

Java
Workshop

- Java 编程基础
- Java 编程技巧
- Java 编程工具

(美) Clayton Walnum 著
华译工作室 译

Using Java Workshop

Java Workshop

使用指南

12
04

机械工业出版社



西蒙与舒斯特国际出版公司

最新 Internet 技术基础与应用系列丛书

最新 Internet 技术基础与应用系列丛书

Java Workshop 使用指南

(美) Clayton Walnum 著

华译工作室 译

机 械 工 业 出 版 社
西蒙与舒斯特国际出版公司

JS160/26

本书主要介绍了 Java 编程语言的特点及在 Internet 中的使用方法。从最基础的编程开始,到编写具有丰富特点的小应用程序和独立应用程序,系统介绍了 Java 编程的全过程。并详细讲述了其系列产品——Java Workshop 的各种工具,诸如公文包管理器、项目管理器、项目测试器等。

本书图文并茂,内容由浅入深,文字流畅,引人入胜,具有很强的可读性和实用性,特别适合初级和中级编程人员使用,也是各类培训班的首选教材。

Clayton Walnum; Using Java Workshop

Authorized translation from the English Language edition published by Que.

Copyright 1996 by Que.

All rights reserved. For sale in Mainland China only.

本书中文简体字版由机械工业出版社和美国西蒙与舒斯特国际出版公司合作出版,未经出版者书面许可,本书的任何部分不得以任何方式复制或抄袭。

本书封面贴有 Prentice Hall 防伪标签,无标签者不得销售。

版权所有,翻印必究。

本书版权登记号:图字:01-97-0141

图书在版编目(CIP)数据

Java Workshop 使用指南/(美)瓦内姆(Walnum,c.);华译工作室译.-北京:机械工业出版社,1997.5

(最新 Internet 技术基础与应用系列丛书)

书名原文:Using Java Workshop

ISBN 7-111-05588-8

I. J… II. ①瓦… ②华… III. Java 语言-程序设计 IV. TP312Ja

中国版本图书馆 CIP 数据核字(97)第 01450 号

出版人:马九荣(北京市百万庄南街 1 号 邮政编码:100037)

责任编辑:何伟新

三河永和印刷有限公司印刷·新华书店北京发行所发行

1997 年 5 月第 1 版第 1 次印刷

787mm×1092mm 1/16·18 印张·427 千字

0001-7000 册

定价:32.00 元

凡购本书,如有倒页、脱页、缺页,由本社发行部调换

前 言

随着国际互联网(Internet)的飞速发展,在不久的将来,不仅可以和朋友在网上聊天,甚至还可以和每一家大公司进行联系。这种不可思议的发展,带来的是机遇。处在信息时代的人们,都在想尽办法连上 Internet。Sun Microsystem 公司就是一家在国际互联网上一鸣惊人的公司,最近发布了一种受人瞩目的程序设计语言——Java。一旦人们接触了 Java,就会发现 Internet 带给他们的将是一个崭新的世界。

那么,Java 到底有什么特别的地方呢? 它给人们带来了一种叫做 applet 的小应用程序,它可以嵌在用户自己的网页程序里。当用户在自己的机器上浏览一个包含 applet 的网页时,机器就会自动地在网上找到那个小应用程序,并在自己的浏览器上运行它,当然这个浏览器程序是要支持 Java 语言的。因为小应用程序能够以机器无关的形式传输,所以能够在拥有 Java 解释器的所有机器上运行。

利用 Java 编程语言,可以在网页中加入简单的动画效果,也可以编写复杂的程序在网上供别人使用。总之,只要使用 Java,没有什么你做不到的。现在已经发布了各种各样的小应用程序,包括游戏、电子表格、绘图程序、动画控制器和模拟器等。以至于业界的巨人们,包括 Microsoft, Netscape, 都先后宣布支持 Java,并提供支持 Java 的软件。

现在,Sun 公司发布了 Java 编程语言,又开发出了 Java 的系列产品——Java Workshop。这是一个复杂的 Java 开发平台。几个月前,利用 Java 编写小应用程序的编程人员还不得不使用 JDK 中的命令行工具,现在有了 Java Workshop,就可以非常方便地使用这些工具了,因为它带有一个非常友好的图形界面,而且,它还带有一套它自己的工具集,包括一个具有丰富特点的源编辑器和一个项目管理器。

从本书中,可以了解到 Java 小应用程序是如何在 Internet 上工作的,如何把 Java 小应用程序嵌入到自己的网页中。更重要的是,还可以学会如何使用 Java Workshop 一步一步编写自己的小应用程序,不但可以自己用,也可以放在 Internet 上发布,供大家使用。

