



普通高等教育“十一五”规划教材
高等职业教育计算机技术系列教材



Java

程序设计教程

张小波 陈子聪 宋 晖 编著

冶金工业出版社

HIGHER TECHNICAL
AND
VOCATIONAL
EDUCATION

普通高等教育“十一五”规划教材
高等职业教育计算机技术系列教材

Java 程序设计教程

张小波 陈子聪 宋 晖 编著

北 京

冶金工业出版社

内 容 简 介

本书是根据普通高等教育“十一五”国家级规划教材的指导精神而编写的。

本书以通俗的语言,系统地介绍了 Java 语言基础和开发应用程序的设计思想以及编程方法,结合实例深入浅出地介绍了 Java 程序设计语言。内容主要包括:Java 语言概述、Java 程序设计基础、流程控制结构、面向对象程序设计、用户界面、图形、异常处理、输入输出、多线程与 Applet 基础、网络编程、数据库编程。

本书内容丰富,文字叙述简明易懂,注重实用性和可操作性,各章配有精心设计的习题。本书既可作为高职高专学校和大学计算机专业的教材,也可作为各类 Java 培训班及全国计算机等级考试用书。

图书在版编目(CIP)数据

Java 程序设计教程 / 张小波, 陈子聪, 宋晖编著.
北京: 冶金工业出版社, 2006.3
ISBN 7-5024-3945-5

I. J... II. ①张...②陈...③宋... III. JAVA 语言—程序设计—教材 IV. TP312

中国版本图书馆 CIP 数据核字(2006)第 017197 号

出版人 曹胜利(北京沙滩嵩祝院北巷 39 号, 邮编 100009)

责任编辑 戈兰

佛山市新粤中印刷有限公司印刷; 冶金工业出版社发行; 各地新华书店经销

2006 年 3 月第 1 版第 1 次印刷

787mm×1092mm 1/16; 17.5 印张; 403 千字; 272 页

25.00 元

冶金工业出版社发行部 电话:(010) 64044283 传真:(010) 64027893

冶金书店 地址: 北京东四西大街 46 号(100711) 电话:(010) 65289081

(本社图书如有印装质量问题, 本社发行部负责退换)

前 言

一、关于本书

本书是根据普通高等教育“十一五”国家级规划教材的指导精神而编写的。

目前,全国各地高职高专院校普遍扩招,高职高专学生人数迅速增长,这给他们的就业带来了巨大的压力。而当前高职高专学生的就业情况不容乐观,究其原因,所用教材与实际应用脱轨是一主要因素。针对现有教材质量较差、品种单一、版本陈旧、实用性和可操作性不强等原因,肩负着应用型人才培养的高职高专院校急需一系列符合当前教学改革需要的教材。

Java 语言是一种面向对象的网络编程语言,其强大的网络功能和其他语言难以比拟的,Java 语言特别适合开发 Internet 应用程序。Java 程序具有与硬件无关、与平台无关、可靠性高、安全稳定和支持多线程等特点,在当今的计算机和网络技术中占据重要的地位。

以网络为中心的计算是当今计算机发展的主流,网络程序设计已经成为应用程序设计的主体。Java 程序设计是网络应用的重要组成部分。此课程是网络技术应用与服务专业学生的必修课。此课程与程序设计基础、网络数据库及网页制作课程相互联系,相互补充。它们分别给读者提供基于 Web 的 C/S 程序设计方法和技能,面向对象程序设计方法,网络数据库的设计和管理及网页制作技巧的学习。使读者能够利用 Java 语言的程序设计将网络数据库、动态网页信息有机地结合起来,使读者的技能得到综合、系统的训练和培养,成为社会专用人才。

二、本书结构

全书的安排如下:

第 1 章:Java 语言概述。主要包括 Java 语言的发展历史和特点。

第 2 章:Java 程序设计基础。主要包括数据类型、常量和变量、运算符和表达式以及数组等。

第 3 章:流程控制结构。主要介绍 Java 语言的几种控制结构,包括顺序、选择和循环等。

第 4 章:面向对象程序设计。主要包括面向对象程序设计的基本概念、继承、多态、接口、包等。

第 5 章:用户界面。主要包括组件介绍、布局设计、菜单设计、对话框设计等。

第 6 章:图形。主要包括图形坐标、图形绘制、文本、属性设置等。

第 7 章:异常处理。主要包括基本概念介绍、异常的处理、创建自己的异常类等。

第 8 章:输入输出。主要包括输入输出的基础知识、字节流、字符流、文件流、对象流、文件的随机存取等。

第 9 章:多线程与 Applet 基础。主要包括多线程基础知识、创建线程、线程间通信、HTML 基础、Applet 类、Applet 的主要方法等。

第 10 章：网络编程。主要包括使用 URL 定位资源、Socket 通信、数据报通信等。

第 11 章：数据库编程。主要包括数据库程序设计基础及方法。

三、本书特点

本书系统、全面地研究和借鉴了国外相关教材先进的教学方法，结合国内院校教学实际和先进的教学成果，根据教育部“十一五”国家级规划教材应用型高职高专教育的指导思想编写，具有实用性和可操作性，与时俱进，与当前就业市场结合得更加紧密。

