

高等院校
计算机教材系列

施霞萍 张欢欢 王瑾德 马可幸 编著

Java

程序设计教程



机械工业出版社
China Machine Press

高等院校
计算机教材系列

施霞萍 张欢欢 王瑾德 马可幸 编著

Java

程序设计教程



机械工业出版社
China Machine Press

Java 程序设计语言及编程技术是随互联网的发展而被推广的,它是目前国内外广泛应用的程序设计语言。本书由浅入深、循序渐进地介绍 Java 程序设计的基本概念、方法和应用。内容包括结构化程序设计、面向对象程序设计、图形用户界面开发、错误处理机制、多线程设计、输入输出流等 Java 技术和应用。本书把概念和方法与详细的程序代码紧密地连接起来,并结合图示,使读者易学易用。本书可以作为高等院校各专业学生学习 Java 程序设计的教材和参考书,也可以作为自学教材。

本书提供学习配套材料,可以到下列网址下载: www.hzbook.com。

版权所有,侵权必究。

图书在版编目(CIP)数据

Java 程序设计教程/施霞萍等编著. —北京:机械工业出版社, 2004.9

(高等院校计算机教材系列)

ISBN 7-111-14950-5

I .J… II .施… 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(2004)第 076852 号

机械工业出版社(北京市西城区百万庄大街 22 号 邮政编码 100037)

策划编辑:温莉芳

责任编辑:华 章

北京瑞德印刷有限公司印刷·新华书店北京发行所发行

2004 年 9 月第 1 版第 1 次印刷

787mm×1092mm 1/16·18 印张

印数:0 001—5 000 册

定价:28.00 元

凡购本书,如有倒页、脱页、缺页,由本社发行部调换
本社购书热线:(010)68326294

前 言

Java 程序设计语言及编程技术是随着互联网的发展而被推广应用的,它是目前国内外广泛使用的程序设计语言。Java 语言是面向对象技术成功应用在程序设计语言中的著名典范。它的内容和特点与以往的其他程序设计语言有很大的不同。面向对象技术被认为是程序设计方法学的一场革命,它已经逐步替代面向过程的程序设计技术,从而成为计算机应用开发领域的主流趋势。

由于 Java 程序设计进入我国高校计算机基础系列课程时间较短,而我国的高等教育正从“精英教育”发展为“大众化教育”,在这种形势下,为推动 Java 程序设计语言的应用普及,同时为配合《上海市高等学校计算机等级考试(二级)》在 2003 年秋推出的 Java 程序设计考试科目,在上海市教育考试院的组织领导下,由多位具有多年丰富教学经验的教师集体编著了本教材。本教材紧密配合上海市高等学校计算机等级考试的 Java 考试大纲,由浅入深、循序渐进地介绍 Java 程序设计语言及应用的基本概念和基本方法,在内容的选择和编排上力争主次分明,避免烦琐细节的罗列,但又不忘记把 Java 的一些先进内容和思想方法介绍给大家,使读者通过有限的篇幅便能比较完整地掌握 Java 程序设计思想和方法。要学好程序设计语言只是看书是不够的,另一个重要步骤是动手实践。为此,本书在讲解概念和方法的同时,配有大量具有说服力和浅显易懂的例子进行演示,读者在使用本书时千万不要忘了上机练习这一过程。你可以直接从 www.hzbook.com 下载书本上的例子来加深对知识点的理解和认识,也可以使用结合本书的内容所出的练习题进行实践,我们还提供练习题的参考答案帮助读者学习(练习题及其答案可从 www.hzbook.com 下载,本书由于篇幅所限并未包含这部分内容)。

全书共分十章,前三章主要涵盖了 Java 在结构化程序设计中的基本概念和应用。通过前三章,没有学过任何程序设计语言的初学者将学会程序设计语言共同具有的概念和知识。第 4 章全面讨论了面向对象程序设计的思想方法以及在 Java 语言中的实现。通过这部分的学习,读者对面向对象程序设计的思想在 Java 中的应用就会有比较完整的认识。第 5 章介绍了小应用程序和用 AWT、Swing 组件进行图形用户界面设计的方法。其中 Swing 组件在技术上要比 AWT 组件先进。完成这部分的学习将使读者具备设计专业化图形用户界面的能力。第 6~8 章分别介绍了 Java 特有的错误处理机制、多线程设计以及输入输出流的实现方法。掌握这些 Java 所具有的功能将为设计出完整、实用的程序提供保证。最后两章讨论了 Java 在网络和多媒体方面的应用。这既是 Java 综合应用的举例,也是实用性的体现。这将进一步提高学生学习 Java 的兴趣。

为了更好、更方便地学习和掌握 Java 程序设计方法,除本教材外,我们还提供教材中所有的程序代码、练习题和练习答案,可以到下列网址下载:www.hzbook.com。为帮助教师的教学,我们将提供教学课件演示,需要者直接与作者联系,邮件地址为 xpshi@mail.dhu.edu.cn。

