



教育部考试中心

授权独家出版

全国计算机等级考试

二级教程

——Java语言程序设计

(2008年版)



高等教育出版社
Higher Education Press

全国计算机等级考试

二级教程

——Java 语言程序设计(2008 年版)

教育部考试中心



高等教育出版社

内容提要

由国家教育部考试中心推出的计算机等级考试是一种客观、公正、科学的专门测试计算机应用人员的计算机知识与技能的全国性考试,它面向社会,服务于社会。

本书在教育部考试中心组织下、在全国计算机等级考试委员会指导下,由有关专家执笔编写而成。本书按照《全国计算机等级考试二级 Java 语言程序设计考试大纲(2007 年版)》的要求编写,内容包括:Java 体系结构、基本数据类型、流程控制语句、类、数组和字符串操作、输入/输出及文件操作、图形用户界面编写、线程和串行化技术、Applet 程序设计以及应用开发工具和安装使用等。本书是参加全国计算机等级考试二级 Java 语言程序设计的考生的良师益友,是教育部考试中心指定教材,也可作为欲学习 Java 编程的读者的参考书。

图书在版编目(CIP)数据

全国计算机等级考试二级教程. Java 语言程序设计:2008
年版/教育部考试中心. —北京:高等教育出版社,2007.9
ISBN 978 - 7 - 04 - 022947 - 9

I. 全… II. 教… III. ①电子计算机 - 水平考试 - 教材
②Java 语言 - 程序设计 - 水平考试 - 教材 IV. TP3

中国版本图书馆 CIP 数据核字(2007)第 132527 号

策划编辑 田晓兰 责任编辑 古锋 何新权 封面设计 张志奇
责任校对 王效珍 责任印制 朱学忠

出版发行 高等教育出版社
社 址 北京市西城区德外大街 4 号
邮政编码 100011
总 机 010 - 58581000

经 销 蓝色畅想图书发行有限公司
印 刷 中青印刷厂

开 本 787 × 1092 1/16
印 张 22.75
字 数 560 000

购书热线 010 - 58581118
免费咨询 800 - 810 - 0598
网 址 <http://www.hep.edu.cn>
<http://www.hep.com.cn>
网上订购 <http://www.landraco.com>
<http://www.landraco.com.cn>
畅想教育 <http://www.widedu.com>

版 次 2007 年 9 月第 1 版
印 次 2007 年 9 月第 1 次印刷
定 价 37.40 元

本书如有缺页、倒页、脱页等质量问题,请到所购图书销售部门联系调换。

版权所有 侵权必究

物料号 22947 - 00

大力推行全国计算机等级考试 为发展知识经济、信息产业和培养计算机 专门人才作出贡献 (代序)

中国科学院院士 北京大学信息与工程科学学部主任
全国计算机等级考试委员会主任委员
杨芙清

当今,人类正在步入一个以智力资源的占有和配置,知识生产、分配和使用为最重要因素的知识经济时代,也就是小平同志提出的“科学技术是第一生产力”的时代。科教是经济发展的基础,知识是人类创新的源泉。基础研究的科学发现、应用研究的原理探索和开发研究的技术发明,三者之间的联系愈来愈紧密,转换周期日趋缩短。世界各国的竞争已成为以经济为基础、以科技(特别是高科技)为先导的综合国力的竞争。

在高科技中,信息科学技术是知识高度密集、学科高度综合,具有科学与技术融合特征的学科。它直接渗透到经济、文化和社会的各个领域,迅速改变着人们的观念、生活和社会的结构,是当代发展知识经济的支柱之一。

在信息科学技术中,微电子是基础,计算机硬件及通信设施是载体,计算机软件是核心。软件是人类知识的固化,是知识经济的基本表征,软件已成为信息时代的新型“物理设施”。人类抽象的经验、知识正逐步由软件予以精确地体现。在信息时代,软件是信息化的核心,国民经济和国防建设、社会发展、人民生活都离不开软件,软件无处不在。软件产业是增长最快的朝阳产业,是具有高额附加值、高投入/高产出、无污染、低能耗的绿色产业。软件产业的发展将推动知识经济的进程,促进从注重量的增长向注重质的提高的方向发展,是典型的知识型产业。软件产业是关系到国家经济安全和文化安全,体现国家综合实力,决定21世纪国际竞争地位的战略产业。

为了适应知识经济发展的需要,大力推动信息产业的发展,需要在全民中普及计算机的基本知识,广开渠道,培养和造就一批又一批能熟练运用计算机和软件技术的各行各业的专门人才。

1994年,原国家教委(现教育部)推出了全国计算机等级考试,它是一种重视应试人员对计算机和软件的实际掌握能力的考试。它不限制报考人员的学历背景,任何年龄段的人员都可以报考。这就为培养各行各业计算机的应用人才开辟了一条广阔的道路。