本书深入浅出地介绍了 Java 程序设计语言。语言通俗，系统性强，内容丰富、实用，文字叙述简明易懂，注重实用性和可操作性。使读者能轻松而快速地学好 Java 语言的语法知识及应用。为了巩固对每章的学习，每章最后都附有一定数量精心设计的练习题。

四、适用对象

本书由广东工业大学的张小波、陈子聪和宋晖编写。

本书在结构和内容的编排上注重深入浅出、循序渐进。可作为高职高专学校和大学计算机专业的教材，也可作为 Java 开发人员的读物或参考资料。

本书第 1、2、3、6、7、9 章由张小波编写，第 4、5、11 章由陈子聪编写，第 8、10 章由宋晖编写，全书由张小波统稿并校对。

由于编写时间仓促，编者水平有限，书中如有疏漏和不足之处，敬请各位读者批评指正，联系方法如下：

电子邮箱：service@cnbook.net

网址：www.cnbook.net

本书电子教案、源代码及习题参考答案可在该网站免费下载，此外，该网站还有一些其他相关书籍的介绍，可以方便读者选购参考。

编 者

2005 年 12 月

目 录

第 1 章 Java 语言概述	1	2.4.5 位运算	29
1.1 Java 语言简介	1	2.4.6 条件运算	32
1.1.1 Java 历史	1	2.4.7 运算优先级	33
1.1.2 前景与发展	2	2.5 数组	34
1.2 Java 语言特点	3	2.5.1 一维数组	35
1.3 Java 程序开发	4	2.5.2 多维数组	39
1.3.1 Java 程序运行环境	4	2.5.3 数组作为方法参数和返回值	43
1.3.2 JDK 开发工具的安装与使用	6	小结	46
1.3.3 Java 集成开发环境	11	综合练习二	47
1.3.4 Java 程序的开发过程		一、选择题	47
及其运行	13	二、填空题	47
小结	16	三、简答题	48
综合练习一	17	四、程序设计题	48
一、选择题	17	第 3 章 流程控制结构	49
二、填空题	17	3.1 流程控制结构概述	49
三、简答题	18	3.2 选择结构	50
四、程序设计题	18	3.2.1 if 语句形式 1	50
第 2 章 Java 程序设计基础	19	3.2.2 if 语句形式 2	51
2.1 符号集	19	3.2.3 switch 语句	54
2.1.1 标识符	19	3.3 循环结构	56
2.1.2 保留字	20	3.3.1 for 循环语句	57
2.2 数据类型	20	3.3.2 while 循环语句	58
2.2.1 整型数据	20	3.3.3 do-while 循环语句	59
2.2.2 浮点型数据	21	3.3.4 for 和 while 的转化	60
2.2.3 字符型数据	22	3.3.5 循环结构举例	60
2.2.4 布尔类型	22	3.4 跳转语句	63
2.3 常量与变量	23	3.4.1 break 语句	63
2.3.1 常量	23	3.4.2 continue 语句	64
2.3.2 变量	23	3.4.3 return 语句	64
2.4 运算符与表达式	24	小结	65
2.4.1 算术运算	24	综合练习三	66
2.4.2 关系运算	27	一、选择题	66
2.4.3 逻辑运算	28	二、填空题	66
2.4.4 赋值运算	28	三、简答题	66

四、程序设计题	67	JComponent 类	102
第 4 章 面向对象程序设计	68	5.2.2 标签	102
4.1 面向对象的基本概念	68	5.2.3 按钮	103
4.1.1 对象	68	5.2.4 复选框	104
4.1.2 类	69	5.2.5 单选按钮	105
4.1.3 接口	70	5.2.6 选择框	105
4.1.4 包	70	5.2.7 列表框	105
4.1.5 方法	71	5.2.8 文本字段	106
4.2 对象和类	73	5.2.9 文本区域	107
4.2.1 类的定义	74	5.2.10 滚动条	107
4.2.2 对象实例的创建	75	5.2.11 其他原子组件	108
4.2.3 封装、继承与多态	81	5.2.12 容器	108
4.3 接口	89	5.3 布局设计	115
4.3.1 接口的作用	89	5.3.1 顺序布局管理器	115
4.3.2 接口的声明	89	5.3.2 网格布局管理器	116
4.3.3 接口的实现	90	5.3.3 网格包布局管理器	117
4.3.4 接口的继承	92	5.3.4 边框布局管理器	120
4.4 包	92	5.3.5 卡片布局管理器	121
4.4.1 Java 程序的组织	92	5.3.6 盒式布局管理器	122
4.4.2 定义包	93	5.4 菜单与工具栏设计	124
4.4.3 引用包	93	5.4.1 菜单	124
4.4.4 包级访问控制	94	5.4.2 工具栏	130
小结	95	5.5 对话框设计	131
综合练习四	95	5.5.1 对话框分类	131
一、选择题	95	5.5.2 对话框的创建及使用	131
二、填空题	96	5.6 图形用户界面设计综合示例	135
三、简答题	96	小结	142
四、程序设计题	96	综合练习五	143
第 5 章 用户界面	97	一、选择题	143
5.1 图形用户界面设计概述	97	二、填空题	144
5.1.1 Java.awt 包和 javax.swing 包	97	三、简答题	144
5.1.2 组件、容器和布局	99	四、程序设计题	144
5.1.3 事件驱动编程方法	100	第 6 章 图形	145
5.1.4 编写 GUI 应用程序的 一般步骤	101	6.1 图形坐标	145
5.2 组件	102	6.2 图形绘制	145
5.2.1 Component 类、Container 类和		6.2.1 直线	146
		6.2.2 矩形	146
		6.2.3 椭圆	146

