

高等院校计算机应用技术系列教材

# Java 程序设计

潘 浩 李芙玲 编著  
王志丽 易 久

清华大学出版社  
北京交通大学出版社  
·北京·

## 内 容 简 介

本书是一本介绍 Java2 编程基本知识的教材。

全书由 12 章组成,分两部分来实现。第一部分从第 1 章到第 5 章,讲述 Java 语言的基础知识,包括 Java 标识符、Java 运算符、表达式和 Java 语句、Java 的面向对象编程、继承、抽象、接口和包的应用;第二部分从第 6 章到第 12 章,讲述 Java 类库的知识,包括 Java 语言包、Java 工具包、图形、输入输出流、多线程、网络编程、JDBC 等应用。

本书的语言浅显易懂,内容介绍由浅入深,并辅以大量实例讲解 Java 程序的实现。本书适用于 Java 语言的初学者,可作为大专院校计算机或非计算机专业学生的教材,也可作为 Java 语言的培训和自学教材。

版权所有,翻印必究。举报电话:010-62782989 13501256678 13801310933

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

本书防伪标签采用特殊防伪技术,用户可通过在图案表面涂抹清水,图案消失,水干后图案复现;或将表面膜揭下,放在白纸上用彩笔涂抹,图案在白纸上再现的方法识别真伪。

### 图书在版编目(CIP)数据

Java 程序设计 /潘浩等编著. —北京:清华大学出版社,北京交通大学出版社,2005.11  
(高等院校计算机应用技术系列教材)

ISBN 7-81082-628-X

I. J... II. 潘... 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 (2005) 第 109183 号

责任编辑:谭文芳 特邀编辑:刘卫华

出版者:清华大学出版社 邮编:100084 电话:010-62776969 <http://www.tup.com.cn>

北京交通大学出版社 邮编:100044 电话:010-51686414 <http://press.bjtu.edu.cn>

印刷者:北京东光印刷厂

发 行 者:新华书店总店北京发行所

开 本:185×260 印张:18.75 字数:477 千字

版 次:2005 年 11 月第 1 版 2005 年 11 月第 1 次印刷

书 号:ISBN 7-81082-628-X/TP·237

印 数:1~5000 册 定价:26.00 元

---

本书如有质量问题,请向北京交通大学出版社质监组反映。对您的意见和批评,我们表示欢迎和感谢。

投诉电话:010-51686043, 51686008; 传真:010-62225406; E-mail: [press@center.bjtu.edu.cn](mailto:press@center.bjtu.edu.cn)

# 前 言

Java 语言是当前最流行的网络编程语言之一,它具有面向对象、与平台无关、安全、多线程等特点。使用 Java 语言不仅可以实现大型企业级的分布式应用系统,还能够为小型的、嵌入式设备进行应用程序的开发。为了适应如此广泛的应用范围,存在大量的、复杂的 Java 技术。而 Java 平台标准版本(J2SE)是所有 Java 技术的基础,只有掌握 J2SE,才能进一步进行深入的 Java 技术的学习。

本书主要讲述了 J2SE(1.4 版本)的相关知识。作为一本适合于 Java 初学者的入门教材,本书主要具有以下几个特点。

(1) 本书的读者不要求具有专门的计算机专业的基础知识和 Java 编程经验,通过本书的学习,读者可以进行 Java 程序的编写。

(2) 本书的结构经过精心安排,内容的讲述由浅入深,基本上按照大多数人的学习习惯来安排每章的内容结构。

(3) 本书对每个知识点,都辅以图形或具体实例的方式进行讲述,使读者能够从具体应用中掌握知识,能够很容易地将所学的知识应用于实践。

(4) 每章后面都有习题,读者通过做习题,巩固并能灵活运用所学知识。

本书共有 12 章。第 1 章讲述 Java 语言的产生和发展过程、Java 语言的特点,以及对 Java 的三大平台(J2SE、J2EE 和 J2ME)作了简要介绍。第 2 章介绍 Java 的开发工具,以及编写和运行一个简单的 Java 程序的完整过程。第 3 章讲述 Java 编程的基本语法知识。第 4 章和第 5 章讲述 Java 的面向对象技术,体现了 Java 作为一种纯粹的面向对象编程语言的编程特点。从第 6 章开始,讲述 Java 类库的知识,从而实现功能丰富的 Java 程序。第 6 章从总体上介绍 Java 类库的组成,并讲述 Java 语言包和工具包中的几个类用类。第 7 章讲述 Java 小程序的实现。第 8 章讲述如何使用图形包实现绘图功能。第 9 章讲述输入输出流的实现。第 10 章讲述 Java 的多线程技术。第 11 章讲述用 Java 实现网络编程的方法。第 12 章讲述如何用 Java 实现对关系型数据库的操作。

