



普通高等教育“十五”国家级规划教材

高等职业教育技能型紧缺人才培养培训系列教材



Java 面向对象 程序设计

聂 哲 袁梅冷 杨淑萍



高等教育出版社

普通高等教育“十五”国家级规划教材
高等职业教育技能型紧缺人才培养培训系列教材

Java 面向对象程序设计

聂 哲 袁梅冷 杨淑萍

高等教育出版社

内容提要

本书是普通高等教育“十五”国家级规划教材。

本书根据项目需要,通过问题提出、实例引入、知识讲解、分析提高的递进方式讲解 Java 面向对象程序设计方法,并通过项目拓展的方式,启发学生对相关知识的学习,以进一步掌握面向对象程序设计的思想。本书内容基础性强、图文并茂,注重内容的编排和例题的选择,尽量做到通俗易懂、方便学习。

本书共 7 章,主要包括:打开 Java 之门,Java 语法基础,面向对象编程基础,面向对象编程高级技术,包、异常与输入/输出,图形用户界面,网络编程基础等内容。附录部分还包括 Applet 简介、Java API 文档、javadoc 的使用。

本书适合于高等职业学校、高等专科学校、成人高校、本科院校举办的二级职业技术学院、示范性软件职业技术学院、继续教育学院、民办高校、技能型紧缺人才培养使用,不仅可作为高等院校相关专业的教材,也可供从事 Java 开发、应用人员学习参考使用。

图书在版编目(CIP)数据

Java 面向对象程序设计/聂哲,袁梅冷,杨淑萍.

北京:高等教育出版社,2005.6

ISBN 7-04-016989-4

I. J... II. ①聂...②袁...③杨... 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 042192 号

策划编辑 冯 英 责任编辑 胡 纯 封面设计 王凌波
版式设计 胡志萍 责任校对 张 颖 责任印制 朱学忠

出版发行	高等教育出版社	购书热线	010-58581118
社 址	北京市西城区德外大街 4 号	免费咨询	800-810-0598
邮政编码	100011	网 址	http://www.hep.edu.cn
总 机	010-58581000		http://www.hep.com.cn
经 销	北京蓝色畅想图书发行有限公司	网上订购	http://www.landaco.com
印 刷	北京鑫海金澳胶印有限公司		http://www.landaco.com.cn
开 本	787×1092 1/16	版 次	2005 年 6 第 1 版
印 张	16.25	印 次	2005 年 6 月第 1 次印刷
字 数	390 000	定 价	20.60 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 16989-00

郑重声明

高等教育出版社依法对本书享有专有出版权。任何未经许可的复制、销售行为均违反《中华人民共和国著作权法》，其行为人将承担相应的民事责任和行政责任，构成犯罪的，将被依法追究刑事责任。为了维护市场秩序，保护读者的合法权益，避免读者误用盗版书造成不良后果，我社将配合行政执法部门和司法机关对违法犯罪的单位和个人给予严厉打击。社会各界人士如发现上述侵权行为，希望及时举报，本社将奖励举报有功人员。

反盗版举报电话：(010) 58581897/58581896/58581879

传 真：(010) 82086060

E-mail: dd@hep.com.cn

通信地址：北京市西城区德外大街4号

高等教育出版社打击盗版办公室

邮 编：100011

购书请拨打电话：(010)58581118

出版说明

为加强高职高专教育的教材建设工作,2000年教育部高等教育司颁发了《关于加强高职高专教育教材建设的若干意见》(教高司[2000]19号),提出了“力争经过5年的努力,编写、出版500本左右高职高专教育规划教材”的目标,并将高职高专教育规划教材的建设工作分为两步实施:先用2至3年时间,在继承原有教材建设成果的基础上,充分汲取近年来高职高专院校在探索培养高等技术应用性专门人才和教材建设方面取得的成功经验,解决好高职高专教育教材的有无问题;然后,再用2至3年的时间,在实施《新世纪高职高专教育人才培养模式和教学内容体系改革与建设项目计划》立项研究的基础上,推出一批特色鲜明的高质量的高职高专教育教材。根据这一精神,有关院校和出版社从2000年秋季开始,积极组织编写和出版了一批“教育部高职高专规划教材”。这些高职高专规划教材是依据1999年教育部组织制定的《高职高专教育基础课程教学基本要求》(草案)和《高职高专教育专业人才培养目标及规格》(草案)编写的,随着这些教材的陆续出版,基本上解决了高职高专教材的有无问题,完成了教育部高职高专规划教材建设工作的第一步。