6.2.4 多边形	148	小结	182
6.3 文本	150	综合练习八	183
6.4 属性设置	152	一、选择题	183
6.4.1 字体	152	二、填空题	183
6.4.2 颜色	152	三、简答题	183
小结	154	四、程序设计题	183
综合练习六	154	第 9 章 多线程与 Applet 基础	185
一、选择题	154	9.1 线程	185
二、填空题	154	9.1.1 线程的概念	185
三、简答题	154	9.1.2 线程的控制	186
四、程序设计题	154	9.1.3 线程的创建	197
第 7 章 异常处理	155	9.1.4 线程间通信	203
7.1 异常基本概念介绍	155	9.2 HTML 基础	209
7.2 异常的抛出和捕获	156	9.3 Applet	211
7.3 创建自己的异常类	157	9.3.1 Applet 的 HTML 知识	211
小结	159	9.3.2 Applet 类	212
综合练习七	159	9.3.3 Applet 的主要方法	214
一、选择题	159	小结	215
二、填空题	159	综合练习九	215
三、简答题	159	一、选择题	215
四、程序设计题	160	二、填空题	216
第 8 章 输入输出	161	三、简答题	216
8.1 输入输出概述	161	四、程序设计题	216
8.2 各种输入输出流	162	第 10 章 网络编程	217
8.2.1 字节流	163	10.1 计算机网络基础	217
8.2.2 字符流	164	10.1.1 分层的概念	217
8.2.3 管道流	165	10.1.2 通信子网	218
8.2.4 SequenceInputStream	167	10.1.3 传输层	219
8.2.5 PushbackInputStream	168	10.1.4 应用层	219
8.2.6 文件流 FileStream	169	10.2 使用 URL 定位资源	220
8.2.7 数据输入输出流: DataInput Stream&DataOutputStream	176	10.2.1 URL 的概念	220
8.2.8 对象流: ObjectInput Stream&ObjectOutputStream	178	10.2.2 URL 类的构造函数与方法	221
8.2.9 其他输入输出类	180	10.2.3 类 URLConnection	223
8.3 文件随机存取: RandomAccessFile 类	181	10.3 Socket 通信	227
		10.3.1 Socket 类的建立	228
		10.3.2 Socket 类的使用	229
		10.3.3 范例: 简易 Web 服务器	232

10.4 数据报通信.....	234	11.1 数据库基本理论.....	253
10.4.1 DatagramPacket 类和 DatagramSocket 类	235	11.1.1 数据库发展历程.....	253
10.4.2 数据报的发送与接收实例	236	11.1.2 关系数据库管理系统	253
10.4.3 范例：一个简单的聊天程序	237	11.2 SQL 和 JDBC 简介	254
10.5 协议处理机.....	240	11.2.1 结构化查询语言（SQL）	254
10.5.1 WWW 协议.....	241	11.2.2 JDBC	257
10.5.2 FTP 协议.....	244	11.3 数据库程序设计.....	263
10.6 用 Java 实现加密	245	11.3.1 数据库应用程序的多层结构	263
小结	250	11.3.2 数据库应用程序的设计	264
综合练习十	251	11.3.3 数据库编程举例.....	265
一、选择题	251	小结	270
二、填空题	251	综合练习十一	270
三、简答题	251	一、选择题	270
四、程序设计题	251	二、填空题	271
第 11 章 数据库编程	253	三、简答题	271
		四、程序设计题	271

第 1 章 Java 语言概述

Java 语言已成为当前系统开发的主流语言。由于它易掌握、跨平台以及功能强大，所以从一开始就为人们所喜爱。本章主要介绍 Java 语言的发展历史和开发环境。

1.1 Java 语言简介

Java 语言是 Sun Microsystems 公司（简称 Sun 公司）推出的新一代面向对象程序设计语言，它一经推出就紧贴网络化发展的大潮，为软件产业带来了划时代的革命。随着时间的推移，Java 的应用领域已突破了最初在网络上的应用，小如嵌入式手提设备，大到火星探测器、大型分布式计算环境、电子商务等各方面。Java 语言已经造就了一个庞大的计算机产业，推动着现代软件产业的发展。每年都有众多的软件开发人员投入到 Java 开发行列中来，学习 Java 语言已成为大多数开发人员进军软件产业的敲门砖。学习一门语言，可以从了解它的历史开始，从中管窥该语言的某些本质特性。

