

计算机科学与技术系列教材

Java语言与面向对象程序设计

主 编 朱福喜 陆 迟



WUHAN UNIVERSITY PRESS

武汉大学出版社

TP312/2621

2007

计算机科学与技术系列教材

Java语言与面向对象程序设计

主 编 朱福喜 陆 迟



WUHAN UNIVERSITY PRESS

武汉大学出版社

图书在版编目(CIP)数据

Java 语言与面向对象程序设计/朱福喜,陆迟主编. —武汉:武汉大学出版社,2007.10

计算机科学与技术系列教材

ISBN 978-7-307-05867-5

I. J… I. ①朱… ②陆… III. ①JAVA 语言—程序设计—高等学校—教材 ②面向对象语言—程序设计—高等学校—教材
IV. TP312

中国版本图书馆 CIP 数据核字(2007)第 149380 号

责任编辑:林 莉 责任校对:王 建 版式设计:支 笛

出版发行:武汉大学出版社 (430072 武昌 珞珈山)

(电子邮件:wdp4@whu.edu.cn 网址:www.wdp.whu.edu.cn)

印刷:湖北新华印务有限公司

开本:787×1092 1/16 印张:26.126 字数:625 千字

版次:2007 年 10 月第 1 版 2007 年 10 月第 1 次印刷

ISBN 978-7-307-05867-5/TP·280 定价:39.00 元

版权所有,不得翻印;凡购买我社的图书,如有缺页、倒页、脱页等质量问题,请与当地图书销售部门联系调换。



前 言



Java 语言自 1995 年诞生以来,在短短的几年时间内以迅猛的速度席卷全球,已形成一种专门的技术。十几年的历史证明,Java 在 IT 业的应用一直保持着强劲的增长势头,为网络科技和网络经济超常发展提供了强大的动力和支持。

本书跟踪了 Java 语言的最新发展动向。全书共分十三章,介绍了 Java 语言的基础知识与主要功能,系统分析了 Java 语言的面向对象的编程机制,并用软件工程的理论和方法,阐述了使用 Java 语言进行面向对象的程序设计技巧。本书还介绍了一些 Java 语言的高级特性,如网络编程、数据库连接、多线程等。

本书第一章主要介绍了 Java 的发展、语言特点、最新版本和展示 Java 的独立应用程序和 Applet 程序的小实例,使读者对 Java 语言有一个概貌性的了解。第二章介绍了面向对象的基础知识,讨论了一般的面向对象的分析、设计和编程的基本概念。第三章介绍了 Java 编程的基础知识,主要包括数据类型、变量、表达式和流程控制语句。第四、五两章介绍了 Java 面向对象编程的知识,以类为中心详细地讨论了面向对象技术的封装、抽象、继承和多态等特征及其在面向对象程序设计中的具体应用。第六章介绍 Java 自定义的常用数据结构,包括字符串类、向量、列表、集合等。第七章介绍异常处理,掌握这一章的内容能够保证编写的程序有足够的强壮性。第八章介绍 Java 图形用户界面的设计和编程实现,通过这章的学习,读者可以编写出丰富多彩程序界面。第九章介绍流和文件,这一章不仅是文件和输入输出操作的基础,也是后续的 Java 高级编程如网络编程的基础。第十章介绍 Applet 的设计,Applet 能够使 Java 语言在 Web 上充分展示其魅力。第十一章介绍 Java 的多线程编程和异常处理,掌握这一章的内容可以编写出功能复杂的多线程程序,实现某些需要并行完成的任务。第十二章介绍网络编程,这一章充分显示了 Java 的强大网络编程功能。第十三章介绍 Java 数据库连接(JDBC),掌握 JDBC 可以很方便地在 Java 程序中引入数据库应用。

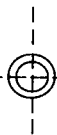
本书可广泛适用于计算机及相关专业的本科生和研究生作为学习 Java 语言的教材,也适合软件开发人员及其他有关人员作为自学的参考书或培训教程。