本书由施霞萍主编,张欢欢、王瑾德、马可幸参加编著,张增林参与了教材资料收集和程序调

试工作,最后由施霞萍通审、修改定稿。机械工业出版社温莉芳等同志对本书的出版给予了充分重视和周到的安排,使本书能在短时间内出版。对所有曾经鼓励、支持和帮助过我们的领导、组织、朋友,在此表示真挚的感谢。

尽管本书作者都有十年以上的大学教龄,积累了程序设计方面的教学经验,但由于时间仓促和水平有限,书中难免有不妥之处,竭诚欢迎读者多提宝贵意见。

施霞萍

2004年7月

作者介绍



施霞萍 1982年和1984年先后毕业于东华大学(原中国纺织大学)电气自动化专业(本科)和工业自动化专业(研究生)。现为东华大学计算机学院副教授、硕士生导师、计算机基础教学部副主任。主要研究方向包括网络及数据库系统集成、数据挖掘、数字图像处理等。主讲数字信号处理、Java程序设计与应用、VB程序设计、计算机组成与网络等课程。作为项目主要参加者,承担和完成了多项国家级和上海市级的科研项目。曾获国家纺织工业局三等奖。目前兼任上海市高等学校计算机等级考试命题组成员。



张欢欢 1990年毕业于中国人民解放军国防科技大学三系,同年到上海华东理工大学计算中心任教。1994年开始在华东理工大学计算机系攻读在职研究生并于1997年毕业获得硕士学位。2002年开始攻读在职博士。主要研究方向是形式化方法与验证技术、通用程序设计。在教学上主要从事程序设计语言的教学。



王瑾德 1983年毕业于上海复旦大学数学系。现任上海中医药大学现代教育技术中心、计算机教学部教师。曾经开设高等数学、数理统计学、模糊数学等课程。1992年转向计算机应用教学,曾经开设的课程有:计算机文化基础(初级、中级)、Pascal程序设计、C语言程序设计、Visual Basic程序设计、Java程序设计以及Web设计与应用等。曾经参与《现代医学信息学》、《高校计算机等级考试实用指南》等教材与辅导书的编写工作。



马可幸 1987年毕业于同济大学。现为上海东华大学计算机科学与技术学院讲师,从事计算机基础教学,讲授程序设计和多媒体技术等课程。感兴趣的研究方向是面向对象程序设计技术和网络多媒体教学。曾参加教育部多媒体教学课件项目的研制工作。

目 录

前言	
作者介绍	
第 1 章 Java 程序设计概述	1
1.1 Java 与程序	1
1.1.1 概述	1
1.1.2 Java 的发展	2
1.2 Java 的特点	3
1.3 Java 程序介绍	3
1.3.1 Java 虚拟机	3
1.3.2 Java 开发工具	4
1.3.3 SDK 目录结构	4
1.3.4 Java 2 开发工具简介	5
1.3.5 Java 的 API 类库与 API 文档	5
1.3.6 Java 程序结构	7
1.3.7 最简单的 Java 程序	8
1.4 Java 程序的开发过程	9
1.4.1 编辑源程序	9
1.4.2 编译源程序	10
1.4.3 运行 Java 程序	10
1.4.4 编写第一个小应用程序	11
1.5 Java 资源的获取与安装	12
1.5.1 下载 SDK	12
1.5.2 安装 SDK	15
1.5.3 设置 SDK 的操作环境	16
第 2 章 Java 程序设计的基本概念	19
2.1 常量值	19
2.2 变量与标识符	21
2.2.1 变量概述	21
2.2.2 变量的定义及赋值	22
2.2.3 变量的作用域	23
2.2.4 标识符	24
2.3 基本数据类型	25
2.3.1 Java 基本数据类型	25
2.3.2 基本数据类型的默认值	25
2.3.3 数据类型转换	26
2.4 运算符与表达式	28
2.4.1 赋值运算符与赋值表达式	28
2.4.2 算术运算符与算术表达式	28
2.4.3 关系运算符与关系表达式	30
2.4.4 逻辑运算符与逻辑表达式	30
2.4.5 复合赋值运算符	32
2.4.6 其他运算符	32
2.4.7 运算符的优先级与结合性	33
2.5 对象和方法的使用	34
2.5.1 对象的使用	35
2.5.2 方法的使用	39
第 3 章 Java 结构化程序设计	41
3.1 顺序结构及基本语句	41
3.1.1 注释语句	41
3.1.2 输入语句	41
3.2 选择结构	43
3.2.1 if 语句	44
3.2.2 if-else 语句	45
3.2.3 if-else if 语句	47
3.2.4 if 语句的嵌套	49
3.2.5 switch 语句	49
3.3 循环语句	52
3.3.1 for 循环语句	53
3.3.2 while 循环结构语句	54
3.3.3 do while 循环结构语句	55
3.3.4 循环结构语句的嵌套	56
3.4 转移语句	59
3.4.1 break 语句	59
3.4.2 continue 语句	60
3.4.3 return 语句	61
3.5 方法	61
3.5.1 Java 的程序模块化	61
3.5.2 方法的定义	62
3.5.3 参数的传递	62
3.5.4 作用域	63

