

21 世纪 高等学校本科系列教材

总主编 吴中福

JAVA 程序设计指导

(37)

汪学明 钟 诚 编著



重庆大学出版社

JAVA 程序设计指导

汪学明 钟 诚 编著

重 庆 大 学 出 版 社

内 容 简 介

本书是 Java 程序设计指导书。全书共分三部分:第一部分简单介绍了 Java 程序设计的基础知识,包括 Java 语言的运行环境和 Java 特有的异常处理、多线程、网络通信及图形用户界面(GUI)编程方法;第二部分详细介绍了各种 Java 高级应用程序设计方法与技巧,内容涉及 Java 数据库连接技术(JDBC)、多媒体技术、组件技术(Java Bean)、服务器技术(Java Servlet)、JSP 技术、分布式应用系统开发技术及多种语言混合编程技术等;第三部分提供了两个典型的综合应用程序设计实例。

书中配有大量范例,内容新颖,实用性强,读者通过对书中内容及范例的学习,可以快速掌握 Java 语言程序设计技术。本书可作为计算机科学与技术及其相关专业 Java 程序设计的指导书,也可供广大 Java 程序设计爱好者和有关工程技术人员使用。

图书在版编目(CIP)数据

JAVA 程序设计指导/汪学明,钟诚编著. —重庆:重庆大学出版社,2001.9

计算机科学与技术专业本科系列教材

ISBN 7-5624-2331-8

I. J... II. ①汪...②钟... III. JAVA 语言—程序设计—高等学校—教材 IV. TP312

中国版本图书馆 CIP 数据核字(2001)第 059819 号

JAVA 程序设计指导

汪学明 钟 诚 编著

责任编辑 曾显跃 吴德银

*

重庆大学出版社出版发行

新华书店经销

重庆大学建大印刷厂印刷

※

开本:787×1092 1/16 印张:18 字数:449 千

2001 年 9 月第 1 版 2001 年 9 月第 1 次印刷

印数:1—5 000

ISBN 7-5624-2331-8/TP·291 定价:25.00 元

前言

Java 语言由于具有面向对象、平台无关、多线程和面向网络等特性而受到世人瞩目。Java 为用户提供了良好的程序设计运行环境,特别适合 Internet 的应用开发,已成为网络时代最重要的程序设计语言之一。

随着 Java 应用领域的不断扩大,Java 新技术层出不穷,如 Java 组件技术、Java 多媒体技术、Servlet、JSP 和 Java 分布式应用系统开发技术等。Java 的开发工具包也由 JDK1.0.1 发展到了 JDK1.2.2,同时还出现了一些新的开发工具包,如 BDK、JSDK 和 JSWDK 等。为了适应 Java 发展的需要,我们在介绍各种程序设计方法与技巧时尽量采用 Java 新技术和新方法。如 JDBC 在 JDK1.0.1 版本中只是一个可选部件,到了 JDK1.1 公布时,SQL 类包(即 JDBC API)开始成为 Java 语言的标准部件。因此,我们在介绍 Java 数据库应用等 Java 高级应用程序设计方法与技巧时都采用了最新的 JDK1.2.2 开发工具。另外,我们还对一些新的开发工具如 BDK、JSDK 和 JSWDK 等的使用作了详细介绍。

全书共分三部分,第一部分(基础篇):包括第 1 章至第 5 章,简要介绍 Java 的开发运行环境和 Java 特有的异常处理程序、多线程程序、网络应用程序及图形用户界面程序的编程方法;第二部分(提高篇):包括第 6 章至第 12 章,详细介绍 Java 数据库连接技术(JDBC)、Java 多媒体技术、Java 组件技术(JavaBean)、Java 服务器技术(Servlet)、JSP 技术、Java 分布式应用系统开发技术及多种语言混合编程技术等 Java 高级应用程序设计方法与技巧;第三部分(实践篇):包括两个典型的 Java 综合应用程序设计实例和附录,供实际编程时参考。

