

Java

编程技术 全接触

孙一林 彭波 主编



清华大学出版社

TP312/2909

2008

Java 编程技术全接触

孙一林 彭 波 主 编



清华大学出版社

北 京

内 容 简 介

本书是一本适合大专院校学生学习 Java 语言以及编写 Java 应用程序的实用型教科书或教学参考书,主要阐述了 Java 语言的关键字、语句、语法规则、类库等的使用方法,其内容主要包括 Java 语言开发工具的介绍、面向对象编程的概念、Java 基础语句、Java 类、类的继承与多态等特性、接口、包、异常、Java 基础类库的应用,以及使用 Java 语言实现创建图形界面、输入输出操作、多线程控制、网络数据传输、多媒体数据处理、操作数据库等功能的编程方法。

本书可作为大专院校计算机相关专业的本科或专科生使用的教材,也可作为学习 Java 语言的初、中级读者使用的参考书。

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。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

Java 编程技术全接触/孙一林,彭波主编. —北京:清华大学出版社,2008.6
ISBN 978-7-302-17585-8

I.J… II.①孙…②彭… 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(2008)第 066075 号

责任编辑:彭 欣

封面设计:山鹰工作室

版式设计:杨玉兰

责任校对:周剑云

责任印制:何 芊

出版发行:清华大学出版社

地 址:北京清华大学学研大厦 A 座

<http://www.tup.com.cn>

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

印 装 者:北京宏伟双华印刷有限公司

经 销:全国新华书店

开 本:185×260 印 张:29.75 字 数:717 千字

版 次:2008 年 6 月第 1 版 印 次:2008 年 6 月第 1 次印刷

印 数:1~4000

定 价:46.00 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:(010)62770177 转 3103 产品编号:026402-01

前言

本书作者都是多年从事 Java 语言教学和使用 Java 开发项目的教育工作者和软件工程师,对 Java 语言有着深入的理解,在从事 Java 语言教学的过程中,他们都或多或少地了解了学生在学习 Java 语言时遇到的难点以及如何使学生能够更快、更准确地掌握和使用 Java 语言,在此基础上,作者在编写本书时尽量做到通俗、易懂,使其符合循序渐进、由浅入深的学习规则,通过学习本书,读者可以规范、系统地学习 Java 语言,为后续深入学习 Java 语言奠定扎实的基础。

计算机语言与人类交流使用的自然语言功能是一样的,只是计算机语言是人与计算机交流时使用的,更确切地说是人命令计算机做某些事情的语言。计算机语言也有单词(被称为关键字)、词组(被称为类库或函数库)、语法规则等,但是计算机语言的单词是有限的,因为计算机 CPU 中的指令译码器只能理解很少的单词,而语法规则也是规定好的,只可以按照语法规则来编写一些语句,否则计算机不会认识,因此,只要我们掌握计算机语言的关键字和语法规则就可以编写计算机认识的可执行语句,而计算机程序就是一些语句遵循语法规则的逻辑组合,它可以让计算机完成特定的功能。计算机顾名思义只能实现计算功能,为什么它能用在各行各业?因为几乎所有的事物以及事物的改变都可以用“数据”来描述,例如,最熟悉的阿拉伯数字,ASCII 码(描述文字)、PCM 码(描述视频、音频信息)、表格、图等“广义数据”,处理这些“数据”正是计算机的专长,因此,编写计算机程序首先需要通过“数据”和对数据的操作来建立一些模型(称为数学模型),然后计算机执行程序则是针对数据的操作来实现要达到的目的。目前的面向对象编程的理念就是帮助我们实现或建立这样的模型,它可以使编写计算机程序变得更加简单,掌握面向对象编程思想是编写应用程序的基础,而 Java 语言就是适用于面向对象编程的语言,另外,一种计算机语言除了关键字和规定的语法规则外,它还提供了大量的类库或函数库,这些类库或函数库则是针对各种类型的“数据”实施处理的,在编写应用程序时,几乎所有需要处理的数据类型都可以在类库或函数库中找到处理方法,类库或函数库可以说是包罗万象。总之学习计算机语言在理解了其使用的编程思想后,剩下的就是学习关键字、语法规则和类库或函数库的使用了。

本书前 6 章主要是讲解 Java 语言的关键字、由关键字组成的表达式、程序流控制、类定义、接口定义、对象的使用、异常的处理等语句以及 Java 语言的语法规则。第 7~12 章则是讲解 Java 基础类库 JFC 的应用,每一个类库都是针对一类数据进行的操作,类库提供了许多数据操作方法,每一类操作都制定了操作机制,了解操作机制是更好地使用类库的前提,Sun 公司提供的 J2SDK 中的类库使用说明文档是最直接的使用类库的学习资料,本书则是从类库的操作机制和使用方法讲解其类库的应用。第 13 章多媒体编程和第 14 章数据库编程讲解的是具有一定难度的 Java API 的使用,多媒体编程讲解的内容是局限在使用 JFC 处理多媒体数据,而数据库编程讲解的内容则是关于数据库操作的 JDBC API 应用程