本书第 1 和第 2 章由易久编写,第 3、4、5、6、9 章由潘浩编写,第 7 和第 8 章由李芙玲编写,第 10 和第 11 章由王志丽编写,第 12 章由李永编写。全书最后由潘浩统稿。本书的程序全部经过严格测试,可以正常运行。耿子林教授和王新生教授在百忙之中认真审阅了书稿,并提出了宝贵的意见。在本书的编写过程中,崔林博士、郑志蕴博士、陈建斌博士、曹德胜、何立也为本书的编写和修改做了许多细致的工作,在此表示深深的感谢。

本书的编写还存在一些纰漏之处,请读者不吝指出。

编 者  
2005 年 9 月

# 目 录

第 1 章 Java 概述 .....	1
1.1 Java 的产生与发展 .....	1
1.2 Java 语言的特点 .....	2
1.3 Java 语言平台及相关应用 .....	4
1.3.1 Java 2 标准版本——J2SE .....	4
1.3.2 Java 2 企业版本——J2EE .....	5
1.3.3 Java 2 微型版本——J2ME .....	7
小结 .....	8
习题 .....	8
第 2 章 Java 编程概述 .....	9
2.1 JDK 的安装与配置 .....	9
2.2 Java 程序编辑环境的建立 .....	10
2.3 Java 应用程序实例 .....	13
2.3.1 Java 应用程序的编辑 .....	13
2.3.2 Java 应用程序的编译和运行 .....	14
2.4 Java Applet 实例 .....	17
2.4.1 Java Applet 源程序的编辑与编译 .....	17
2.4.2 HTML 文件的编辑与运行 .....	18
小结 .....	19
习题 .....	19
第 3 章 Java 基本语法结构 .....	21
3.1 标识符 .....	21
3.1.1 自定义标识符 .....	21
3.1.2 关键字 .....	21
3.1.3 类库中的标识符 .....	22
3.2 分隔符 .....	22
3.2.1 空白符 .....	22
3.2.2 普通分隔符 .....	23
3.3 注释 .....	23
3.3.1 单行注释 .....	24
3.3.2 多行注释 .....	24
3.3.3 文档注释 .....	24
3.4 基本数据类型 .....	24

3.4.1	整数类型 .....	25
3.4.2	浮点类型 .....	25
3.4.3	字符类型 .....	26
3.4.4	布尔类型 .....	26
3.4.5	数据类型的相互转换 .....	27
3.5	常量与变量.....	28
3.5.1	常量 .....	28
3.5.2	变量 .....	30
3.6	运算符与表达式.....	31
3.6.1	赋值运算符 .....	31
3.6.2	算术运算符 .....	32
3.6.3	关系运算符 .....	34
3.6.4	逻辑运算符 .....	35
3.6.5	位运算符 .....	36
3.6.6	其他运算符 .....	38
3.6.7	运算符的优先级 .....	39
3.7	数组.....	39
3.7.1	数组的创建 .....	40
3.7.2	数组的使用 .....	41
3.7.3	多维数组 .....	42
3.8	Java 语句.....	43
3.8.1	表达式语句 .....	43
3.8.2	方法调用语句 .....	43
3.8.3	空语句 .....	44
3.8.4	复合语句 .....	44
3.9	控制语句.....	45
3.9.1	条件语句 .....	46
3.9.2	循环语句 .....	49
3.9.3	转移语句 .....	51
小结	.....	53
习题	.....	53
第 4 章	类与面向对象编程 .....	56
4.1	面向对象编程基础.....	56
4.1.1	面向对象编程的基本概念 .....	56
4.1.2	面向对象编程的优点 .....	57
4.1.3	对象的抽象 .....	57
4.1.4	封装 .....	58
4.2	类的编程实现.....	59
4.2.1	类定义 .....	59

4.2.2	域的定义 .....	60
4.2.3	方法的定义 .....	61
4.2.4	main()方法 .....	66
4.3	对象的定义与使用.....	67
4.3.1	对象的创建 .....	67
4.3.2	构造方法 .....	68
4.3.3	对象的使用 .....	70
4.4	重载.....	73
4.4.1	方法的重载 .....	73
4.4.2	构造方法的重载 .....	74
4.4.3	关键字 this 的应用 .....	75
4.5	访问控制符.....	76
4.5.1	public 修饰符 .....	76
4.5.2	private 修饰符 .....	77
4.5.3	protected 修饰符.....	78
4.5.4	默认的修饰符 .....	78
4.6	其他修饰符.....	78
4.6.1	static 修饰符 .....	78
4.6.2	final 修饰符 .....	81
小结	.....	82
习题	.....	82
第 5 章	类间的关系 .....	84
5.1	类与类之间的关系.....	84
5.1.1	使用关系 .....	84
5.1.2	包含关系 .....	84
5.1.3	继承关系 .....	84
5.2	类的继承.....	85
5.2.1	继承的基本概念 .....	85
5.2.2	子类的创建和使用 .....	86
5.2.3	域的隐藏 .....	88
5.2.4	方法的覆盖 .....	90
5.2.5	super 关键字 .....	91
5.2.6	类对象之间的类型转换 .....	93
5.3	抽象类与抽象方法.....	95
5.3.1	抽象类 .....	95
5.3.2	抽象方法 .....	97
5.4	接口.....	98
5.4.1	接口的定义 .....	99
5.4.2	接口的实现 .....	100