本书以介绍 Java 程序设计方法与开发技巧为主要目的,书中配有大量范例,内容新颖,实用性强,通过对书中内容及范例的学习,读者可以很快掌握 Java 程序设计方法和技巧。因此,本书既可作为专业程序设计人员的指导书,也可供广大 Java 程序设计爱好者使用。

本书由汪学明主编。全书共 13 章,其中第 3 章和第 5 章由钟诚编写,其余部分由汪学明编写。全书的审校及程序调试工作由汪学明负责完成。另外,在本书的编写过程中得到了汪萍的大力支持,在此深表感谢。

由于水平和时间有限,书中难免有错误和不妥之处,恳请广大读者批评指正。

编 者
2001 年 5 月

目 录

第 1 篇 基础篇

第 1 章	Java 运行环境及开发工具	1
1.1	Java 程序的运行环境	2
1.2	Java 开发工具包(JDK)	5
1.3	Java 第三方开发工具	8
1.4	Java 应用程序和小应用程序	14
1.5	Java2 的主要特性	16
第 2 章	Java 异常处理程序设计	18
2.1	基本概念	18
2.2	异常分类	19
2.3	异常处理	19
2.4	异常处理程序设计举例	22
第 3 章	Java 多线程程序设计	28
3.1	线程的基本概念	28
3.2	多线程程序设计的特点与方法	28
3.3	线程的优先级	30
3.4	多线程并发同步程序的设计与实现	31
3.5	多线程应用程序实例的设计与实现	34
第 4 章	Java 网络程序设计	37
4.1	Java 网络通信机制	37
4.2	Socket 通信程序设计	38
4.3	数据报通信程序设计	46
4.4	URL 网络资源访问程序设计	49
第 5 章	Java 图形用户界面设计	52
5.1	Java 抽象窗口工具集简介	52

5.2	AWT 绘图程序设计实例	54
5.3	AWT 的事件处理机制	62
5.4	Java 图形用户界面程序设计实例	64

第2篇 提高篇

第6章	Java 数据库应用程序设计	77
6.1	Java 数据库连接(JDBC)简介	77
6.2	JDBC 应用程序接口	80
6.3	使用 JDBC 访问数据库的机制	83
6.4	Java 数据库应用程序设计举例	87
第7章	Java 多媒体程序设计	97
7.1	多媒体技术概述	97
7.2	图像处理	98
7.3	声音处理	104
7.4	动画技术	111
7.5	多媒体合成技术	120
7.6	Java Media APIs 简介	127
第8章	Java 组件程序设计——JavaBean	132
8.1	概述	132
8.2	JavaBean 的设计特点	135
8.3	BDK 和 Bean 的创建与测试	136
8.4	用 Java Studio 开发 JavaBean 组件	144
8.5	JavaBean 程序设计举例	149
第9章	Java 服务器程序设计	154
9.1	Java Servlet 技术简介	154
9.2	Servlet 的特点及应用范围	156
9.3	Java Servlet 开发工具包(JSDK)	157
9.4	Servlet 程序设计举例	159
第10章	JSP 应用程序设计	168
10.1	JSP 技术简介	168
10.2	JSP 的运行环境及配置	171
10.3	JSP 常用指令	173
10.4	JSP 程序设计举例	177
第11章	Java 分布式应用系统程序设计	184
11.1	Java 远程方法调用	184
11.2	Java RMI 分布式应用程序设计	185

11.3	Java 接口定义语言	193
11.4	Java IDL 分布式应用程序设计例	194
11.5	Java RMI 与 Java IDL 的比较	204
第 12 章	多种语言混合编程	205
12.1	Java 与 C 的连接	205
12.2	Java 与 C 连接的实现	207
12.3	Java 与 C 的参数传递	209
12.4	Java applet 与 HTML 文件的连接	211
12.5	Java applet 与 HTML 文件参数的传递	213
12.6	Applet、Application 和 HTML 混合编程	215