2002年教育部确定了普通高等教育“十五”国家级教材规划选题,将高职高专教育规划教材纳入其中。“十五”国家级规划教材的建设将以“实施精品战略,抓好重点规划”为指导方针,重点抓好公共基础课、专业基础课和专业主干课教材的建设,特别要注意选择一部分原来基础较好的优秀教材进行修订,使其逐步形成精品教材;同时还要扩大教材品种,实现教材系列配套,并处理好教材的统一性与多样化、基本教材与辅助教材、文字教材与软件教材的关系,在此基础上形成特色鲜明、一纲多本、优化配套的高职高专教育教材体系。

普通高等教育“十五”国家级规划教材(高职高专教育)适用于高等职业学校、高等专科学校、成人高校及本科院校举办的二级职业技术学院、继续教育学院和民办高校使用。

教育部高等教育司

2002年11月30日

前 言

Java 语言自 1995 年出现以来,就以迅猛的发展速度成为计算机主流编程语言,在商业、通信和工业等很多领域都开始大量应用。

Java 语言作为一种完全的面向对象程序设计的语言,很好地支持和采用了面向对象设计的思想。因此,采用 Java 语言作为面向对象教学语言,可以浅显、快速、透彻地培养读者面向对象程序设计的思想。

在编写风格上,本书不追求对 Java 技术的全面介绍,而是根据项目需要,通过问题提出、实例引入、知识讲解、分析提高的递进方式适当讲解,并通过项目拓展的方式,启发读者对相关知识的学习,符合人们对事物的认识规律。

在内容编排上,本书始终通过经典的实际项目来讲述面向对象和 Java 编程的基本思想、方法和技术,每章节都配备有与单元知识相适应的练习项目,使读者通过项目的开发,巩固相关的内容,并掌握实际编程技能。教材内容以最新 Java 技术版本为背景,紧跟 Java 技术发展潮流。本书层次分明,对基本知识、技术论述详尽,又注重深度与广度的拓展,可按需进行取舍,适应个性化教育的需要。

本书的特色体现在:

(1) 选择经典实例,切合读者认知

本书尽可能选用来自读者所接触到的现实世界的实例,以提高读者的学习兴趣。同时,对一些综合性比较强的章节,所选用的实例均来自于实际应用,读者在掌握的基础上可以转化为自己的实际编程经验。

(2) 面向对象程序设计思想贯穿全书

由于本教材的落脚点是“面向对象程序设计”,因此在所有实例应用中,均不断强调面向对象程序设计的思想,从“为什么要这样去设计,怎样去设计及编程”的角度去训练读者的面向对象思想。希望通过对本教材的学习,能真正帮助读者树立起面向对象程序设计的思想。

(3) 根据 Java 技术发展,内容取舍符合实际应用需求

在传统的教材中,Applet 及 AWT 在整个教材中占了相当的比重,而随着 Java 技术的不断发展,这些内容在逐渐被淡化。因此,本教材根据 Java 的企业应用情况,主要介绍 Application 和 Swing 的应用,真正实现“学以致用”的目标。

(4) 重点突出,章节编排注重对读者面向对象思想的培养

由于面向对象程序设计思想的建立是一个逐步实现的过程,从人的认知规律出发,教材在编排上遵循“从简单到复杂”、“从概括到具体”的原则,较好地将对面向对象程序设计的思想通过层层拓展的方式,展现给读者。

作为“面向对象程序设计”课程的教材,建议开设 72 学时,着重培养读者面向对象程序设计的思想与方法。其中,第一~二章建议学时为 12 学时,主要通过比较教学方式,让读者掌握 Java 的基本语法;第三~四章建议学时为 24 学时,主要让读者掌握面向对象的概念、高级特性以及如

何用 Java 语言来实现类;第五章建议学时为 12 学时,主要让读者掌握异常处理、输入/输出流的使用;第六章建议学时为 14 学时,主要让读者掌握面向对象思想的具体应用及图形用户界面设计。第七章建议学时为 12 学时,主要让读者掌握多线程的概念及网络编程的基本应用,为后续课程的学习打下基础。

在本课程的基础上,可以开设 JBuilder 网络编程技术、JSP 动态 Web 技术、中间件技术、移动应用开发技术等后续课程,以进一步训练和强化读者对 Java 技术的掌握,真正能利用 Java 开发具体的商业应用。

参与本书编写的教师均有 4 年以上的 Java 应用开发以及 Java 课程的高职教学经验。其中,第一、二章由杨淑萍执笔,第三~五章由袁梅冷执笔,第六、七章以及附录由聂哲执笔。全书由聂哲负责统稿。肖正兴、范新灿两位老师通读了全书内容,并提出了非常宝贵的修改意见,在此表示衷心感谢。