在本书的编写过程中,得到了武汉大学出版社黄金文老师的大力支持,在此谨向他表示衷心感谢。

由于时间和水平所限,难免有错,恳请读者批评指正,使本书得以改进和完善。

作 者

2007 年 8 月



内 容 简 介

Java 目前不仅是一门最为流行的计算机语言，而且是一个优秀的纯面向对象语言。本书从 Java 的基本概念入手，介绍了 Java 语言的基础知识与主要功能，系统分析了 Java 语言的面向对象的编程机制，并用软件工程的理论和方法，阐述了使用 Java 语言进行面向对象的程序设计技巧。本书还介绍了一些 Java 语言的高级特性，如网络编程、数据库连接、多线程等。可广泛适用于计算机及相关专业的本科生作为学习 Java 语言的教材，也适合软件开发人员及其他有关人员参考学习。

目 录

第一章 Java 概述	1
1.1 Java 技术的出现与形成.....	1
1.1.1 Java 技术的发展.....	1
1.1.2 Java 技术的应用.....	2
1.2 Java 语言的特色.....	3
1.2.1 Java 语言的特点.....	3
1.2.2 Java 程序的工作机制.....	3
1.3 Java 的开发和执行环境.....	4
1.3.1 JDK 的下载和安装.....	4
1.3.2 JDK 的环境设置.....	5
1.3.3 Java SE 常用开发工具简介.....	6
1.3.4 为自己的开发准备文件夹.....	7
1.4 简单的 Java 程序及其开发方法.....	7
1.4.1 Java 应用程序.....	8
1.4.2 Java 小程序.....	9
1.5 Java 程序的结构剖析.....	10
习题一.....	11
第二章 面向对象基础知识	12
2.1 面向对象程序设计概述.....	12
2.2 对象.....	14
2.3 消息.....	15
2.4 类.....	15
2.5 类成员与实例成员.....	16
2.6 继承.....	17
2.6.1 继承的定义.....	17
2.6.2 继承的优越性.....	18
2.7 抽象与封装.....	18
2.7.1 抽象.....	18
2.7.2 封装.....	19
2.8 多态性.....	19
习题二.....	20



第三章 Java 语言基础	21
3.1 数据和数据类型	21
3.1.1 数据类型概述	21
3.1.2 常量与变量	22
3.1.3 基本类型	22
3.2 运算符	26
3.2.1 算术运算符	26
3.2.2 关系运算符和逻辑运算符	28
3.2.3 位运算符	31
3.2.4 赋值运算符	32
3.2.5 条件运算符	33
3.2.6 字符串连接运算符	33
3.2.7 new 运算符	34
3.2.8 instanceof 运算符	34
3.3 常用 Java 数值计算方法	34
3.4 表达式和语句	35
3.4.1 表达式	35
3.4.2 语句	39
3.5 输入输出初步	40
3.5.1 标准输出	40
3.5.2 标准输入	41
3.6 控制语句	42
3.6.1 if 语句	42
3.6.2 switch 语句	45
3.6.3 while 语句和 do-while 语句	47
3.6.4 for 语句	49
3.6.5 跳转语句	52
3.7 数组	56
3.7.1 数组的创建和使用	56
3.7.2 对象数组	58
3.7.3 多维数组	62
3.7.4 数组操作的常用方法	65
习题三	68
第四章 类与对象	72
4.1 类与对象的创建	72
4.1.1 类的定义	72
4.1.2 创建对象和构造方法	75
4.1.3 销毁对象与垃圾回收	79



4.2 类成员的定义.....	80
4.2.1 成员变量的定义.....	81
4.2.2 成员方法的定义.....	81
4.2.3 变量的作用域.....	86
4.3 类的访问与封装.....	87
4.3.1 访问成员变量与成员方法.....	87
4.3.2 成员的访问权限的控制.....	89
4.3.3 静态成员变量与静态成员方法.....	94
习题四.....	99
第五章 继承与多态.....	100
5.1 继承性.....	100
5.1.1 类的继承层次.....	100
5.1.2 继承与构造方法.....	104
5.1.3 抽象与终结.....	106
5.1.4 Java 的基类 Object.....	108
5.2 多态性.....	112
5.2.1 重载实现多态性.....	112
5.2.2 覆盖实现多态性.....	115
5.3 接口与多态.....	120
5.3.1 多级继承与多重继承.....	120
5.3.2 接口.....	121
5.4 利用抽象类、接口和 Object 类实现多态性下的计算.....	122
5.4.1 用抽象类实现多种形状面积的累加.....	122
5.4.2 用接口实现多种形状面积的累加.....	123
5.4.3 用一个 Object 数组实现多种形状面积的累加.....	126
5.5 包.....	127
5.5.1 Java 平台定义的包.....	128
5.5.2 包的创建.....	128
5.5.3 包的使用.....	129
5.5.4 编译和生成包.....	130
习题五.....	130
第六章 Java 自定义的常用数据结构.....	132
6.1 String 类.....	132
6.1.1 String 类字符串的定义.....	132
6.1.2 String 类的常用方法及其应用.....	134
6.1.3 命令行参数.....	140
6.2 StringBuffer 类.....	142
6.2.1 StringBuffer 类的定义.....	142