5.5	包 .....	101
5.5.1	包的创建 .....	101
5.5.2	包的引用 .....	102
	小结.....	104
	习题.....	104
第 6 章	Java 基本类库 .....	106
6.1	Java 类库 .....	106
6.1.1	Java 类库的分类 .....	106
6.1.2	Java 类库的使用 .....	109
6.1.3	Java 文档的使用 .....	110
6.2	Object 类与 System 类 .....	114
6.2.1	Object 类 .....	114
6.2.2	System 类 .....	116
6.3	String 类及 StringBuffer 类 .....	116
6.3.1	String 类 .....	116
6.3.2	StringBuffer 类 .....	120
6.4	Java 类型包装类 .....	122
6.5	Math 类.....	124
6.6	Exception 类 .....	125
6.6.1	Java 中的异常类 .....	125
6.6.2	异常的使用与处理 .....	126
6.7	Vector 类 .....	130
6.7.1	Vector 类的基本方法 .....	130
6.7.2	数据插入与访问方法 .....	132
6.7.3	修改和删除数据方法 .....	133
6.7.4	数据的查找方法 .....	134
	小结.....	135
	习题.....	136
第 7 章	Java Applet .....	138
7.1	Applet 概述 .....	138
7.2	一个 Applet 小程序开发的技巧 .....	138
7.3	Applet 类 .....	139
7.3.1	Applet 类中主要方法和 Applet 生命周期 .....	139
7.3.2	什么是 JApplet .....	141
7.3.3	Applet 的运行机制 .....	143
7.3.4	HTML 文件与 Applet 之间的参数传递 .....	145
7.4	Applet 的应用举例 .....	146
7.4.1	实例 1 鼠标事件 .....	147

7.4.2 实例2 加载图像 .....	148
小结 .....	150
习题.....	150
<b>第8章 图形用户接口设计</b> .....	<b>151</b>
8.1 Swing 简介 .....	151
8.1.1 AWT 和 Swing 概述 .....	151
8.1.2 Swing 类层次结构 .....	152
8.2 Swing 组件与容器 .....	153
8.2.1 Swing 组件概述 .....	153
8.2.2 顶层容器.....	153
8.3 标签与按钮 .....	155
8.3.1 标签 .....	155
8.3.2 按钮 .....	156
8.4 事件与文本输入 .....	158
8.4.1 事件处理.....	158
8.4.2 文本域 .....	159
8.4.3 文本区 .....	161
8.5 复选框与单选按钮 .....	163
8.5.1 复选框 .....	163
8.5.2 单选按钮.....	165
8.6 GUI 布局管理器 .....	167
8.6.1 基本布局管理器 .....	167
8.6.2 面板 .....	169
8.6.3 高级布局管理器 .....	171
8.7 组合框与列表框 .....	174
8.7.1 组合框 .....	174
8.7.2 列表框 .....	176
8.8 其他组件 .....	180
8.8.1 滚动条和滑动块 .....	180
8.8.2 Swing 对话框 .....	183
8.8.3 菜单 .....	187
小结.....	191
习题.....	191
<b>第9章 输入输出流与文件</b> .....	<b>193</b>
9.1 输入输出流简介 .....	193
9.2 字节输入输出流 .....	193
9.2.1 字节输入流 .....	194

9.2.2	字节输出流 .....	195
9.3	字符输入输出流 .....	196
9.3.1	字符输入流 .....	197
9.3.2	字符输出流 .....	197
9.4	文件的创建与访问 .....	198
9.4.1	File 类 .....	198
9.4.2	输入输出文件流 .....	201
9.4.3	随机读写文件流的输入输出 .....	205
9.4.4	文件流的功能扩展 .....	212
9.5	标准输入输出流 .....	217
	小结 .....	219
	习题 .....	219
第 10 章	多线程 .....	220
10.1	线程的相关概念 .....	220
10.1.1	程序、进程与线程 .....	220
10.1.2	多线程 .....	220
10.2	线程的生命周期 .....	221
10.3	Java 中多线程的实现 .....	222
10.3.1	Thread 类 .....	222
10.3.2	Runnable 接口 .....	225
10.4	线程的同步 .....	227
10.4.1	同步方法 .....	227
10.4.2	同步语句 .....	229
10.5	线程的调度 .....	231
10.5.1	线程的优先级调度 .....	231
10.5.2	线程调度的管理 .....	233
10.6	多线程之间的通信 .....	236
10.6.1	线程之间的通信问题 .....	236
10.6.2	线程之间的通信解决方法 .....	240
	小结 .....	242
	习题 .....	242
第 11 章	网络通信程序设计 .....	244
11.1	网络通信基础 .....	244
11.1.1	套接字 .....	244
11.1.2	TCP 和 UDP 传输协议 .....	245
11.1.3	客户-服务器编程模型 .....	245
11.2	Java 网络包 .....	246