本书所有实例均在 JDK1.4 环境下调试通过,需要实例电子档的读者,请与编者联系:niezhe@oa.szpt.net。

编 者

于深圳职业技术学院

2005 年 5 月

目 录

第一章 打开 Java 之门	1	2.4.1 类型转换	25
1.1 Java 语言简介	1	2.4.2 运算符的优先级	26
1.1.1 Java 语言产生的背景	1	2.5 控制语句	27
1.1.2 Java 的工作原理	2	2.5.1 分支语句	27
1.1.3 Java 语言的特点	3	2.5.2 循环语句	33
1.1.4 Java 与 C 及 C++ 的比较	4	2.5.3 跳转语句	37
1.1.5 Java 语言的应用前景	6	2.6 总结与提高	39
1.1.6 Java 语言新进展	6	习题	42
1.2 建立 Java 语言开发与运行环境	7	第三章 面向对象编程基础	43
1.2.1 Java 语言开发平台	7	3.1 面向对象编程的基本思想与方法	43
1.2.2 建立 Java 语言开发与运行环境	8	3.1.1 面向对象方法的基本思想	43
1.3 第一个 Java 程序——Hello World	11	3.1.2 面向过程编程与面向对象编程比较	44
1.4 总结与提高	13	3.2 对象、类与 Java 实现	45
习题	14	3.2.1 对象与类	45
第二章 Java 语法基础	15	3.2.2 Java 类	45
2.1 Java 语言的基本组成	15	3.2.3 对象实例化及对象的使用	47
2.1.1 标识符	15	3.3 封装与抽象	49
2.1.2 关键字	16	3.4 对象的初始化——构造器方法	52
2.1.3 分隔符	16	3.5 static 变量与方法	54
2.2 数据类型	17	3.5.1 static 变量	54
2.2.1 基本数据类型	17	3.5.2 static 方法	55
2.2.2 常量和变量	17	3.6 方法重载与静态多态性	56
2.3 运算符	20	3.7 对象的高级使用	60
2.3.1 算术运算符	21	3.7.1 对象作为类的成员	60
2.3.2 关系运算符	21	3.7.2 对象作为方法的参数和方法的返回值	63
2.3.3 逻辑运算符	22	3.8 数组	64
2.3.4 位运算符	23	3.8.1 一维数组	64
2.3.5 条件运算符	24	3.8.2 多维数组	66
2.3.6 赋值运算符	24	3.8.3 数组的应用	68
2.3.7 字符串连接运算符	25	习题	71
2.4 表达式	25		

第四章 面向对象编程高级技术	73
4.1 类的继承	73
4.1.1 继承的基本概念	73
4.1.2 类继承的 Java 实现	74
4.1.3 Java 的单继承结构和 Object 类	83
4.2 多态性	84
4.2.1 继承结构中类对象之间的类型转换	84
4.2.2 运行时多态	85
4.3 抽象方法与抽象类	87
4.4 接口	88
4.4.1 接口的定义、继承与实现	88
4.4.2 接口与多继承	92
4.4.3 接口与多态	95
习题	96
第五章 包、异常与输入/输出	98
5.1 Java 包	98
5.1.1 包的创建与使用	98
5.1.2 基于包的访问权限	102
5.1.3 Jar 归档文件	103
5.2 Java API 包与常用 API 的使用	104
5.2.1 Java API 包	104
5.2.2 常用 API 的使用	105
5.3 异常	111
5.3.1 Java 异常处理机制与 try-catch-finally 语句	112
5.3.2 throws 子句与 throw 语句的使用	115
5.3.3 Java 异常的分类与 Java 异常类的层次结构	117
5.3.4 自定义异常	119
5.4 输入/输出流与文件	120
5.4.1 输入/输出流概念	120
5.4.2 输入/输出节点流	121
5.4.3 过滤流	125
5.4.4 InputStreamReader 和 OutputStreamWriter 类	129
5.4.5 PrintWriter 类	130

5.4.6 文件的随机读/写	131
5.4.7 文件与目录管理	133
习题	137
第六章 图形用户界面	139
6.1 Java GUI 基础	139
6.2 Java 的布局管理	141
6.2.1 顺序布局	141
6.2.2 边界布局	142
6.2.3 网格布局	143
6.2.4 手工布局	144
6.3 Java 的事件处理机制	146
6.4 Java 的常用组件及事件处理	149
6.4.1 Font 和 Color	149
6.4.2 标签和按钮	149
6.4.3 文本输入框和密码输入框	152
6.4.4 单选按钮和复选框	155
6.4.5 下拉列表框	159
6.4.6 菜单	161
6.4.7 标准对话框	164
6.4.8 用户自定义对话框	167
6.4.9 文件框	169
6.4.10 常用容器	172
6.5 综合应用	173
6.5.1 游戏的主要功能	173
6.5.2 游戏的界面设计	173
6.5.3 游戏的设计思想	173
6.5.4 游戏的实现代码	175
6.6 总结与提高	182
6.6.1 javax.swing 包与 java.awt 包	182
6.6.2 Swing 简介	183
6.6.3 Java 的事件处理	184
习题	185
第七章 网络编程基础	187
7.1 网络编程的基本知识	187
7.1.1 网络协议	187
7.1.2 机器标识	188
7.1.3 服务器和客户	188