6.2.2	StringBuffer 类的常用方法及其应用	143
6.3	Vector<E>类	147
6.3.1	Vector 类的定义	147
6.3.2	Vector 类的常用方法及其应用	147
6.4	List	149
6.4.1	LinkedList 类	149
6.4.2	ArrayList 类	151
6.5	Set 接口	151
6.5.1	HashSet<E>类	151
6.5.2	TreeSet<E>类	152
	习题六	153
第七章	异常处理	156
7.1	Java 中出错类型	156
7.1.1	异常的概念	156
7.1.2	异常类的层次和主要子类	157
7.1.3	Exception 异常类的方法	158
7.2	异常的抛出	159
7.3	异常的处理	160
7.3.1	运行时异常	160
7.3.2	try-catch-finally 语句	160
7.3.3	throw 语句和 throws 子句	162
7.4	Finally 子句	163
7.5	创建自己的异常	165
	习题七	166
第八章	图形用户界面设计	168
8.1	抽象窗口工具 (AWT) 简介	168
8.2	基本的窗口类	169
8.2.1	Frame 类	169
8.2.2	Window 类	169
8.2.3	Container 类	169
8.2.4	Panel 类	170
8.2.5	Dialog	170
8.2.6	简单窗口举例	170
8.3	简单的 GUI 构件类	172
8.3.1	Button 类	172
8.3.2	TextField 类	172
8.3.3	Label 类	173
8.3.4	TextArea 类	173



8.3.5 List 类	173
8.4 简单的事件处理	174
8.4.1 动作事件响应的一般步骤	175
8.4.2 ActionListener 接口	175
8.5 事件类型及其处理	177
8.5.1 事件源	177
8.5.2 事件监听接口	179
8.5.3 低级事件处理	179
8.5.4 高级事件(语义事件)	188
8.5.5 事件、事件源与监听接口	196
8.5.6 监听接口适配器	198
8.6 布局管理	199
8.6.1 FlowLayout	199
8.6.2 BorderLayout	199
8.6.3 GridLayout	201
8.6.4 CardLayout	203
8.6.5 GridBagLayout	205
8.7 菜单系统	209
8.8 对话框	212
8.8.1 Dialog 类	212
8.8.2 FileDialog 类	214
8.9 图形类与图形的绘制	217
8.9.1 画线	217
8.9.2 画矩形	222
8.9.3 设置色彩和画其他图形	225
8.10 AWT 与 Swing	228
习题八	230
第九章 流和文件	231
9.1 流的基本概念	231
9.2 文件类及其操作	233
9.2.1 建立与删除文件	234
9.2.2 获取文件或目录属性	234
9.2.3 获取目录成员	236
9.2.4 文件名过滤	237
9.3 字节级输入输出类	239
9.3.1 文件输入流(FileInputStream)	241
9.3.2 文件输出流(FileOutputStream)	242
9.3.3 字节数组输入流(ByteArrayInputStream)	244
9.3.4 字节数组输出流(ByteArrayOutputStream)	245