读者对象

本书是学习 Java 的入门书,同时对有一定编程经验的人来说,尤其是用过 C 或 C++ 的,也有很大的帮助。这本书系统介绍了 Java 语言的特点和如何利用 Java Workshop 编写小应用程序,还详细讲述了 Java Workshop 的各种工具,诸如公文包管理器(Portfolio Manager),项目管理器(Project Manager),项目测试器(Project Tester)——这些都是创建小应用程序时所用到的工具。从学习最基础的编程开始,到编写具有丰富特点的小应用程序和独立应用程序,学习 Java 编程。

尽管本书是面向初学者的,但是有经验的编程人员也会发现很多有趣的东西。如果对 C 或 C++ 已经很熟悉了,可以跳过 Java 简介,直接学习创建小应用程序的内容。尽管 Java 语言和 C 语言很相象,但 Java 有自己的特点。在 Java 出现以前,还从来没有过象 Java 小应用程序

这样的东西。

总之,本书对初级和中级编程人员都是适用的。初学者可以了解一下 Java 语言,有经验的编程人员可以集中精力学习如何编写功能强大的小应用程序。高级编程人员也许会发现,本书对于了解 Java 也是有一定帮助的。当然,为了更好的读懂这本书,一些基本的编程知识还是必须的,诸如函数、循环等等。

硬件和软件要求

目前,Windows 95、Windows NT、Sun Solaris、Macintosh、Unix 系统都支持 Java,本书的大部分内容对于那些能运行 JDK 的机器都是适用的。但是 Java Workshop 目前只能运行于 Windows 95、Windows NT 和 Solaris 系统上。因为 Windows 95 毫无疑问将会成为拥有最多小应用程序的操作系统,所以本书中的例子和程序都是用 Windows 95 版的 Java Workshop 来编写的。

Windows 95 或 NT 用户的最低配置如下:

- 一台 IBM 486 或其兼容机,最少 16M 内存。
- 操作系统是 Windows 95 或 Windows NT。
- 一个硬盘驱动器。
- 一个 Microsoft 兼容的鼠标。
- 256 色的彩显。
- 一个 Windows 兼容的声卡(如果不太在意听声音文件的话,也可以不要)。
- 可以从 Microsystem 站点 <http://www.sun.com> 上得到一份 JDK 的拷贝。

编译本书的程序

当要用到本书的例子时,需要了解如何安装 Java Workshop,如何编译每章提供的例子程序。通常,可以按照以下步骤来编译本书的程序。

1. 在系统上,使用默认的目录 C:\Java-Workshop 安装 Java Workshop(另外,可再安装一个 Netscape Navigator 2.0,可以从 Netscape 的站点 <http://www.netscape.com> 上得到该浏览器的一份拷贝,这个浏览器支持 Java)。

2. 用 SYSEDIT.EXE 加载系统文件(可以在 Windows\system 目录中找到 SYSEDIT.EXE)。

3. 找到 AUTOEXEC.BAT 文件中的路径设置,在后面加上 C:\Java-Workshop\JDK\Bin。

4. 保存这些设置,重新启动机器。

增加路径是为了确保在命令行状态下使用这些工具时,系统可以找到它们。

最后,还要创建一个目录 C:\Classes,用于保存 Java 文件。

编译和运行一个小应用程序,按以下步骤进行:

1. 在根目录下建立一个名为 Classes 的文件夹,存放 Java 文件。

2. 在 Classes 文件夹里再建立一个子文件夹,用小应用程序的类名夹命名,子文件夹里存放小应用程序的源文件。

例如,如果要为一个小应用程序 Applet1 的类建立一个文件夹,则在 Classes 里还要建立一个名为 Applet1 的子文件夹。

3. 录入小应用程序的源代码并把它存放在刚建立的文件夹中。源文件通常以 .java 为后缀名,包含声音或图像文件的小应用程序,也是这样。例如,一个小应用程序 Applet1,它的源文件就是 Applet1.java。

4. 启动 Java Workshop,显示如图 0-1:

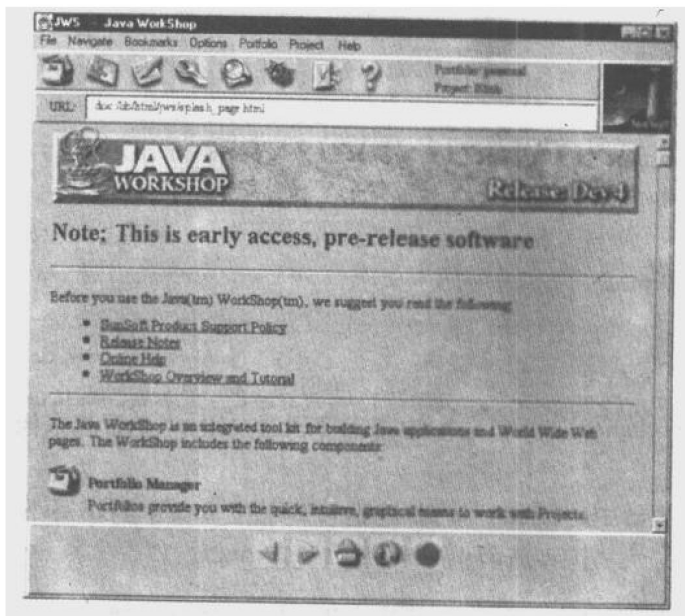


图 0-1 首次运行 Java Workshop 时的样子

5. 如果已经创建了一个公文包(portfolio),跳到第 7 步。

6. 从 Java Workshop 菜单条上选择 Portfolio|Create,弹出创建公文包对话框。

7. 在文本框内敲入 C:\Classes\Using Java Workshop.psf,如图 0-2 所示。然后,单击 Create 按钮,创建一个新的公文包。利用公文包组织源文件,利用这些源文件构造各种 Java 项目。在这里,可以使用 Using Java Workshop 这个公文包来组织创建的所有程序。

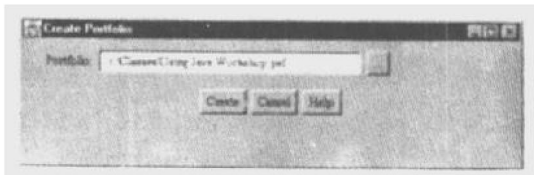


图 0-2 使用 Create Portfolio 对话框创建一个公文包

8. 从 Java Workshop 菜单条上选择 Project|Create|Applet,弹出小应用程序的属性表单。

9. 在名字框中敲入小应用程序的名字,在源目录框中敲入 C:\Classes\,后面再紧接着敲入这个小应用程序的文件夹名。例如,如果要为一个名为 Applet1 的小应用程序创建一个项目,则在名字框中敲入 Applet1,在源目录框中敲入 C:\Classes\Applet1。

10. 在“Existing Sources”项,单击“Yes”按钮,然后单击“Add All in Directory”按钮,这个小应用程序的源文件就出现在源代码框内。

11. 最后,在主类文件名框内敲入小应用程序名(见图 0-3),后跟.class 后缀名(例如 Applet1.class),然后,单击“Apply”按钮,Java Workshop 便开始创建一个新的小应用程序项目,并且打开文本编辑器。项目为公文包中的每一个独立程序组织它们的源代码文件。可以认为一个公文包就是一个文件夹,它包含所有的 Java 项目,而项目就象是公文包这个文件夹里的子文件夹。

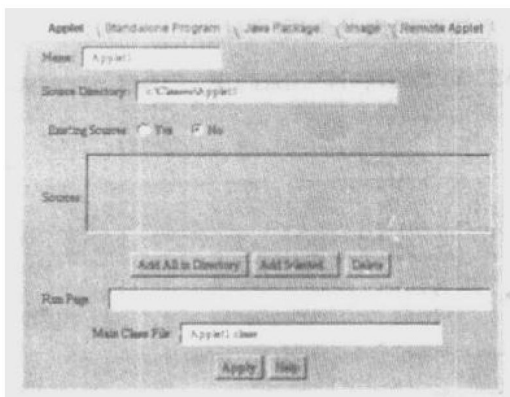


图 0-3 用属性表单创建一个项目

12. 单击 Java Workshop 主工具条中的建造管理器按钮(一个看起来像扳手的按钮),出现建造管理器窗口。

13. 单击建造管理器中的“Build”按钮(建造管理器工具条中的第一个按钮),建造管理器运行 Java 编译器——javac,编译新的小应用程序(见图 0-4)。