7.1.4 端口	188	信息广播系统	203
7.1.5 套接字	189	7.5.1 客户/服务器之间需要 传送的信息内容	203
7.2 基于 TCP 协议的简单 聊天系统	189	7.5.2 客户/服务器协议(信息格式) 的约定	203
7.2.1 Java 的网络编程类	189	7.5.3 信息的分离、存储与显示	204
7.2.2 服务器和客户端的连接过程	190	7.5.4 服务器端功能结构	207
7.2.3 简单聊天系统的实现	190	7.5.5 服务器端程序实现	209
7.3 Java 的多线程机制	196	7.5.6 客户端功能结构	215
7.3.1 线程的创建	197	7.5.7 客户端程序实现	215
7.3.2 线程的生命周期	197	7.6 总结与提高	221
7.3.3 线程的实现	198	7.6.1 多线程的第二种实现途径	221
7.4 基于 TCP 协议的多客户/服务 器信息交互系统	199	7.6.2 线程的同步	221
7.4.1 实现多客户连接的原理	199	7.6.3 基于 UDP 协议的网络连接	227
7.4.2 客户连接线程的实现	200	习题	235
7.4.3 收/发信息线程的实现	200	附录一 Applet 简介	237
7.4.4 多客户/服务器信息交互 系统的实现	201	附录二 Java API 文档	242
7.5 基于 TCP 协议的多客户		附录三 javadoc 的使用	244

第一章 打开 Java 之门

本章要点:

- Java 语言简介
- 建立 Java 语言开发与运行环境
- 第一个 Java 程序——Hello World

1.1 Java 语言简介

Java 语言是 Sun 公司开发的一种网络编程语言,它具有平台无关性、安全性、面向对象、动态性、分布性等卓越的特点,是当今最流行的 Internet 编程语言之一。

1.1.1 Java 语言产生的背景

1991 年,美国 Sun Microsystems 公司的某个研究小组为了能够在消费电子产品上开发应用程序,积极寻找合适的编程语言。消费电子产品种类繁多,包括 PDA、机顶盒、手机等,即使是同一类消费电子产品所采用的处理芯片和操作系统也不相同,也存在着跨平台的问题。当时最流行的编程语言是 C 和 C++ 语言,Sun 公司的研究人员就考虑是否可以采用 C++ 语言来编写消费电子产品的应用程序。但是研究表明,对于消费电子产品而言 C++ 语言过于复杂和庞大,并不适用,安全性也并不令人满意。于是,Bill Joy 先生领导的研究小组就着手设计和开发出一种语言,称之为 Oak。该语言采用了许多 C 语言的语法,提高了安全性,并且是面向对象的语言,但是 Oak 语言在商业上并未获得成功。1994 年下半年,Internet 的迅猛发展和环球信息网 WWW 的快速增长,促进了 Java 语言研制的进展。1995 年,Sun 公司对 Oak 语言的设计进行了改进,取名为 Java 语言,并很快使其成为 Internet 上受欢迎的开发与编程语言。一些著名的计算机公司纷纷购买了 Java 语言的使用权,如 Microsoft、IBM、Netscape、Novell、Apple、DEC、SGI 等。因此,Java 语言被美国的著名杂志 PC Magazine 评为 1995 年十大优秀科技产品。接着出现了大量用 Java 编写的软件产品,受到了工业界的重视和好评,认为“Java 是 20 世纪 80 年代以来计算机界的一件大事”。Microsoft 公司总裁比尔·盖茨在悄悄观察了一段时间后,不无感慨地说:“Java 是长时间以来最卓越的程序设计语言”,并确定微软整个软件开发的战略由 PC 机向以网络为中心的计算机时代转移,而购买 Java 则是他重大战略决策的实施部署。

Java 的诞生对整个计算机产业具有深远的影响,对传统的计算模型提出了新的挑战。Sun Microsystems 公司的总裁 Scott McNearly 认为,Java 为 Internet 和 WWW 开辟了一个崭新的时代。环球信息网 WWW 的创始人 Berners-Lee 说:“计算机事业发展的下一个浪潮就是 Java,并且将会很快发生的”。有人预言:Java 将是网络上的“世界语”,今后所有的用其他语言编写的软件统统都要用 Java 改写。