1.1.1 Java 历史

Java 语言的历史可以追溯到 1991 年。当时美国 Sun 公司的一群工程师在 Patrick Naughton 和 James Gosling 的领导下，开始了一个代号为“Green”的项目。这个项目的初衷是为消费类电子产品，例如电视机机顶盒，设计一种通用的开发语言。由于针对的目标设备在内存和处理器方面的能力都十分有限，因此客观上要求这种语言必须非常小巧和紧凑；另外，由于不同的厂家生产的产品在体系结构方面千差万别，这就要求这种语言必须不依赖于某一平台或者架构，而必须平台中立。“Green”项目组原先打算采用当时非常流行的 C++ 语言作为开发语言，但是在仔细考量了 C++ 语言以及其他各种语言的优缺点之后，他们发现竟然没有一种语言能够胜任这项工作。为了让项目顺利开展，他们最后决定开发一门新语言来满足开发要求，Java 语言就这样诞生了。

Java 语言最初的名字不是“Java”，而是“Oka”，据说是 Java 之父 James Gosling 根据他在 Sun 公司的办公室窗外那棵 Oka 树命名的。在历尽开发的艰辛之后，“Green”项目组终于在 1992 年提交了他们的第一个产品，然而不幸的是 Sun 公司上下竟然没有几个人对它有兴趣，项目面临着被搁置的危险。无奈之下，Green 小组不得不自己为该产品开拓市场渠道，可是噩耗依然，根本没有一家消费类电子产品公司对该产品有兴趣。从 1993 年到 1994 上半年，Green 小组一直在寻找市场渠道中苦苦挣扎。

正当 Green 小组苦于找不到出路的时候，互联网正在悄然兴起并一步一步发展壮大。与今天风靡世界的动态网页技术不同，当时网络上的网页都是静态的页面，而且各个网络节点上的机器及操作系统差别很大，网络传输能力十分有限。Green 小组敏锐地感觉到 Java 大展身手的机会来了，因为当初设计 Java 语言的初衷之一就是让同一个用 Java 编写的程序不用（或很少）修改就能轻松地在各种软硬件环境下运行，而且 Java 小巧紧凑的程序也很适应在当时的网络状况下传输和部署。为了尽展 Java 的魅力，Green 小组决定采用 Java 编写了一个网页浏览器。该浏览器不仅能够运行在不同的平台上，并且还支持一种叫 Applet

的技术,它使得浏览器能够执行嵌入在网页中的 Java 小程序,从而能够展现出动态的网页,极大地提高了网页和用户之间的交互性。该浏览器一经发布,就吸引了很多人的目光,人们终于看到了 Java 在异构网络环境下实现跨平台互操作的强大威力。

1996 初, Sun 发布了 Java 的第一个版本 Java 1.0,不久人们发现了该版本中存在很多不尽人意的地方。于是 Sun 又接着发布了 Java 1.1,该版本修正了 1.0 版本中的很多缺陷,还引入了一些新的特性,如新的事件处理模型,但是该版本依然还不够完善。

1998 年, Sun 发布了 Java 1.2,该版本替换了早期版本中的 GUI(Graphics User Interface) 和 graphics 工具包,采用了更加复杂和更具伸缩性的处理模型,从而更好地体现了 Java 的信条——“Write Once, Run Anywhere”。不久, Sun 出于商业考虑,采用了“Java 2 Standard Edition Software Development Kit Version 1.2”来命名 Java。之后, Sun 又陆续推出 Java 的升级版本。目前, Java 已经发展到了 1.4 版本,其支持泛型编程的 1.5 版本也已经有了测试版,预计不久就会发布正式版。现在普遍称 1.2 版本及之后版本的 Java 为 Java 2。

1.1.2 前景与发展

随着 Java 技术的不断发展,它根据市场需要又进一步细分为:针对企业级应用的 J2EE (Java 2 Enterprise Edition)、针对普通 PC 应用的 J2SE (Java 2 Standard Edition) 和针对嵌入式设备及消费类电器的 J2ME(Java 2 Micro Edition)三个版本,相应的三部分分别为 J2EE (Java 2 企业版)、J2SE (Java 2 标准版) 和 J2ME (Java 2 移动版)。这三个部分的关系如图 1-1 所示。