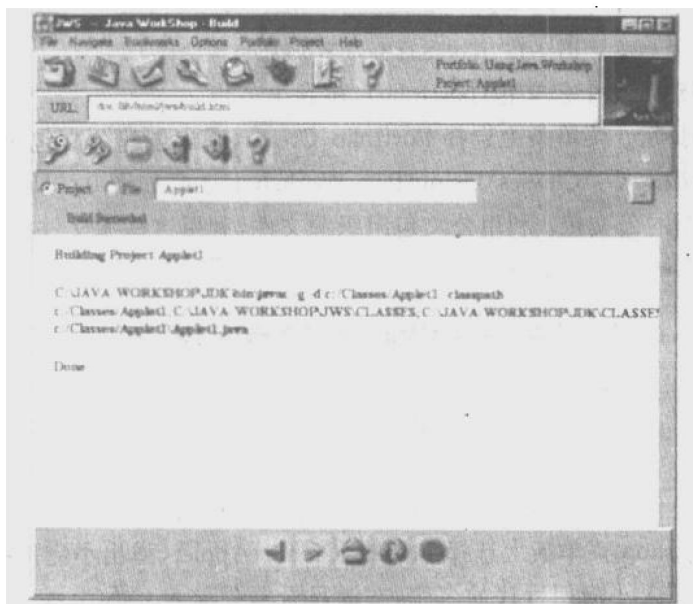


图 0-4 Build Nanager 运行 Java 编译器编译一个小应用程序

14. 单击 Java Workshop 主工具条中的项目测试器按钮(样子象电灯开关),测试小应用

程序。

忠告

每个编程人员都知道,一个好的程序应该是防崩溃的。对于每一个可能导致失败的行为都要进行错误检查,并且要给出合适的错误信息。遗憾的是,好的错误检查需要大量的额外编程。对于开发人员来讲,这仅仅是其工作的一部分,但对于写编程书的作者而言,把查错程序全列出来,不免会事倍功半。一本编程书应尽量使其主题清楚明了,也就是说,所描述的程序,其源码不必从许多与当前主题并不直接相关的细节中获得。正是这个原因,本书里的程序并不是都有错误检查,例如,用户输入的数据就没有进行验证,对象的动态构造(dynamic construction)也被认为是成功的。

简而言之,如果想把本书列出来的代码用到自己的程序中时,应该加上被漏掉的错误检查,以确保不使程序在用户手中崩溃。

进入美妙的 Java 世界

如果你还在看这个前言,可以确信你确实想从 Java 中学到点什么。如果你对 Internet 感兴趣,这是一个明智的选择。毕竟,Java 在 Internet 的历史上应该是占有一席之地的。

Clayton Walnum

1996 年 6 月

目 录

前言

第一部分 入门篇

第1章 Java 概述	1
1.1 Java 的故事	1
1.2 Java 介绍	2
1.3 Java 程序	3
1.4 Java 开发人员工具箱	5
1.5 如何获得 Java	5
1.6 安装 HotJava	6
1.7 安装 JDK	6
1.8 Java 小应用程序范例	7
1.8.1 Appletviewer 工具	7
1.8.2 执行 TicTacToe	7
1.8.3 小应用程序 Animator	8
1.8.4 小应用程序 BarChart	8
1.8.5 其他的演示小应用程序	8
1.9 在 HTML 文档中添加小应用 程序	9
1.9.1 小应用程序的可选属性	9
1.9.2 小应用程序参数	11
1.9.3 非 Java 浏览器	12
1.10 小应用程序和 Internet	12
1.10.1 本地的和远程的 Applets	12
1.10.2 客户机与服务器	14
1.10.3 安全性	14
1.10.4 网页	15
1.11 小结	17
第2章 Java Workshop 介绍	18
2.1 Java Workshop 工具	18
2.1.1 项目管理器	18
2.1.2 公文包管理器(Portfolio Manager)	19

2.1.3 建造管理器(Build Manager)	20
2.1.4 项目测试器(Project Tester)	20
2.1.5 源编辑器(Source Editor) ...	22
2.1.6 源浏览器(Source Browser)	23
2.1.7 调试器(Debugger)	25
2.1.8 可视化 Java(Visual Java) ...	26
2.2 安装 Java Workshop	26
2.3 小结	27

第二部分 Java 语言

第3章 关键字、数据类型和运算符	29
3.1 关键字	29
3.2 标识符	30
3.3 常量	30
3.4 数据类型	31
3.4.1 整型	31
3.4.2 浮点型	32
3.4.3 字符型	33
3.4.4 布尔型	34
3.5 变量的作用域	34
3.6 数学运算符	35
3.6.1 运算结果的类型转换	35
3.6.2 乘法和数据类型	36
3.6.3 整数和浮点数相除	36
3.6.4 模除运算符	36
3.6.5 自增运算符	36
3.6.6 自减运算符	37
3.7 表达式	37
3.8 比较运算符	38
3.9 逻辑运算符	39
3.10 小结	40