11.3	InetAddress 类 .....	247
11.4	面向连接的 Socket 网络通信 .....	250
11.4.1	Socket 类 .....	250
11.4.2	ServerSocket 类 .....	251
11.4.3	面向连接 Socket 的一个具体应用 .....	252
11.4.4	支持多客户的服务器的设计 .....	255
11.5	基于数据报 Socket 的网络通信 .....	257
11.5.1	DatagramPacket 类 .....	257
11.5.2	DatagramSocket 类 .....	258
11.5.3	基于数据报 Socket 的通信实现 .....	259
11.6	访问 URL 资源 .....	261
11.6.1	URL 类 .....	262
11.6.2	URLConnection 类 .....	264
	小结 .....	267
	习题 .....	267
第 12 章	数据库与 JDBC 编程 .....	269
12.1	JDBC 概述 .....	269
12.2	JDBC 驱动程序的类型 .....	269
12.3	JDBC URL .....	270
12.4	JDBC 中主要的类和接口 .....	271
12.4.1	JDBC 驱动设置类 DriverManager .....	274
12.4.2	连接接口 Connection .....	275
12.4.3	语句接口 Statement .....	275
12.4.4	记录集对象 ResultSet .....	277
12.4.5	PreparedStatement 接口 .....	278
12.4.6	CallableStatement 接口 .....	279
12.5	JDBC 本地驱动连接 SQLServer2000 数据库 .....	280
12.6	综合范例 .....	282
	小结 .....	285
	习题 .....	285
	参考文献 .....	286

# 第 1 章 Java 概述

## 1.1 Java 的产生与发展

Java 语言是当前最受欢迎的编程语言之一,作为美国 Sun Microsystem 公司开发的新一代面向对象的程序语言 Java 以其独有的特点及与网络的紧密结合,成为 Internet 上最流行的编程语言。

然而,最初 Java 语言的产生却不是为了应用于网络方面。Java 语言诞生于 1991 年,是 Sun 公司内部的 Green 项目组开发的。他们的目的是开发一种小型的用于消费类电子设备(如有线电视交换盒)的计算机语言,由于该类设备的内存小,并且不同厂商的产品可能选用不同的中央处理器,因此要求这种新型语言尽可能小,并且要独立于平台。Green 项目组的实现方案是为混合型的机器产生通用的中间码,通过在不同类型的机器上安装对应的虚拟机,这种中间码可以在任何一种特定的机器上解释运行。新语言是面向对象的,通过改造 C++ 语言而成。这种新型语言最初被称为 Oak 语言。由于它与当时已经存在的 Oak 语言的名字发生冲突,改为 Java 语言。Java 语言推出后,却没有受到人们的青睐。甚至到 1994 年,他们的产品没有签订一份合同。

在这段时间里,Internet 正得到快速的发展,作为 Internet 一部分的 WWW 成为最流行的信息共享的平台。对于 Web 来说,关键是要有一个好的浏览器,能够将超文本页面生动地显示到屏幕上。在 Java 出现以前,Internet 上的信息内容都是一些乏味死板的 HTML 文档。这对于那些迷恋于 Web 浏览的人们来说简直无法忍受。他们迫切希望能在 Web 上看到一些交互式的内容,开发人员也极希望能够在 Web 上创建一类无需考虑软硬件平台就可以执行的应用程序,当然这些程序还要有极大的安全保障。

在 1994 年中 Java 语言开发者们敏锐地察觉到了这一点,意识到“我们可以研制一款出色的浏览器,它将追随客户-服务器架构这一主流,提供一些真正有价值的东西,而这些东西我们已经做出来了,其中包括与平台无关性、实时性、可靠、安全,等等——这些问题原来在工作站世界中并不重要。因此我们构建了一个浏览器”。接着他们用 Java 语言开发出 HotJava 浏览器的第一个版本,该浏览器能够解释中间字节码,能够在其上运行 Java 小应用程序。在 1995 年 5 月 23 日召开的 SunWorld '95 大会上首次亮相,从此一举成名。1996 年 1 月, Netscape 公司推出了 Netscape 2.0 版本,提供了对 Java 的支持。注册了 Java 使用许可证的还有 IBM, Inprise 等公司,微软公司的 Internet Explorer 浏览器也具备 Java 能力。Java 的使用进入了流行阶段。

Java 本身从推出到现在,不断地完善和成熟。Sun 公司在 1996 年初发布了 Java 的第一个版本,几个月后,又发布了 Java 1.02 版本。这些版本并不适合进行正规的程序开发,缺乏对很多功能的支持。例如,不支持打印功能。Java 1.1 版本则实现了许多 Java 类库,对原版本进行了很大的改进,基本上实现了 Java 的重要特性。1998 年问世的 Java 1.2 版本,改进了原来像