序接口规范的使用,拓展了 Java API 的学习范围,为深入学习各个应用领域中 Sun 公司提供的 Java API 奠定一些基础。

另外,书中的所有程序示例都是通过编译可运行的 Java 应用程序,并且实现了程序具有的主要功能,一些程序示例稍加修改(添加一些辅助功能)就可以使用到实际应用项目中。为了读者的学习和使用方便,本书还提供了这些 Java 应用程序的源代码和编译后可执行类代码的资料,读者可以到 <http://www.wenyuan.com.cn> 自行下载。

本书由北京师范大学信息科学与技术学院孙一林副研究员主编,参加编著的有中国农业大学信息与电气工程学院彭波教授、北京师范大学物理系黄文宏高级工程师、北京师范大学数学科学学院曾跃进老师、北京联合大学师范学院电子信息系曾文琪老师、江苏警官学院刘群老师、胡治国、崔永普等研究生。

由于作者水平有限,书中难免有错误,敬请读者谅解。如果读者有问题需要与作者讨论时,请发送电子邮件到: sunyilin@elec.bnu.edu.cn。

编 者

目 录

第 1 章 Java 语言综述	1	2.2.4 常量	33
1.1 Java 语言简介	1	2.3 Java 运算符和表达式	33
1.1.1 Java 语言的历史与现状	1	2.3.1 Java 运算符	33
1.1.2 Java 语言的特点	2	2.3.2 Java 表达式	35
1.1.3 Java 程序的运行机制	6	2.3.3 表达式中运算符的使用规则	36
1.2 Java 程序开发环境	7	2.4 Java 程序流控制语句	39
1.2.1 Java 语言程序开发平台	7	2.4.1 分支结构语句	39
1.2.2 Java 程序基础开发运行 平台——J2SDK	8	2.4.2 循环结构语句	42
1.2.3 在 Windows 操作系统中搭建 Java 程序的开发和运行环境	9	2.4.3 辅助控制语句	45
1.2.4 Java 程序的编译和运行	14	2.4.4 流控制语句应用示例	48
1.3 HTML 标记语言简介	15	第 3 章 面向对象编程	50
1.3.1 HTML 语句的特点	16	3.1 面向对象程序设计	50
1.3.2 HTML 文档的语法格式	16	3.1.1 面向对象的程序设计方法	50
1.3.3 启动 Java Applet 小程序的 HTML 文档	18	3.1.2 Java 面向对象程序设计	51
1.3.4 Java Applet 小程序的编译 和运行	19	3.2 Java 类	52
1.3.5 Java 标准的 HTML 帮助 文档	21	3.2.1 Java 类的定义	52
第 2 章 Java 程序设计基础	22	3.2.2 方法的定义	53
2.1 Java 程序结构	22	3.2.3 Java 修饰符与其权限	56
2.1.1 Java 编译单元	22	3.3 Java 对象	57
2.1.2 Java 注释语句	23	3.3.1 对象的创建	57
2.1.3 Java 标识符	25	3.3.2 构造方法的使用	59
2.1.4 Java 关键字	25	3.3.3 对象的使用	60
2.2 Java 基本数据类型	26	3.3.4 Java 对象的清除	62
2.2.1 数据值的表示法	27	3.3.5 Java 类和对象的关系	63
2.2.2 变量	28	3.4 数组对象	64
2.2.3 变量的数据类型转换	31	3.4.1 一维数组的声明和创建	64
		3.4.2 多维数组的声明和创建	66
		3.4.3 数组的应用	68
		第 4 章 Java 类的继承与多态	72
		4.1 Java 类的继承	72
		4.1.1 概念和语法	72

4.1.2	Java 类继承关系的测试	74	6.3	自定义异常类	128
4.1.3	隐藏、覆盖和重载	75	第 7 章	Java 基础类的应用	133
4.1.4	构造方法的重载	78	7.1	java.lang 包	133
4.2	abstract 和 final 修饰符	80	7.2	Object 类	135
4.2.1	abstract 修饰符	80	7.3	基本数据类型类	136
4.2.2	final 修饰符	82	7.3.1	整型类	137
4.3	this 和 super 变量	83	7.3.2	浮点类	139
4.3.1	this 变量	83	7.3.3	其他常用类	141
4.3.2	super 变量	85	7.4	字符串 String 类	143
4.4	Java 的多态性	87	7.4.1	String 类	143
4.4.1	多态的概念	87	7.4.2	创建 String 对象并对其 进行操作	146
4.4.2	多态的应用	90	7.4.3	StringBuffer 类	147
4.4.3	构造方法与多态	92	7.4.4	创建 StringBuffer 对象 并对其进行操作	149
第 5 章	Java 接口和 Java 包	93	7.5	Math 类	150
5.1	Java 接口	93	7.6	Runtime 类	153
5.1.1	接口的定义	93	7.7	System 类	155
5.1.2	接口的实现	94	第 8 章	Java 输入输出操作	158
5.1.3	接口的继承	96	8.1	Java 的输入、输出机制	158
5.1.4	Java 类同时继承父类 并实现接口	97	8.2	java.io 包	159
5.1.5	接口与 Java 抽象类	98	8.3	java.io 包中的数据流操作根类	162
5.1.6	接口的应用	100	8.3.1	字节输入流 InputStream 类	162
5.2	Java 包	103	8.3.2	字节输出流 OutputStream 类	164
5.2.1	package 语句	104	8.3.3	字符输入流 Reader 类	165
5.2.2	Java 包与路径	105	8.3.4	字符输出流 Writer 类	166
5.2.3	import 语句	106	8.4	静态输入、输出操作	168
5.2.4	直接引用 Java 包中的类 和接口	107	8.4.1	静态输出操作	169
5.2.5	Java 包的应用	108	8.4.2	静态输入操作	170
第 6 章	Java 异常处理	112	8.5	文件的输入、输出操作	172
6.1	Java 异常处理	112	8.5.1	创建文件对象	173
6.1.1	异常处理机制	112	8.5.2	读文件操作	175
6.1.2	异常的捕获与处理	114	8.5.3	写文件操作	179
6.1.3	finally 语句	117	8.5.4	文件的随机读写操作	183
6.1.4	异常对象的抛出	119	8.5.5	对象的序列化	187
6.2	Java 基础包中定义的常用 异常类	122			

