

高等学校计算机科学与技术教材

Java 程序设计实用教程

Computer
马迪芳 宋泾舸 王 芳 编著



清华大学出版社

<http://www.tup.tsinghua.edu.cn>



北京交通大学出版社

<http://press.bjtu.edu.cn>

高等学校计算机科学与技术教材

Java 程序设计实用教程

马迪芳 宋泾舸 王 芳 编著

清华大学出版社
北京交通大学出版社
·北京·

内 容 简 介

本书是基于 Java 2 的基础教程,对 Java 语言的内容、功能、特性和实际运用作了深入浅出、系统全面的介绍。本书共分 10 章,前 4 章主要介绍了 Java 基本语法和面向对象技术,其中包括 Java 语言的词法、数据类型、操作符与表达式、流程控制、数组、字符串处理、集合框架、Java 类和对象、包、接口等基础知识。后面 6 章深入地介绍了 Java 的编程技术,包括异常处理、线程、输入输出流、图形用户界面设计、Applet 小应用程序、网络通信等内容。本书提供了丰富的典型实例,各章附有一定量的习题,便于读者思考和复习。

本书可作为高等院校计算机及其相关专业的教学用书,同时也适用于 Java 2 的初学者和具有一定 Java 编程经验的开发人员。

版权所有,翻印必究。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

图书在版编目(CIP)数据

Java 程序设计实用教程/马迪芳,宋泾舸,王芳编著. —北京:清华大学出版社;北京交通大学出版社,2004.9

(高等学校计算机科学与技术教材)

ISBN 7-81082-333-7

I. J… II. ①马… ②宋… ③王… 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 063503 号

责任编辑:谭文芳 特邀编辑:陈晓莉

出版者:清华大学出版社 邮编:100084 电话:010-62776969

北京交通大学出版社 邮编:100044 电话:010-51686045, 62237564

印刷者:北京瑞达方舟印务有限公司

发行者:新华书店总店北京发行所

开本:185×260 印张:20.5 字数:525 千字

版次:2004 年 9 月第 1 版 2004 年 9 月第 1 次印刷

书号:ISBN 7-81082-333-7/TP·131

印数:1~5000 册 定价:27.00 元

前 言

Java 语言的诞生给 IT 产业带来了一次变革, 由于 Java 语言的平台无关性与互联网发展的紧密结合, 使它成为目前最流行的软件开发语言之一。Java 具有简单高效、面向对象、不依赖于机器结构的特点, 用其开发的应用软件在可移植性、健壮性、安全性等方面明显优于已存在的其他语言。Java 语言有着广阔的应用前景, 它不但适合于网络程序设计, 也适合于一般的大规模软件工程项目。目前, Java 产业在国内正在如火如荼地发展着, 广泛地应用在电子商务、金融、证券、邮电、电信等行业领域。因此为了迎接信息时代的挑战, 学习和掌握 Java 语言无疑会带来更多的机遇。

本书是基于 Java 2 的基础教程, 对 Java 语言的内容、功能、特性和实际运用作了深入浅出、系统全面的介绍。全书布局合理, 结构严谨, 重点突出。通过本书的学习, 能够很快地掌握 Java 语言程序设计的方法和技巧, 同时对面向对象程序的设计也会有较深入的理解。本书既可以作为高等院校本科生和研究生学习 Java 语言的教材和参考书, 也可作为软件开发者的自学用书。

本书共分 10 章, 前 4 章主要介绍了 Java 基本语法和面向对象技术, 其中包括 Java 语言的词法、数据类型、操作符与表达式、流程控制、数组、字符串处理、集合框架、Java 类和对象、包、接口等基础知识。后面 6 章深入地介绍了 Java 的编程技术, 包括异常处理、线程、输入输出流、图形用户界面设计、Applet 小应用程序、网络通信等内容。

为了方便读者学习, 提供了丰富的典型实例, 各章附有习题。

本书第 1, 7, 8, 9 章由宋泾舸编写, 第 3, 5, 6 章由马迪芳编写, 第 2, 4, 10 章由王芳编写, 全书最后由马迪芳进行统校定稿。田小萍、曲和政和毛华帮助测试了程序, 特此致谢。