玩具一样的图形用户接口和图形工具包,使它的功能更加全面。

自从 1995 年 Sun 公司正式推出 Java 以来,Java 技术迅猛发展。目前,全球有 300 多万开发者采用 Java 技术,全球有 90% 以上的应用服务器基于 Java 平台。IDC(International Data Corporation, 国际数据公司)报告显示,2001 年北美有近 74% 的专业开发商将 Java 用做他们的主要语言,全球 5.5 亿台个人计算机中有一半以上采用了 Java 技术。目前,采用 Java 技术的电话数量已经突破 1 亿,有关分析机构预测,到 2007 年,几乎所有的手机都将采用 Java 技术。

## 1.2 Java 语言的特点

Java 是一种跨平台、适合于分布式计算环境的面向对象编程语言。具体来说,它具有简单性、面向对象、分布式、解释型、可靠、安全、与平台无关、可移植、高性能、多线程、动态性等特点。

### 1. 简单

Java 最初是为对家用电器进行集成控制而设计的一种语言,因此它必须简单明了。Java 语言的简单性主要体现在以下三个方面。

- ∞ Java 的风格类似于 C++ ,因而 C++ 程序员很容易掌握 Java 语法。
- ∞ Java 的语法去除了 C++ 的头文件、指针、结构、多重继承等功能,使得 Java 语言的出错概率大大减少。
- ∞ Java 提供了丰富的类库。Java 程序员通过调用类库中的方法,引用类库中已经实现的功能,实现了代码的重复利用。

### 2. 面向对象

面向对象是 Java 最重要的特性。与 C++ 语言相比较,Java 语言的设计完全是面向对象的,它不支持类似 C 语言那样的面向过程的程序设计技术。面向对象其实是现实世界模型的自然延伸。现实世界中任何实体都可以看作是对象。对象之间通过消息相互作用。如果说传统的面向过程编程语言是以过程为中心、以算法为驱动的话,面向对象的编程语言则是以对象为中心、以消息为驱动。用公式表示如下。

过程式编程语言为 程序 = 算法 + 数据

面向对象编程语言为 程序 = 对象 + 消息

作为面向对象编程语言,Java 支持封装、多态和继承等面向对象特性。

#### (1) 封装

现实世界中的对象均有属性和行为,映射到计算机程序上,属性则表示对象的数据,行为表示对象的方法。所谓封装,就是用一个自主式框架把对象的数据和方法联在一起形成一个整体。可以说,对象是支持封装的手段,是封装的基本单位。Java 语言提供了类机制,以类对象为基本单位,封装了状态数据和方法。封装特性将对象的状态数据很好地保护起来,外界不能直接访问这些数据,只能通过调用对象的方法获得。

#### (2) 多态

多态性就是多种表现形式,可以用“一个对外接口,多个内在实现方法”表示。Java 通过同一类中方法的重载及子类对父类方法的覆盖实现多态性。方法的重载是指一个类中可以存在相同名字的多个方法,但这些方法中参数的数量和参数的类型不能完全相同。方法的覆盖发

生在子类对父类的继承中,子类可以对父类中的方法进行重新定义与实现。多态性使方法的调用更加容易、灵活和方便。

### (3) 继承

继承是指一个对象直接使用另一对象的属性和方法。客观世界中的很多实体都有继承的含义。例如,若把电视看成一个实体,它可以分成多个子实体,如黑白电视机、彩色电视机等。这些子实体都具有电视的特性,因此,电视是它们的“父亲”,而这些子实体则是电视的“孩子”。Java 提供的类继承机制只支持单继承,一个父类可以拥有多个子类,而一个子类只能有一个父类。子类不但拥有父类的所有特性,还可以增加新的特性。类继承提高了程序的可重复利用性,减轻了编程的负担。

## 3. 安全性和可靠性

Java 语言源于 C++ ,它去除了 C++ 的许多不可靠因素,可以防止许多编程错误,在此基础上还采取了其他一些安全措施。

- ✎ Java 不支持指针操作,这杜绝了对内存的非法访问,从而不会发生由于编码失误造成有效数据被破坏的现象。
- ✎ Java 的自动单元收集机制防止了由于动态内存分配导致的内存丢失等问题。
- ✎ Java 解释器运行时实施检查机制。在字节码装载过程中使用字节代码检验器对字节代码的信息进行检查,从而保证指令参数类型的正确性、对象域访问的合理性,及时发现数组和字符串访问的越界等错误。
- ✎ Java 提供了异常处理机制,可以把一组错误处理代码放在一个地方,简化错误处理任务,便于恢复错误。
- ✎ Java 主要用于网络应用程序开发时,通过自身的安全机制防止了病毒程序的产生和下载程序对本地系统的威胁和破坏。使用类装载器,将来自网络的类装载到单独的内存区域,避免应用程序之间相互干扰破坏。从网络上装载的类只能访问某些文件系统,不能通过任何途径对本地类进行操作。