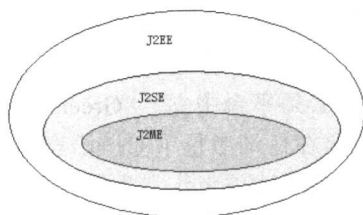


图 1-1 J2EE、J2SE 和 J2ME 的关系

Java 所有平台的划分均以 J2SE 为基准点, J2SE 支持所有的核心类库以及八种基本数据类型。J2SE 外面的 Java 平台 (J2EE), 也支持完整的核心类库及其八种基本数据类型。反之, 位于同心圆内部 Java 平台 (J2ME) 所支持的就只是核心类库的子集合。J2SE 定位在客户端 (Client-side) 程序的应用上; 相比于 J2SE, J2EE 增加了许多支持企业内部计算的扩充类库, 其定位是服务器端 (Server-side) 程序的应用; 而 J2ME 所支持的核心类库比 J2SE 少, 但增加了一些支持嵌入式系统的扩充类库, 其定位是嵌入式系统的应用。

Java 作为一种面向对象的程序设计语言, 特别适合于 Internet 应用程序的开发, 它的平台无关性直接打破了 Wintel 的垄断地位。Java 作为软件开发的一种革命性技术, 其地位已经确立。主要表现在:

(1) 计算机产业的很多厂商, 包括 IBM、Oracle、Borland 甚至是 Microsoft 都购买了 Java 的许可证。

(2) 众多支持 Java 的软件已经推出, 大大促进了 Java 的应用普及。

(3) 目前, 企业信息系统已经出现了众多成熟的解决方案, 而 Java 在其中发挥了不

可替代的作用。

1.2 Java 语言特点

按照 Java 设计者的说法, 可以用一句话来概括 Java 语言的特点: “Java 是一种简单的 (Simple)、面向对象的 (Object Oriented, OO)、分布式的 (Distributed)、解释的 (Interpreted)、健壮的 (Stronger, Robust)、安全的 (Secure)、结构中立的 (Architecture Neutral)、可移植的 (Portable)、性能优异的 (High Performance)、多线程的 (Multithread)、动态的 (Dynamic) 语言”。下面分别解释这些特点:

1. 简单

首先, Java 语言的语法和 C++ 非常接近, 一定程度上可以说它就是 C++ 的一个变体, 熟悉 C++, 甚至是 C 语言的程序员都可以很快掌握 Java 的基本语法。当然, Java 毕竟不是 C++, 它抛弃了 C++ 一些容易出错和晦涩难懂的地方, 使得程序员更容易写出安全高效的代码。其次, Java 把编程时经常用到的一些功能封装成类库提供给程序员使用。程序员可以在现有类库的基础上搭建起应用程序, 避免了重复编写相同代码的麻烦, 从而使程序员能更好地把精力集中在如何实现业务逻辑上。再次, Java 语言编译出来的程序非常小巧。由于 Java 最初的目标之一是作为嵌入式设备上软件开发的语言, 这决定了 Java 程序必须是非常精简小巧, 以便应对嵌入式设备上有限的资源。

2. 面向对象

简单地说, 面向对象设计就是这样的一种技术, 它关注的是数据 (即对象的状态) 和在数据上的操作 (即对象的动作), 程序围绕被操作的数据来设计, 而不是围绕操作本身。

3. 分布式

Java 拥有一个包含 HTTP、FTP 协议的子库, Java 应用程序能够凭借 URL, 通过网络访问其他对象, 其在语言层面的访问方式和访问本地文件系统几乎相同。采用 Java 编写网络应用程序既方便高效又健壮。

4. 健壮的

Java 能编写出可靠的程序以应对各种各样的环境。Java 致力于检查程序在编译和运行时的错误。此外, Java 还抛弃了直接内存操作, 从而避免了程序运行时覆盖或修改某些重要数据。

5. 安全

Java 为了应对网络环境下的安全问题, 建立了严格的安全检查机制, 有效地避免了病毒、非法篡改、窃取等危害系统安全情况的发生。例如: Java 可以防止应用程序运行时造成的内存或堆栈溢出的问题, 而溢出是病毒攻击系统的一个主要手段; Java 中的沙箱操作可以禁止 Applet (嵌入在网页中的 Java 小应用程序) 访问本地文件系统, 从而防止信息的泄密和被篡改。此外, Java 还是类型安全的, 编译器将检查程序中所有的类型转换, 避免不合法的类转换。

6. 解释的

采用 Java 语言编写的源程序经过 Java 编译器编译后并不是直接产生目标代码, 而是转化成一种平台无关的二进制字节码 (Byte code) 并存放于相应的 .class 文件中。运行 Java 应用程序时, 由解释器根据字节码一边解释一边执行, 并根据需要载入相应的类库。

7. 平台中立

Java 将它的程序编译成一种平台中立的中间代码文件格式,只要安装有 Java 运行环境的机器都能够运行这种中间代码。现在,Java 运行环境已经有了 Solaris、MacOS、Linux 和 Windows 等版本。