第 9 章 图形用户界面.....	191
9.1 AWT 包.....	191
9.2 java.awt 包中的组件根类.....	192
9.2.1 Component 组件类.....	193
9.2.2 MenuComponent 菜单 组件类.....	198
9.2.3 Container 容器类.....	198
9.3 图形用户界面.....	200
9.3.1 AWT 中的窗体类.....	201
9.3.2 AWT 中的组件类.....	205
9.3.3 AWT 中的菜单类.....	211
9.4 Java 布局管理器.....	214
9.5 Java 组件的事件处理.....	217
9.5.1 AWTEvent 事件类.....	218
9.5.2 java.awt.event 包中的接口 和事件类.....	219
9.5.3 组件事件处理机制.....	227
9.5.4 组件事件处理示例.....	231
9.6 javax.Swing 包中常用组件.....	237
第 10 章 Java Applet 小程序.....	244
10.1 Applet 小程序类.....	244
10.2 Applet 程序的工作原理.....	246
10.2.1 Applet 小程序编程框架.....	246
10.2.2 Applet 小程序的 运行过程.....	249
10.2.3 Applet 小程序的测试.....	251
10.3 合并 Application 和 Applet 小程序.....	252
10.4 Applet 小程序的应用.....	254
10.4.1 显示外部参数的 Applet 小程序.....	254
10.4.2 显示时间的 Applet 小程序.....	255
10.4.3 播放声音的 Applet 小程序.....	256
10.4.4 在 Applet 小程序界面中 添加菜单.....	258

10.5 Java 的图形绘制.....	259
10.5.1 Graphics 图形类.....	259
10.5.2 绘图坐标体系.....	261
10.5.3 图形绘制示例.....	264
第 11 章 多线程编程.....	277
11.1 线程对象.....	277
11.1.1 Runnable 接口和 Thread 类.....	277
11.1.2 创建启动线程对象.....	279
11.1.3 线程组 ThreadGroup 类.....	281
11.1.4 创建具有多线程功能的 Applet 小程序对象.....	283
11.2 Java 多线程机制.....	284
11.2.1 线程对象生命周期和 状态.....	284
11.2.2 线程对象的基本控制.....	285
11.2.3 多线程的问题.....	285
11.2.4 线程间的同步控制机制.....	286
11.3 多线程应用程序.....	291
11.3.1 Thread 类定义的 sleep 方法.....	291
11.3.2 Object 类定义的线程 控制方法.....	292
11.3.3 账户数据操作问题.....	295
11.3.4 计数并显示计数值的 Applet 小程序.....	296
11.3.5 实时显示时钟的 Applet 小程序.....	297
11.3.6 滚动显示文字信息的 Applet 小程序.....	299
11.3.7 简单图形动画显示.....	302
第 12 章 网络编程.....	306
12.1 java.net 包.....	306
12.2 建立 URL 网络通信.....	307
12.2.1 URL、URLConnection、 HttpURLConnection 类.....	308
12.2.2 创建并连接 URL 对象.....	311

12.2.3 获取各种网络资源.....	313	14.1.2 JDBC 的任务.....	401
12.3 建立 Socket 网络通信.....	319	14.1.3 JDBC 的主要接口和类.....	402
12.3.1 Socket 和 ServerSocket 类.....	319	14.2 数据库操作命令 SQL.....	409
12.3.2 建立服务器和客户机 Socket 通信程序框架.....	324	14.2.1 创建、删除数据库.....	409
12.3.3 Socket 通信应用程序.....	329	14.2.2 创建、删除、修改 基本表格.....	410
12.4 建立 UDP 网络通信.....	339	14.2.3 创建、删除索引.....	411
12.4.1 DatagramSocket 和 DatagramPacket 类.....	340	14.2.4 创建、删除视图.....	411
12.4.2 建立 UDP 通信程序框架.....	342	14.2.5 数据查询.....	412
12.4.3 UDP 通信应用程序.....	345	14.2.6 数据更新.....	412
第 13 章 多媒体编程.....	356	14.3 创建 Java 数据库应用模型.....	413
13.1 Java 多媒体处理类简介.....	356	14.3.1 创建数据源.....	413
13.2 Java 2D API 的应用.....	358	14.3.2 加载数据库驱动程序.....	416
13.2.1 绘制及处理二维图形.....	358	14.3.3 连接数据库.....	417
13.2.2 绘制及处理二维文字.....	367	14.3.4 操作数据库.....	418
13.2.3 处理二维图像.....	369	14.3.5 获取数据结果集.....	419
13.3 Java 动画制作技术.....	386	14.4 Java 数据库应用示例.....	421
13.3.1 动态文字的显示.....	386	14.4.1 显示查询数据库结果.....	421
13.3.2 动态显示图像.....	388	14.4.2 向数据库中添加记录.....	424
13.3.3 利用缓冲技术动态 显示图像.....	392	14.4.3 数据库的测试.....	429
第 14 章 数据库编程.....	399	14.4.4 Applet 数据库应用程序.....	442
14.1 JDBC API.....	399	附录 A Java 程序编写规则.....	449
14.1.1 JDBC 的组成.....	400	附录 B J2SDK 常用文档的 使用摘要.....	457