## 4. 平台无关性

Java 是与平台无关的语言,用 Java 写的应用程序不用修改就可在不同的软硬件平台上运行,而不受计算机硬件和操作系统的限制。Java 基本数据类型的长度是固定的,与具体的计算机无关。Java 源程序经过编译产生的二进制文件称为字节码,它是一种与具体机器指令无关的指令集合。字节码由 Java 虚拟机(JVM)在不同平台上解释执行。Java 虚拟机是一种抽象机器,附着在具体操作系统之上,本身具有一套虚机器指令,并有自己的栈、寄存器组等。JVM 是 Java 平台无关的基础,在 JVM 上,有一个 Java 解释器用来解释 Java 编译器编译后的程序。Java 编程人员编写完软件后,通过 Java 编译器将 Java 源程序编译为 JVM 的字节代码。任何一台机器只要配备了 Java 解释器,就可以运行这个程序,而不管这种字节码是在何种平台上生成的。Java 还采用了基于 IEEE 标准的数据类型。通过 JVM 保证数据类型的一致性,也确保了 Java 的平台无关性。图 1-1 描述了源程序 MyApp.java 在不同平台上的编译与执行过程。将源程序 MyApp.java 编译成类文件 MyApp.class 后,通过在不同平台上安装相应的 Java 虚拟机,使类文件 MyApp.class 在不同平台上可以解释执行,体现了 Java 的跨平台性。

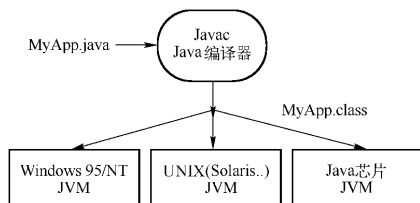


图 1-1 Java 语言的编译执行过程

## 5. 多线程

C 和 C++ 采用单线程体系结构,而 Java 却提供了多线程支持。Java 在两方面支持多线程。一方面 Java 环境本身就是多线程的。若干个系统线程运行负责必要的无用单元回收、系统维护等系统级操作;另一方面 Java 语言内置多线程控制,可以大大简化多线程应用程序开发,实现支持多任务功能。

Java 提供了 Thread 类和一组内置的方法,负责启动运行、终止线程,并可检查线程状态。Java 的线程还包括一组同步原语。这些原语负责对线程实行并发控制。利用 Java 的多线程编程接口,开发人员只需继承 Thread 类和调用相应方法,可以方便地写出支持多线程的应用程序,提高程序执行效率。然而 Java 的多线程支持在一定程度上受运行时支持平台的限制。如果操作系统本身不支持多线程,Java 的多线程特性可能就表现不出来。

## 6. 分布性

分布包括数据分布和操作分布。数据分布是指数据可以分散在网络中的不同主机上,操作分布是指把一个计算分散在不同主机上处理。Java 支持这两种分布性。Java 提供了内容丰富的网络类库,可以处理 TCP/IP 协议。Java 提供了一个叫作 URL 的对象,利用这个对象,可以打开并访问网络资源,访问方式就像访问本地文件系统一样简单。Java 的 Applet 小程序可以从服务器下载到客户端,将部分计算在客户端进行,提高系统的执行效率。

# 1.3 Java 语言平台及相关应用

Java 平台发展到 Java 2 版本以后,为了适应不同级别计算机硬件的开发需要,按应用主要分为三个主要分支:Java 2, Standard Edition(J2SE);Java 2, Enterprise Edition (J2EE);Java 2, Micro Edition (J2ME)。这三块相互补充,分别应用在不同的范围。J2SE 是 Java 2 的标准版,为台式机而设计,主要用于桌面应用程序的编程,运行在 Linux, Solaris, 或 Microsoft Windows 等操作系统上。J2EE 是 Java 2 的企业版,是一个适合分布式的、多用户、企业级应用系统运转的平台,主要用于分布式网络程序的开发,如电子商务网站和 ERP 系统。J2ME 是为了支持像掌上电脑、手机等小型的嵌入式或移动设备而推出的一系列的技术和规范的总称,主要应用于嵌入式系统的开发。

### 1.3.1 Java 2 标准版本——J2SE

Java 之所以能够发展得很快,是因为互联网的发展非常快。Java 可以和浏览器结合在一

起,因而很自然地在桌面系统得到广泛应用。Java 2 标准版是一种开发和部署平台,提供了编写桌面、工作站应用程序所有必需的功能。J2SE 软件是快速开发、部署关键任务和企业应用程序的首选解决方案。J2SE 1.4 版本是基于 Java 跨平台技术和强有力的安全模块而开发的,其最新的特征和功能极大地提高了 Java 语言的伸缩性、灵活性、适用性及可靠性。本书是以 J2SE 1.4 版本为基础的。以下是 J2SE 1.4 的特点。