第 3 篇 实践篇

第 13 章	Java 综合应用实例	219
13.1	基于 Client/Server 的电话号码本	219
13.2	聊天室	231
附录		258
附录 A	Java 基本语法	258
附录 B	Java 标准 API 类库	262
附录 C	Visual J++ 6.0 类库	266
附录 D	Java 编程中应注意的问题	278
参考文献		280

第 **1** 篇

基础篇

第 **1** 章

Java 运行环境及开发工具

Java 语言是 Sun MicroSystem 公司开发的一种新型的计算机高级语言。由于它采用了面向对象程序设计技术,而且具有健壮、安全、多线程、高度可移植等许多优点,所以很快风靡全球,成为网络编程的首选语言。本章主要介绍 Java 程序的运行环境、Java 开发工具包(JDK)的安装与使用、Java 第三方开发工具及 Java 应用程序和小应用程序的开发,并简单地介绍了 Java 2 的一些新特性。

1.1 Java 程序的运行环境

1.1.1 Java 技术及其应用前景

(1) Java 技术概述

在计算机的发展史上,很少有一种语言能够像新一代的编程语言 Java 那样如此快速地产生产深远的影响。现在 Java 已经不仅仅是一种编程语言,它还创造了一种全新的计算方式——网络计算。

Java 作为一门日趋成熟的新技术,它主要包括以下内容:

①Java 软件技术:Java 语言及其编程技术、Java 操作系统 (JavaOS)、Java 开发环境与开发工具、Java 数据库接口 JDBC、Java 服务器程序 (Java servlet)、Java 组件 (bean) 技术、JSP 程序设计、Java 多媒体程序设计、Java 软件包等等;

②Java 虚拟机及其规范;

③Java 嵌入技术:Java 芯片、基于 Java 技术的新型计算机系统 (网络计算机 NC、基于网络计算的计算机 NetPC、网络计算机体系结构 NCA、网络工作站等);

④Java 计算:基于 Java 处理功能的信息综合处理系统。

(2) Java 技术的应用前景

由于 Java 本身所具有的特性,以及各界人士对 Java 技术的认可,其应用前景极为宽广。大体上可以从以下几个方面考虑其应用:

①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、分析等;

②计算机过程的可视化、可听化、可操作化软件的开发;

③动态画面的设计,包括图形、图像的调用;

④交互操作的设计 (选择交互、定向交互、控制流程等);

⑤Internet 的系统管理功能模块的设计,包括 Web 页面的动态设计、多媒体设计、管理和交互操作设计等;

⑥Intranet 上的软件开发 (直接面向企业内部用户的软件开发);

⑦与各类数据库连接查询的 SQL 语句的实现;

⑧电子商务;

⑨分布式应用系统的开发;

⑩家用电器的嵌入技术 (基于 Java 芯片的技术、尤其是 WebTV 或 InternetTV 的开发使用);

⑪其他应用类型的程序。

1.1.2 Java 技术的特点

Java 语言是一种简单的、面向对象的、分布式的、解释型的、健壮的、安全的、结构中立的、可移植的、高效的、多线程的以及动态的计算机语言。Java 之所以能取得今天的成功,是由于它具有以下突出特点:

(1) 简单性

简单性是指 Java 语言是一种简单易学的语言,它继承了 C、C++ 语言中大量语法成分,因此,Java 的程序风格与 C、C++ 语言非常相似;另一方面,Java 语言又对 C、C++ 语言做了一定的简化,抛弃了很多容易引起混乱的地方,如头文件、预处理器、指针、结构、联合、隐式的类型转换、操作符重载等,这样就使 Java 程序更加简明,且程序的可读性更强。

(2) 面向对象

Java 是一种完全面向对象的程序设计语言,它除了数值、布尔和字符三种基本数据类型外,其他类型都是对象。Java 语言不再支持过程式程序设计方法,也不再支持全局变量。其最基本的模块单位是类,由类来定义程序中各种对象的状态和行为。Java 支持面向对象程序设计方法的全部特性,如数据封装、继承、多态性等。但是它不支持多重继承,即每个类只能从一个类中继承。在 Java 中要实现多重继承功能,可以通过接口编程来解决。