由于我们水平有限, 书中难免存在许多不足之处, 恳请读者批评指正。

编 者

2004 年 7 月

目 录

第1章 Java 概述	(1)
1.1 Java 语言的发展历史	(1)
1.2 Java 语言的特点	(2)
1.3 Java 的运行机制	(4)
1.4 Java 运行平台	(6)
1.4.1 Java 运行环境	(6)
1.4.2 Java 应用程序接口	(6)
1.4.3 Java 2 SDK 的安装与组成	(7)
1.5 常用 Java 开发工具简介	(10)
1.6 一个简单的 Java 程序	(11)
小结	(15)
习题	(15)
第2章 Java 语言基础 (一)	(16)
2.1 Java 语言的词法	(16)
2.1.1 空格与分号	(16)
2.1.2 注释	(16)
2.1.3 标识符	(17)
2.1.4 关键字	(17)
2.2 变量和常量	(18)
2.2.1 常量	(18)
2.2.2 变量	(19)
2.3 简单数据类型	(20)
2.3.1 数据类型概述	(20)
2.3.2 整型数据	(21)
2.3.3 实型数据	(22)
2.3.4 字符型数据	(23)
2.3.5 布尔型数据	(24)
2.4 操作符与表达式	(24)
2.4.1 运算符与表达式概述	(24)
2.4.2 算术运算符与算术表达式	(25)
2.4.3 关系运算符与关系表达式	(26)
2.4.4 布尔逻辑运算符和布尔逻辑运算表达式	(26)
2.4.5 位运算符和位运算表达式	(27)
2.4.6 赋值运算符和赋值表达式	(29)
2.4.7 条件运算符和条件表达式	(30)

2.4.8 各类运算符的优先级	(31)
2.4.9 各类型数据间的相互转换	(32)
2.5 流程控制	(32)
2.5.1 流程控制结构概述	(32)
2.5.2 分支结构	(34)
2.5.3 循环结构	(38)
2.5.4 break 语句和 continue 语句	(41)
习题	(43)

第3章 Java 面向对象技术	(45)
3.1 面向对象编程的基本概念	(45)
3.1.1 对象、类和消息	(45)
3.1.2 继承和多态	(46)
3.1.3 面向对象程序设计	(47)
3.2 类	(48)
3.2.1 类的定义	(48)
3.2.2 成员变量	(49)
3.2.3 定义方法	(50)
3.2.4 构造方法	(54)
3.3 对象	(56)
3.3.1 创建对象	(56)
3.3.2 对象的使用	(59)
3.3.3 清除对象	(60)
3.4 类成员和实例成员	(60)
3.4.1 类变量和实例变量	(61)
3.4.2 类方法和实例方法	(61)
3.4.3 类变量和实例变量的初始化	(63)
3.5 包	(65)
3.5.1 创建包	(65)
3.5.2 包的使用	(66)
3.5.3 包对文件的管理	(67)
3.6 类成员的访问权限控制	(68)
3.7 类的继承	(72)
3.7.1 继承的实现	(72)
3.7.2 成员变量的隐藏和方法重写	(73)
3.7.3 super 关键字	(74)
3.7.4 运行时多态	(77)
3.7.5 instanceof 操作符	(81)
3.7.6 java.lang.Object 类	(81)
3.7.7 final 方法和类	(84)
3.7.8 abstract 类和方法	(84)
3.8 内部类	(86)
3.8.1 内部类的定义和使用	(86)

3.8.2	创建内部类对象的几种方式	(87)
3.8.3	方法中定义的内部类	(88)
3.8.4	static 内部类	(89)
3.9	接口	(90)
3.9.1	接口概念	(90)
3.9.2	定义接口	(91)
3.9.3	实现接口	(92)
3.9.4	接口类型的使用	(93)
3.10	Java 程序结构	(94)
习题		(95)
第 4 章	Java 语言基础 (二)	(96)
4.1	基本数据类型的类包装	(96)
4.2	数组	(97)
4.2.1	一维数组	(97)
4.2.2	多维数组	(100)
4.3	字符串	(104)
4.3.1	基本概念	(104)
4.3.2	String 类和 StringBuffer 类	(104)
4.4	集合框架	(118)
4.4.1	Collection, Set 和 List 接口	(119)
4.4.2	Map 接口	(123)
4.4.3	Iterators 反复器	(126)
4.4.4	排序和搜索	(127)
习题		(132)
第 5 章	异常处理	(134)
5.1	异常	(134)
5.1.1	基本概念	(134)
5.1.2	异常处理机制	(134)
5.2	Throwable 类及其子类	(135)
5.3	异常处理	(138)
5.3.1	捕获异常	(138)
5.3.2	声明异常	(143)
5.4	抛出异常	(144)
5.5	自定义异常类	(146)
5.6	综合应用举例	(148)
5.7	Java 异常处理的优点	(150)
习题		(150)
第 6 章	线程	(151)
6.1	线程概念	(151)