1994年是推出计算机等级考试的第一年,当年参加考试的有1万余人;而2006年,年报考人数已近356万人。截至2007年上半年,全国计算机等级考试共开考25次,考生人数累计达2269万人,其中有826万人获得了不同级别的计算机等级证书。

事实说明,鼓励社会各阶层的人士通过各种途径掌握计算机应用技术,并运用等级考试对他们的才干予以认真的、有权威性的认证,是一种人才培养的有效途径,是比较符合我国具体情况



的。等级考试也为用人单位录用和考核人员提供了一种测评手段。从有关公司对等级考试所作的社会抽样调查结果看,不论是管理人员还是应试人员,对该项考试的内容和形式都给予了充分的肯定。

计算机等级考试所取得的良好效果,也同全国各有关单位专家们在等级考试的大纲编写、试题设计、阅卷评分及效果分析等多项工作中所付出的大量心血和辛勤劳动密切相关,他们为这项工作的顺利开展作出了重要的贡献。

计算机与软件技术是一项日新月异的高新技术。计算机等级考试大纲有必要根据计算机与软件技术在近年的新发展,进行适当的修正,从而使等级考试更能反映当前计算机与软件技术的应用实际,使培养计算机应用人才的基础工作更健康地向前发展。

从面临知识经济的机遇与挑战这样一个社会大环境的背景出发,考察全国计算机等级考试,就会看到,这一举措是符合知识经济和信息产业的发展方向的,是值得大力推行的。

我们相信,在 21 世纪知识经济和信息产业加快发展的形势下,在教育部考试中心的精心组织领导下,在全国各有关专家们的大力配合下,全国计算机等级考试一定会以更新的面貌出现,从而为我国培养计算机应用专门人才的宏大事业作出更多的贡献。

2007 年 7 月

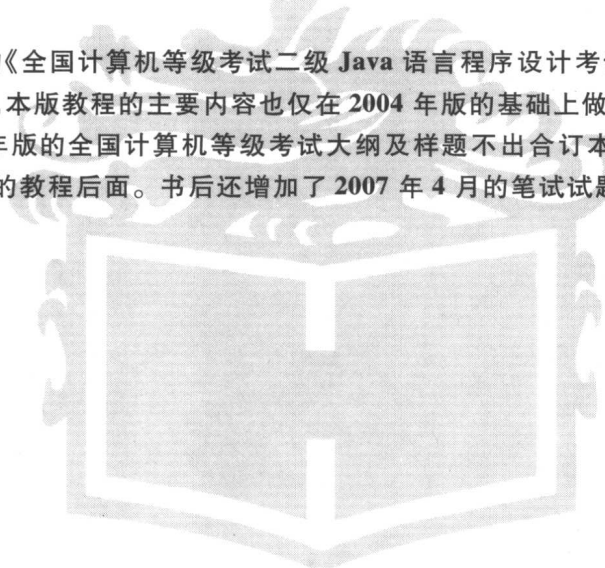
前 言

本书是根据教育部考试中心制定的《全国计算机等级考试二级 Java 语言程序设计考试大纲(2007 年版)》的要求,在 2004 年版的基础上修订而成。本书以最新 JDK1.4.2 版本为基础,介绍 Java 语言编程基础知识以及应用开发技术。本书中的实例均通过编译并可直接在 JDK1.4.2 上运行,每章包括练习题,供考生练习使用,是考生复习备考的必备教材。

本书由清华大学计算机系柳西玲教授主编,参加编写的作者有:第 1、2、6、12 章由柳西玲编写;第 3、4、9 章由许斌编写;第 8、10、11 章由郎波编写;第 5、7 章由金铁鹰编写。全书经南开大学辛运伟教授审阅。由于编写时间仓促,难免有疏误之处,请读者提出宝贵意见,以便修订时改进。

鉴于 2007 年版的《全国计算机等级考试二级 Java 语言程序设计考试大纲》在 2004 年版的基础上未做大的变动,本版教程的主要内容也仅在 2004 年版的基础上做了少许修改和完善。为方便考生使用,2007 年版的全国计算机等级考试大纲及样题不出合订本,各科目大纲及样题以附录的形式附于相应的教程后面。书后还增加了 2007 年 4 月的笔试试题,供考生参考。