(3) 可移植性

Java 自己定义了一套虚拟机,以及虚拟机上所使用的字节码。字节码非常类似机器指令,但它不是为某个特定机器定义的,因此,一般不能在某个平台上直接执行,而须要由 Java 运行系统中的解释器来解释执行,即首先由编译器将 Java 程序编译成字节码,然后再由 Java 运行系统解释执行。可见,Java 语言的程序设计并不依赖于具体的软硬件环境。因而,Java 程序具有非常好的可移植性。

(4) 稳定性和安全性

Java 语言定位于 Internet 网络编程,因此,它是一种在分布式环境下使用的语言。分布式计算环境要求软件具有高度的稳定性与安全性。为了保证稳定性,Java 采取了三项措施:①不支持指针数据类型;②提供数组下标的检查机制;③提供内存管理机制。

为了保证安全,Java 除了在字节码传输过程中使用公开密钥加密机制(PKC)外,还提供了四级安全性保障机制,字节码验证、类装载器、运行时内存布局 and 文件访问机制等。

(5) 结构中立性

Java 语言的设计不针对某种具体结构,采取了结构中立。除了字节码外,Java 还规定了统一的语言文本,保证其基本语法不会随着开发环境的变化而变化。另外,Java 语言还提供了一个用于访问底层操作系统功能的可扩展类库,以确保 Java 应用程序能运行在任意支持 Java 语言的平台上。

(6) 高效能

Java 编译器先将程序编译成为与机器指令非常接近的字节码,它可以非常容易地对应于机器指令,从而可以大大提高执行效率。另外,为了提高程序性能,在 Java 程序中还采取了下列措施:引入多线程机制;对程序进行及时编译,直接将源代码编译成可执行程序;在程序中嵌入 C 代码等。

(7) 动态性

Java 的动态性是其面向对象设计的延伸。Java 程序的基本结构单元是类,而 Java 的类又是在运行中动态装载的,因此,在 Java 类中可以自由加入新的方法和成员变量而不会影响程序的执行,这样就使得 Java 语言可以在分布式环境中动态地维护应用程序和其支持类库之间的一致性。Java 的动态性还体现在使你能够真正拥有“即插即用”的软件模块功能。

(8) 分布性

分布性主要是指数据分布和操作分布。数据分布是指数据可以分散存放在网络的不同主机上;操作分布则把计算分散在不同主机上进行处理。Java 应用程序接口(API)中包含了支持 HTTP 和 FTP 等 TCP/IP 协议的子类库,这样就使用户在 Java 应用程序中通过 URL 地址可以直接访问网络上的对象,其访问方式和访问本地文件系统几乎一样。

1.1.3 Java 运行环境和 Java 虚拟机

Java 语言具有平台无关性,它的目标代码是字节码,而不是针对某种软硬件环境的机器指令,因此在执行时需要将字节码转换成为具体环境下的机器指令。另外,在 Java 语言中,内存的布局和分配要在运行时确定,在字节码中只是采用符号来引用内存,因此在运行时要将符号具体解释为实际的内存地址。出于这两方面的原因,在 Java 中引入了运行系统的概念,由运行系统负责装载程序运行时所需要的类并安排程序运行时的内存布局。

(1) Java 运行系统

Java 语言主要提供了两种形式的应用程序:Java application 和 Java applet。这两种应用程序的执行过程稍有不同。Java application 拥有自己的程序入口,它先由 Java 编译器编译成为,独立于平台的字节码(Byte Code),而后由 Java 解释器来运行,因此其运行系统为 Java 解释器。而 Java applet 没有自己的程序入口,因此不能独立运行。其执行过程如下:

首先由 Java 编译器将其编译为字节码,并把字节码嵌入到 Web 页面中;在客户端将字节码下载,并由 Web 浏览器来控制执行。因此,其运行系统是 Java 兼容的 Web 浏览器。