6.2	线程的实现	(151)
6.2.1	Thread 类	(152)
6.2.2	Runnable 接口	(153)
6.3	线程的生命周期	(156)
6.4	线程优先级和线程调度	(158)
6.5	多线程资源共享与同步	(161)
6.5.1	多线程访问共享资源的加锁机制	(162)
6.5.2	多线程的协作同步运行机制	(165)
6.5.3	死锁问题	(172)
6.5	Daemon 线程	(172)
6.6	线程组	(172)
	习题	(174)
第 7 章	Java 输入输出流	(175)
7.1	输入输出流的基本概念	(175)
7.1.1	理解流的概念	(175)
7.1.2	标准输入输出流	(176)
7.2	基本字节流	(178)
7.2.1	InputStream 类和 OutputStream 类	(178)
7.2.2	FileInputStream 和 FileOutputStream	(179)
7.2.3	ByteArrayInputStream 和 ByteArrayOutputStream	(181)
7.3	基本字符流	(183)
7.3.1	Reader 类和 Writer 类	(183)
7.3.2	FileReader 和 FileWriter	(184)
7.3.3	CharArrayReader 和 CharArrayWriter	(186)
7.4	文件操作	(188)
7.4.1	File 类	(189)
7.4.2	文件过滤器	(190)
7.4.3	随机存取文件流类	(191)
7.5	java.io 包中其他流类	(194)
7.5.1	缓冲流	(194)
7.5.2	字符串流	(196)
7.5.3	带格式数据流	(197)
7.5.4	管道流	(200)
7.5.5	打印流	(203)
	小结	(206)
	习题	(206)
第 8 章	编写图形用户界面应用程序	(208)
8.1	Swing 概述	(208)
8.1.1	什么是 Swing	(208)
8.1.2	Swing 的特点	(208)

8.1.3	Swing 包的组成	(209)
8.2	Swing 组件和容器	(210)
8.2.1	样题分析	(210)
8.2.2	Swing 组件分类	(212)
8.2.3	顶层容器	(219)
8.2.4	创建窗体	(221)
8.2.5	创建对话框	(222)
8.2.6	面板 (Panel)	(224)
8.2.7	创建菜单 (Menu)	(224)
8.2.8	JComponent 类	(227)
8.3	布局管理器	(229)
8.3.1	FlowLayout (流式布局)	(229)
8.3.2	BorderLayout (边界布局)	(230)
8.3.3	GridLayout (网格布局)	(232)
8.3.4	BoxLayout (框式布局)	(234)
8.3.5	GridBagLayout (网格包布局)	(236)
8.4	事件处理	(242)
8.4.1	基本概念	(242)
8.4.2	事件处理机制	(242)
8.4.3	实现事件处理的几种方式	(243)
8.4.4	Swing 组件支持的监听器类型	(245)
8.4.5	监听器接口 API 总结	(247)
8.5	绘画与动画显示	(248)
8.5.1	Swing 组件的显示过程	(248)
8.5.2	基本绘图方法	(249)
8.5.3	绘制简单图形和文字	(251)
8.5.4	使用图像	(258)
8.5.5	实现动画	(260)
小结		(263)
习题		(263)
第 9 章	Applet 小应用程序	(266)
9.1	Applet 概述	(266)
9.1.1	什么是 Applet	(266)
9.1.2	运行 Applet 程序的步骤	(266)
9.1.3	Applet 的安全性限制	(268)
9.1.4	Applet 的类层次结构	(268)
9.1.5	Applet 程序与应用程序的区别	(269)
9.2	Applet 的生命周期及主要方法	(270)
9.3	Applet 中添加组件和绘画	(272)
9.4	Applet 的 HTML 标记	(275)
9.5	Applet 应用	(277)
9.5.1	显示图像	(277)