编 者
2007 年 7 月



目 录

第 1 章 Java 语言概论	1	3.2.2 二元算术运算符	27
1.1 Java 语言简介	1	3.2.3 算术运算符的优先级	29
1.1.1 Java 语言的由来	1	3.3 关系运算符和关系表达式	30
1.1.2 Java 语言的特点及优势	2	3.4 布尔逻辑运算符和布尔逻辑表达式	32
1.1.3 Java 语言实现的机制	4	3.5 位运算符和位运算表达式	34
1.2 Java 的体系结构	5	3.5.1 位逻辑运算符	35
1.2.1 JDK 目录结构	5	3.5.2 移位运算符	36
1.2.2 Java 的 API 结构	6	3.5.3 位运算符的优先级	37
1.2.3 Java 源程序结构	8	3.6 赋值运算符和赋值表达式	38
1.2.4 Java 程序编写及运行的过程	9	3.6.1 赋值运算符	38
1.2.5 Java 程序举例	11	3.6.2 扩展赋值运算符	38
习题	14	3.7 条件运算符与条件表达式	39
第 2 章 简单数据类型	15	3.8 运算符的优先级和复杂表达式	39
2.1 概述	15	3.9 表达式语句	40
2.2 简单数据类型	17	3.10 完整的应用程序	41
2.2.1 整型数据	17	习题	47
2.2.2 浮点型数据	18	第 4 章 流程控制	50
2.2.3 布尔型数据	18	4.1 概述	50
2.2.4 字符型数据	19	4.2 分支语句	50
2.3 各类数据之间的转换	19	4.2.1 条件语句	50
2.3.1 优先关系	20	4.2.2 多分支语句	54
2.3.2 类型的自动转换规则	20	4.3 循环语句	56
2.3.3 类型的强制转换	21	4.3.1 while 循环	56
2.3.4 实例	21	4.3.2 do - while 循环	57
2.4 Java 类库中对简单类型数据的类包装	21	4.3.3 for 循环	58
习题	22	4.4 跳转语句	60
第 3 章 运算符和表达式	24	4.4.1 break 语句	60
3.1 概述	24	4.4.2 continue 语句	61
3.1.1 运算符	24	4.4.3 return 语句	62
3.1.2 表达式	25	4.5 循环语句与分支语句的嵌套	63
3.2 算术运算符和算术表达式	25	4.6 递归	67
3.2.1 一元算术运算符	25	习题	68

第 5 章 类、数组和字符串操作	72	7.2.1 创建文件	127
5.1 概述	72	7.2.2 File 类提供的方法	128
5.1.1 类定义	72	7.2.3 程序实例	129
5.1.2 对象、接口与包	77	7.2.4 随机文件流	130
5.1.3 类成员修饰符、继承、内部类、类库 ..	81	7.2.5 程序实例	130
5.1.4 内部类的应用实例	91	7.2.6 压缩文件流	134
5.2 一维数组	92	7.2.7 程序实例	134
5.2.1 定义数组、创建数组	92	7.3 字节 I/O 流	139
5.2.2 初始化数组	92	7.3.1 字节输入流	139
5.2.3 实例	94	7.3.2 字节输出流	139
5.3 多维数组	94	7.3.3 内存的读写	140
5.3.1 数组边界	98	7.4 字符类 I/O 流	140
5.3.2 复制数组	98	7.4.1 字符类输入流	141
5.3.3 调整数组大小	99	7.4.2 字符类输出流	141
5.4 字符串操作	99	7.4.3 程序实例	142
5.4.1 字符串的表示	99	7.5 过滤流	145
5.4.2 字符串的访问	100	7.6 管道流	146
5.4.3 字符串的修改	101	7.7 J2SE1.4 提供的新的 I/O 功能 ..	146
5.4.4 字符串的比较	102	7.7.1 内存映射文件	146
5.4.5 其他操作	102	7.7.2 文件通道	147
5.4.6 程序实例	102	7.7.3 CRC32 类	147
习题	103	7.7.4 程序实例	147
第 6 章 异常处理	105	7.8 输入输出流和正则表达式	148
6.1 概述	105	7.8.1 Pattern 类	148
6.2 异常类型	107	7.8.2 Matcher 类	148
6.2.1 捕获异常	108	7.8.3 程序实例	149
6.2.2 声明异常	109	习题	150
6.2.3 抛出异常	110	第 8 章 线程与对象串行化	155
6.3 处理异常	111	8.1 线程的概念	155
6.3.1 try 和 catch 语句	111	8.1.1 什么是线程	155
6.3.2 finally 语句	114	8.1.2 Java 中的线程模型	156
6.3.3 异常处理的原则	115	8.2 线程的创建	156
习题	118	8.3 线程的调度与线程控制	159
第 7 章 输入输出及文件操作	121	8.3.1 线程优先级与线程调度策略	159
7.1 概述	121	8.3.2 线程的基本控制	160
7.1.1 计算机数据的 I/O 方向	121	8.4 线程同步	163
7.1.2 Java 中包含的输入/输出流的类 ..	122	8.4.1 多线程并发操作中的问题	163
7.2 文件	127	8.4.2 对象的加锁及其操作	166