3.5.5 递归	65	第 5 章 Java 图形用户界面	105
3.6 数组	67	5.1 小应用程序概述	105
3.6.1 一维数组的声明与引用	68	5.1.1 一个简单的小应用程序例子	105
3.6.2 数组的赋值	69	5.1.2 小应用程序的调试	107
3.6.3 一维数组程序举例	69	5.1.3 小应用程序的安全模型	111
3.6.4 二维数组的声明及引用	72	5.1.4 java. applet. Applet 类与其他 类的关系	112
3.6.5 数组的复制	74	5.1.5 小应用程序的生命期	112
3.6.6 字符串处理	75	5.2 java. awt 与图形用户界面	116
第 4 章 Java 面向对象程序设计	77	5.2.1 标签和文本域	116
4.1 面向对象程序设计的基本概念	77	5.2.2 Java AWT 中的事件处理机制	118
4.2 类的创建	78	5.2.3 按钮	124
4.2.1 类的声明格式	78	5.2.4 布局	124
4.2.2 成员变量	79	5.2.5 面板	126
4.2.3 成员方法	79	5.2.6 文本区域	128
4.3 对象的创建和使用	81	5.2.7 复选框和单选钮	129
4.3.1 创建对象	81	5.2.8 下拉列表	132
4.3.2 构造方法和对象的初始化	81	5.2.9 列表	134
4.3.3 对象的使用	83	5.2.10 窗口与菜单	137
4.3.4 对象的销毁	84	5.2.11 对话框	150
4.4 类的封装	84	5.3 Swing 与图形用户界面	153
4.4.1 封装的目的	84	5.3.1 Swing 的优点	153
4.4.2 访问权限的设置	85	5.3.2 Swing 类的继承关系	153
4.4.3 类成员(静态成员)	86	5.3.3 Swing 中的容器	155
4.5 类的继承	89	5.3.4 Swing 中的常用组件	169
4.5.1 继承的基本概念	89	5.3.5 Swing 中的事件概述	184
4.5.2 子类的创建	89	5.4 2D 图形设计	186
4.5.3 null, this, super 对象运算符	92	5.4.1 2D 图形的坐标系统	186
4.5.4 最终类和抽象类	94	5.4.2 字体	186
4.6 类的多态性	95	5.4.3 颜色	187
4.6.1 方法的重载	95	5.4.4 绘图	187
4.6.2 方法的覆盖	97	第 6 章 Java 异常处理	193
4.6.3 前期绑定和后期绑定	98	6.1 异常和异常对象	193
4.7 接口	99	6.2 异常的捕获与处理	193
4.7.1 接口的声明	99	6.3 try 语句的嵌套	196
4.7.2 接口的实现	99	6.4 throw 语句	198
4.8 包	100	6.5 throws 语句	200
4.8.1 Java 的类和包	100	6.6 使用异常处理的准则	201
4.8.2 引用 Java 定义的包	102	第 7 章 Java 多线程程序设计	203
4.8.3 自定义包	102	7.1 线程的概念	203
4.8.4 包和访问权限	104		