9.3.5	回退输入字节流 (PushbackInputStream)	247
9.3.6	数据输出流 (DataOutputStream)	249
9.3.7	数据输入流 (DataInputStream)	251
9.3.8	缓存输出流 (BufferedOutputStream)	252
9.3.9	缓存输入流 (BufferedInputStream)	254
9.3.10	格式化输出流(PrintStream)	256
9.4	字符级输入输出类	257
9.4.1	字符输入流 (Reader)	259
9.4.2	字符输出流 (Writer)	259
9.4.3	FileReader 类	260
9.4.4	BufferedReader 类	260
9.4.5	FileWriter 类	262
9.4.6	BufferedWriter 类	262
9.4.7	PrintWriter 类	263
9.4.8	读取字符数组类 (CharArrayReader)	265
9.4.9	写字符数组类 (CharArrayWriter)	266
9.4.10	回退输入字符流 (PushbackReader)	268
9.5	流的标记分解类 (StreamTokenizer)	270
9.6	对象级输入输出类	272
9.6.1	Serializable 接口	272
9.6.2	对象输出流 (ObjectOutputStream)	273
9.6.3	对象类输入流 (ObjectInputStream)	274
9.7	RandomAccessFile	276
9.8	系统 I/O 流	278
	习题九	281
第十章 Applet		283
10.1	Applet 的基本概念	283
10.1.1	Applet 的构架	283
10.1.2	Applet 类的实例变量	285
10.1.3	Applet 类的成员方法	285
10.1.4	Applet 的生命周期	286
10.2	Applet 标记	288
10.2.1	Applet 标记中的属性	288
10.2.2	利用标记向 Applet 传递参数	290
10.3	Applet 的安全限制与文件读取方法	292
10.3.1	统一资源定位器 URL	293
10.3.2	获取 URL	293
10.3.3	使用 URL 载入数据	296
10.4	Applet 环境与状态显示	298

10.5 Applet 的绘图与控制	303
10.5.1 在 Applet 中利用 Graphics 类绘图	303
10.5.2 在 Applet 显示与更新	307
10.5.3 在 Applet 中加入控制项	309
习题十	315
第十一章 多线程程序设计	317
11.1 线程的基本概念	317
11.1.1 线程类 Thread	317
11.1.2 线程的状态	318
11.1.3 线程的阻塞	319
11.1.4 线程的优先级	320
11.2 线程的使用方法	320
11.2.1 通过继承 Thread 类创建线程	320
11.2.2 通过实现 Runnable 接口来创建线程	322
11.3 多线程的使用方法	325
11.3.1 多线程的创建方法	325
11.3.2 判断线程是否存在	328
11.3.3 线程的合并 (join)	329
11.3.4 线程的优先级的获取和设置	331
11.4 多线程的同步处理	332
11.4.1 线程 synchronized 方法	333
11.4.2 Applet 中使用线程	335
11.4.3 线程的等待与唤起	338
习题十一	342
第十二章 Java 网络编程	343
12.1 Java 网络应用基础	343
12.1.1 IP 地址	343
12.1.2 端口	345
12.1.3 套接字	345
12.1.4 数据包	345
12.1.5 Internet 协议	346
12.2 基于 URL 网络应用	347
12.2.1 直接从 URL 读取内容	347
12.2.2 建立一个 URL 连接并从中读取内容	348
12.3 Socket 方式实现通信	349
12.3.1 客户端 Socket	350
12.3.2 服务器端 Socket	350
12.3.3 简单邮件系统	357



12.3.4 目录客户/服务系统	360
12.4 Datagram 方式实现通信	373
12.4.1 数据包和套节字	374
12.4.2 Datagram 实现客户服务模式	375
习题十二	381
第十三章 Java 数据库连接	382
13.1 JDBC 概述	382
13.1.1 JDBC 简介	382
13.1.2 JDBC 和 ODBC	383
13.2 JDBC 访问数据库的方法	386
13.2.1 装载驱动程序	386
13.2.2 定义连接 URL	386
13.2.3 获取数据库信息和创建接口 Statement 对象	387
13.2.4 执行 SQL 语句以访问数据库	387
13.2.5 对执行 SQL 语句的结果进行处理	388
13.2.6 关闭操作	389
13.3 JDBC 数据库操作实例	389
13.3.1 建表和查询操作	389
13.3.2 数据库表的交互操作	391
13.3.3 数据库操作的 GUI 程序	393
习题十三	397
参考文献	399