Java applet 的执行过程如图 1.1 所示。

Java 运行系统一般都包括以下几个组成部分:类装配器、字节码校验器、解释器、代码生成器和运行支持库。其基本结构如图 1.2 所示。

Java 字节码的执行过程可以分为三步:装入字节码、校验字节码和执行字节码。

1) 装入字节码

装入字节码的工作由类装配器来完成。类装配器负责装入程序运行所需要的代码,包括程序代码中调用到的所有类。这些类都被安放在自己的名字空间中,不会对其他类所在的空间造成影响。在装人类之后,运行系统就可以确定程序执行的内存布局,即建立符号引用和具体内存地址之间的查找表。

2) 校验字节码

由字节码校验器对字节码进行检查,以确保所装入的字节码不违背 Java 语言的安全性规则。

3) 执行字节码

在经过了装入和校验两个步骤之后,字节码就可以提交运行了。在 Java 语言中,字节码的运行有两种方式:编译方式和解释执行方式。通常采用解释执行方式,即由 Java 解释器通过每次翻译并执行一小段代码来完成字节码程序的所有操作。但是,如果对程序的运行速度有较高的要求,就应该采用编译方式,即由代码生成器先将整个程序转化为机器指令,而后全速执行。

(2) Java 平台

由于 Java 字节码在执行时须要有 Java 运行系统的支持,因此,虽然 Java 运行系统是建立

在不同的平台上,但是为了做到 Java 程序的可移植性,对 Java 运行系统的功能要求是统一的,即该系统可以看做是一个与具体的软硬件环境无关的“Java 平台”。

Java 平台是 Java 语言编程和操作环境的基础,它由两个部分组成:Java 虚拟机(JVM)和 Java 应用程序接口(API)。

1) Java 虚拟机

Java 虚拟机是 Java 平台的核心部件,它可以看做是一个想象中的、能运行 Java 字节码的操作平台。

2) Java 应用程序接口

Java 应用程序接口是供 Java 应用程序使用的与 JVM 的标准接口,它又可以分为 Java 基本 API 和 Java 标准扩展 API 两大部分。Java 标准扩展 API 包含了面向不同领域和范畴的 API 接口规范,它们随着 Java 语言的发展而不断完善,并将逐渐成为 Java 基本 API 的组成部分。

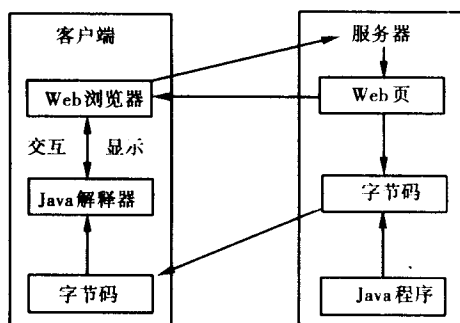


图 1.1 Java applet 的执行过程

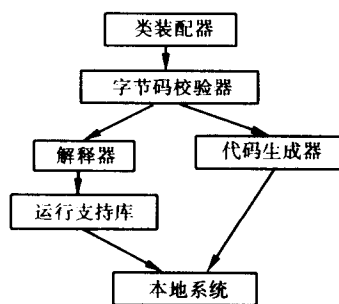


图 1.2 Java 运行系统的基本结构

1.2 Java 开发工具包 (JDK)

Sun Microsystems 公司在推出 Java 语言的同时,推出了一套 Java 的开发工具包(JDK)。JDK 提供了 Java 的编译器和解释器等开发程序时所必需的工具及可供 Java 程序设计人员使用的一些标准包及程序范例。

要编写 Java 程序必须先获得并安装 JDK,可以从 <http://java.sun.com/j2se/> 站点免费下载 JDK。JDK 开发工具包有 UNIX、Windows98/Windows NT、Mac 等多种版本,版本间功能相差不大。

1.2.1 JDK 的安装

JDK 在不同系统下的安装过程不尽相同,以 Windows98/NT 下的 JDK 为例,介绍 JDK 的安装过程。

首先,从 Sun 公司的 Web 站点上下载 Windows 版本的 JDK,得到一个自解压的程序;