7.1.1 进程和线程	203	8.3.3 创建字符文件流对象并打开文件 ...	230
7.1.2 线程和多任务	203	8.3.4 字符文件流的读写	230
7.1.3 Java 对多线程的支持	203	8.3.5 字符文件流的关闭	230
7.2 线程的创建	204	8.3.6 字符缓冲流 <code>BufferedReader</code> 和 <code>BufferedWriter</code>	231
7.2.1 <code>Runnable</code> 接口	204	8.4 文件类	232
7.2.2 <code>Thread</code> 类	204	8.4.1 文件类 <code>File</code>	232
7.2.3 创建线程的方法	205	8.4.2 <code>File</code> 类的方法	233
7.3 线程的状态与控制	206	8.4.3 文件过滤	234
7.3.1 线程的状态	207	8.5 文件的随机读写	236
7.3.2 对线程状态的控制	207	8.5.1 <code>RandomAccessFile</code> 类	236
7.4 线程的优先级和调度	211	8.5.2 <code>RandomAccessFile</code> 的构造方法	236
7.4.1 线程的优先级	211	8.5.3 <code>RandomAccessFile</code> 的方法	236
7.4.2 线程的调度	211	第 9 章 Java 网络应用	239
7.5 线程组	211	9.1 网络的基本概念	239
7.5.1 线程组	211	9.1.1 IP 地址和端口号	239
7.5.2 <code>ThreadGroup</code> 类	212	9.1.2 URL	240
7.6 线程的同步	212	9.1.3 UDP 与 TCP	240
7.6.1 线程的同步机制	212	9.1.4 <code>Socket</code>	241
7.6.2 共享数据的互斥锁定	214	9.2 URL 的使用	241
7.6.3 数据传送时的同步控制	217	9.2.1 使用 URL 的方法	242
7.6.4 死锁	219	9.2.2 应用举例	244
第 8 章 Java 输入输出流	223	9.3 <code>Socket</code> 的应用	251
8.1 流的基本概念	223	9.3.1 TCP 套接字通信基本步骤	251
8.1.1 数据流	223	9.3.2 服务器端程序设计举例	253
8.1.2 Java 的标准输入输出	223	9.3.3 客户端程序设计举例	257
8.1.3 <code>java.io</code> 包中的数据流	225	9.4 网络安全处理	260
8.2 字节流和顺序文件的读写	225	9.4.1 Java 的安全特性	260
8.2.1 <code>InputStream</code> 和 <code>OutputStream</code> 类	225	9.4.2 缓存溢出	261
8.2.2 文件字节输入输出流	225	9.4.3 竞争状态	262
8.2.3 创建文件输入输出流对象并 打开文件	226	9.4.4 建立安全性策略	263
8.2.4 对文件进行读写的方法	227	9.4.5 安全基本原则	265
8.2.5 字节文件流的关闭	227	第 10 章 Java 多媒体应用	267
8.3 字符流	228	10.1 图像显示	267
8.3.1 <code>Reader</code> 和 <code>Writer</code> 类	228	10.2 动画设计	269
8.3.2 字符文件流 <code>FileReader</code> 和 <code>File-</code> <code>Writer</code> 类	229	10.3 声音播放	275

第 1 章 Java 程序设计概述

1.1 Java 与程序

1.1.1 概述

程序设计是伴随着电子计算机的出现而产生的一门技术。程序设计,简单地说,就是根据所提出的任务,写成一种能让计算机直接或间接接受并立即执行即完成该任务的语句的过程,整个任务所对应的一系列语句的集合即被称为一段程序。随着计算机科学的飞速发展,程序设计的方式和水平也在不断提高。程序设计所用的语言(称为程序设计语言)从低级语言发展到高级语言。与计算机硬件的发展一样,程序设计语言作为计算机软件的一部分也有它的几个发展阶段。

机器语言和汇编语言:在计算机问世的初期,人们直接使用计算机能够识别的二进制代码按一定的规则进行程序编写工作。这种用二进制代码表示的规则就是“机器语言”。由于机器语言表示形式的不直观,语义单一,从而给编写程序带来很大的麻烦,阻碍了计算机的广泛应用。为此软件设计者们用一些简单而又形象的符号来替代每一条具体的机器语言,这就形成了“符号语言”,也就是“汇编语言”。但对计算机来说,它无法直接识别“符号语言”,所以从汇编语言到机器语言,中间要有一个翻译过程,这由翻译程序——“汇编程序”来完成。机器语言和汇编语言是与具体所用的计算机(确切地说是与计算机指令系统)相关的,是为特定的机器服务的,所以称为面向机器的语言。

高级语言:人们在汇编语言的基础上,设想能否避开具体的机器,用一些接近于自然语言和数学公式的符号来描述自己的解题意图,能够通过各类机器对应的翻译程序即可以在各类机器上运行。这便出现了各种高级语言。20 世纪 80 年代初期,国内外比较通用的计算机语言有十几种,常见的且应用最普遍的有 BASIC、FORTRAN、ALGOL、COBOL、PL/1、PASCAL 以及 C 等。在支持十六位应用程序的 DOS 操作系统支持下,用如上所述的计算机语言所编的程序都是按事先设计的流程运行的,因而这些计算机语言被称为是面向过程的程序设计语言。与汇编程序作用一样,高级语言所编的程序必须翻译成机器语言才能使计算机执行,担负这一任务的程序称为“语言处理程序”。它分为两大类:解释程序和编译程序。解释程序逐句地接受所输入的源程序语言(源语言)书写的程序(源程序),然后逐句直接解释执行源程序,大家所熟知的 BASIC 和 APL 等会话型语言就是采用解释方法实现的。编译程序(有些书上也把它称为编译系统),是把用高级语言编写的面向过程的整个源程序翻译成目标程序的一种语言处理程序(目标程序即为机器语言构成的程序)。80 年代后期曾经应用相当广泛的 FORTRAN、PL/1、PASCAL 和 C 等语言皆采用编译方式实现。

20 世纪 90 年代,操作系统 Windows 9.x 以其新颖的图形用户界面、卓越的多任务操作系统性能和高层次的软件开发平台而迅速风靡全球,继续使用面向过程的程序设计语言在与用户