第一章 Java 概述

1.1 Java 技术的出现与形成

1.1.1 Java 技术的发展

1990年,美国 Sun 公司的 James Gosling、Bill Joe 等人,为在电视、控制烤箱等家用消费类电子产品上进行交互式操作而开发了一种与平台无关、可靠性强、小而灵活的编程语言,但当时并没有引起人们的注意。直到 1994 年下半年,Internet 的迅猛发展,环球信息网 WWW 的快速增长,人们发现 Java 这种中性平台及可靠性强的语言恰恰就是全球信息网在等待的语言。Java 的开发人员基于网络对 Java 进行了一系列的改进,融合了 C 和 C++ 等语言的优点,形成了现在这套与众不同的面向对象的通用程序设计语言。

Java 的原名叫 Oak (橡树),但在申请注册商标时,发现已经有人使用 Oak 这个名字了。在比较了一系列名字后,最终使用了提议者在喝 Java 咖啡时无意提到的 Java。

Java (JDK 1.0 - Java 开发工具)正式发表于 1995 年 5 月。Java 的“Write Once, Run Anywhere (一次编写,到处运行)”的口号使得 Java 一出现就引起广泛的注意。移植性强是 Java 语言的主要特点,用 Java 开发的软件可以不用修改或重新编译而直接应用于任何计算机上。随着 IT 业和 Java 十多年来的发展,Java 语言和随之发展起来的 Java 技术的众多优点使得它逐渐成为 Internet 上受欢迎的开发工具。Java 的诞生对传统的计算模型提出了新的挑战。

Java 的发展历史可以简述如下:

- 1995 年 5 月 23 日,Java 语言诞生。
- 1996 年 1 月,第一个 JDK - JDK1.0 诞生。
- 1997 年 2 月 18 日, JDK1.1 发布。
- 1998 年 12 月 4 日, JDK 1.2 发布,称为 Java 2。
- 1998 年 12 月 8 日, Java 2 企业平台 J2EE 发布。
- 1999 年 6 月, Sun 公司发布 Java 的三个版本:标准版 J2SE、企业版 J2EE 和微型版 J2ME,分别对应 Java 桌面应用程序和低端服务器程序系统开发、企业级的服务应用和环境资源有限的嵌入式消费产品(如掌上电脑、手机等移动设备等不同系统)的 Java 开发。
- 2000 年 5 月 8 日, J2SE1.3 发布。
- 2001 年 9 月 24 日, J2EE1.3 发布。
- 2002 年 2 月 26 日, J2SE1.4 发布,自此 Java 的计算能力有了大幅提高。
- 2004 年 9 月 30 日, J2SE1.5 发布,是 Java 语言的发展史上的又一里程碑事件。为了



表示这个版本的重要性, J2SE1.5 更名为 J2SE5.0。

- 2005 年 6 月, Sun 公司公开 Java SE 6。此时, Java 的各种版本已经更名, 取消了其中的数字“2”: J2EE 更名为 Java EE, J2SE 更名为 Java SE, J2ME 更名为 Java ME。
- 2006 年 12 月 11 日, Java SE 6 正式版发布。Java SE 6 提供了包括 XML Digital Signature 在内的一些新 API, 更新了 JDBC 4.0 和 JAXB 2.0 等 API, 重新设计了图形渲染管道, 提高了 Swing 的 Windows 的精度 GTK。内置地支持脚本语言, 如 JavaScript。

今天, Java 已经形成一门技术, 主要体现在:

- Java 软件技术: Java JDK、Java Runtime Environment、HotJava、Java OS、JDBC、JavaBean、Java 虚拟机及规范。
- Java 嵌入技术: Java 芯片 (如 MicroJava701), 基于 Java 技术的 NC、Java Server、Java Station、WebTV (机顶盒)、Java 汽车、Java 手机、PDA (个人数字助理) 等。
- Java Computing: 基于处理功能的综合处理系统, 如 J2EE Server 或 Web Application Server, 其典型产品有 Tomcat、WebLogic、Websphere 等。

1.1.2 Java 技术的应用

现在, 在 Java 诞生后的第 12 个年头, Java 平台已经吸引了近 500 万软件开发商, 全世界的每个主要行业领域都在使用它, 任何使用编程技术的设备、计算机和网络都在大范围地应用它。