8. 可移植

平台无关性使任何配备了 Java 运行环境的计算机系统都可以运行同一 Java 应用程序,实现了 Java 应用软件的可移植性。为了保证可移植性,Java 标准规定了独立于具体平台的基本数据类型及其上的运算,使得不同平台上的数据保持一致。在任何平台的任何 Java 解释器上,数据类型都是依据上述规定表示的。采用 Java 编写的应用程序,一般都有“一次编写,到处运行”的效果。

9. 高性能

如果解释器的速度不够快的话,可以把 Java 代码直接转换成相应 CPU 的机器代码再执行。为了提高解释执行的速度,当前的 Java 解释器普遍采用了即时编译技术(Just In Time Compiler, JIT),该技术把字节码转换成机器码并缓存,当再次运行该段字节码时,直接采用已经缓存的机器码执行,避免了每次执行时都要解释执行,从而大大提高了程序的执行效率。

10. 多线程

多线程使得一个程序里面可同时执行多个任务,更重要的是多线程提高了程序的交互性能和实时响应性能,此外还可以充分利用多处理器系统带来的优异特性。Java 语言进行线程编程十分方便,它内置了对多线程的支持,很容易就能实现线程的有关操作。

11. 动态

Java 语言允许在运行时动态地装入所需要的类,这些类既可以位于本地,也可以是从 Internet 上下载的,这一特性是 C++等其他编译型语言所无法实现的。在 Java 中对象的类型识别延迟到运行时才进行,因而可以在运行时轻松地加入新的方法和实例变量或者更改现有方法的实现,而不影响现有系统的运行。

12. 垃圾回收

Java 中所有的内存分配和回收都交由 Java 运行环境负责,Java 应用程序根本不用手工分配和管理内存。一直以来,由程序自己管理内存是比较麻烦且容易出错的工作。Java 语言提供了自动的垃圾回收机制(Garbage Collection),程序员不再需要处理内存管理的问题,Java 系统会自动回收不再使用的内存。

1.3 Java 程序开发

要开发 Java 应用程序,必须先建立开发运行环境。最简单的开发环境就是用于编译和运行程序的 Java 2 平台标准版(The Java™ 2 Platform, Standard Edition),以及一个用于编辑源程序的文本编辑器,此外还有很多集成化开发环境(Integrated Development Environment, IDE)可以使用。下面将分别加以介绍。

1.3.1 Java 程序运行环境

了解 Java 程序的运行环境必须先弄清楚一个概念——Java 平台(The Java Platform)。

所谓平台就是指应用程序运行时所处的软件或硬件环境,大多数平台都是特定软件和硬件的结合体。一些比较流行的平台有:Windows XP、Linux、Solaris 和 MacOS 等等。与上面提到的平台不同,Java 平台是一个运行于其他硬件平台上面的纯软件平台。Java 平台实际上是由两部分组成的:Java 虚拟机 (Java Virtual Machine, JVM) 和 Java 应用程序编程接口 (Java Application Programming Interface, Java API)。

JVM 也就是 Java 解释器,它负责在不同的硬件平台上解释执行 Java 应用程序。Java API 则是一组庞大的、功能强大的软件组件集合,这些组件提供各种各样的功能供应用程序调用。这些 API 分布于很多不同的类 (Class) 和接口 (Interface) 中,由这些类和接口再组织成很多不同的包 (Package)。图 1-2 表示了应用程序、Java Platform 和硬件的关系。



图 1-2 Java 运行平台结构

由此可见,Java Platform 屏蔽了硬件平台的差别,为应用程序提供了统一的软硬件界面。

不同于解释型语言和编译型语言,Java 是一种混合型语言。Java 程序的工作机制可以简单表述如下:

首先,Java 源程序经过 Java 编译器,被转换成一种平台无关的中间代码形式,称为 Java 字节码 (Java Byte code),并保存在相应的类文件中。

接着,当要运行 Java 应用程序时,该应用程序对应的字节码被提交给 JVM,由 JVM 解释器转化成本地机器的可执行代码并执行。在 JVM 解释执行这些字节码时,它会动态地追踪应用程序中类的使用情况,适时地载入所需的类。图 1-3 展示了上述过程。

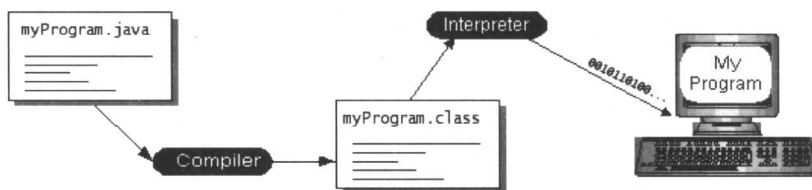


图 1-3 Java 程序解释执行过程

最常见的 Java 程序主要是 Applet 和 Application。

Applet 是一种嵌入在 HTML 页面中的 Java 应用程序片,它必须运行在支持 Java 技术的浏览器中。当支持 Java 技术的浏览器遇到包含有 Applet 标记的 HTML 页面时,会自动下载网页中指定的 Java 字节码文件,并提交给浏览器中的 JVM 解释器解释执行,并将执行结果展现在该网页中。