的交流过程中,其用户界面便显得有些“不合时宜”,特别是随着计算机网络的诞生和发展,面向对象的程序设计语言纷纷应运而生。面向对象的程序设计是近年来出现的程序设计技术,它以一种全新的设计和构造软件的思维方法,开拓了程序设计方法史上的新世纪,而Java程序设计便是其中一位佼佼者。

Java是美国Sun Microsystems公司于1995年5月正式发布的程序设计语言,它的前身是Sun Microsystems公司为智能消费类家用电器研究而开发的项目的一部分,但该项目的开发过程却并不顺利,直到1993年Web开始在Internet上盛行,开发小组试着将这一技术转移到Web网络上,没想到这一举动竟使Java在Internet上获得了空前的成功,使它成为了Web世界富有创造性的工具。时至今日,尽管Web开发人员仍然在利用Java使站点更生动、活泼,但它已远远超过了Web技术的范围。

1.1.2 Java的发展

Java的发展过程见表1-1。

表 1-1 Java的发展历史

时 间	描 述
1991 年	Sun Microsystems 公司进军消费电子产品 (IA) 市场
1991. 4	Sun 成立 “Green” 小组, 以 C++ 为基础开发新的程序设计语言, 并将其命名为 Oak
1992. 10	Green 小组升格为 First Person 公司, 他们将 Oak 的技术转移到 Web 上, 并把 Oak 改名为 Java
1993~1994	Web 在 Internet 上开始流行, 致使 Java 得以迅速发展并获得成功
1995. 5	Sun Microsystems 公司正式发表 Java 与 HotJava 产品
1995. 10	Netscape 与 Sun Microsystems 合作, 在 Netscape Navigator 中支持 Java
1995. 12	微软 (Microsoft) IE 加入支持 Java 的行列
1996. 2	Java Beta 测试版结束, Java 1.0 版正式诞生
1997. 2	Java 发展至 1.1 版。Java 的第一个开发包 JDK (Java Development Kit) 发布
1999. 7	Java 升级至 1.2 版
2000. 9	Java 升级至 1.3 版
2001. 7	Java 升级至 1.4 版

由表1-1可知,Java的最新版本为1.4版。Java的早期版本称为JDK,1.2版之后改名为Java 2,Java 2平台的发布被称为是Java发展史的新里程碑,目前社会上流行的很多Java教程包括本书都是基于Java 2技术的。

1999年下半年,Sun Microsystems公司重新组织Java平台的集成方法,并将Java企业级应用平台作为发展方向,因此,到今天Java的大家庭中已有了三个主要成员:

1) J2ME——Java 2 Micro Edition 用于嵌入式Java消费电子平台。不论是无线通讯、手机、PDA等小型电子装置都可采用其作为开发工具及应用平台。

2) J2SE——Java 2 (Software Development Kit) Standard Edition 是Java最通行的版本,是

用于工作站、PC 机的 Java 标准平台。因此也是本书应用程序的使用版本。

3) J2EE——Java 2 Enterprise Edition 可扩展的企业应用 Java 2 平台，它提供了企业 e-Business 架构及 Web Services 服务，其深受广大企业用户欢迎之处是开放的标准和优越的跨平台能力。

1.2 Java 的特点

Java 程序设计语言是新一代语言的代表，它强调了面向对象的特性，可以用来开发范围广泛的软件，它具有支持图形化的用户界面、网络以及数据库连接等复杂的功能。Java 语言的主要特点有：

1. 易于学习

Java 语言很简单，对于熟悉 C++ 的程序设计人员更不会对其感到陌生，因为它的语法非常类似于 C++，但是它摒弃了 C++ 许多低级、困难、容易混淆、容易出错或不经常被使用的功能，例如运算符重载、指针运算、程序的预处理、结构、多重继承以及其他一系列内容，这样便使其比其他许多编程语言更有利于初学者的学习。利用 Java 语言同样能够编制出非常复杂的系统，且运行时占用很少内存资源。

2. 高效率的执行方式

Java 源程序的执行方法是采用先经过编译器编译，再利用解释器解释的方式来运行的。它综合了解释性语言与编译语言的众多优点，使其执行效率较以往的程序设计语言有了大幅度的提高。

3. 与平台无关性

Java 源程序经过编译器编译，会被转换成一种“字节码”的目标程序。“字节码”的最大特点便是可以跨平台运行，即程序设计人员们常说的“编写一次，到处运行”，正是这一特性使 Java 得以迅速普及的重要原因。

4. 分布式与安全性

Java 从诞生时起就与网络联系在一起，它强调网络特性，内置 TCP/IP、HTTP、FTP 协议类库，便于开发网上应用系统，Java 2 平台采用了域管理方式的安全模型，无论是本地代码还是远程代码都可以通过配置的策略，设定可访问的资源域。可以说，在 Internet 上没有比 Java 更安全的技术了。