8.4.3 死锁的防治	168	9.6.1 构件的分类	226
8.4.4 线程间的交互 wait()和 notify()	169	9.6.2 使用 Swing 的基本规则	227
8.4.5 不建议使用的一些方法	172	9.6.3 各种容器面板和构件	228
8.5 线程状态与生命周期	173	9.6.4 布局管理器	240
8.6 线程相关的其他类与方法	174	9.7 Swing 的事件处理机制	241
8.6.1 支持线程的类	174	习题	242
8.6.2 线程组	174	第 10 章 Applet 程序设计	247
8.6.3 Thread 类的其他方法	175	10.1 Applet 概述	247
8.7 对象的串行化	176	10.1.1 Applet 的概念	247
8.7.1 串行化概念和目的	176	10.1.2 Applet 的生命周期概念	248
8.7.2 串行化对象的方法	176	10.1.3 Applet 的类层次结构	250
8.7.3 构造可串行化对象的类	179	10.1.4 Applet 类 API 概述	250
8.7.4 定制串行化	180	10.1.5 Applet 的关键方法	252
8.7.5 串行化中对敏感信息的保护	185	10.1.6 Applet 的显示	253
8.7.6 串行化的注意事项	186	10.2 Applet 的编写	255
习题	186	10.2.1 Applet 编写的步骤	255
第 9 章 编写图形用户界面	189	10.2.2 用户 Applet 类的定义	256
9.1 概述	189	10.2.3 在 HTML 页中包含 Applet	256
9.2 用 AWT 编写图形用户界面	189	10.3 Applet 中的图形化用户界面 GUI	261
9.2.1 java.awt 包	189	10.3.1 基于 AWT 构件的 Applet 用户 界面	261
9.2.2 构件和容器	190	10.3.2 Applet 中使用弹出式窗口	262
9.2.3 常用容器	191	10.3.3 基于 Swing 的 Applet 用户界面	265
9.2.4 LayoutManager(布局管理器)	193	10.3.4 Applet 中的事件处理	268
9.3 AWT 事件处理模型	200	10.4 Applet 的多媒体支持	270
9.3.1 事件类	202	10.4.1 显示图像	270
9.3.2 事件监听器	203	10.4.2 动画制作	271
9.3.3 AWT 事件及其相应的监听器 接口	205	10.4.3 播放声音	274
9.3.4 事件适配器	208	10.5 Applet 的安全控制	277
9.4 AWT 构件库	211	10.5.1 Applet 的安全限制	277
9.4.1 基本构件的应用	211	10.5.2 Java 中的沙箱模型	278
9.4.2 构件与监听器的对应关系	219	10.5.3 Java 2 的安全策略定义与实施	278
9.5 用 Swing 编写图形用户界面	220	10.5.4 Java 2 中基于数字签名的安全 控制	283
9.5.1 Swing 概述	220	10.6 Applet 与工作环境的通信	286
9.5.2 Swing 的类层次结构	222	10.6.1 同页面 Applet 之间的通信	286
9.5.3 Swing 的特性	224	10.6.2 Applet 与浏览器之间的通信	290
9.6 Swing 构件和容器	226		

10.6.3 Applet 的网络通信	293	12.2 网络通信(URL、Socket、数据报通信)	312
10.7 Applet 与 Application	293	12.3 J2ME 平台	312
习题	295	12.4 J2SE 平台	313
第 11 章 J2SDK 的下载和操作	298	12.5 J2EE 平台	314
11.1 J2SDK 的下载与安装	298	第 13 章 上机指导	316
11.1.1 J2SDK 的下载	298	13.1 上机考试系统使用说明	316
11.1.2 J2SDK 的安装	298	13.1.1 上机考试环境	316
11.2 J2SDK 的操作命令	299	13.1.2 上机考试时间	317
11.3 Java 编程规范	301	13.1.3 上机考试题型及分值	317
11.3.1 Java 编程规范的作用与意义	301	13.1.4 上机考试登录	317
11.3.2 Java 命名约定	301	13.1.5 试题内容查阅工具的使用	320
11.3.3 Java 注释规则	302	13.1.6 考生文件夹	322
11.3.4 Java 源文件结构规则	304	13.2 上机考试内容	323
11.3.5 Java 源代码排版规则	305	13.2.1 基本操作	323
11.3.6 编程建议	306	13.2.2 简单应用	324
习题	308	13.2.3 综合应用	326
第 12 章 Java 的应用	309	13.3 上机考试样题	327
12.1 JDBC 的概念及利用 JDBC 访问数据库	309		
附录 1 全国计算机等级考试二级 Java 语言程序设计考试大纲(2007 年版)	332		
附录 2 全国计算机等级考试二级 Java 语言程序设计样题及参考答案	335		
附录 3 2007 年 4 月全国计算机等级考试二级笔试试题——Java 语言程序设计	341		
附录 4 习题参考答案	349		