1.1.2 Java 的工作原理

1. Java 虚拟机

Java 虚拟机(JVM - Java Virtual Machine)是软件模拟的计算机,可以在任何处理器上(无论是在计算机中还是在其他电子设备中)安全并且兼容地执行保存在.class 文件中的字节码。Java 虚拟机的“机器码”保存在.class 文件中,称之为字节码文件。

Java 程序的跨平台主要是指字节码文件可以在任何具有 Java 虚拟机的计算机或者电子设备上运行,Java 虚拟机的建立需要针对不同的软硬件平台做专门的实现,既要考虑处理器的型号,也要考虑操作系统的 PPC 等嵌入式处理芯片。目前针对 SPARC 结构、X86 结构和 MIPS,在 Unix、Linux、Windows 和部分实时操作系统上都实现了相应的 Java 虚拟机。

Java 源程序需要通过编译器编译成为.class 文件(字节码文件),然后由 Java 虚拟机中的 Java 解释器负责将字节码文件解释成为特定的机器码进行运行。Java 程序的编译和运行过程见图 1-1。

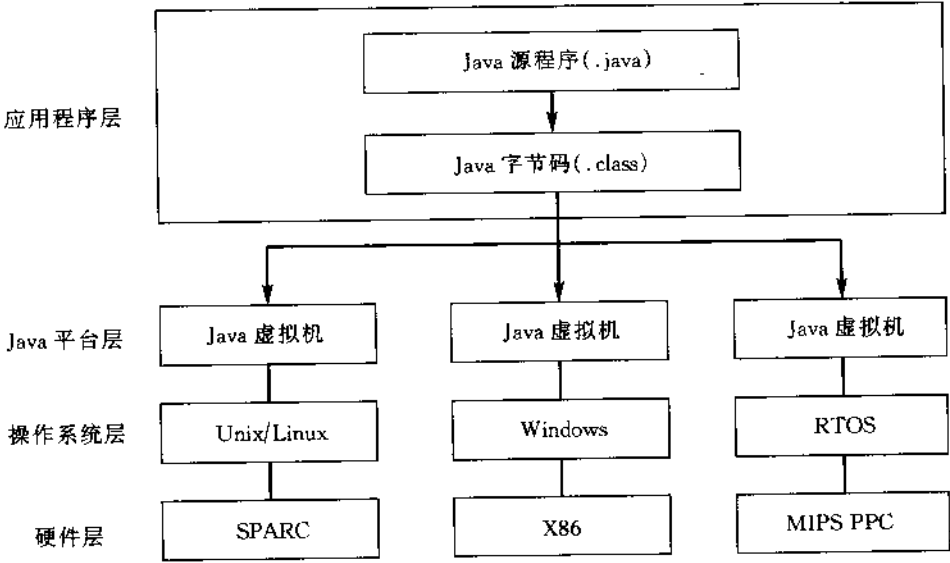


图 1-1 Java 程序的编译和执行过程

2. 无用内存自动回收机制

在程序的执行过程中,部分内存在使用过后就处于废弃状态,如果不及时进行无用内存的回收,就可能出现内存不够的情况,进而导致系统崩溃。在 C++ 语言中是由程序员进行内存回收的,程序员需要在编写程序的时候把不再使用的对象内存释放掉。但是,这种人为的管理内存释放的方法却往往由于程序员的疏忽而致使内存无法回收,同时也增加了程序员的工作量。而在 Java 运行环境中,始终存在着一个系统级的线程,专门跟踪内存的使用情况,定期检测出不再使用的内存,并进行自动回收,从而避免了内存不足的情况,也减轻了程序员的工作量。

3. 代码安全性检查机制

安全和方便总是相对矛盾的。Java 编程语言的出现使得客户端可以方便地从网络上下载 Java 程序到本机上运行。但是,如何保证该 Java 程序不携带病毒或者不怀有其他险恶目的呢?如果 Java 语言不能保证执行的安全性,那么它就不可能发展到今天。虽然有时候少数程序员会抱怨说 Applet 连文件系统也不能访问,但是正是各种安全措施的实行才确保了 Java 语言的生存与发展。