5. 丰富的 API 文档和类库

Java 为用户提供了详尽的 API (Application Program Interface) 文档说明。Java 开发工具包中的类库包罗万象，应有尽有，程序员的开发工作可以在更高的层次上展开，这也正是 Java 受欢迎的重要原因之一。

1.3 Java 程序介绍

1.3.1 Java 虚拟机

任何一种可以运行 Java 程序的软件都可以称为 Java 虚拟机 (Java virtual machine, JVM)，因此，诸如浏览器与 Java 的开发工具等皆可看作为 JVM。当然我们可以把 Java 的字节码看成

是 JVM 所运行的机器码。

1.3.2 Java 开发工具

为了正确且顺利地创建、运行和调试 Java 程序，我们需要的不仅仅是解释器，还需要编译器和其他一系列工具，这些内容都由一系列程序文件（软件）组成。这些文件的集合称为 Java 的开发工具。从 Java 的特点部分可以看到：任何支持用于编写 Java 程序的环境都会给予编写程序的人员提供很多的帮助内容，从这层意义上说，再用“编写”程序这样的词汇便不很妥当了，因此，我们用“开发”两字来代替“编写”，就相当有意思了，随着大家学习的进一步深入，一定会有更深的体会。由 Java 的开发工具所组成的帮助用户开发 Java 程序的环境称为 Java 的集成开发环境（IDE）。

我们要在此特别介绍的是开发工具包（Software Development Kit）Standard Edition 1.4.2，简称 SDK1.4，有些参考书称此开发工具包为 JDK1.4。本书中的所有程序都是用这个版本的工具测试的。它是由 Sun Microsystems 公司所开发的一套 Java 程序开发软件，可从 Sun Microsystems 公司的网站免费获得，在 1.5 节我们将向大家详细介绍 SDK1.4 的下载与安装。

若要安装 Java 开发环境，必须有支持该环境的操作系统，Java 产品面向的主流平台有 Solaris、Macintosh 以及 Windows 等，Java 开发工具包可从 <http://java.sun.com/j2se/1.4/download.html> 下载，由于在大家学习的过程中，所接触的大多是 Windows 操作系统，因此我们现在就以 SDK1.4 Windows 版本为例，下载程序“j2sdk-1_4_2_03-windows-i586-p.exe”，解压安装后会出现如图 1-1 所示的目录结构。

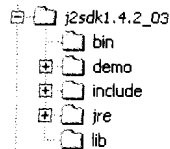


图 1-1 SDK 目录结构

现在来介绍一下 SDK1.4 目录以及所包含的文件。

1.3.3 SDK 目录结构

1. 软件安装的根目录

包含版权、许可以及 README 文件。还包括：src.zip 构成 Java 平台核心 API 的所有类的源文件的归档。

2. bin 目录

包含 SDK 开发工具的可执行文件。

3. lib 目录

开发工具使用的归档包文件。其中 tools.jar 包含支持 SDK 的工具和实用程序的非核心类，dt.jar 是 BeanInfo 文件的 DesignTime 归档，BeanInfo 文件用来告诉交互开发环境（IDE）如何显示 Java 组件以及如何让开发人员根据应用程序自定义它们。

4. jre 目录

Java 运行时环境的根目录。其子目录 \jre\bin 中包含 Java 平台使用的工具和库的可执行文件及 DLL。其中 DLL 文件是指经典虚拟机使用的 DLL 文件。经典虚拟机是 Java 虚拟机的语言注释版本。当新虚拟机可用时，它们的 DLL 文件将被安装在 jre/bin 的某个新子目录中。

子目录 jre\lib 是 Java 运行时环境使用的代码库、属性设置和资源文件。包括 rt.jar 为自

举类（构成 Java 平台核心 API 的 RunTime 类）。charsets.jar 为字符转换类及其他与国际化和本地化有关的类。

子目录 jre \ lib \ ext 是 Java 平台扩展的默认安装目录。

子目录 jre \ lib \ security 包含用于安全管理的文件。这些文件包括安全策略（java.policy）和安全属性（java.security）文件。