第 1 章

Java 语言概论

Java 语言是 1995 年 5 月由 Sun 公司在 SunWorld 大会上发布的,从此,这一新一代的网络计算机语言受到广泛青睐,很快兴起了 Java 的热潮。仅仅短短的几年,它为 Web 提供了简便而功能强大的 API 接口,为 Web 提供了动态内容的交互页面技术,使 Web 网页翻开了新的一页,使互联网从以通信为主的功能扩展到网络应用系统服务。这种突破性的技术,对企业产生了革命性的影响,也显示了 Java 语言的强大生命力。

1.1 Java 语言简介

1.1.1 Java 语言的由来

1991 年, Sun 公司的 Jame Gosling、Bill Joe 等人,为电视、控制烤面包机等家用电器的交互操作开发了一个 Oak(一种橡树名字)软件,它是 Java 的前身。当时, Oak 并没有引起人们的注意,直到 1994 年,随着互联网和 3W 的飞速发展,他们用 Java 编制了 HotJava 浏览器,得到了 Sun 公司首席执行官 Scott McNealy 的支持,得以研制和发展。为了促销和法律的原因,1995 年 Oak 更名为 Java。Java 的得名还有段小插曲,一天, Java 小组成员正在喝 Java 咖啡时,议论给新语言取名字的问题,有人提议用 Java(Java 是印度尼西亚盛产咖啡的一个岛屿),这个提议得到其他成员的赞同,于是就采用了 Java 来命名此新语言。很快 Java 被工业界认可,许多大公司如 IBM、Microsoft、DEC 等购买了 Java 的使用权,并被美国著名杂志 PC Magazine 评为 1995 年十大优秀科技产品。从此,开始了 Java 应用的新篇章。

Java 的诞生是对传统计算机模式的挑战,对计算机软件开发和软件产业都产生了深远影响:

(1) 软件 4A 目标要求软件能达到任何人(Any One)在任何地方(Any Where)在任何时间(Any Time)对任何电子设备(Any Device)都能应用。这样能满足软件在异构平台上互操作、具有可伸缩性和重用性并可即插即用等分布式计算模式的需求。

注:为与计算机程序保持一致,本书中的所有变量均为正体。

(2) 基于构件开发方法的崛起,引出了 CORBA 国际标准软件体系结构和多层应用体系框架。在此基础上形成了 Java 2 平台和 .NET 平台两大派系,推动了整个 IT 业的发展。

(3) 对软件产业和工业企业产生了影响深远,软件从以开发为中心转向以服务为中心。中间件提供商、构件提供商、服务器软件提供商以及咨询服务商出现。企业必须重塑自我,B2B 的电子商务将带动整个新经济市场,使企业获得新的价值、新的增长、新的商机、新的管理。

(4) 对软件开发方法带来新的革命,重视使用第三方构件集成,利用平台的基础设施服务,实现开发各阶段的重用技术,重视开发团队的组织和文化理念,协作、创造、责任、诚信是人才的基本素质。

总之,目前已看到 Java 对信息时代的重要作用,未来还会不断发展,Java 在应用方面将会有更广阔的前景。

1.1.2 Java 语言的特点及优势

Java 是一个网络编程语言,它避免了许多其他编程语言的缺点,更好地利用了当前软件新技术,是一种新概念。首先,它作为编程语言,简单易学,利用了面向对象的技术基础,但又独立于硬件结构,具有可移植性、健壮性、安全性、高性能。其次,它围绕网络应用开发,最大限度地利用网络资源。它的小应用程序(Applet)在网络上的传输不受计算机 CPU 和环境限制。第三,Java 提供了丰富的类库,为编程人员提供快速和标准的应用接口,提高了应用软件的生产率。下面将详细地讨论 Java 语言的特点。

1. 简单易学

Java 由 Oak 发展而来,简单易学。其简单性首先表现在自身系统的精炼,它的基本解释程序和类库仅占 40 KB 空间,附加的基本标准类库和多线程支持也仅占 175 KB 的空间,力图用最少的系统完成尽可能多的功能。对硬件环境的要求不高,在很普通的计算机上就能运行。其次,它只要求理解一些基本概念就能编程,无需任何编程基础就可学会。它的基本语法与 C++ 类似,但删除了 C++ 中许多不常用而又难以理解的内容,如运算符重载、多重继承、指针操作等,大大降低了编程学习的难度。

2. 利用面向对象技术

过去的高级语言大多数是面向过程的语言,这类语言通过数据结构和算法去描述客观世界。它的致命缺点是问题的解决与程序是一一对应的关系,问题如有任何一点变更,程序也随之要修改。因此,可维护性和重用性很差。而面向对象的语言将客观世界看成由各种对象组成,对象客观实体可定义成大的结构——类(class),每个类有自己的特性(属性)和操作(方法)。面向对象技术使复杂的问题可以分解简化,大大地提高了软件重用性,能适应变化快速的需求。与其他面向对象语言相比,Java 利用面向对象技术更彻底。它把所有的 Java 应用和 Applet 都看做对象,按类封装。虽然 Java 的简单类型不是对象,但它提供了封装对象,使这些简单类型可以作为类来实现。其封装性实现了模块化和信息隐藏,继承性实现了代码重用,让用户可以自由地定义数据类型,建立自己的类库。