然后,选择安装的目标盘和安装目录,并将下载的 JDK 自解压程序拷入,执行该程序即可把 JDK 解压并安装到硬盘上,并创建 Java 目录及其子目录;

最后,在 AUTOEXEC. BAT 中加入以下语句(假定 Windows 目录为 C:\Windows,JDK 安装

在 C:\JAVA) 并重新启动计算机。

```
PATH C:\WINDOWS;C:\WINDOWS\COMMAND;C:\JAVA\BIN
SET INCLUDE = C:\JAVA\INCLUDE\;C:\JAVA\INCLUDE\WIN32% INCLUDE%
SET LIB = C:\JAVA\LIB% LIB%
SET CLASSPATH = C:\JAVA\LIB\CLASSES.ZIP
```

安装完成后,在 JDK 的主目录 C:\JAVA 中包含以下文件和子目录:

C:\java\COPYRIGHT	JDK 版本说明
C:\java\index.html	JDK 的 HTML 介绍文档
C:\java\README	JDK 基本内容介绍及功能介绍
C:\java\src.zip	JDK 程序源代码压缩文件
C:\java\bin	包含了一些常用的 JDK 工具,如 javac、java、jdb、javah、javap、javadoc、appletviewer 等。

C:\java\lib 包含了一些在执行 JDK 可执行文件时所要用到的函数库。注意其中的 Classes.zip 不要解压,Java 编译器和 Java 解释器会直接使用这个压缩过的函数库文件。

C:\java\include 包含了一些与 C 连接时所需头文件

C:\java\demo 包含了许多 Sun 公司已经完成的 Java 小应用程序(applet)范例。

1.2.2 JDK 开发工具介绍

(1) Java 编译器(javac)

javac 是 Java 语言的编译器,它能将用 Java 语言编写的程序源代码编译成字节码,并保存于 class 文件中。

格式:javac [选项] 文件名.java

如:c:\java\bin\javac Example1.java

注意:javac 编译器要求待编译的文件具有.java 扩展名,并且区分文件名的大小写。

(2) Java 解释器(java)

java 命令用于解释执行 Java 字节代码。因为编译后的 Java 字节码是不能在机器上直接运行的,必须由 Java 解释器对其解释执行。

格式:Java [选项] 类名<参数表>

如:c:\java\bin\java HelloWorld 运行后在屏幕上输出“Hello World!”。

(3) Java 语言调试器(jdb)

jdb 命令用于调试 Java 语言编写的程序。

格式:jdb [选项]

jdb 是一个类似于 dbx 的 Java 语言类命令行调试器,通过使用 Java DebuggerAPI 能够对本地的或远程的 Java 解释器进行调试。

启动 jdb 的方法有两种:常用的方法是用 jdb 解释执行被调试的类,格式与由 Java 解释器执行相类似;第二种方法是把 jdb 附加到一个已运行的 Java 解释器上,该解释器必须是带 -debug 项启动的。

如:jdb-host<主机名>-passwd<口令>

则口令在 Java 解释器带 -debug 选项运行后显示。

(4) C 头文件和源文件生成器(javah)

javah 命令用于从一个 Java 类生成实现 native 方法所需要的 C 头文件和 C 源程序文件,用作 Java 语言与 C 语言之间的连接。

格式:javah [选项] 类名...

(5) Java 类文件反汇编器(javap)

javap 用于对类文件进行反汇编。

格式:javap [选项] 类名

假如想了解 HelloWorldApp.class 类中的情况,可以用 javap 命令:

c:\java\bin\>javap-c-v HelloWorldApp

(6) Java API 文档生成器(javadoc)

javadoc 从 java 源文件生成 HTML 格式的 API 文档。

格式:javadoc [选项] 包 class1.java class2.java

如:c:\java\bin>javadoc-version 1.02 HelloWorldApp.java

(7) Java 小应用程序浏览器(appletviewer)

appletviewer 使用户不需要使用 WWW 浏览器就可以运行 Java 语言编制的 Applet。

格式:appletviewer [选项] urls

如:c:\java\bin>appletviewer HelloWorldApplet.html

1.2.3 Java 档案文件

(1) Jar 文件的功能

Java 档案文件 Jar(Java Archive)可以把多个文件捆绑在一起作为一个文件,典型地,一个 Jar 文件包含几个类文件和一些 Java 应用程序或 applet 用到的资源文件,Jar 文件格式在 JDK1.1 被引入。在 Java2 中,Jar 文件的功能得到进一步的扩充。