第4章 程序流程控制	42	6.1 最简单的 Java 小应用程序	69
4.1 程序流程和分支	42	6.2 小应用程序的生命周期	70
4.2 if 语句	42	6.2.1 生命周期的五个阶段	70
4.2.1 简单 if 语句	42	6.2.2 重置生命周期的方法	71
4.2.2 多行 if 语句	43	6.3 画图	72
4.2.3 else 子句	43	6.3.1 小应用程序的画布	72
4.3 switch 语句	44	6.3.2 使用坐标系	72
4.4 while 循环	46	6.3.3 画形状	73
4.5 do-while 循环	46	6.3.4 画一个矩形	74
4.6 for 循环	47	6.4 形状绘制小应用程序	76
4.7 在循环中使用变量	48	6.4.1 小应用程序 shapeApplet	76
4.8 小结	48	6.4.2 理解小应用程序	
第5章 面向对象的程序设计和 Java		shapeApplet	78
语言	50	6.4.3 在小应用程序中画弧形	79
5.1 历史回顾	50	6.4.4 画多边形	81
5.2 现代程序设计的思想	52	6.5 图形文本	82
5.3 面向对象的程序设计	52	6.5.1 在小应用程序中显示文本	82
5.3.1 封装	52	6.5.2 获取字体属性	83
5.3.2 把类作为一种数据类型	53	6.5.3 显示字体属性	83
5.3.3 继承	53	6.5.4 获取字体的规格	
5.3.4 多态性	53	(metrics)	85
5.4 封装、继承和多态性的使用	53	6.5.5 显示字体的规格	86
5.5 类和对象	55	6.5.6 建立字体	87
5.5.1 定义一个简单的类	55	6.5.7 使用字体	88
5.5.2 声明类的数据域	55	6.5.8 显示不同大小的字体	88
5.5.3 定义构造函数	56	6.6 小结	91
5.5.4 定义方法	56	第7章 Java 控件	92
5.6 在小应用程序中使用类	57	7.1 标签	92
5.7 理解小应用程序 Applet1	61	7.2 按钮	93
5.8 使用继承	62	7.2.1 处理多按钮事件	94
5.8.1 建立子类	62	7.2.2 小应用程序 Button	
5.8.2 向子类中添加数据域和		Applet	94
方法	62	7.3 复选框	95
5.8.3 在程序中使用子类	63	7.3.1 建立非互斥使用的	
5.9 重置(override)超类中的方法	66	复选框	96
5.10 关键字 this	66	7.3.2 复选框组	97
5.11 小结	66	7.3.3 复选框的方法	98
		7.3.4 小应用程序 Checkox	
		Applet	98
		7.3.5 响应复选框的事件	100
		7.3.6 小应用程序 Checkbox	
第三部分 编写 Java 小应用程序			
第6章 小应用程序和图形	69		

Applet2	101	8.7.2 建立和设置网格包约束 对象	131
7.4 文本域	102	8.7.3 在小应用程序中使用 网格包布局管理器	132
7.4.1 文本域的方法	103	8.7.4 理解小应用程序 GridBag Applet	134
7.4.2 小应用程序 EchoApplet ..	103	8.8 小结	137
7.5 选项菜单(choice menu)	104	第9章 窗口、菜单条和对话框	139
7.5.1 选项菜单的方法	106	9.1 显示窗口	139
7.5.2 小应用程序 ChoiceApplet...	106	9.1.1 在小应用程序 FrameApplet 中显示一个窗口	139
7.6 滚动式列表	108	9.1.2 建立一个窗口类	140
7.6.1 建立一个单选列表	108	9.1.3 向窗口中添加组件	142
7.6.2 建立一个多选列表	109	9.2 使用菜单条	145
7.6.3 建立滚动式列表	109	9.2.1 建立和设置类 MenuBar 的一个对象	145
7.6.4 类 List 的方法	110	9.2.2 向菜单条中添加菜单	146
7.6.5 小应用程序 ListApplet ..	111	9.2.3 向菜单中添加菜单项	146
7.7 文本区控件	112	9.2.4 在框架窗口中使用菜 单条	147
7.7.1 小应用程序 TextArea Applet	112	9.3 使用对话框	150
7.7.2 类 TextArea 的方法	114	9.3.1 建立对话框	150
7.8 滚动条	114	9.3.2 建立对话框的布局	150
7.8.1 响应滚动条事件	116	9.3.3 显示对话框	151
7.8.2 小应用程序 Crollbar Applet	117	9.3.4 撤走对话框	151
7.9 画布	119	9.3.5 类 Dialog 的方法	151
7.10 小结	121	9.3.6 建立一个用于文本输出 的对话框	151
第8章 面板和布局管理器	123	9.4 小结	154
8.1 面板	123	第10章 事件和可配置的小应用 程序	156
8.2 布局管理器	125	10.1 类 Event 的对象	156
8.3 流式布局管理器(FlowLayout Manager)	125	10.2 鼠标	156
8.4 网格布局管理器(GridLayout Manager)	126	10.2.1 处理鼠标单击事件	157
8.5 边框式布局管理器(BorderLayout Manager)	127	10.2.2 在小应用程序中使用 鼠标单击	157
8.6 卡片式布局管理器(CardLayout Manager)	129	10.2.3 处理鼠标移动事件	158
8.6.1 卡片式布局管理器的 方法	129	10.2.4 在小应用程序中响应 鼠标移动事件	159
8.6.2 建立卡片式布局管理器 ..	129	10.3 键盘	161
8.7 网格包布局管理器(GridBag Layout Manager)	131	10.3.1 响应按键事件	161
8.7.1 建立和设置网格包布局 管理器	131		