#### (1) 良好的计算性能和缩放能力

J2SE 1.4 版是一个快速开发和配置跨平台的企业级应用与服务的综合性平台。它提升了 Java 的计算性能与缩放能力,代表了 Java 技术的大跨步进展。有了 J2SE 1.4 版,企业就可以通过更简易的步骤,花费更少的时间,采用 Java 技术开发与配置要求更高的应用了。J2SE 1.4 版在 Java 客户机应用方面得到了重大性能提升。基于 Swing 的 Java 小程序和应用程序在图形用户接口的响应速度上也大有提高。

#### (2) 跨平台

J2SE 1.4 版是一个具有综合性能的平台,能使一个完整的、可缩放的高性能应用配置在任何种类的平台之上。

#### (3) 支持 Web 服务

J2SE 1.4 版与以前各 J2SE 版全面兼容。J2SE 1.4 版对 SSL、LDAP 和 CORBA 等业界标准技术也给予支持,因此保证了在不同平台、系统和环境间的互操作能力。J2SE 1.4 版支持 XML 技术和一整套安全服务,因此为基于各类标准的可互操作应用与服务的创建奠定了重要基础。有了 J2SE 1.4 版与 Java Web Services Developer Pack 的结合,开发商就能创建、测试和配置 Web 服务了。

#### (4) 完整性

J2SE 1.4 版还是一个完整的开发环境。开发商可基于这一环境创建功能丰富的交互式应用。这些应用可以配置在任何平台上。J2SE 1.4 具有新的 GUI 控制功能、快速的 Java 2DT 图形功能、支持国际化与本地化扩展和新的配置选项,并对 Windows XP 提供扩展支持。

J2SE 1.4 版是多种不同风格软件的开发基础,包括客户端 Java 小程序和应用程序,以及独立的服务器应用程序等。J2SE 1.4 版也是 J2ME 的基础,是 J2EE 执行程序的推动力。

### 1.3.2 Java 2 企业版本——J2EE

J2EE 是一个开放的用于构建以服务端计算为核心的、基于 Web 的、N 层模块化的、企业应用开发和部署的平台,同时也是所有兼容 J2EE 标准的应用服务器产品的统一标识。J2EE 技术在 J2SE 的基础之上,提供了企业计算所必须的服务,如事务服务、安全性服务、消息服务等。J2EE 应用组件如企业 Enterprise JavaBean(EJB)、JSP 和 Servlet 运行于 J2EE 容器之中,通过连接器访问企业信息系统,如数据库系统、ERP 系统和其他应用程序系统。J2EE 应用可以集成一系列的客户端,包括独立运行的台式客户端、无线客户端,以及基于 Web 浏览器的客户端等,为开发企业应用提供了高性能、高可靠性和可伸缩性的运行支撑环境。

J2EE 将组成一个完整企业级应用的不同部分纳入不同的容器(Container),每个容器中都包含若干组件(这些组件是需要部署在相应容器中的),同时各种组件都能使用各种 J2EE Service/API。图 1-2 是 J2EE 的结构。J2EE 容器包括以下容器。

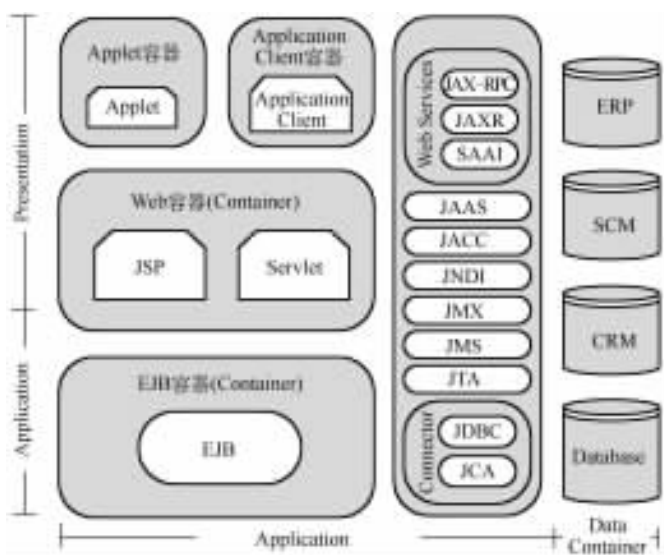


图 1-2 J2EE 的结构

### (1) Web 容器

是服务器端容器,包括 JSP 和 Servlet 两种组件。JSP 和 Servlet 都是 Web 服务器的功能扩展,接受 Web 请求,返回动态的 Web 页面。Web 容器中的组件可使用 EJB 容器中的组件完成复杂的商务逻辑。

### (2) EJB 容器

是服务器端容器,包含的组件为 EJB,它是 J2EE 的核心之一,主要用于服务器端商业逻辑的实现。EJB 规范定义了一个开发和部署分布式商业逻辑的框架,以简化企业级应用的开发,使其较容易地具备可伸缩性、可移植性、分布式事务处理、多用户和安全性等。