Jar 文件是一项意义重大的 Java 技术,它提供了许多功能:

①安全性 可以在 Jar 文件中加上数字化签名,用户可以在识别数字化签名之后对 Jar 文件进行安全授权,允许访问其私有资源或系统功能。

②减少了下载时间 如果 applet 及其支持文件捆绑在一起,则 applet 类文件和相关的资源文件可以在一次 HTTP 事务处理过程中下载完,在 Jar 文件出现之前,applet 的每一个支持文件都需要分别传送,分别带来一次 HTTP 事务处理的开销,从而使具有多个类和资源的 applet 开销很大。

③压缩 Jar 格式支持对文件的压缩。Jar 文件所用的压缩方法也是常见的 WinZip 等应用程序中所用的方法,因此,可以像对 .zip 文件一样进行压缩、建档、解压缩等操作。

④用于扩展 Java 平台(Java2 的功能) 利用 Jar 文件格式和扩展框架可以对 Java 核心平台的功能进行扩展,如 Java 3D 和 JavaMail 就是 Sun 公司提供的扩展功能大的例子。

⑤包封装(Java2 功能) 以 Jar 文件格式保存的包(package)可以被封装以便能保持版本上的一致性,把包封装在 Jar 文件中指的是在一个包中定义的所有类必须放在同一个 Jar 文件中。

⑥包的版本(Java2 功能) 一个 Jar 文件可以包含一些和其包含的文件有关的信息,如开发商信息、版本号信息等。

⑦可移植性 处理 Jar 文件的机制是 Java 平台核心 API 的标准部分。

(2) Jar 工具的使用

为了使用 Jar 文件格式的一些基本功能,可以用 Java 开发工具包中提供的 Java 档案工具。由于 Java 档案工具以 jar 命令行的形式调用,Java 档案工具又称为 jar 工具。

jar 工具的语法如下:

jar options Files

命令行中 options 是 jar 工具的选项,根据不同的选项,jar 工具执行不同的功能,Files 是 jar 工具的操作对象。表 1.1 给出了 jar 工具所提供的一些主要功能。

需要注意的是,Jar 工具与大多数 Java 开发工具包中提供的工具是不同的,jar 工具在指定选项时不需要使用“-”符号。如:

jar cvf example.jar example.class simple image

其中 simple 和 image 是目录。Jar 工具把这些目录下的所有文件及 examples.class 文件一起放入 jar 文件中,生成 example.jar 文件。

表 1.1 jar 工具的一些主要功能及相应的命令

功 能	命 令
创建一个 Jar 文件	jar cf jar-file input-files(s)
浏览 Jar 文件中的内容	jar rf jar-file
抽取 Jar 文件中的内容	jar xf jar-file
抽取 Jar 文件中的指定文件	jar xf jar-file archived-file(s)
运行打包在 Jar 文件中的应用程序(JDK1.1)	jre-cp app.jar MainClass
运行打包在 Jar 文件中的应用程序(Java2,需要有声明文件的初始部分及 MainClass)	jre-cp app.jar MainClass
调用打包在 Jar 文件中的 applet	<applet code = AppletClassName.class archive = "JarFileName.jar" width = width height = height> </applet>

1.3 Java 第三方开发工具

1.3.1 Java WorkShop 简介

Java WorkShop 1.0 是 Sun 公司 1996 年推出的一组跨平台的开发工具,它拥有非常多的优点,便于用户开发应用程序。其界面如图 1.3 所示。