9.5.2 播放声音	(278)
9.5.3 Applet 与浏览器之间的通信	(279)
小结	(281)
习题	(282)
第 10 章 网络通信	(284)
10.1 网络基础知识	(284)
10.1.1 计算机网络的概念	(284)
10.1.2 TCP/IP 协议集	(284)
10.2 使用 URL 访问网络资源	(287)
10.2.1 什么是 URL	(287)
10.2.2 创建 URL	(288)
10.2.3 使用 URL 来读取 WWW 信息	(290)
10.2.4 使用 URLConnection 读取网络资源	(291)
10.3 Socket 通信	(293)
10.3.1 什么是 Socket	(293)
10.3.2 Socket 通信机制	(294)
10.3.3 建立一个服务器端和客户端的 Socket 通信	(298)
10.3.4 服务器支持多客户机通信	(304)
10.4 数据报通信	(307)
10.4.1 什么是数据报	(307)
10.4.2 数据报通信机制	(307)
10.4.3 建立一个服务器端和客户端的数据报通信	(308)
10.4.4 广播式通信	(311)
习题	(313)
附录 A JCP 提供的主要方法列表	(314)
参考文献	(317)

第 1 章 Java 概述

随着网络技术的发展，一种基于网络的程序设计语言——Java 语言正受到越来越多的开发人员和企业的重视。Java 语言是网络时代的一种重要的程序设计语言，Java 技术是重要的网络开发技术。在基于网络的应用不断扩大的今天，Java 语言和基于 Java 的技术得到了广泛的应用，成为继 C/C++ 后的又一通用的程序开发技术。

1.1 Java 语言的发展历史

1995 年，当 Java 1.0 首次发布时，Java 语言及相关技术还只是作为编写网页中的可交互组件而提出的“小玩意儿”，经过近十年的发展，目前 Java 技术已涉及与网络应用相关的众多领域，成为一种网络时代通用的程序设计语言。为了更好地了解 Java 技术的发展状况，有必要对其发展史作一简要介绍。

Java 语言的产生可以追溯到 1990 年 12 月，当时美国 Sun Microsystem 公司的高级专家 Patrick Naughton、Mike Sheridan、James Gosling 成立了一个叫做 Green Team 的小组，开发一种分散式系统架构，使其能在电冰箱、电视机、PDA、手机等消费类电子产品的操作平台上运行，特别是对交互式电视机的解决方案（但在交互式电视机的项目上没有中标）。1992 年 9 月，Green Team 开发了一种名为 Star 7 的产品，类似于今天使用的 PDA，但功能比现在的 PDA 更强大。最初的方案是采用 C++ 语言进行基于 Star 7 的系统与应用开发。但是，在开发过程中，James Gosling 感到 C++ 语言存在复杂和易出错的缺点，于是产生了建立一种新的程序设计语言的想法，经过一段时间的工作，在 C/C++ 语言基本语法的基础上创立了一种新的面向对象程序设计语言，并从他办公室窗外的橡树获得灵感将这种语言取名为 Oak（橡树）语言，希望它像橡树一样生机勃勃，在未来的市场上茁壮成长。但是在申请知识产权的过程中发现 Oak 已被另外一个产品注册，于是将这种语言改名为 Java 语言，这就是 Java 语言最早的雏形，James Gosling 也因此被称为“Java 之父”。

在以后的几年里，Java 语言在 Sun Microsystem 公司内不断得到发展和完善。1995 年，Java 语言的第一个正式版本 Java 1.0 向世界发布，重点是帮助网页设计人员建立可嵌入网页进行交互工作的组件程序——Applet（小应用程序）。网景公司在其著名的浏览器 Netscape Navigator 中首先支持了 Java 技术，从而使这一新的语言开始为人们所认识。