### (3) Applet 容器

是客户端容器,包含的组件为 Applet。Applet 是嵌在浏览器中的一种轻量级客户端,一般而言,仅当使用 Web 页面无法充分地表现数据或应用接口的时候,才使用它。Applet 是一种替代 Web 页面的手段,只能使用 J2SE 开发,无法使用 J2EE 的各种 Service 和 API,这是为了安全性的考虑。

### (4) Application 容器

Application 是客户端容器,包含的组件为 Application Client。Application Client 相对 Applet 而言是一种较重量级的客户端,它能够使用 J2EE 的大多数 Service 和 API。

通过这 4 个容器, J2EE 能够灵活地实现面向企业级应用的模型-视图-控制器(Model-View-Controller)架构。

在客户端, J2EE 实现视图功能,它提供了三种手段: JSP(或 Servlet)、Applet 和 Application Client,分别能够实现面向浏览器的数据表示和面向桌面应用的数据表示。Servlet 是实现 Controller 部分业务流程控制的主要手段,而 EJB 则主要针对 Model 部分的业务逻辑实现。至于与各种企业资源和企业级应用相连接,则是依靠 J2EE 的各种服务和 API。

J2EE 提供了一个建立在开放和标准的技术之上、非常灵活的端到端的多层体系架构,从而满足各种不同企业应用的需求,由于 J2EE 规范和标准的制定主要由 Sun 公司领导,IBM,

BEA 等许多公司参与下进行。因此 J2EE 每一个新规范的推出,都体现着整个业界对技术的共同认同。大家都遵守着“在标准上进行合作,在产品上进行竞争”的原则,从而使 J2EE 在技术规范上日臻完善和进步。在同一标准的基础上,为用户提供多样化的产品选择。

### 1.3.3 Java 2 微型版本——J2ME

J2ME 是为了支持像 PDA、手机等小型嵌入式或移动设备而推出的一系列的技术和规范的总称。它借用了 J2SE 类库的一部分,使用了更少量的 API。由于 J2ME 用于嵌入式设备应用的开发,它采用的 JAVA 虚拟机(JVM)比 J2SE 的 JVM 也小得多。

通过 J2ME 实现无线应用平台,通过将 Java 程序下载到本地执行,可以解决以前手机等无线设备内置了应用程序却无法改变与更新的问题。以手机为例,现在手机厂商还存在两个普遍的问题:一个是缺乏开放的编程平台,另一个是只能通过 WAP 方式访问 Internet,没有其他方式实现。通过应用 J2ME 技术开发手机的应用软件,就可以解决这两个问题:首先,Java 语言是跨平台运行的,软件开发商可以很容易地开发应用程序,也可以方便地安装到手机中;其次, J2ME 提供了 HTTP、TCP 等 Internet 协议,Java 程序可以自由地访问 Internet。使手机的应用多样化。如手机游戏,用户再也不用翻来覆去地玩有限的几款手机内置游戏。当用户玩腻了一款游戏之后,可以放心地将它删除,然后用手机到相关网站去下载更新的游戏。它也能为更多程序创造环境,对于一些商务人士,可以将涉及金融、证券和保险的各种应用程序安装到手机中。用户还可以通过手机进行购物、酒店和机票预订、阅读新闻、查看天气等信息和下载电子出版物。

J2ME 内部分三层,从下到上分别是虚拟机(Java Virtual Machine JVM)、配置层(Configuration)和框架层(Profile)。JVM 负责建立 Java 虚拟机,解释 Java 字节码,Configuration 建立了核心类库,实现基本功能,Profile 建立了高级类库,提供更加丰富的功能。由于 Java 的跨平台特点,它可以应用于异构的平台。J2ME 要支持的硬件平台也有很大差异,其中包括高端设备,例如电视机的机顶盒、网络电视等;以及比较低端的设备,如手机、寻呼机等。为了满足不同硬件的开发要求, J2ME 规定了按需配置的概念,即针对不同级别的硬件,所使用的 JVM 和基础 API 集合方面做了相应配置。高端设备采用 CDC(Connected Device Configuration)标准配置,所使用的 JVM 称为 CVM;低端设备采用 CLDC(Connected Limited Device Configuration)配置,所用的 JVM 为 KVM。CDC 和 CLDC 仅仅是对各类设备中最具共性的配置提供了基本的功能集合。MIDP(Mobile Information Device Profile)以 CLDC 为基础,是第一个制订完成并可供使用的 J2ME 应用程序运行环境。在 MIDP 的基础上可以进行手机等移动设备的 Java 应用的开发。

目前,支持 MIDP,可以运行 J2ME 程序的设备很多,其中,手机包括 Motorola A388、T720, Nokia 6800、7210 等。到 2003 年 2 月,已经有 21 个生产厂家推出了 100 多种支持 Java 的手持设备。根据 ARC Group 2002 年 6 月的《无线 Java 报告》到 2007 年,几乎所有的手持设备都将提供对 Java 的支持。J2ME 在未来的几年里仍然将持续升温。