10.3.2 预定义的键常量	161	第12章 程序包、接口、异常情况和线程	194
10.3.3 键修饰符	162	12.1 程序包	194
10.3.4 在小应用程序中使用 按键事件	162	12.1.1 创建程序包	195
10.4 直接处理事件	164	12.1.2 使用新建的程序包	197
10.5 可配置的小应用程序	166	12.1.3 扩充程序包	199
10.6 用户类型	166	12.2 接口	201
10.7 参数和小应用程序	167	12.2.1 基本接口	201
10.7.1 设置并检索参数值	168	12.2.2 创建接口	201
10.7.2 在小应用程序中使用 参数	168	12.2.3 实现接口	202
10.7.3 多个参数	172	12.3 异常情况	203
10.7.4 默认的参数值	174	12.3.1 抛出异常情况	205
10.8 小结	177	12.3.2 异常情况的类型	206
第11章 图像、声音和通信	178	12.3.3 确定要处理的异常 情况	207
11.1 图像类型	178	12.3.4 抓住运行时的异常 情况	209
11.2 加载和显示图像	178	12.3.5 处理多个异常情况	210
11.2.1 使用 getDocumentBasec() 方法	178	12.4 线程	212
11.2.2 使用 getCodeBasec() 方法	179	12.4.1 把类转变为线程	212
11.2.3 加载图像	179	12.4.2 同步多线程(Synchronizing Multiple Threads	218
11.2.4 显示图像	179	12.4.3 理解 ThreadApplet3	220
11.3 声音播放	181	12.5 小结	221
11.3.1 在小应用程序中播放 声音	182	第13章 编写 Java 独立应用程序	222
11.3.2 控制声音播放	183	13.1 什么是 Java 独立应用程序	222
11.3.3 在小应用程序中使用 声音	183	13.2 最简单的 Java 独立应用 程序	222
11.4 通信	185	13.3 建造一个独立应用程序	223
11.4.1 URL 对象	185	13.4 获取独立应用程序的 命令行参数	225
11.4.2 URL 异常	186	13.5 窗口式独立应用程序	226
11.4.3 小应用程序上下文	186	13.5.1 把小应用程序改为 独立应用程序	226
11.4.4 用 Applet Context 对象连接 URL	186	13.5.2 理解 Face App 独立 应用程序	228
11.4.5 在小应用程序中使用 类 Applet Context 的 对象	187	13.5.3 最后一个独立应用程序 范例	229
11.4.6 建立一个设置好 URL 的小应用程序	189	13.6 小结	232
11.5 小结	192		

第四部分 Java Workshop 的工具

第14章 公文包管理器(Portfolio Manager)和项目管理器

(Project Manager)	233
14.1 使用公文包管理器	233
14.1.1 创建新公文包	233
14.1.2 选择公文包	235
14.1.3 移走(remove)公文包 ..	236
14.1.4 装入公文包	237
14.2 使用项目管理器	237
14.2.1 创建项目	237
14.2.2 选择项目	240
14.2.3 装入项目	240
14.2.4 编辑项目	241
14.2.5 移走(remove)项目	241
14.2.6 运行项目	243
14.3 小结	243
第15章 源编辑器	244
15.1 运行源编辑器	244
15.2 状态条	245
15.3 文件操作	246
15.4 编辑命令	246
15.5 查找和替换	248
15.6 其他菜单	248
15.7 控制文本光标	249
15.8 小结	249
第16章 建造管理器和源浏览器	250
16.1 使用“建造管理器”	250
16.1.1 启动建造管理器	250
16.1.2 编译项目	251
16.1.3 修改语法错误	251

16.1.4 选择项目或文件	252
16.2 使用源浏览器	253
16.2.1 启动源浏览器	253
16.2.2 浏览项目	253
16.2.3 浏览目录	255
16.2.4 查找字符串	256
16.3 小结	257
第17章 调试器	258
17.1 启动调试器	258
17.2 调试器的显示页	258
17.2.1 线程/堆栈页(Threads/ stacks page)	259
17.2.2 表达式页(Expressions page)	260
17.2.3 断点页(Breakpoints page)	262
17.2.4 异常情况页(Exceptions page)	263
17.2.5 类页(Classes page)	264
17.2.6 消息页(Messagepage)	265
17.3 从源编辑器中调试	265
17.4 小结	266
第18章 Visual Java	267
18.1 使用 Visual Java	267
18.1.1 为新程序创建项目	267
18.1.2 构造小应用程序的 界面	268
18.1.3 产生代码	271
18.1.4 向小应用程序加入 代码	272
18.2 小结	274

第一部分 入 门 篇

第 1 章 Java 概述

任何一个使用计算机的人,都知道当今最流行的是 Internet。这个国际互联网的飞速发展,使人们不仅能够方便地下载文件,而且还导致了关于言论自由、版权和计算机安全的激烈争论。几乎每一天,Internet 上都在制造着新闻。

但是并不是所有 Internet 上的举动都会引起争论,比如说 Java 语言的发布,就得到了人们的一致认同。使用 Java 语言,可以开发出 Internet 上的独立应用程序,而不必考虑操作平台的兼容性和网络上的安全性。两种类型的 Java 程序——用于 3W 网页上的小应用程序和独立应用程序,使得 3W 变得充满生机。