字节码的执行需要经过 3 个步骤:首先由类装载器(Class Loader)负责把类文件(.class 文件)加载到 Java 虚拟机中,在此过程需要检验该类文件是否符合类文件规范;其次字节码校验器(Bytecode Berifier)检查该类文件的代码中是否存在着某些非法操作,例如 Applet 程序中写本机文件系统的操作;如果字节码校验器检验通过,由 Java 解释器负责把该类文件解释成为机器码进行执行。Java 虚拟机采用的是“沙箱”运行模式,即把 Java 程序的代码和数据都限制在一定内存空间里执行,不允许程序访问该内存空间外的内存,如果是 Applet 程序,还禁止访问客户端机器的文件系统。

1.1.3 Java 语言的特点

Java 是一门迅速发展的网络编程语言,是一种全新的计算概念。它具有下面一些特点:简单性、面向对象、分布式、健壮性、安全性、平台无关性、可移植性、解释执行、高性能、多线程、动态性,以及丰富的类库。

1. 简单性

Java 语言的简单性首先体现在精简的系统上,力图用最小的系统实现足够多的功能。对硬件的要求不高,在小型的计算机上便可以良好地运行。而且 Java 采用的是相对简单的面向对象技术,与 C++ 相比,Java 去掉了运算符重载、多继承的复杂概念,而采用了单一继承、类强制转换、多线程、引用(非指针)等方式。无用内存自动回收机制也使得程序员不必费心管理内存,使程序设计更加简单,同时大大减少了出错的可能。Java 语言采用了 C 语言中的大部分语法,熟悉 C 语言的程序员会发现 Java 语言在语法上与 C 语言极其相似。

2. 面向对象

Java 语言的设计集中于对象及其接口,它提供了简单的类机制以及动态的接口模型,所有的 Java 程序和 Applet 程序均是对象。对象封装了它的成员属性及其相应的方法,实现了模块化和信息隐藏;而类则提供了一类对象的原型,并且通过继承机制,使得子类可以使用父类提供的方法,实现了代码的复用。

3. 分布式

Java 作为一种面向网络的语言,通过它提供的类库可以处理 TCP/IP 协议,用户可通过 URL 地址在网络上很方便地访问其他对象。

4. 健壮性

Java 提供的异常处理机制可以在编译运行程序时,对可能出现的问题进行检查和捕获,以避免错误的产生。自动垃圾回收机制可以进行内存管理,防止程序员在管理内存时产生错误,极大地提高了程序的健壮性。

5. 安全性

用于网络、分布式环境下的 Java 程序必须要防止病毒的人侵。Java 不支持指针,一切对内存的访问都必须通过对象的实例变量来实现,这样就防止黑客使用“特洛伊木马”等欺骗手段访问对象的私有成员,同时也避免了指针操作中容易产生的错误。

6. 平台无关性

Java 编译器编译生成了与体系结构无关的字节码文件,任何种类的计算机,只要在其处理器和操作系统上有 Java 运行环境,字节码文件就可以在该计算机上运行,从而同一版本的 Java 应用程序可以运行在所有的平台上。

7. 可移植性

体系结构的中立也使得 Java 系统具有可移植性。Java 运行时系统可以移植到不同的处理器和操作系统上,Java 的编译器是由 Java 语言实现的,解释器是由 Java 语言 and 标准 C 语言实现的,因此可以较为方便的进行移植工作。

8. 解释执行

Java 解释器直接对 Java 字节码进行解释执行。字节码本身携带了许多编译时的信息,使连接过程更加简单。

9. 高性能

虽然 Java 是解释执行的,但它仍然具有非常高的性能,在一些特定的 CPU 上,Java 字节码可以快速地转换成为机器码进行执行。而且 Java 字节码格式的设计就是针对机器码的转换,实际转换时相当简便,自动的寄存器分配与编译器对字节码的一些优化可使之生成高质量的代码。随着 Java 虚拟机的改进和“即时编译”(just in time)技术的出现,使得 Java 的执行速度有了更大的提高。

10. 多线程

多线程机制使应用程序能够并行执行,而且同步机制保证了对共享数据的正确操作。通过使用多线程,程序员可以分别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,这些特点非常适合于网上实时交互行为的实现。

11. 动态性

Java 的设计适合于一个不断发展的环境。在类库中可以自由地加入新方法和实例变量而不会影响用户程序的执行。并且 Java 通过接口来支持多继承,使之比严格地类继承具有更灵活的方式和扩展性。

12. 丰富的类库

Java 提供了大量的类以满足网络化、多线程、面向对象系统的需要。

1.1.4 Java 与 C 及 C++ 的比较

如果了解 C 语言和 C++ 语言,可以参考下列 Java 与 C/C++ 语言的比较;如果不了解 C 语言和 C++ 语言,可以忽略本部分知识。

1. 全局变量

Java 程序不能定义程序的全局变量,而类中的公共、静态变量就相当于这个类的全局变量,这样就使全局变量封装在类中,保证了安全性。而在 C/C++ 语言中,不加封装的全局变量往往会由于使用不当而造成系统的崩溃。它们的条件转移指令也不相同。C/C++ 语言中用 goto 语