5. demo 目录

含有资源代码的程序示例。

6. include 目录

包含 C 语言头文件，支持 Java 本地接口和 Java 虚拟机调试程序接口的本地代码编程技术。

1.3.4 Java 2 开发工具简介

在 SDK 的 bin 目录下，存放着 Java 2 提供的一些可执行应用程序，为我们开发和测试自己的程序提供了工具：

- 1) javac: Java 语言的编译器。
- 2) java: Java 应用程序执行引擎，类名后面可带一个或多个字符串参数。
- 3) javadoc: API 生成器，对当前类生成 HTML 格式的 API 文档。
- 4) appletviewer: 小应用程序浏览器。
- 5) jar: Java Archive 文件归档工具，用它可以把类文件以及其他格式的文件进行压缩，以便于程序中引入以及网络传输等。
- 6) jdb: Java 调试工具 Debugger。
- 7) javap: 类文件的分解器。
- 8) extcheck: 判断和检测 jar 冲突。
- 9) rmic: 为远程对象调用产生存根和框架。
- 10) rmiregistry: 提供远程对象注册服务。
- 11) rmid: 激活远程方法的系统后台服务。
- 12) serialver: 返回类的序列化版本号。
- 13) native2ascii: 将本地编码的文本文件转化成 Unicode 编码文件。
- 14) keytool: 按键和键盘识别的管理工具。
- 15) jarsigner: 产生和识别 jar 文件签名。
- 16) policytool: 图形化用户界面工具的管理策略文件。
- 17) tnameserv: 提供可访问的 CORBA 命名服务。
- 18) idlj: 生成 .java 文件使它映射到一个 OMG RDL 接口并且使一个应用可以利用 CORBA 功能写入 Java 程序。

1.3.5 Java 的 API 类库与 API 文档

在开发 Java 程序时，需要设计和构造类集合。当程序运行时，对象从那些类进行实例化，并按照需要使用。作为 Java 程序员，其主要任务就是创建正确的类集合，以完成程序需要完

成的工作。非常幸运的是，一切工作并不需要都从零开始，SDK 给出了一套标准的类库，这些类为执行大部分的编程任务提供了方法和接口。类库被组织成许多包，每个包又包含一些子包和多个类。形成树型结构的类层次，其中包括核心包 java、扩展包 javax 和 org 等，如图 1-2 所示。

下面简单介绍一些重要包及其类：

1) java.lang——这个包包含了一些形成语言核心的类，提供了类似 Character、Integer 和 Double 这样的封装类。它还提供了系统标准类，如 String 和 StringBuffer。Java 编辑器总是自动装载这个包。因而一般不必显示导入 java.lang 中的任何类。本书后续内容将涉及这个包中的许多类。

2) java.applet——这个包提供了创建 Java applet 的途径，Java applet 运行在 Web 浏览器下，通常通过 Internet 下载。

3) java.awt——是由许多组成 Java 的抽象视窗工具 (AWT) 的类所组成的包，它提供了基于类的图形用户界面，可以为 Java applet 和应用程序编程提供视窗、按钮、对话框和其他控件。

4) java.net——这个包提供了网络、套接字处理器和 Internet 实用工具类。

5) java.io——这个包中的类提供了输入输出服务，用于读出和写入文件数据，访问键盘输入和打印输出。

6) java.util——这个包包含为任务设置的实用程序类和集合框架类，每一个 Java 应用程序和 Java applet 可能至少会用到这个包中的一个类。另外它还提供了 Collection 接口和它的实现容器类，如 List 和 Set。

7) java.rmi——远程方法启用包，在这个包中的类提供了通过远程接口控制的分布式代码的支持。通过该包中的类，可以创建 Java 应用程序，使它的不同部分在不同的系统中一起运行。

8) java.sql——这个包提供了结构化查询语言数据库字段类型和方法的实现。根据系统的不同，这个包的类可能会通过一个特定的数据库系统实现，或者默认时通过 ODBC (开放数据库连接) 标准的直接映射实现。

Java 的 API 文档是 Java 程序开发的最好帮手，当你从 Sun Microsystems 公司的网站 <http://java.sun.com/j2se/1.4.2/download.html> 下载 SDK 参考文件“j2sdk-1_4_2-doc.zip”，解压缩后，可以在名为“j2sdk-1_4_2-doc”的文件夹下看到如图 1-2 所示的目录结构，其中的文本是 HTML 格式文件，提供了所有有关 API 类库的信息及链接，API 文档内容主要包括：类层次结构、类及其一般目的的说明、成员变量表、构造函数表、方法表、变量详细说明表及每一个变量使用目的的详细描述、构造方法的详细说明及进一步的描述等等。为了方便使用，我们通常将 C:\j2sdk-1_4_2-doc\docs\api\index.html 放置到浏览器的收藏夹中，以供随时查阅。它在浏览器下的界面形式如图 1-3 所示。