第 1 章 Java 语言综述

本章将介绍 Java 语言的主要特点、Java 语言程序的运行机制,以及 Java 语言程序的开发平台和运行环境的安装和使用等内容。

1.1 Java 语言简介

Java 语言的出现以及它与计算机系统平台无关的独特性从根本上改变了计算机应用程序的使用方式,由于 Java 程序可以不依赖任何现有的某个操作系统而直接运行在某种机器或 CPU(中央处理器)上,例如,Java 程序可以运行在 Internet 网络中的任何一款计算机上,以及像手机等嵌入式设备上,因此,Java 语言已经成为广受欢迎的开发和编写计算机程序的语言之一,Java 语言除了可以实现其他计算机语言所能实现的功能外,其突出的表现是在网络和嵌入式设备中的应用。

1.1.1 Java 语言的历史与现状

Java 语言来自于 Sun Microsystem 公司一个称为 Green 的项目。1991 年,Green 项目的最初目的是为家用消费电子产品开发一个分布式代码系统,它可以把 E-mail(电子邮件)等信息发送给电冰箱、电视机等家用电器,以及嵌入式设备上,对它们进行可编程控制,并和它们进行交互式信息交流。该项目组成员开始准备采用 C++语言,但他们感到 C++语言太复杂,安全性也差,无法满足项目设计的需要,最后决定基于 C++语言开发一种新的编程语言,并临时为它起名为 Oak 语言(一种橡树的名字),后来正式改名为 Java 语言。Oak 语言是一种用于嵌入式设备和网络上的精巧而安全的语言,Sun 公司曾依此投标一个交互式电视项目,但结果被 SGI 公司击败。1994 年,WWW(World Wide Web)在 Internet 上如火如荼地发展起来,这时网景公司开发的 Netscape(Internet 网络浏览器)启发了 Oak 项目组成员,与此同时他们又意识到在 Internet 上浏览网页需要的是一个中性浏览器,即不依赖于任何硬件平台和软件平台,应是一种实时性较高、可靠安全、有交互功能的浏览器,于是 Oak 项目组成员决定用 Java 语言开发一个新的 Web 浏览器,他们使用 Java 语言编制的 HotJava 浏览器得到了 Sun 公司当时的首席执行官 Scott McNealy 的大力支持,1995 年,HotJava 浏览器发布后立刻引起了计算机产业界的轰动,Java 语言得到了肯定,从此 Java 进军了 Internet 广域网。

Java 语言是 Sun 公司推出的一种新一代面向对象的程序设计语言,特别适合 Internet 应用程序的开发,它的与硬、软件平台的无关性直接威胁到 Windows 和 Intel 的垄断地位。1996 年初,Java 语言 1.0 版本正式发表,目前,Java 语言已成为最卓越的网络程序设计语言之一,它的出现对整个计算机软件业的发展产生了重大及深远的影响,对传统的计算模

型提出了新的挑战。Java 语言作为软件开发的一种革命性的技术，其地位已被确立，主要表现在以下几个方面。

(1) 计算机产业的许多大公司购买了 Java 语言的使用许可证，其中包括 IBM、Apple、DEC、Adobe、Silicon Graphics、HP、Oracel、Toshiba、Netscape、Novell、SGI 等公司以及最不愿的 Microsoft 公司，这一点说明了 Java 语言已得到了软件工业界的认可。

(2) 众多的软件开发商开始支持 Java 语言的软件产品。例如，Sun 公司自己的 Java 语言开发环境 Java Workshop，以及 Borland 公司开发的基于 Java 语言的快速应用程序开发环境 JBuilder 等，它们的产品直接推动了 Java 语言进入 PC 机软件市场。一些数据库厂商，例如，Illustra、Sysbase、Versant、Oracle 等公司也都在开发支持 HTML 和 Java 语言的接口，以适用于当今以 Internet 网络为中心的计算机时代。

(3) Java 语言不仅仅是一种程序设计语言，它还可作为一种计算机操作系统来应用，Java 语言可以不依赖于任何现有的操作系统，直接运行在某种机器或 CPU 上，是搭建 Internet 和 Intranet(企业内部网)等网络时首选的语言之一。

经过十几年的发展，Java 语言已经形成了比较完整的、功能齐全的计算机语言体系，并成功地应用到电子商务、网站建设等大型软件应用系统中，以及手机应用软件等嵌入式设备中。目前，Sun 公司已发布了 Java 2 标准，同时支持 Java 2 开发与运行的工具已发展到 J2SDK 1.6 版本。

