像这样的字节代码程序强在类型上, 所以在执行之前它可能验证 Java 程序。被验证的 Java 程序得以保证不突破 Java 语言的任何限制, 而且可以安全地执行。Java 字节代码验证被 Web 浏览器用来确保小程序不含病毒。

1.3 几种 Java 语言的开发工具简介

平台中立是 Java 的一个重要特性, 正是这个特性使得 Java 非常适用于以网络为中心的计算环境。因此 Java 一经推出就受到了全世界的欢迎。随着 Java 语言的迅速发展, Java 开发工具也越来越多。本节对四种最流行的 Java 开发工具进行了比较, 介绍了它们在平台

中立应用的可视化开发方面的各种表现：Sun Java Workshop 2.0, Borland JBuilder Professional 1.0, IBM VisualAge for Java 1.0 Enterprise Edition 和 Symantec Visual Cafe for Database Development Edition 2.0。

1.3.1 Java 如何进行布局管理

基于 Windows 的工具往往在可视化开发中一直都使用“所见即所得 (WYSIWYG)”的方式。这对众多的 Windows 开发人员来讲当然是一个很熟悉的开发环境。但是这种方法开发的应用放到其他的平台上时几乎总是会出现这样或那样的问题。因此在 Java 的可视化开发中就不能再继续沿用这种开发方式了。

Java Workshop 2.0 在基于网格布局创建具有吸引力的用户接口方面功能最强，因为 Workshop 是惟一的围绕网格布局创建可视化设计器的工具。然而，正是因为这一点使得 Java Workshop 2.0 有时在灵活性方面就略显不足，但是总的来看，它还是这几种工具中综合表现最好的。

如果用户既要求易于进行可视化布局，又要求有较大的灵活性，JBuilder 就是最好的选择。JBuilder 允许程序员使用绝对坐标设计结构，然后再将其很方便地转换成网格，这使得管理网格的问题变得相当容易。

在可视化布局方面 VisualAge for Java 和 Visual Cafe for Java 要略胜一筹。如果要想利用这些产品设计网格布局，需要不断摸索和实践才能设置好网格的限制变量。

在可视化编程方面，VisualAge for Java 要比其他几种工具好得多。利用 VisualAge Visual Composition Editor 编辑器再加上一个鼠标用户几乎就可以完成一切开发工作了。这个编辑器功能强大，但是熟练掌握它却不太容易。而 Visual Cafe for Java 不但能够提供 VisualAge for Java 在可视化方面的绝大部分功能并且容易学习。Java Workshop 在可视化编程方面不如 Visual Cafe，因为使用它许多代码还是需要手工编写。而 JBuilder 在可视化编程方面的表现就更不如人意。

如果不是创建大规模的应用，JBuilder 和 Visual Cafe for Java 提供的缺省组件调色板很难比较出个高低来。Visual Cafe 调色板提供的计时器控制（用于激发事件）很能引起人们的兴趣。JBuilder 也具有一些很好的图表控制。VisualAge 在这方面要弱一些，但它包括了很丰富的有用的 Java 类。Java Workshop 在调色板方面的功能最差。

1.3.2 Java Workshop 2.0

1. 结构易于创建

在创建平台中立的网格结构方面，Java Workshop 2.0 比其他任何一种 Java 开发工具都要方便。但是它的这一点好处却经常被一些复杂的设计选择项给掩盖了，要熟练掌握基于网格布局的可视化设计器需要下一些功夫，但这是很值得的。一旦用户掌握了这些设计方法，工作起来就会相当轻松。

用户可以很容易地拖放组件并调整其所占的网格单元区域。用鼠标单击网格外的部分就可以控制网格限制的设置（如挤压和填充物等）。运行过程中不管用户在何时想检查结构是个什么样子，都可立即转换到预览模式。用户还可对预览结构进行调整和控制以观察这些变化能否立刻反映到自己的设计中。