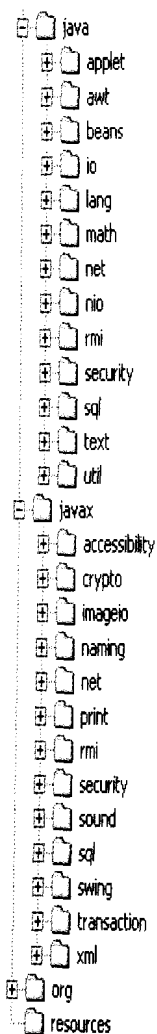


图 1-2 Java 类库结构

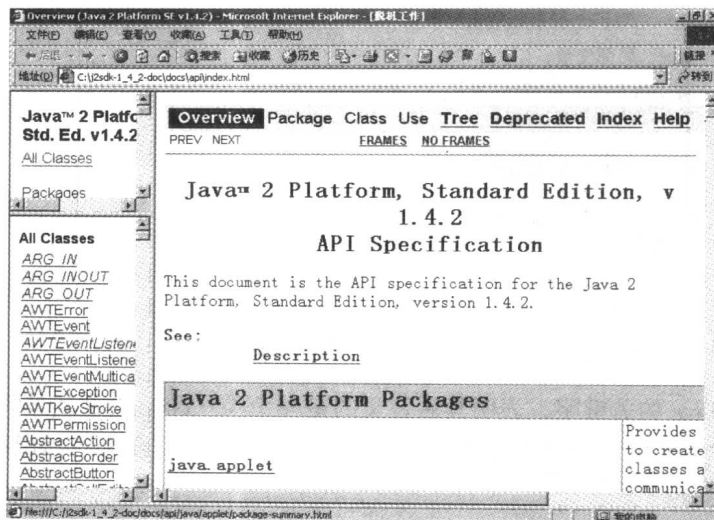


图 1-3 API 文档索引页

1.3.6 Java 程序结构

要编写 Java 程序，首先应该知道 Java 程序文件中必须包括什么内容，Java 程序的源代码文件结构如下：

- package 语句，0 ~ 1 句，必须放在文件开始，作用是把当前文件放入所指向的包中。
- import 语句，0 ~ 多句，必须放在所有类定义之前，引入标准类或已有类。
- public classDefintion, 0 ~ 1 句，文件名必须与类的类名完全相同。
- classDefintion, 0 ~ 多句，类定义的个数不受限制。
- interfaceDefintion, 0 ~ 多句，接口定义的个数不受限制。

Java 程序的源代码文件要求包含三个要素：

- 1) 一个可选的包声明语句，此句为可选。若有，且只能有一个 package 语句且只能是源文件的第一个语句，若没有，此文件将放到默认的当前目录下。
- 2) 任意数量的引入类声明语句。
- 3) 类和接口的声明，只能有一个可见性为 public 的类，且要求源文件名必须和 public 类名相同，Java 语言对字符的大小写敏感，因此文件名相同意味着字母大小写也完全相同。如果源文件中有主方法 main()，它应放在 public 类中。

这三个要素在程序中必须严格按上述顺序出现。

Java 程序都是由类 (class) 组成的，Java 中类定义的语法形式为：

```
修饰符 class 类名 [extend 父类名]
```

```
type 类变量 1;
```

```
type 类变量 2;
```

```
.....
```

```

    修饰符  type  类方法 1(参数列表){
        type  局部变量;
        方法体
    }

    修饰符  type  类方法 2(参数列表){
        type  局部变量;
        方法体
    }

    .....
}

```

其中 `class` 是 Java 的关键字，表明其后定义的是一个类。`class` 前面的修饰符用来限定所定义的类的使用方式。类名是用户为该类起的名字，它应该是一个合法的标识符。`type` 指的是变量或方法的数据类型。

以上概念在后面的章节中将详细为大家介绍。

在开发 Java 程序的过程中，经常需要在适当的地方加上注释语句，以便其他人阅读你的程序，一般来说，Java 语言的源文件中，任何地方都可以加注释语句。

1.3.7 最简单的 Java 程序

在这里首先介绍一个最简单的 Java 应用程序。

例 1.3.1 简单的 java 程序，它的功能是在屏幕上输出这样的内容：“Let us begin to study Java!”。

程序清单如下：

```

public class BegintoLearn
{
    public static void main(String args[])
    {
        System.out.println("Let us begin to study Java ! ");
    }
}

```

上面的程序定义了一个新的 `public` 类 `BegintoLearn`，这个类的源程序文件名为 `BegintoLearn.java`，类 `BegintoLearn` 的范围由一对左、右大括号包含，大括号中间的内容称为类主体。`public` 是 Java 的关键字，用来表示该类为公有，也就是在整个程序里都可以访问到它。在完整的 Java 程序里，至少需要有一个类。一般情况下，一个独立的 Java 程序，只能有一个 `public` 类，因此该源程序的文件名必须与 `public` 类的名称一致，其他的类都为 `non-public` 类（若是在源程序文件中没有一个类是 `public` 类，则该源程序文件名就可以不必和其中的任何类名称相同）。类主体由许多语句组成，语句一般有两种类型——简单语句和复合语句，对简单语句来说，习惯约定一个语句书写一行，语句必须以分号“`;`”来表示结束；而复合语句则是由左、右大括号括起来的一组简单语句的集合。对于复合语句在以后的内容中将进一步为大家介绍。