Java 语言正在不断发展和完善，Sun 公司是该语言发展的主要推动者，它提供的较为通用的开发、编译、运行环境有 JDK(Java Develop Kit)、J2SDK(Java 2 Software Development Kit)与 JWS(Java Workshop)等，此外，还有很多公司也研制了 Java 语言的开发、编译、运行、调试的可视化集成环境，例如，Microsoft 公司的 Visual J++、Borland 公司的 JBuilder、Symmatec 公司的 Visual Cafe、IBM 公司的 Eclipse 和 IBM VisualAge for Java 等。Java 语言的正确性与运行效率的不断提高，以及开发环境的逐步完善，使得计算机软件开发人员使用 Java 语言编程和使用 C++等语言编程一样的方便、快捷。

目前，Java 2 标准规定了 Java 语言的应用领域，其主要为 4 个方面：网络服务器应用程序、一般计算机应用程序、嵌入式设备应用程序、智能卡应用程序。

1.1.2 Java 语言的特点

Java 语言是由 C++语言发展而来的，因此，它继承了 C++语言的所有优良特征，它是一种彻底的纯面向对象的程序设计语言，同时，它又去掉了 C++语言的一些容易引起错误的特性，例如，Java 语言没有指针，避免了使用指针直接访问物理存储器带来的风险，以及 Java 语言用接口取代了 C++语言中容易引起混乱的多重继承机制等。

用一句高度概括的话来描述 Java 语言的特征则是“Java 是一个简单的、面向对象的、分布的、解释的、健壮的、安全的、独立于平台的、可移植的、可扩展的、高性能的、多线程的以及动态的程序设计语言。”

1. 简单性

Java 语言最初是为对家用电器等嵌入式设备进行集成控制而设计的一种语言，因此它

简单明了,易于学习,与 C++语言风格极为相似,但却比 C++语言简单得多。从某种意义上讲,Java 语言是 C/C++语言的一个变种,因此,C++语言程序员可以很快掌握 Java 语言编程技术。

Java 语言摒弃了 C++语言中容易引发程序错误的地方。例如,指针和内存管理,在高级编程语言的所有特性中,不是绝对需要的,Java 语言都已删去。此外,Java 语言没有算符过载、标题(头)文件、预处理、指针、指针运算、结构、联合、多维数组、模板及隐式类型变换以及 goto 等不合乎逻辑的语句等。

Java 语言通过提供最基本的“方法”(函数)来完成指定任务,读者只需理解一些基本概念,就可以用它编写出适合于各种情况的应用程序。另外,Java 语言为程序开发者提供了丰富的类库,使程序编写变得更加容易、简单。

2. 小巧型

Java 语言是一种小巧型的语言,适合于在小型机上运行。它能有效地在 4MB 内存的嵌入式 PC 机上运行,其基本解释器及对类的支持系统只有 40KB 左右,加上标准类库和线程的支持也只有 220KB 左右,因此,它很适合应用在电话机、手机、电视机、电冰箱等家电设备中,以及在 Internet 广域网上进行动态发布。

3. 面向对象编程特性

Java 语言由 C++发展而来,是一种彻底的纯面向对象(Object-Oriented)的程序设计语言。它具备面向对象的 4 大特点:封装、继承、多态和动态,非常适用于大型软件的开发。

与其他面向对象编程语言一样,Java 语言的代码也是按类组织的,每个类除定义描述类属性的数据外,还定义了一组规定对象行为的方法。Java 语言的程序设计集中于对象的类封装及接口的定义上,它提供了简单的类机制以及动态的接口模型,对象中封装了它的状态变量以及相应的方法,实现了程序的模块化和信息隐藏。与其他面向对象的语言不同的是,Java 语言只支持单一类的继承,即每个类一次只能继承一个其他的类。Java 语言是通过接口来实现多重继承的。面向对象可以说是 Java 语言最重要的特性,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。Java 语言支持静态和动态风格的代码继承和重用。

4. 分布式特性

Java 语言支持网络应用程序的特性,这使得它成为一种分布式程序设计语言。Java 语言提供了一个支持 HTTP 和 FTP 等基于 TCP/IP 协议的子库(java.net 包),通过它可以完成各种层次上的网络连接,因此,使用 Java 语言编写的应用程序访问网络上的对象与访问本地对象几乎完全相同。Java 语言使程序设计者可以非常方便地创建分布式的 Client(客户机)和 Server(服务器)应用程序。

5. 编译特性

完成编写 Java 语言程序后,在开始运行之前,需要通过 Java 编译器编译 Java 源程序。编译器将 Java 源程序编译输出为一种被称为 bytecode(字节代码)的中间代码或类代码,它

并不是二进制的机器指令代码，但非常类似于机器指令代码。字节代码并非专对某一种特定机器或某种 CPU，它是 Java 语言程序在 Java 虚拟机支持下的可执行代码，因此，Java 语言程序可以在不同的计算机上执行而不需要重新编译源程序。