事实上, Java 技术的多功能性、有效性、平台的可移植性以及安全性已经使它成为网络计算领域最完美的技术之一。因此到今天为止, Java 技术已经为 25 亿台设备提供支持:

- 7 亿台以上的 PC。
- 7 亿 8 百万部移动电话以及其他手持式设备。
- 10 亿个智能卡。
- 不计其数的机顶盒、打印机、网络照相机、游戏、汽车导航系统、彩票终端、医疗设备、收费站等。

日臻完善、极度强大而且功能繁多的 Java 技术已经成了开发商的无价之宝, 利用它可以:

- 在一个平台上编写软件, 然后在另一个平台上运行。
- 创建可在 Web 浏览器和 Web 服务中运行的程序。
- 开发适用于联机论坛、存储、投票、HTML 格式处理以及其他用途的服务器端应用程序。
- 将基于 Java 技术的应用程序或服务组合在一起, 以生成高度自定义的应用程序或服务。
- 为移动电话、远程处理器、低成本的消费产品以及任何具有数字核心的设备编写强大而高效的应用程序。

今天, 无论是互联网和科学超级计算机还是膝上型计算机和手机, 无论是华尔街的市场模拟器还是家庭游戏机和信用卡, 在所有网络和设备上都会看到 Java 技术的身影, 它已经无处不在。



1.2 Java 语言的特色

1.2.1 Java 语言的特点

Java 是一个迅速发展的网络编程语言，它是一种新的计算概念。Java 的创建者 Sun 公司将 Java 描述为：

A simple, object-oriented, distributed, robust, secure, architecture-neutral, portable, high-performance, multi-threaded and dynamic language.

即：Java 是一种简单的、面向对象的、分布式的、强壮的、安全的、体系结构中立的、可移植的、高性能的、多线程的和动态的语言。

Java 最大限度地利用了网络。一种称为 Java 小程序(Applet)的 Java 程序是动态、安全、跨平台的网络应用程序，可在网络上运行而不受 CPU 和环境的限制。Java Applet 嵌入 HTML 语言中，通过 Web 页发布到 Internet。网络用户访问服务器的 Applet 时，这些 Applet 从网络上进行传输，然后在支持 Java 的浏览器中运行。由于 Java 语言的安全机制，用户一旦载入 Applet，就可以放心地来生成多媒体的用户界面或完成复杂的计算而不必担心病毒的入侵。虽然 Applet 可以和图像、声音、动画等一样从网络上下载，但它并不同于这些多媒体文件格式，它可以接收用户的输入，动态地进行改变，而不仅仅是动画的显示和声音的播放。

另外，Java 还提供了丰富的类库，以满足网络化、多线程、面向对象系统的需要，使程序设计者可以很方便地建立自己的系统。

① 语言包提供的支持包括字符串处理、多线程处理、异常处理、数学函数处理等，可以用它简单地实现 Java 程序的运行平台。

② 实用程序包提供的支持包括哈希表、堆栈、可变数组、时间和日期等。

③ 输入输出包用统一的“流”模型来实现所有格式的输入/输出，包括文件系统、网络、输入/输出设备等。

④ 低级网络包用于实现 Socket 编程。

⑤ 抽象图形用户接口包实现了不同平台的计算机的图形用户接口组件，包括窗口、菜单、滚动条、对话框等，使得 Java 可以移植到不同平台的机器。

⑥ 网络包支持 Internet 的 TCP / IP 协议，提供了与 Internet 的接口。它支持 URL 链接和 WWW 的即时访问，并且简化了客户/服务器模型的程序设计。

1.2.2 Java 程序的工作机制

学习 Java 语言，有必要了解 Java 的工作机制，这将更有助于理解 Java 语言的特点。

对于运行在 Internet 上的网络应用程序，需要有良好的可移植性。因为 Internet 是由各种各样不同类型的终端、服务器和 PC 等硬件设备组成的，而且在这些设备上运行的软件系统也是多种多样的，所以 Internet 上的网络应用程序应该具有在各种不同的软硬件平台上均可正常工作的能力。Java 的工作机制使得它具有了这样的能力。