此后，Java 作为一种网络环境下的编程语言和开发技术开始了它的全面发展的历程。下面是 Java 发展中的一些主要事件。

- 1996 年，Java Development Kit (JDK) v1.0 发布，包含核心层的功能（套接字编程，文件 I/O，图形用户接口 GUI 等）。
- 1997 年，JDK 1.1 发布，在 Java GUI、JDBC 数据控制、RMI 分布对象和其他一些方面进行了增强。
- 1998 年，JDK 1.2（称 Java 2）发布，包括 JFC、Swing 等。当年 JFC/Swing 被下载了 50 多万次，促进了 Java 语言在世界范围的传播。
- 1999 年，Java 技术被分成 J2SE（标准版）、J2EE（企业版）、J2ME（微型版）三个平台。

采用 Java HotSpot 技术提高了虚拟机的执行性能。同年, Java Server Pages (JSP) 技术公诸于众, 进一步扩展了 Java 技术在服务器端的应用。针对 Linux 操作系统的 J2SE 平台也在当年发布, 拓展了 Java 作为平台无关技术的应用范围。

- 2000 年, JDK 1.3 发布, 包括 Java APIs for XML。同时, Sun Microsystem 公司将 J2EE 授权给 ATG、BEA Systems、Gemstone、Silverstream 和一些其他公司共同开发基于 J2EE 框架的网络技术。当年世界范围的 Java 开发者成员上升到 150 万。
- 2003 年, Java 在 Web 服务和移动通信领域有了更大发展。世界范围的开发者成员上升到 300 万。2003 年 10 月, Sun Microsystem 公司副总裁、“Java 之父” James Gosling 在中国第六届全国 Java 技术大会上表示, 到 2005 年, Java 技术团队将从目前的 300 万个发展到 1000 万个。

现在, Java 语言已发展成为应用广泛的程序设计语言, 主要应用领域包括:

- 因特网 (网页、Web 服务);
- 关系数据库;
- 大型计算机;
- 嵌入式与移动计算 (如手机、个人数字助理等);
- 智能卡、信用卡;
- 企业信息平台;
- 科技研究与交流;
- 商务;
- 娱乐。

随着我国信息化建设步伐的加快, Java 语言及相关技术正成为 IT 企业建立网络化信息平台的有力工具。

1.2 Java 语言的特点

概括起来, Java 语言有如下特点。

1. 完全的面向对象性

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为对象。对象之间通过消息相互作用。另外, 现实世界中任何实体都可归属于某类事物, 任何对象都是某一类事物的实例。如果说传统的过程式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。也就是说, 过程式编程语言是“程序 = 算法 + 数据”, 而面向对象编程语言则为“程序 = 对象 + 消息”。

C++语言是一种功能强大的面向对象语言, 但考虑到同 C 语言 (过程式语言) 的兼容性, 不能做到完全的面向对象。Java 语言放弃了对其他语言的兼容, 在继承了 C/C++语言基本语法的基础上, 形成了完全面向对象的程序结构, 更好地体现了面向对象程序的特征。

2. 平台无关性

所谓平台无关性是指用 Java 语言编写并生成的应用程序 (Java 程序) 不用修改就可在不同的软硬件平台上运行。Java 采用虚拟机方式在目标码级实现平台无关性。

Java 的平台无关性具有深远意义。首先,它使得编程人员所梦寐以求的事情(开发一次软件在大多数平台上运行)变成事实,这将大大加快和促进软件产品的开发。其次,Java 的平台无关性正好迎合了“网络计算机”(NC)的思想。如果大量常用的应用软件(如字处理软件等)都用 Java 重新编写,并且放在某个 Internet 服务器上,那么具有 NC 的用户将不需要占用大量空间安装软件,他们只需要一个 Java 解释器,每当需要使用某种应用软件时,下载该软件的字节码即可,运行结果也可以发回服务器。另外,Java 采用的是基于 IEEE 标准的数据类型。通过虚拟机保证数据类型的一致性,也确保了 Java 的平台无关性。

目前,已有一些公司开始使用这种新型的计算模式构筑企业内部的信息系统。

3. 分布性

分布性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 WWW 客户-服务器计算模式,因此它支持这两种分布性。对于前者,Java 提供了一个叫做 URL 的对象,利用这个对象,可以打开并访问具有相同 URL 地址的对象,访问方式与访问本地文件系统相同。对于后者,Java 的 Applet 小应用程序可以从服务器下载到客户端,即部分计算在客户端进行,提高系统执行效率。