6. 解释特性

Java 语言通常又称为解释型语言。它编译后生成的不是特定的 CPU 机器代码，而是 Java 字节代码。Java 语言运行时要借助 Java 语言运行系统(Java 语言解释器，又称为 Java 虚拟机)。首先该系统对 Java 语言编译后的字节代码进行逐条解释，然后再执行它们。Java 语言解释器与硬件、软件平台有关，它可在某一特定硬、软件平台环境中直接运行 Java 语言程序的字节代码。Java 语言以字节代码形式发布的 Java 程序运行在 JVM(Java Virtual Machine，虚拟机)环境中，JVM 将字节代码翻译成具体的 CPU 机器指令，在不同的平台上用不同的 JVM 实现，与平台相关部分的工作是由 JVM 而不是由 Java 编译器来实现的。而平台的种类比起应用软件的数量毕竟要少得多。

7. 健壮特性

Java 语言是一种比 C++ 语言还强的强类型语言。Java 语言要求显式的方法声明，从而保证编译器可以发现方法调用错误，保证了程序更加可靠，Java 语言程序不可能造成计算机系统的崩溃。Java 语言系统仔细检测对内存的每次访问，确认它是合法的，而且不致引起任何问题，如果出现某种出乎意料之事，程序不会崩溃，而是将错误的异常类抛弃，应用程序或 JVM 会发现这类异常，并加以处理。

传统的计算机语言程序可以任意访问计算机的全部内存，程序可能(无意识地)修改内存中的任何值，这将造成不可预见或致命的问题。而 Java 语言不支持指针等操作，因此从语言体系中杜绝内存的非法访问，Java 语言程序只能访问内存中允许它们访问的那些部分，所以它不可能修改不适宜修改的内存值。

在一个 Java 语言程序中，需要定义所用对象(数字、字符、数组等)的类型，当程序编译时就可以检测类型的错误，该项功能可以帮助程序员发现应用程序的早期错误。

8. 安全性

Java 语言是一种安全的程序设计语言。Java 语言程序分为应用程序(Application)和小程序(Applet)两种类型。应用程序一般在本地执行，小程序可在网络上发布，但需要在 Web 浏览器中执行。为了保证执行从远端下载的 Applet 小程序不会对用户造成危害，它限制了 Applet 小程序访问本地资源，从而确保了 Applet 小程序的安全性。对应用于网络、分布环境下的 Java 语言程序需要防止病毒的入侵，一切对内存的访问都需要通过对象的实例变量来实现，这样就防止了程序员使用“特洛伊木马”等欺骗手段访问对象的私有成员，同时，在 Java 语言里删除了像指针和释放内存等 C++ 语言的功能，避免了指针操作中容易产生的安全性错误。

Java 语言程序在计算机上执行前，要经过很多次的测试，它经过代码校验，检查代码段的格式，检测对象操作是否过分以及试图改变一个对象的类型等。如果字节代码通过代码校验，没有返回错误，则表示代码没有堆栈上溢出和下溢出、所有操作代码参数类型都

是正确的、没有发生非法数据转换、访问对象操作是合法的、被验证的 Java 语言程序得以保证不突破 Java 语言的任何限制、可以安全地执行等。Java 语言的这种字节代码的验证方式可以确保发布在网络上的 Applet 小程序不含计算机病毒。

9. 中立性

Java 语言程序应用于每种计算机都是一样的。例如,简单的类型都是不变的。整数总是 32 位字长,长整数总是 64 位字长,这就可以使 Java 语言程序的移植变得容易,而且不需要重新编译。另外,为了使 Java 语言成为网络的一个组成部分,Java 语言源程序被编译成一种高层次的与机器无关的以及结构中立的中间文件格式(字节代码),该格式文件被设计在虚拟机(JVM)上运行,当计算机中安装了 Java 语言运行系统(JVM)时,该机器就可以执行这种中间代码。现在,支持 Java 语言程序运行的系统有 UNIX、Linux、Windows 等操作系统,即这些操作系统中已经嵌入了 Java 虚拟机。

10. 与平台无关性

Java 语言的与平台无关性是该语言的重要特性。Java 语言程序编译后生成的字节代码是一种与硬、软件平台无关的代码。在任何一种特定的硬、软件平台上,只要实现了 Java 虚拟机,Java 语言程序就可以运行。现在在 Internet 网络上或各种操作系统上都带有 Java 虚拟机,这使得 Java 语言开发人员开发的应用程序很容易在多种硬、软件平台上使用。

11. 可移植性

与语言本身体系结构的无关性使得 Java 语言应用程序可以在配备了 Java 语言解释器和运行环境的任何计算机系统中运行,这为 Java 语言应用软件的移植提供了良好基础。但仅如此还不够,如果基本数据类型设计依赖于具体的计算机和操作系统,例如,在 Windows 3.2 中整数(Integer)为 16 位,在 Windows 98 中整数为 32 位,在 DEC Alpha 中整数为 64 位,同样也为程序的移植带来很大不便。Java 语言是通过定义独立于硬、软件平台的基本数据类型及其相关运算来实现应用程序的可移植特性的。

目前,Java 语言的类库也实现了与不同平台的应用接口,而 Java 编译器本身也是用 Java 语言编写的,同样 Java 虚拟机也是由 Java 语言实现的,这使得 Java 语言系统本身具备了整体的可移植性。

12. 可扩展性

Java 语言程序可与用其他语言编写的已经存在的程序库相连接。由于 Java 语言的数据结构与 C 语言的数据结构的类型极为相似,使得程序库之间的连接变得相当方便。同时,Java 语言还可以把内部方法映射成软件库所定义的功能,动态地链接到虚拟机上供其他程序使用。