句实现无条件跳转。而 Java 语言没有 goto 语言,通过例外处理语句 try、catch、finally 来取代之,提高了程序的可读性,也增强了程序的健壮性。

2. 指针

指针是 C/C++ 语言中最灵活,但也是最容易出错的数据类型。用指针进行内存操作往往造成不可预知的错误,而且通过指针对内存地址进行显示类型转换后,可以访问类的私有成员,破坏了其安全性。在 Java 中,程序员不能进行任何指针操作,同时 Java 中的数组是通过类来实现的,很好地解决了数组越界这一 C/C++ 语言中不做检查的缺点。

3. 内存管理

在 C 语言中,程序员使用库函数 malloc() 和 free() 来分配和释放内存,C++ 语言中则是运算符 new 和 delete。再次释放已经释放的内存块或者释放未被分配的内存块,会造成系统的崩溃,而忘记释放不再使用的内存块也会逐渐耗尽系统资源。在 Java 中,所有的数据结构都是对象,通过运算符 new 分配内存并得到对象的使用权。无用内存回收机制保证了系统资源的完整,避免了内存管理不周而引起的系统崩溃。

4. 数据类型的一致性

在 C/C++ 语言中,不同的平台上,编译器对简单的数据类型如 int、float 等分别分配不同的字节数。例如:int 在 IBM PC 上为 16 位,在 VAX-11 上就为 32 位,导致了代码数据的不可移植。在 Java 中,对数据类型的位数分配总是固定的,不管是在哪个计算机平台上,因此就保证了 Java 数据的平台无关性和可移植性。

5. 类型转换

在 C/C++ 语言中,可以通过指针进行任意的类型转换,不安全因素大大增加。而在 Java 语言中系统要对对象的处理进行严格的相容性检查,防止不安全的转换。

6. 头文件

在 C/C++ 语言中使用头文件声明类的原型和全局变量及库函数等,在大的系统中,维护这些头文件是非常困难的。Java 不支持头文件,类成员的类型和访问权限都封装在一个类中,运行时系统对访问进行控制,防止非法的访问,同时,Java 中用 import 语句与其他类进行通信,以便访问其他类的对象。

7. 结构和联合

C/C++ 语言中用结构和联合来表示一定的数据结构,但是由于其成员均为公有的,安全性上存在问题。Java 不支持结构和联合,通过类把数据结构及对该数据的操作都封装在类里面。

8. 预处理

C/C++ 语言中有宏定义,而用宏定义实现的代码往往影响程序的可读性,而 Java 不支持宏定义。

为易于实现跨平台性,Java 设计成为解释执行,字节码本身包含了许多编译时生成的信息,使连接过程更加简单。而多线程使应用程序可以同时进行不同的操作,处理不同的事件。在多线程机制中,不同的线程处理不同的任务,互不干涉,不会由于某一任务处于等待状态而影响了其他任务的执行,这样就可以容易地实现网络上的实时交互操作。Java 在执行过程中可以动态地加载各种类库,这一特点使之非常适合于网络运行,同时也非常有利于软件的开发,即使是更新类库也不必重新编译使用这一类库的应用程序。

1.1.5 Java 语言的应用前景

Java 语言有着广泛的应用前景,迄今为止,Java 语言已作为一门综合技术在多个领域得到发展和应用,包括:

- ① 所有面向对象的应用开发,包括面向对象的事件描述、处理、综合等。
- ② 计算过程的可视化、可操作化的软件的开发。
- ③ 动态画面的设计,包括图形图像的调用。
- ④ 交互操作的设计(选择交互、定向交互、控制流程等)。
- ⑤ Internet 的系统管理功能模块的设计,包括 Web 页面的动态设计、管理和交互操作设计等。
- ⑥ Intranet(企业内部网)上的软件开发(直接面向企业内部用户的软件)。
- ⑦ 与各类数据库连接查询的 SQL 语句的实现。
- ⑧ 可重用的软件组件的开发。
- ⑨ 嵌入式应用与开发。将 Java 运行环境固化到硬件芯片中,实现 Java 程序在小型电子产品上的自由运行,提供增强功能。
- ⑩ 设备互联。不需要找驱动程序、没有 OS 问题、没有电缆和连接器问题,即只需在网络上插入能使用 Jini 技术的设备,就可以工作。
- ⑪ 部署企业应用服务器。
- ⑫ 中间件产品的开发与应用。