Java 真正强大的功能还在于它所编写出来的面向各种应用的应用程序 (Application)。Java Application 是指那些可以独立地在 JVM 上直接运行的 Java 程序。这些应用程序种类繁多,既有桌面应用,也有服务器端的应用 (如 Servlet); 既包括网络应用 (如 Applet, NetWorking), 也有数据库方面的应用 (如 JDBC), 还包括了图形图像方面的应用。时至

今日, Java 已经发展为一个庞大的体系结构, 但 Java 的发展脚步仍然没有停止。

1.3.2 JDK 开发工具的安装与使用

1. 安装 JDK

JDK (或 JSDK) 即 Java Software Development Kits, 是 Java 的开发工具, 包括了一系列开发调试工具。最新的 JDK 版本是 1.4.2 (实际上目前还有 1.5 版的测试版本), 可以到如下网址免费下载: <http://java.sun.com/j2se/1.4.2/download.html>。

选择针对相应平台的 JDK 并下载后 (此处以 Windows 平台为例), 双击安装文件, 进入如图 1-4 所示的安装界面。

选择接受上面的条款, 单击 **Next >** 按钮进入如图 1-5 所示的画面。

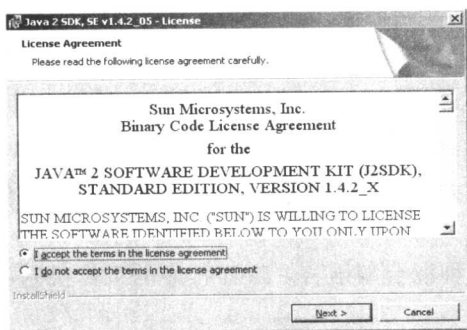


图 1-4 JDK 安装过程

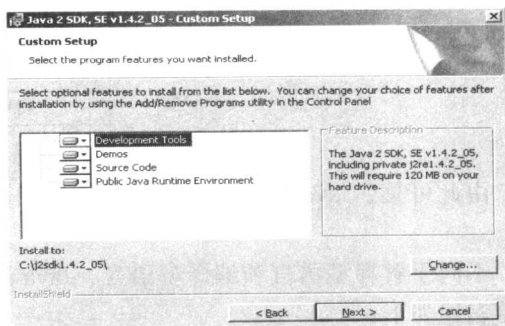


图 1-5 JDK 安装过程

选择要安装的内容和安装的路径, 这里采用默认的方式, 单击 **Next >** 按钮进入如图 1-6 所示的画面。

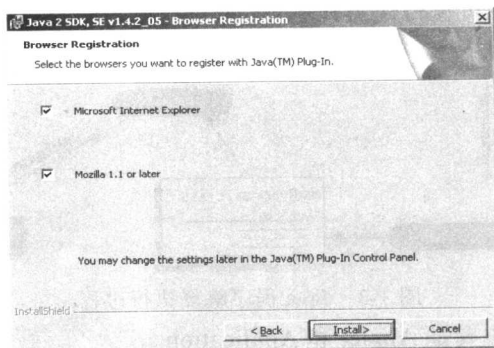


图 1-6 JDK 安装过程

选取将要安装 JVM 插件的浏览器 (以便该浏览器能够解释含有 Applet 的网页), 单击 **Install >** 按钮开始 JDK 的安装。默认安装完后会在 “C:\Program Files\Java\” 目录下安装一套 JRE (供浏览器使用), 在 “C:\j2sdk1.4.2_05\” 下安装一套 JDK (也包括一套 JRE)。

为了在命令行方式使用 JDK 工具时, 系统能够自动找到 JDK 工具的位置, 还需要修改系统的环境变量 PATH 和设置 CLASSPATH, 在 PATH 的最前面增加 Java 的路径。这样 JDK 便安装好了。

在 Windows 环境下设置 PATH 的方法为：

对于 Windows95/98，打开“开始”菜单，选择“运行”，输入命令“sysedit”（调用系统配置编辑程序），在 C:\AUTOEXEC.BAT 窗口中找到 PATH 设置并追加一项如下内容“c:\jdk1.4.2_05\bin”。

对于 Windows NT、Windows 2000 和 Windows XP 等系统，进入“开始”菜单，选择“设置/控制面板/系统”。对于 Windows NT，选择“环境”页，对于 Windows 2000/XP，选择“高级”页，在环境变量设置中的用户变量栏中增加（如果原来没有 PATH）或修改 PATH 环境变量，使其包含“c:\jdk1.4.2_05\bin”路径，下面以 Windows 2000 下的设置作为示例：