13. 高性能性

Java 语言是一个解释型语言。解释型语言比编译型语言的运行速度慢。实际上 Java 语言程序平均运行速度比 C 程序慢 20 倍左右。为了解决运行速度问题,Java 语言的设计者们开发了“Just in time”编译器(也称为代码生成器),这种编译器可以在运行时把 Java 语

言的字节代码翻译成特定 CPU 的机器码, Sun 公司称转化成机器码的字节代码在性能上接近于 C 或 C++ 语言程序。Java 语言解释器与其他解释执行语言(如 BASIC)不同, Java 语言字节码的设计使之能很容易地直接转换成对应于特定 CPU 的机器码, 从而得到较高的性能。

14. 多线程特性

多线程特性使应用程序能够并行执行, 且多线程的同步机制保证了对共享数据的正确操作, 通过使用多线程技术, Java 应用程序设计者可以分别使用不同的线程完成特定的行为, 而不需要采用全局的事件循环机制, 容易实现网络上的实时交互行为。Java 语言提供的多线程功能使得在一个程序中可同时执行多个小任务, 例如, 在一个线程中完成某一耗时计算的同时另一个线程可以实现与用户进行交互对话等操作, 而多线程带来的更大的好处是使应用程序具有更好的交互性能和实时控制性能, 当然多线程以及多线程的性能还取决于计算机操作系统本身(UNIX, Windows, Macintosh 等)。

15. 动态性

Java 语言可在类库中自由地加入新的方法和实例变量而不影响用户程序的执行。Java 语言程序可以把一个类动态地链接到运行系统中, Java 语言通过接口支持多重继承使之比严格的类继承具有更灵活的方式和扩展性, 使其具有动态特性。Java 语言的动态特性是其面向对象设计方法的扩展, 它允许程序动态地装入运行过程中所需要的类, 这是其他面向对象程序设计语言 C++ 等所无法实现的。

16. 收集无用存储单元

用 C 及 C++ 语言编写软件的编程人员需要仔细跟踪所用的内存块, 当一个内存块不再使用时, 需要让程序释放它后方可再用。在大型软件项目中很难控制内存操作, 因此往往会成为程序出错或内存不足等问题的根源。使用 Java 语言编写程序时, 编程人员不必为管理内存而担心, 因为 Java 语言运行系统中有一个称为“无用单元收集器”的内置程序, 它通过扫描内存自动释放那些不再使用的内存块, Java 语言的这种自动垃圾收集机制, 对不再被程序引用的对象自动取消其所占资源, 这就彻底消除了出现存储器泄漏之类的错误, 并免去了程序员管理存储器的繁琐工作。

1.1.3 Java 程序的运行机制

Java 语言源程序在编译后生成的不是某种 CPU 的指令码, 而是 Java 语言特有的字节代码。Java 字节代码运行在 Java 虚拟机上。Java 虚拟机类似一个小巧而高效的 CPU。Java 虚拟机底层的运行系统把字节代码转化成实际硬件设备的调用, 但是, Java 虚拟机未必非要运行在某种操作系统之上, 它的下面可以直接是各种 CPU 芯片。

Java 语言应用程序可分为两种形式: Application 应用程序和 Applet 小程序。两种 Java 源程序(以 .java 为后缀的文件)通过 Java 编译器生成可执行的 Java 字节代码(以 .class 为后缀的类代码文件), Java 的类代码文件即是 Java 的可执行文件。Application 应用程序是由 Java 虚拟机支持运行的, Applet 小程序是由嵌入了 Java 虚拟机的 Web 浏览器支持运行的。目前

几乎所有的 Web 浏览器都嵌入了 Java 虚拟机, 计算机 CPU 和 Web 浏览器支持 Java 应用程序运行的层次关系如图 1-1 所示。

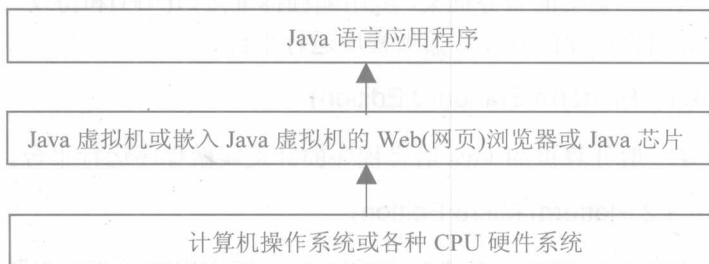


图 1-1 支持 Java 程序运行层次图

对于每一种操作系统的 Java 解释器(面向 Java 语言程序的一个独立运行系统)是不同的, 但实现的 Java 虚拟机是相同的, 这就是 Java 语言与平台无关的关键所在。Java 虚拟机由 5 个部分组成: 一组指令集、一组寄存器、一个堆栈、一个无用内存单元收集器、一个方法区域, 这 5 个部分是 Java 虚拟机的逻辑抽象成分, 它不依赖于任何实现技术或组织, 但是它们的功能在某种真实机器上是一致的。