当初构思 Java 时,并没有想到日后它会如此适用于 Internet,你也许会感到很奇怪,开发 Java 竟是为了另外的用途。实际上,“Java”都不是这个语言的原名。在本章里,可以大致了解一下 Java 的历史,得知 Java 为什么会成为用于创建 Internet 上应用程序的杰出工具。

1.1 Java 的故事

Java 的产生可以追溯到 1990 年,当时,Java 的开发者之一——James Gosling 接受了一项任务,开发一个面向家用电器市场的软件产品。Gosling 和他的开发小组刚开始是使用 C++ 进行开发,因为 C++ 语言以它面向对象编程的特色为软件开发人员所称道。但是,Gosling 很快就发现 C++ 并不能胜任这个工作,因为 C++ 在某些方面显得过于复杂,诸如类的多重继承等;而且存在程序臭虫(bugs),诸如内存漏洞等。因此,Gosling 决定开发一种新的语言。尽管 Gosling 对 C++ 语言的复杂性不感兴趣,但对它的基本语法和面向对象的特性还是非常欣赏的。所以,他在设计自己的新语言时,就以 C++ 为模板完成了新语言的设计,Gosling 把它命名为 Oak。

Oak 首先被用于“Green”项目,这个项目的目标是设计一种家用控制系统。这个控制系统可以使用户通过一个名为 * 7 的手持计算机控制多个电器,诸如电视、录像机、电灯、电话等等,* 7 系统采用了触摸屏技术,用户可以通过它选择和控制设备。

注:* 7 的屏幕上有许多动画图形,Duke 就是其中之一。一旦接触到 Java,肯定会看到许多的 Duke 在 Sun 公司的站点上跳来跳去,它还是一些 Java 小应用程序范例的特色呢。

接着,Oak 又用在了 VOD 项目中,在这个项目中,Oak 被用作控制交互式电视系统的软件基础。尽管 * 7 和 VOD 这两个项目都没产生什么实际产品,但它们给了 Oak 一个发展和成熟

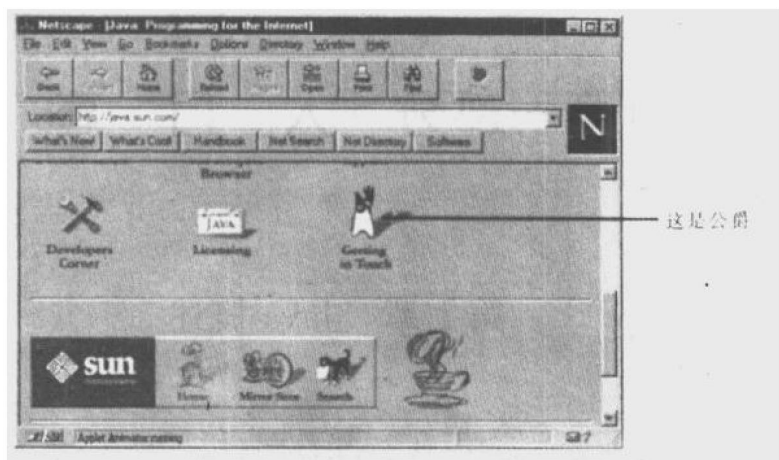


图 1-1 Duke 已经成了 Java 吉祥物

的机会。当 Sun 公司发现“Oak”已经被注册过了,便起了一个新名字“Java”。于是,有了一个功能强大,简易安全的编程语言——Java。更重要的是,Java 是一种平台无关性的编程语言,意味着用它开发出来的程序能够在任何系统中无差别地运行。平台独立性(platform independence)是因为 Java 程序可以被编译成一种特殊的格式——字节码(byte-code),有 Java 解释器的机器都可以读懂和运行它。当然,不同系统中的 Java 解释器是要分别编写的。

1993 年,随着 3W 的发展,Internet 由文本方式转向了丰富多彩的图形方式,Java 开发小组意识到 Java 非常适合于 3W 编程,于是他们提出了小应用程序(Applet)的概念,这种小应用程序可以嵌在网页程序中。他们还用 Java 开发出了一个浏览器程序——HotJava。

到了 1995 年的第二个季度, Sun 公司公布了 Java 的完整技术规范,并迅速成为开发 Internet 应用程序的强有力工具。Java 的发布得到了 3W 厂商的广泛支持, Netscape 公司的 Netscape Navigator 2.0(见图 1-2), Microsoft 公司的 Internet Explorer 3 β 版都增加了支持 Java 的功能。经过五年多的发展,Java 终于找到了它的归宿。



图 1-2 新的 Netscape Navigator 2.0 是一个支持 Java 的浏览器

1.2 Java 介绍

直到现在,也许称 Java 是 Internet 上最强大的开发工具仍然使不少人感到很迷惑。其实,

Java 就是 C++ 的简化版,它的特色就在于它的简洁性。C++ 比 C 增加了许多东西,就连专业的编程人员都很难一下子搞清楚。

根据 Sun 公司的定义,Java 是一种具有“简单、面向对象、分布式、解释型、健壮、安全、体系结构中立、可移植、高性能、多线程和动态”等各种特性的语言,这种口语化的描述,也许并不能使人对 Java 了解多少。下面通过列举 Java 的这些特性,以便理解。