Java 提供了一整套网络类库,开发人员可以利用类库进行网络程序设计,方便地实现 Java 的分布式特性。

4. 可靠性和安全性

Java 最初的设计目的是将它应用于电子类消费产品,因此要求较高的可靠性。Java 虽然源于 C++,但它去除了许多 C++语言的不可靠因素,可以防止许多编程错误。首先,Java 是强类型的语言,要求显式的方法声明,这保证了编译器可以发现方法调用错误,保证程序更加可靠。其次,Java 不支持指针,这杜绝了内存的非法访问。第三,Java 的自动垃圾回收机制防止了内存丢失等动态内存分配导致的问题。第四,Java 解释器运行时实施检查,可以发现数组和字符串访问的越界。最后,Java 提供了异常处理机制,程序员可以把一组可能产生运行异常的代码放在一个地方,这样可以简化异常的处理过程,使运行任务便于恢复。

网络应用程序对安全性有较高的要求。如果没有安全保证,用户从网络下载程序执行就变得非常危险,系统容易受到恶意程序的破坏。Java 通过自己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破坏。当 Java 字节码进入解释器时,首先必须经过字节码校验器的检查,然后,Java 解释器将决定程序中类的内存布局,随后,类装载器负责把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。最后,客户端用户还可以限制从网络上装载的类只能访问某些文件系统。上述几种机制结合起来,使得 Java 成为安全的编程语言。

5. 动态性

Java 语言是为一个开放的、可扩展的环境而设计的。在不改变用户程序的情况下,采用包和归档文件机制对应用程序及类库进行管理,使它更容易实现类库的扩充,包括加入新的成员方法和成员变量。由于 Java 语言的这种良好的动态特性,使得许多开发人员可以在别人开发的应用程序和类库的基础上工作,比较容易地扩充自己的功能和应用,从而提高大型软件项目的开发效率和质量。

6. 多线程

Java 在两方面支持多线程。一方面, Java 运行环境本身就是多线程的, 若干个运行的系统线程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。Java 提供了一个类 Thread, 由它负责线程的启动、运行和终止, 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。当然, 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 的多线程特性就表现不出来。

7. 易用性

前面提到, Java 语言是在 C/C++语言基本语法的基础上经简化而发展起来的。它的创立目的是在保证程序安全性的基础上, 使程序设计过程更加方便、容易。因此语言中摒弃了 C++语言的复杂、易产生错误的语法特征, 如指针、全局变量等。此外, Java 除保留了一些简单的数据类型外(如整数、浮点数和字符等), 其他较复杂的数据类型都被处理成对象, 更加突出了面向对象的编程特征。

值得注意的是, Java 语言的易用性是相对其他通用程序设计语言而言的, 对于初学者而言, 要想掌握这样一种功能强大的编程语言, 仍然不是一件十分容易的事, 需要不断的实践。

1.3 Java 的运行机制

1. Java 虚拟机

Java 语言是一种解释型的程序设计语言, 即 Java 程序需要通过 Java 程序解释器的解释之后才能执行, 这个解释器就是前面提到过的 Java 虚拟机的一部分。

虚拟机(Virtual Machine)是一种用软件模拟硬件功能实现的“虚拟”计算机。Java 语言采用虚拟机技术实现 Java 程序的跨平台运行, 这就是 Java 虚拟机(Java Virtual Machine, JVM)。通常 JVM 在形式上表现为一个软件。与常见的计算机硬件结构类似, 虚拟机中定义了指令集、寄存器集、类文件结构栈、垃圾收集堆、内存等, 从而实现了计算机的基本框架。

与传统的解释型语言如 BASIC 等不同的是, Java 语言源程序(.java 文件)需要首先被编译成一种称为字节码的二进制文件(.class 文件), 而不是机器码文件, 然后 Java 解释器解释执行这种字节码文件。这一过程如图 1-1 所示。