Java 虚拟机支持 240 多个字节代码, 每个字节代码执行一种基本的 CPU 运算。Java 语言指令集相当于 Java 语言程序的“汇编语言”, Java 语言指令集的一条指令有一个操作码、0 个或多个操作数, 操作码指定实施的操作, 操作数是操作码所需的参数, Java 语言的操作码均为一个字节长, 操作数没有限制。Java 虚拟机也有类似微处理器中的寄存器, 它们用于保存机器的运行状态, Java 虚拟机寄存器有 4 种: Java 语言程序计数器(pc), 指向操作数栈顶端的指针(optop), 指向当前执行方法的执行环境指针(frame), 指向当前执行方法局部变量区第一个变量的指针(var)。Java 虚拟机是用堆栈来处理数据传递的, 它不定义或使用寄存器来传递或接受参数, 其目的是为了保证指令集的简洁性和实现时的高效性。Java 虚拟机的堆栈有 3 个区域: 局部变量区、执行环境区和操作数区。Java 虚拟机的堆栈是 Java 应用程序运行时动态分配的对象存储区域并是可自动回收的。

1.2 Java 程序开发环境

J2SDK 是 Java 2 Software Development Kit 的简称, 其前身是 JDK(Java Development Kit), J2SDK 和 JDK 都是 Sun Microsystems 公司推出的一套 Java 语言程序开发工具兼做运行 Java 语言程序的平台, Java 运行环境 JRE(Java Run Environment)也被称为 Java 虚拟机 JVM, 支持 Java 应用程序的执行, 因此, 编写 Java 语言程序, J2SDK 是必备的。

1.2.1 Java 语言程序开发平台

针对 Java 语言程序主要应用于网络服务器、一般计算机、嵌入式设备、智能卡这 4 个领域, Java 2 提供了开发这 4 类应用程序的标准框架和运行环境, 其开发平台有如下几种。

1. J2EE(Java 2 Platform Enterprise Edition)

J2EE 平台包含有一整套的服务框架、应用编程标准接口(API)和协议等,它适用于在网络中服务器端 Java 语言程序的开发并兼做程序运行平台。

2. J2SE(Java 2 Platform Standard Edition)

J2SE 适用于在一般计算机的 Java 语言程序的开发兼做程序运行平台。

3. J2ME(Java 2 Platform Micro Edition)

J2ME 适用于消费类电子产品中嵌入式系统 Java 语言程序的开发兼做程序运行平台,适合某种产品的单一要求,例如,手机、固定电话等设备。

4. JavaCard

JavaCard 适用于在 Smart Card(智能 IC 卡)上运行的 Java 语言程序的开发兼做程序运行平台。

1.2.2 Java 程序基础开发运行平台——J2SDK

J2SDK 包含了针对 Java 语言程序的基本编译器、一些实用工具、运行环境,以及用来开发和运行 Java Application 和 Java Applet 程序的 API, J2SDK 1.6 版本包括的内容如图 1-2 所示。

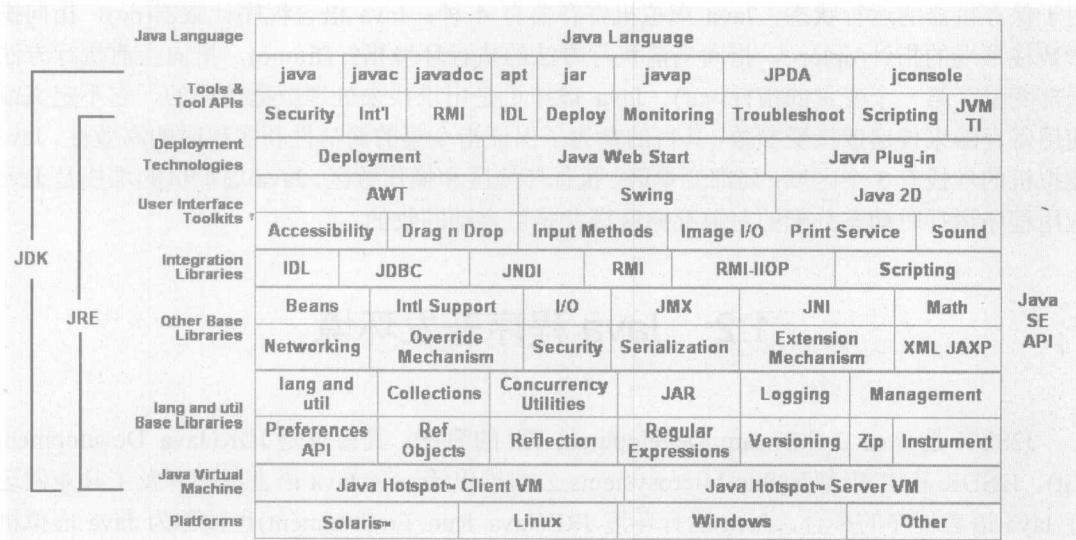


图 1-2 J2SDK 1.6 版本的内容

其中, JRE 为 Java 语言程序(java language)运行环境,包括两部分,一部分为运行 Java 程序的 Java 虚拟机(Java Virtual Machine, JVM);另一部分是支持 Java 程序运行的类库(简称 Java API),类库有基础类库(Base Libraries)和工具类库(Toolkits)等,当在 Java 应用程序中有调用某个 Java 类库时,该类库则被加载,为应用程序提供调用。JDK 为开发环境,包