3. 分布式计算

Java 提供的类库支持 TCP/IP 协议,应用程序可通过 URL 地址,在访问网络上任何地方的对象时,如同访问本地文件一样简单。

4. 健壮性(也称鲁棒性)

Java 语言在编译和运行时都有比较严格的检查,过去许多编程语言的出错统计资料说明,最容易出错的是数据类型的不匹配和内存地址计算出错。Java 在这方面有独特的措施。首先,Java 是一种强类型语言,即它在编译和连接时进行大量的类型检查,防止不匹配数据类型的发生,对非法数据类型都将在编译和解释时指出。其次,Java 语言不允许使用指针访问内存,更不允许使用指针数组访问内存。另外 Java 设计有自动收集垃圾功能,这不仅防止了内存地址计算出错的问题,也省去了编程时对内存进行分配的烦恼。

5. 安全性

面向网络、分布式环境的 Java 语言,对非法侵入的防范是至关重要的,Java 语言必须提供充分的安全保障,它在运行程序时,有严格的访问权限检查。对字节代码执行前要检查,不允许使用指针,可防止对内存的非法入侵,它是目前安全性最佳的语言。

6. 跨平台(即体系结构中立)

Java 解释器采用生成与体系结构无关的字节代码指令的技术,只需安装 Java 运行系统就可保证 Java 程序可以在网络的任何地方运行。这对 Web 应用是很重要的,也是必需的,即同一个 Java 程序可在不同的处理器上运行,大大地降低了软件维护成本,提高了软件生产效率。

7. 可移植性

跨平台的特点保证了软件的可移植性。Java 的类库也具有可移植性。另外,Java 本身的编译器也用 Java 语言编写,运行系统的虚拟机用 C 语言实现,这样,Java 系统本身也具有可移植性。

8. 解释执行

Java 语言用字节码进行解释执行。字节本身带有许多编译时产生的信息,使它的连接过程更简单。

9. 高性能

60 年代,由于硬件条件限制,给人们留下解释器很慢的印象,另外,在 Java 刚问世时 JVM (Java 虚拟机)速度也不大理想,因此,至今在有些人的认知中也还存在误区,认为 Java 的缺点是性能慢。实际上,大量 Java 应用系统的成功案例,足以证明高性能已成为它的一大特点。字节码的设计很容易直接转换成一些特定 CPU 的机器码,其性能高到使人不感觉到与编译有什么差别。而且编译时还可对字节码优化,生成高质量的代码。另外,它的多线程技术也是性能提高的重要因素。

10. 多线程

多线程是 Java 程序的并发机制,它能同步共享数据、处理不同的事件。网络连接需要时间,如在网上采用事件循环机制会造成网上瓶颈。因此,多线程技术使网上实时交互很容易实现,为解决网上大数量的客户访问提供技术基础。

11. 动态性

Web 应用系统中最大的挑战是快速变化的需求,即软件的演化性。软件可扩充性、可伸缩性强,才能适应不断发展变化的环境。Java 语言带动了软件构件化方法,它本身的类库就可以自由地增加新方法或实例。Java 通过接口支持多重继承,使类继承具有更灵活的扩展性。它还可以随时插入构件和数据库。

12. Applet 的特点

Applet 是 Java 的一类特殊应用程序,它嵌入 HTML 中,随主页发布到互联网上。利用它可以实现多媒体的用户界面或复杂的计算。Applet 要求在支持 Java 的浏览器上运行,它使互联网上的信息能很容易地实现动态性和交互性。Java 类库提供的 Applet 类是所有 Applet 程序的根,Applet 类为编程准备好程序框架,编写时只需填入相应的方法实现,无需考虑窗口创建、事件处理等问题,大大简化了编程的复杂性。

1.1.3 Java 语言实现的机制

1. Java 虚拟机

Java 语言的执行模式是半编译和半解释型。Java 编写好的程序首先由编译器转换为标准字节代码,然后由 Java 虚拟机去解释执行。字节代码也是一种二进制文件,但不能直接在操作系统上运行,它可看做虚拟机的机器码。虚拟机把字节代码程序与各操作系统和硬件分开,使 Java 程序独立于平台。虚拟机可以用软件实现,也可用硬件实现,但在无线技术中,都用硬件实现。

Java 程序的下载和执行步骤如图 1-1 所示:

- (1) 程序经编译器得到字节代码;
- (2) 浏览器与服务器连接,要求下载字节码文件;
- (3) 服务器将字节代码文件传给客户机;
- (4) 客户机上的解释器执行字节代码文件;
- (5) 在浏览器上显示并交互。

虚拟机(VM)的执行过程有 3 个特点:

- (1) 多线程;
- (2) 动态连接;
- (3) 异常处理。

2. 垃圾回收机制

在 Java 语言中,所有事物都封装在类中,需要时创建类的实例(对象)来处理,这种动态的实例都存储在内存堆中。为了充分利用资源,Java 有一个系统级的线程,对内存的使用进行跟踪,使程序员从繁忙的内存管理中解放出来。该线程可以在系统空闲时对不用的内存进行回收。

3. 代码安全检测

Java 程序的安全性体现在多个层次上,在编译层,有语法检查;在解释层,有字节码校验器、测试代码段格式和规则检查,访问权限和类型转换合法性检查,操作数堆栈的上溢或下溢,代码参数类型合法性等;在平台层上,通过配置策略,可设定访问资源域,而无需区分本地或远程。

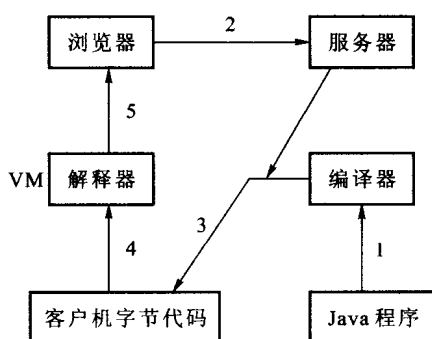


图 1-1 Java 程序的下载和执行

1.2 Java 的体系结构

前面已说到 Java 以 JVM 为基础,实际上,许多 JVM 也是由软件实现的。Java 的体系结构如图 1-2 所示。最下层是移植接口,由适配器和 Java OS 组成,保证 Java 体系结构可以跨平台。虚拟机的上层是 Java 基本类和基本 API,它们都具有可扩展性。最上一层是 Java 应用程序和 Applet 小程序。

Java 的产品主流操作系统平台是 Solaris、Windows 和 Macintosh。目前最新的 JDK 是 j2sdk1.4.2 版,可到 SUN 公司网站下载。本书第 11 章将叙述如何具体下载,全书所有例子都用 j2sdk1.4.2 版编写。

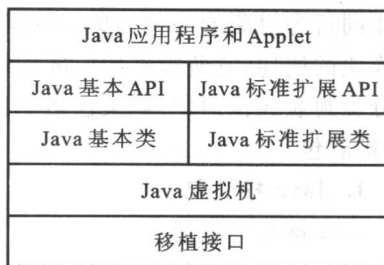


图 1-2 Java 的体系结构

1.2.1 JDK 目录结构

以 j2sdk1.4.2 版为例,解压后得目录结构如图 1-3 所示。

bin	文件夹	2003-8-11 10:45
demo	文件夹	2003-8-11 10:45
include	文件夹	2003-8-11 10:45
jre	文件夹	2003-8-11 10:45
lib	文件夹	2003-8-11 10:45
COPYRIGHT	5 KB 文件	2003-3-24 6:55
LICENSE	11 KB 文件	2003-8-11 10:45
README.html	17 KB Microsoft HTML Doc...	2003-8-11 10:45
README.txt	9 KB TXT 文件	2003-8-11 10:45
src.zip	11,561 KB WinZip File	2003-3-24 6:56

图 1-3 JDK 目录结构

其中:

bin 目录下有编译器、解释器和许多工具(如服务器工具、IDL、package 工具和 jdb 等)。

demo 目录下有各种演示例子。

include 目录下是 Win32 子目录,都是本地方法文件。

jre 目录是 Java 程序运行用环境的根目录,它下面有 bin 子目录,包括平台所用工具和库的可执行文件和 DLL 文件;lib 子目录,包括 java 运行环境的代码库、属性设置和资源文件,默认安装目录,安全管理。

lib 目录下都是库文件。

src.zip 是源码压缩文件。

1.2.2 Java 的 API 结构

Java 是面向对象语言,对象是客观事物的实体,对象与实体是一一对应的,它是很具体的概念。在面向对象语言中,对象是程序的基本单位。Java 语言以类为程序的基本单位,类(Class)是具有某些共同特性实体的集合,是一种抽象的概念,它实质上是一种对象类型的定义,即对具有相同行为对象的一种抽象,说明该类型对象的性质。在 Java 语言中,类是具有某种功能的基本模块的描述,它所提供的标准类库,为编程所需的底层模块提供了常用的方法和接口,并将它们分类封装成包,每个包又包括子包,形成树结构的类层次。类库主要包括核心 java 包、javax 和 org 扩展包。

1. Java 核心包

Java 核心包包括:

(1) java.lang 包:封装所有编程应用的基本类,如 Object、Class、String、System、Integer、Thread 等。而 Object 是所有类的根,它所包含的属性和方法被所有类继承,它的方法有:

- protected Object clone() throws CloneNotSupportedException,该方法生成当前对象的一个拷贝,并将它返回。该对象类型是 Object,它要求使用该方法的类都必须实现 cloneable 接口,否则视为异常。
- public final Class getClass(),该方法返回当前对象的运行对象,该对象类型是 Class 类。
- public final void notify(),该方法唤醒等待对象监视器的多个线程中的一个,是用于多线程技术的方法。
- public final void notifyAll(),该方法唤醒所有等待监视器的线程。
- public final void wait(long timeout) throws InterruptedException,该方法让当前线程放弃对象的锁定。形参 timeout 为超时,如当前线程在等待中被中断是异常。
- public final void wait() throws InterruptedException,该方法与 wait(0)作用相同。
- public int hashCode(),该方法返回一个 hash code 值。
- public boolean equals(Object obj),如当前对象与形参对象相同则返回 true,否则返回 false,所有类型对象的比较都用此方法。
- public String toString(),返回该对象信息的字符串,该字符串与对象相同类型。
- protected void finalize() throws Throwable,该方法把对象从内存中清除,由垃圾收集器自动调用。
- public final void wait(long timeout, int nanos) throws InterruptedException,该方法的形参 nanos 是十亿分之一秒的时间,超时计算更精确。

Class 类是由编译器自动生成对象的一个特殊类,它伴随每个类。这个 Class 对象包含所属类的所有信息,可通过 Class 类的方法访问这些信息。Class 的方法有:

- public static Class forName(String className) throws ClassNotFoundException,该方法返回形参所指类的 Class 对象,形参是一个类名。
- public String getName(),该方法返回 Class 对象的名字。
- public Class getSuperclass(),返回一个数组,其成员是 Class 类的实例。
- public ClassLoader getClassLoader(),对类加载内存。

- `public Class getComponentType()`, 返回数组成员的类型。如对象不是数组, 返回 `null` (空值)。
- `public int getModifiers()`, 返回类/接口的修饰符, 如 `public`, `final` 等。
- `public Class class getDeclaringClass()`, 如当前对象是另一个类的成员, 则返回该类的 `Class` 对象, 否则为空。
- `public Class[] getSuperclass()`, 返回的是一个数组, 数组成员是 `Class` 对象。
- `public Package getPackage()`, 返回当前类所在包。
- `public Class[] getInterfaces()`, 返回接口。
- `public Object[] getSigners()`, 返回信号。
- `public Field[] getFields() throws SecurityException`, 正常返回项, 出错抛出异常。
- `public Method[] getMethods() throws SecurityException`, 正常返回方法结果, 出错抛出异常。
- `public Constructor[] getConstructors() throws SecurityException`, 正常返回结构文件名, 出错抛出异常。

`System` 类是一个特殊类, 它是一个 `final` 类, 所有的方法都用类变量来调用, 即对 `System` 类不可能实例化, 它主要提供了标准输入/输出和系统环境信息的访问、设置。它的属性有:

- `public static final InputStream in`, 标准输入。
- `public static final PrintStream out`, 标准输出。
- `public static final PrintStream err`, 标准错误输出。

有关环境信息的方法有:

- `public static Properties getProperties(argument)`, 返回系统环境信息。
- `public static String setProperty(String key, String value)`, 设置系统变量的值。
- `public static long currentTimeMillis()`, 返回系统时间(毫秒)。
- `public static void exit(int status)`, 强制关闭 JVM 并把状态 `status` 传给操作系统, `status` 非零时, 是非正常退出。
- `public static void gc()`, 运行垃圾收集器。

- `public static void setIn(InputStream in)`, 设置输入信息。
- `public static void setOut(PrintStream out)`, 设置输出信息。
- `public static void setErr(PrintStream err)`, 设置错误信息。

(2) `java. awt` 包: 封装抽象窗口工具包, 提供构建和管理用户图形界面功能。

(3) `java. applet` 包: 为 Applet 提供执行需要的所有类, 主要是访问 Applet 内容的通信类。

(4) `java. io` 包: 提供程序输入/输出文件操作的类。

(5) `java. net` 包: 提供程序执行网络通信应用及 URL 处理的类。

(6) `java. rmi` 包: 提供程序远程方法调用所需的类。

(7) `java. math` 包: 提供程序常用的整数算术以及十进制算术的基本方法类。

(8) `java. util` 包: 提供实用程序类和集合类, 如系统特性定义和使用、日期方法类、集合 `Collection`、`Map`、`List`、`Arrays` 等常用工具类。

(9) `java. sql` 包: 提供访问和处理标准数据源数据的类。

(10) `java. security` 包: 提供网络安全操作类。