(1) 特点

Java WorkShop 是一个价格低廉但又应用极广的开发软件。它为各个层次、各个领域的用户提供了舒适的环境。Java WorkShop 就像 Web 浏览器一样,非常便于使用。在其漂亮的集成环境中可以编辑、编译、测试和调试程序,还可以在网上实现应用程序的共享。

Java WorkShop 完全用 Java 语言开发而成。其开发环境本身就结合容纳了一个网络浏览器,其中的每个工具都是一个小应用程序。

另外,通过集成环境中的浏览器可以直接到 Sun 公司的大本营以及世界各地去,共享许多应用程序,还可以通过 Java WorkShop 和超文本链接来查看更多的网络信息。

(2) 组成

Java WorkShop 是一个完整的用于开发 Java 应用程序和 WWW 主页的工具。它具体包括以下几部分:

1) 文件夹管理器 (Portfolio Manager)

文件夹管理器提供了一种迅速、直观的图形方式,使用户直接面对 Project 进行工作。事实上,portfolio 就是 Project 的存储处,在第一次运行 Java WorkShop 时,系统会为每个用户自动建立一个名为 Personal 的文件夹。

2) 项目管理器 (Project Manager)

项目管理器提供了编译、浏览、调试一个项目的集成环境。用户通过它可以建立或修改自己的项目。

3) 资源编辑器 (Source Editor)

资源编辑器提供了对具体程序进行编辑的界面,除了具有绝大多数的文本编辑器都具有的功能以外,它还提供了一些新的工具以更好、更快地编写程序。

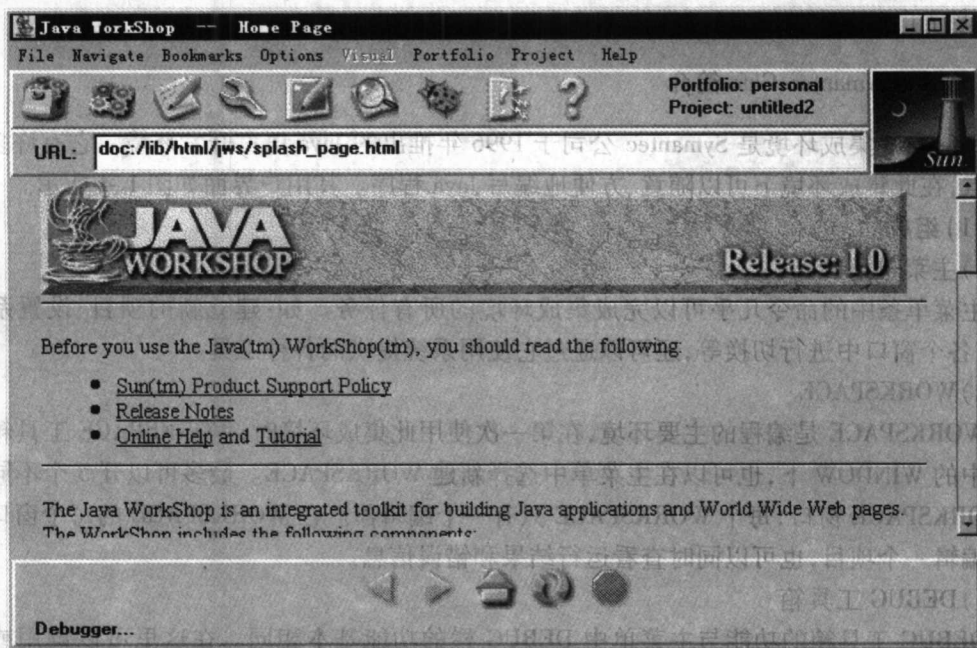


图 1.3 Java WorkShop 用户界面

4) 编译管理器 (Build Manager)

无论是编译一个项目,还是编译一个单独的文件,利用 Java WorkShop 的编译器都会使人感觉更加轻松和方便。

5) 可视化 Java (Visual Java)

利用 Visual Java 不仅为 Applet 也可以为单独的 Project 建立图形用户接口 (GUIs)。