首先进入“开始”菜单，选择“控制面板”，再选择“系统”，出现如图 1-7 所示的窗口。

单击“环境变量(E)”按钮，弹出如图 1-8 所示的窗口。

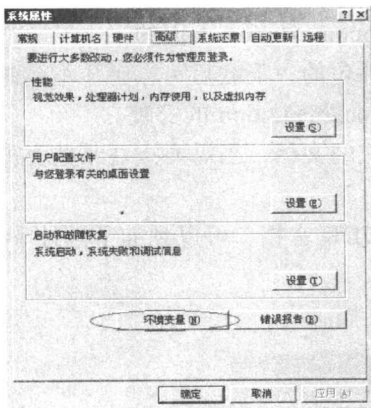


图 1-7 JDK 环境变量配置过程

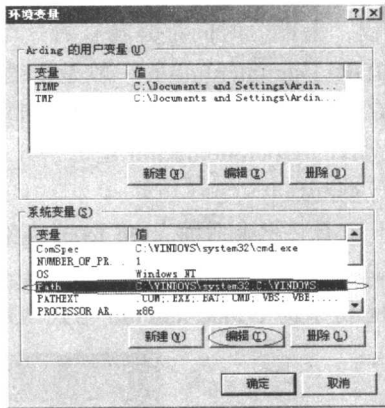


图 1-8 JDK 环境变量配置过程

在“系统变量”中选中“Path”，单击“编辑”按钮，弹出如图 1-9 所示的窗口。

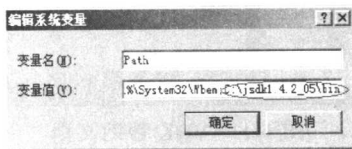


图 1-9 JDK 环境变量配置过程

在“变量值”中添加“C:\jdk1.4.2_05\bin”（具体添加内容根据你的 JDK 安装路径而不同，例如当把 JDK 安装在“D:\JavaSoft\jdk1.4.2_05”时，则添加应为“D:\JavaSoft\jdk1.4.2_05\bin”），注意添加的内容与前面已有的内容之间要用“;”分隔开来（如图 1-9 中 C:\前面有一个“;”）。

设置完 Path 之后接着还要更新（或添加/新建）CLASSPATH 环境变量，这是为了使系统能找到用户定义的类，需要将用户类所在的目录（通常是当前目录，加入到 CLASSPATH 中）放入 CLASSPATH 变量中。具体更改方法同 PATH 修改过程，只是需找到 CLASSPATH 环境变量进行修改。注意设置 CLASSPATH 是可选的，如果没有设置，则在运行 Java 程序时必须显式指明 CLASSPATH。

设置完上述环境变量之后, 可以用 DOS 命令“set”查看设置后的值, 再输入 `java -version`, 会出现如图 1-10 所示的窗口。这表明上面的 PATH 已经被正确设置。

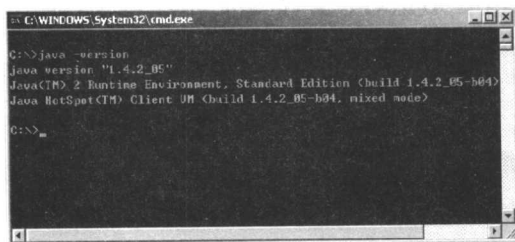


图 1-10 查看环境的设置结果

2. 安装 JDK 文档

Documentation 在 JDK 的下载页面也有下载链接, 建议同时下载 Documentation。Documentation 是最重要的编程手册, 涵盖了整个 Java 所有方面内容的描述。可以这样说, 学习 Java 编程, 大部分时间都是花在看这个 Documentation 上面的。

下载的 JDK Documentation 一般是压缩文件包的, 下载后可以解压缩到某些方便查看目录中 (如 `C:\jdk1.4.2_05\docs`), 完成 JDK Documentation 的安装。

安装完成后的 Documentation 是 HTML 格式的文档, 当需要查看某些类的说明时, 可以打开它进行查阅。

当前网上也有一些制作成 chm 格式的免费 JDK 文档, 可以到如下网址下载:

<http://www.confluent.fr/javadoc/index.html>

如图 1-11 所示为下载后的 chm 格式 JDK 文档。

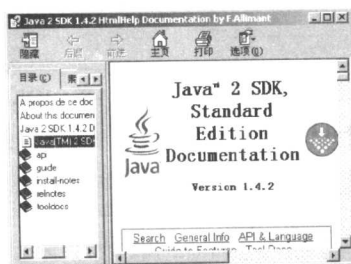


图 1-11 JDK 帮助文档

3. 安装后的 JDK 目录结构

安装完成后, 在 JDK 的根目录 (如 `c:\jdk1.4.2_05`) 下有如表 1-1 所示的文件 (夹):

表 1-1 JDK 目录说明

目录	说明
docs	HTML 格式的 JDK 文档
bin	Java 编译器、解释器和其他工具所在目录
demo	Java 平台开发的示例, 包括源代码
include	存放用于本地方法调用所需的文件
lib	存放必要的库文件
src.zip	存放 Java 核心类库的源代码
jre	存放 Java 运行环境的文件