Java 的工作机制是这样的：编程人员首先编写好源代码，然后经编译生成一种二进制的中间码，称为字节码(byte code)，最后再通过运行与操作系统平台环境相应的一种称为 Java 解释器的运行机构来执行编译生成的字节码。虽然不同的平台环境需要有各自相应的解释器，但是任何一个平台上的解释器，对于一段 Java 程序的字节码来说却是相同的，它们对



Java 字节码呈现出完全相同的面貌。也就是说, Java 的运行机制是利用解释器来隐藏网络上平台环境的差异性的。由此可见, Java 实现了二进制代码级的可移植性, 在网络上实现了跨平台的特性。

Java 的解释器和运行时系统又称为“Java 虚拟机 (JVM, Java Virtual Machine)”, 是驻留于计算机内存的虚拟计算机或逻辑计算机, 实际上是一段负责解释执行 Java 字节码的程序。JVM 能够从字节码流中读取指令并解释指令的含义, 每条指令都含有一个特殊的操作码, JVM 能够识别并执行它。从这个意义上说, Java 可以被称为是一种“解释型”的高级语言。

Java 语言程序的两种形式为 Java 应用程序和 Java 小程序。若 Java 解释器是一个独立的应用程序, 并可以在操作系统下直接启动, 那么它解释执行的程序被称为“Java Application(Java 应用程序)”; 若 Java 解释器包含在一个 WWW 的客户端浏览器内部, 使得这个浏览器能够解释字节码程序, 则这种浏览器能够自动执行的 Java 程序被称为“Java Applet (Java 小程序)”。这两种程序从程序结构到运行机制上都不相同, Application 多在本地或服务器上运行, 而 Applet 则只能通过浏览器从服务器上下载后再运行。

无论是 Java Application 还是 Java Applet, 其程序源代码文件都以 .java 为文件扩展名, 而 .class 则是编译后二进制字节码文件的文件扩展名。

1.3 Java 的开发和执行环境

若要编写 Java 程序, 就需要开发工具。现在可用于开发 Java 程序的工具有很多, 常用的有美国 Sun 公司的 Java SE (EE, ME) Development Kit (JDK) 和 NetBeans IDE、Borland 公司的 JBuilder、IBM 公司的 Eclipse、Microsoft 公司的 Visual J++ 及 Xinox 公司的 JCreator 等。JDK、NetBeans 和 Eclipse 等是免费的, 可以到相应公司的网站或其他提供软件下载的网站去下载; JBuilder、JCreator 和 Visual J++ 等是商业化的产品, 需要付费和注册来使用 (部分提供试用版)。JDK 是很多其他开发工具的核心, 学习 Java 一般从 JDK 的标准版 Java SE 入手, 因此本书基于此来学习采用。Java SE 是一个命令行版本, 设计图形界面程序时方便程度差一些, 但运行速度较快, 对机器的要求相对较低。

1.3.1 JDK 的下载和安装

JDK 的意思是 Java Development Kit, 即 Java 软件开发工具。截止到 2007 年 1 月, 提供下载的 JDK 标准版软件最新版本为 Java SE 6 版, 有不同操作系统的不同版本。下面介绍采用 32 位的 Windows 系统版本。

可以从网址 <http://java.sun.com> 下载最新的 SDK 开发工具: `jdk-6-windows-i586.exe` (53.16MB, 可直接运行) 和 API 说明文档: `jdk-6-doc.zip` (52.36MB, 若需要, 再下载安装)。下载完成后运行 `jdk-6-windows-i586.exe` 即进行开发工具的安装, 安装时, 可指定安装目的盘和文件夹, 也可安装到默认的目的盘和文件夹。默认安装时, 将在 C 盘程序文件夹中创建名为 `java\jdk1.6.0` 开发工具的主文件夹和名为 `jre1.6.0` 的运行时系统主文件夹。程序开发所用的 `jdk1.6.0` 文件夹下创建有一些子文件夹, 主要的有 `bin` 文件夹 (存放可运行的程序开发工具)、`lib` 文件夹 (存放库文件)、`demo` 文件夹 (存放供学习、演示用的 Java 程序)、`jre` 文件夹 (存放运行时环境等支撑文件) 等。为了在任何目录中运行 Java 工具, 可对运行环境