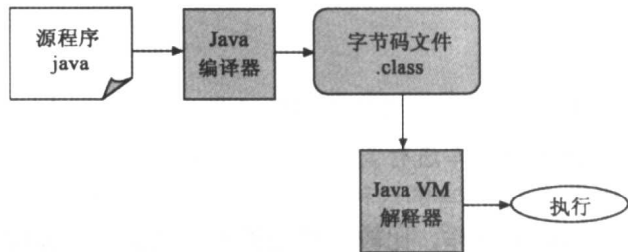


图 1-1 Java 程序的执行机制

也可以用硬件实现 Java 虚拟机。目前, Sun 公司已经设计实现了 Java 芯片, 主要使用在网络计算机上。Java 芯片的出现也会使 Java 更容易嵌入到家用电器中。Java 编程人员在编写完软件后, 通过 Java 编译器将 Java 源程序编译成 JVM 的字节码。任何一台机器只要安装了 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开发环境中生成的。目前, JVM 已有多种软硬件平台的版本, 可以适应绝大多数的开发与应用。

Java 语言是一种网络环境的程序设计语言, 为了提高其安全可靠, Java 虚拟机在运行 Java 字节码程序时提供了垃圾回收机制、代码安全机制等。

2. 垃圾回收机制

垃圾回收机制是 Java 语言为了减轻编程人员在内存资源分配上的负担而设计的一种内存资源自动回收机制。在 C/C++ 语言中, 程序员不仅要通过编程为程序中的数据分配内存, 而且常常为了充分利用有限的内存资源, 通过编程回收不再使用的内存。这一过程一方面增加了程序员的编程负担, 另一方面也会因程序员的疏忽造成资源的浪费。Java 语言中的资源回收是自动完成的。系统在程序执行过程中跟踪分配给对象的内存资源的使用情况, 定期检查出不再使用的内存并由系统自动回收这部分内存做再次分配。这样可以提高内存资源的使用效率, 同时减小了系统和软件出现故障的几率, 提高了软件的安全性与可靠性。

3. 代码安全机制

Java 安全模式的重点在于保护最终用户不受从网上下载来的破坏性程序的干扰。为了达到这个目的, Java 在扩展其技术应用领域的同时, 也在不断改进其安全模型。Java 1.0 提供了一个专用的运行 Java 程序的沙箱 (sandbox)。沙箱内可做任何事情, 但超出此边界就不能有任何操作。例如, 对于运行于网页中的 Applet, 通过限定下载代码的可执行操作的范围, Java 安全模式可使用户免受破坏性程序的威胁。

在安全沙箱中, JVM 的一个重要方面是其类载入程序。该程序负责输入那些定义运行程序的类和接口的二进制数据。图 1-2 是 JVM 类装载的过程。通过类载入程序, Java 系统可以确认被载入程序是否安全。

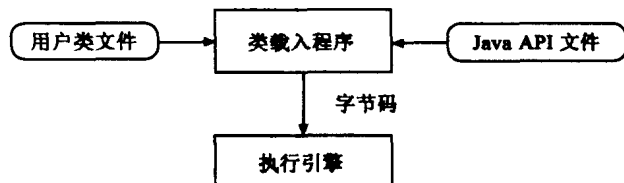


图 1-2 Java 系统的类载入过程

Java 1.1 采用的安全模型是信任度模型, 用户可以有选择地授权给远程代码, 如果远程某地资源可以被信任, 那么带有此地签名的代码将被授权可以访问本地的系统资源, 其他不被信任的代码仍受限在沙箱内。Java 2 采用了域管理模型, 无论是本地代码或远程代码, 都可以通过配置来设定可访问的资源, 从而更好的支持企业及应用。

1.4 Java 运行平台

1.4.1 Java 运行环境

Java 运行时环境 (Java Runtime Environment, JRE) 是由 JVM 和 Java 应用程序接口 (Java API) 组成的一个软件系统。是 JVM 机制的软件实现和跨平台运行的基础。一台计算机要想运行 Java 程序, 必须首先安装 JRE。目前, Sun 公司为了推广 Java 技术, 根据应用领域的不同, 在 JRE 的基础上建立了三个面向不同应用的运行平台。

(1) J2SE: Java 2 标准版 (Standard Edition), 是 Java 系统的标准与核心平台, 主要面向个人用户。

(2) J2EE: Java 2 企业版 (Enterprise Edition), 是标准版的企业级扩展, 面向网络用户。

(3) J2ME: Java 2 微型版 (Micro Edition), 主要面向嵌入式系统应用。

1.4.2 Java 应用程序接口

Java 应用程序接口 (Java API) 是一组 Java 类与接口的总称, 用来提供 Java 语言程序设计的基本方法。这些类与接口涉及数学计算、输入输出操作、网络应用、用户界面与图形绘制等许多领域, 将为应用程序的开发提供极大的便利和更可靠的基础。