简单性

Java 的开发者们删去了 C++ 中大量复杂和冗余的部分。它不支持指针、显式类型转换、结构和联合、操作符重载、模板、头文件、多重继承等 C++ 的语法,使软件人员易于开发、实现和维护。

面向对象

和 C++ 一样,Java 也是使用类来组织逻辑模块中的代码。运行时,程序都是用类来创建对象,Java 的类具有继承性,但多重继承(从多个类中继承方法和数据域)是不允许的。

静态类型定义

程序中用到的所有对象都要预先定义,这样可以使 Java 编译器定位和报告类型冲突。

编译

运行一个 Java 程序前,必须先编译它。编译的结果是产生了一个字节码文件,字节码类似于机器码,可以在有 Java 解释器的所有操作系统中执行,执行的时候,先把字节码译成机器码,再直接运行。在这个意义上说,Java 既是一种编译语言,也是一种解释语言。

多线程

Java 程序支持多线程,可以同时处理多个任务。例如,一个多线程的程序可以在接受用户从键盘上输入信息的同时,在屏幕上绘制一个图形,所有的程序都至少有一个主线程。

内存垃圾收集

Java 提供了自动内存垃圾收集机制,也就是说,程序不必删除内存中分配的对象,从而减轻了编程人员进行内存管理的负担。

健壮性

因为程序运行时 Java 解释器会检查所有的系统访问,所以不会导致系统崩溃。当发现一个严重错误时,程序会创建一个异常,然后捕捉和处理这个异常,不会使系统死掉。

安全性

Java 系统不但检查所有的内存访问,而且还能确保在运行的小应用程序不受病毒的侵入。因为,Java 不支持指针,所以,程序无法进入它无权访问的系统区域。

可扩展性

Java 程序支持用其他语言(通常是 C++)编写的方法,也就是其他语言中所谓的函数。这样,当用 Java 写的方法运行起来较慢时,编程人员完全可以使用其他语言编写出等效的函数来代替这个 Java 方法。Java 的方法和程序之间的链接是动态的,即二者在程序运行时才连在一起。这是因为 Java 语言更注重速度,而方法在这方面不是必需的。

易懂性

Java 语言是建立在使用了多年的技术上的,因而使用过现代编程语言(如 C++)的编程人员可以很容易理解并掌握 Java。

1.3 Java 程序

前面提到过,Java 能够创建两种类型的程序:小应用程序(applet)和独立应用程序(stand-alone application)。小应用程序只是网页中的一部分,就象是一幅画或是一行字而已。一个浏览器显示 HTML 文档中的一幅图像就相当于一个支持 Java 的浏览器运行一个小应用程序。支持 Java 的浏览器加载 HTML 文档的同时,小应用程序也被加载和运行。使用小应用程序,可以做任何想做的事,从在网页中加入动画效果到开发出完整的游戏和工具都可以。现在已经开发出了一些小应用程序,包括:BarChart,嵌在一个 HTML 文档中的可配置的条表(bar chart), Crossword Puzzle,使用户能够解决网上纵横字谜问题;LED Sign 为网页浏览者提供一个滚动的、计算机化的消息。

图 1-3 显示的就是在 Netscape Navigator 2.0 上运行的电子表格小应用程序。

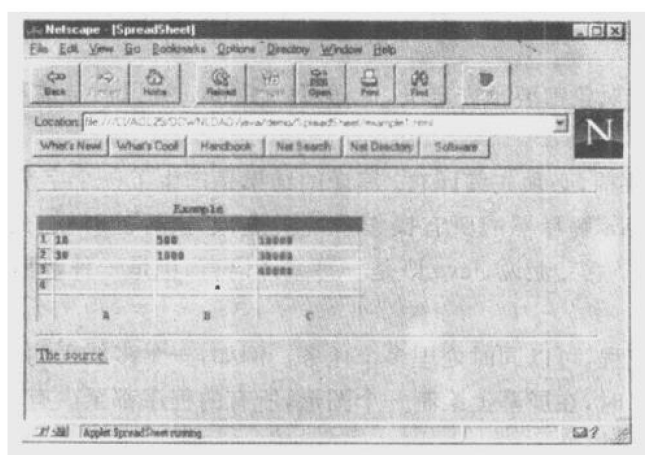


图 1-3 小应用程序是运行在 HTML 文件中的小程序



图 1-4 HotJava 是一个完全用 Java 编写的浏览器

除了可以编写小应用程序外,Java 还可以用来开发独立应用程序——一种不需要嵌在 HTML 文档中的应用程序,最著名的就是 HotJava 浏览器本身了(见图 1-4)。这个浏览器程序