1.1.6 Java 语言新进展

Java 语言发展很快,除了前面提到的之外,在此再介绍如下几个方面:

1. Just-in-Time 编译器

它将 Java 字节码编译成与本机结构相适应的代码,以提高执行速度。

2. Joe

Joe 是 Java 客户/服务器模式的扩展。它是第一个用 Java 语言编写的商用对象请求代理软件(ORB),遵循对象管理组(OMG)的“通用对象请求代理体系结构”(CORBA)。其测试版可由 <http://www.sun.com/sunsoft/neo> 免费下载。

3. 类库的发展

SUN 公司将 Java 类库划分为核心(CORE)和扩展(Standard Extension)两类,前者可以保证适用于所有 Java 平台。SUN 公司与许多第三方软件供应商都在开发类库。扩展类库如果发展较完善,就可以被划定为核心类库,甚至打包到 JSDK(Java 开发工具包)之中。目前正在发展和测试的类库很多,例如,Java Enterprise API 为 Intranet 的开发而设计;Java Commerce API 用于开发商务应用;Java Management API 用于解决网络管理问题;Java Media API 包含对二、三维图形、图像和语音、动画的处理;Java Embedded API 用于嵌入式设备(如电话)的软件开发等。

4. 开发工具

除了 SUN 公司的 JSDK 和 Java WorkShop 外,还有许多公司研制出了一些 Java 的开发环境,

如 Symantec Café、Visual Café、Visual J++、Borland 公司的 JBuilder 等,甚至有的数据库厂商也推出了自己的 Java 开发环境,如 Informix 的 Jwork 和 Sybase 的 Power J。其中 JSDK 反映了 Java 的最新进展,虽然它的界面不如可视化工具友好,但却是各种其他开发环境的基础。

1.2 建立 Java 语言开发与运行环境

1.2.1 Java 语言开发平台

Java 不仅是编程语言,还是一个开发平台和运行平台。Java 提供了一整套开发工具,如编译、解释、文档生成、打包工具等供程序员开发 Java 程序,其 JRE (Java runtime environment) 提供 Java 应用程序的运行平台。目前,Sun 公司把 Java 平台划分成 J2EE、J2ME、J2SE 3 类平台,针对不同的市场目标和设备进行定位。

J2EE (Java2 enterprise edition),主要目的是为企业计算提供一个应用服务器的运行和开发平台。J2EE 本身是一个开放的标准,任何软件厂商都可以推出自己的符合 J2EE 标准的产品,使用户可以有多种选择。IBM、Oracle、BEA、HP 等 29 家公司已经推出了自己的产品,其中尤以 BEA 公司的 weblogic 产品和 IBM 公司的 websphere 最为著名。J2EE 将逐步发展成为可以与微软的 .net 战略相对抗的网络计算平台。

J2ME (Java2 micro edition),主要是面向消费电子产品,为消费电子产品提供一个 Java 的开发与运行平台,使得 Java 程序能够在智能卡、手机、PDA、电视机顶盒等产品上运行。

J2SE (Java2 standard edition),主要目的是为台式机和工作站提供一个开发和运行的平台。

本书在介绍 Java 的过程中,使用 J2SE 平台。上述 3 个 Java 平台的关系如图 1-2 所示。

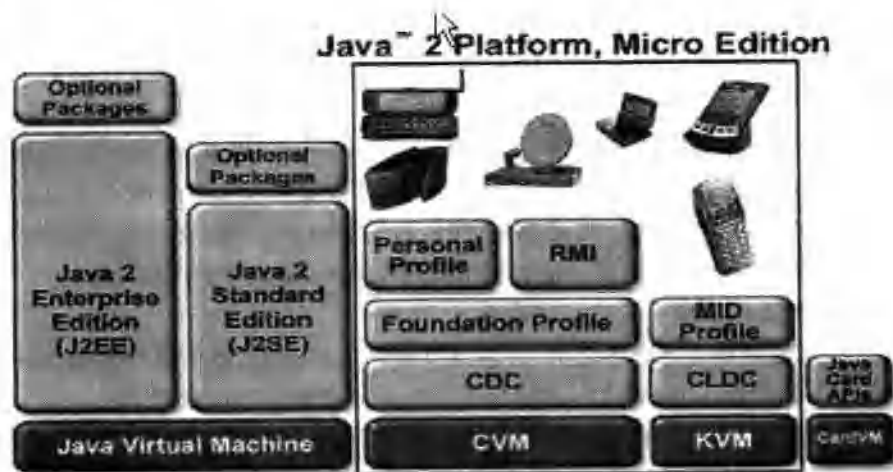


图 1-2 J2SE、J2ME、J2EE 平台