Java API 的结构呈树状, 如图 1-3 所示。

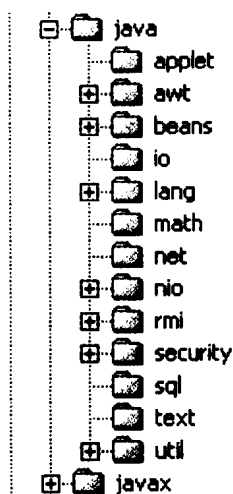


图 1-3 Java API 的树状结构图

Java API 的详细说明包含在 API 文档中, 该文档以 HTML 网页形式编写, 可以通过各种 Web 浏览器或支持网页浏览的软件查看。关于 Java API 文档的获取和安装, 请参考 1.4.3 节“Java 2 SDK 的安装”部分。图 1-4 是 Java API 文档的索引页 (index.html)。

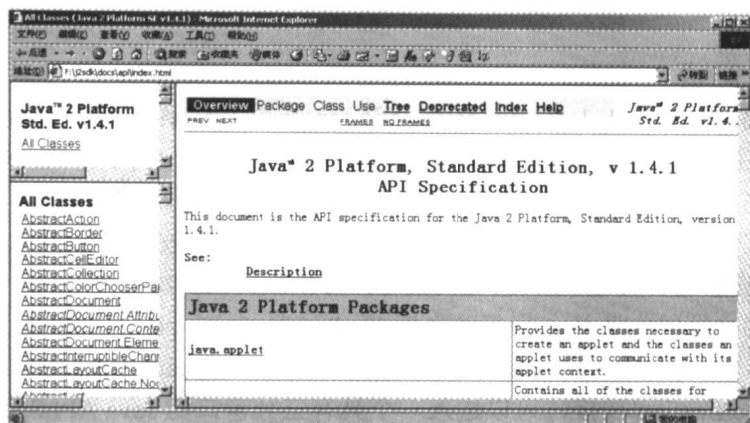


图 1-4 IE 浏览器中显示的 Java API 文档的索引

表 1-1 列出了 Java API 中的一些主要包的功能及其相关的类。

表 1-1 Java API 中的主要包功能及其相关类

包 名	功能及相关类
java.lang	包含一些形成语言核心的类，包括 Object, Math, String, System, Integer, Thread 等
java.lang.ref	提供对垃圾回收功能支持的类，包括 Reference, ReferenceQueue, WeakReference 等
java.lang.reflect	提供程序加载时的动态类检测功能
java.awt	用来构建和管理程序的图形用户界面
java.applet	提供执行 Applet 程序特殊行为的类
java.net	提供与网络应用相关的类和接口
java.io	提供处理输入输出文件的类
java.util	提供为应用任务设置的实用程序类和集合框架集，包括 Collection, Map, Set, List, Arrays 等

1.4.3 Java 2 SDK 的安装与组成

Java 2 软件开发工具包（Software Development Kit, SDK 或 JDK）是 Sun 系统公司提供的标准 Java 程序设计开发工具，它包括 Java 2 运行环境（J2RE）和一组开发工具。由于 Java 2 SDK 是由 Java 技术的创造者 Sun 系统公司建立的，因此成为 Java 语言程序设计和发展的参照标准，许多 Java 技术开发商都将 Java 2 SDK 作为其软件开发工具和技术的基础。Java 2 SDK 是一组基于命令行方式的 Java 语言程序设计工具。下面介绍 Java 2 SDK（J2SDK）的安装和主要工具。

1. J2SDK 的获得

一套完整的 J2SDK 开发包包括开发工具和 API 文档两部分。对于不同的操作系统，可下载的安装文件格式不尽相同，安装方法也存在差异。以 JDK 1.4.1 for MS Windows 为例，应包括如下两个文件。

- （1）j2sdk-1_4_1_02-windows-i586.rar ——JDK 开发包的微软 Windows 版本。
- （2）j2sdk-1_4_1-doc.zip ——相应的帮助文档。

这两个文件可以从 <http://java.sun.com/products/> 下载。下面以 JDK 1.4.1 开发包为工具介绍 Java 2 SDK 的安装。