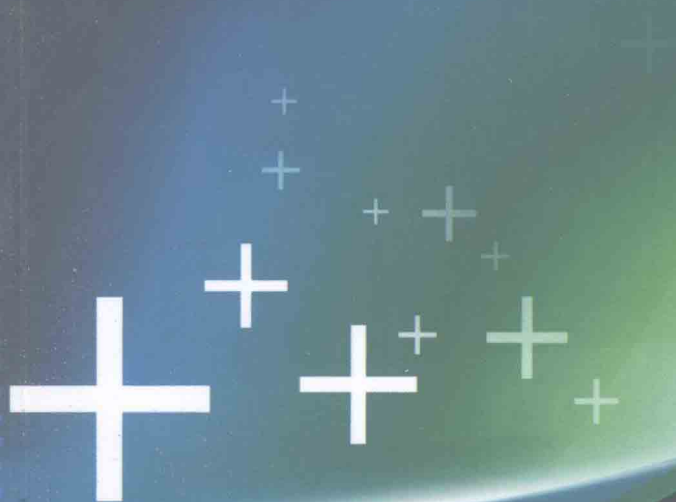




毕树军 主 编
李 晶 副主编
关 心

Java

程序设计实用教程



黑龙江大学出版社
HEILONGJIANG UNIVERSITY PRESS



Java

程序设计实用教程

毕树军 主 编
关 心 李 晶 副主编



黑龙江大学出版社
HEILONGJIANG UNIVERSITY PRESS

图书在版编目(CIP)数据

Java 程序设计实用教程/毕树军主编. —哈尔滨:黑龙江大学出版社,2008. 10

ISBN 978 - 7 - 81129 - 098 - 1

I. J… II. 毕… III. JAVA 语言 - 程序设计 - 教材
IV. TP312

中国版本图书馆 CIP 数据核字(2008)第 157431 号

责任编辑 张怀宇
封面设计 乐然纸尚

Java 程序设计实用教程

JAVA CHENGXUSHEJI SHIYONGJIAOCHENG

毕树军 主 编

关 心 李 晶 副主编

出版发行 黑龙江大学出版社

地 址 哈尔滨市南岗区学府路 74 号 邮编 150080

电 话 0451 - 86608666

经 销 新华书店

印 刷 哈尔滨市石桥印务有限公司

版 次 2008 年 10 月 第 1 版

印 次 2008 年 10 月 第 1 次印刷

开 本 787 × 1092 毫米 1/16

印 张 18.25

字 数 452 千

书 号 ISBN 978 - 7 - 81129 - 098 - 1/T · 17

定 价 32.60 元

凡购买黑龙江大学出版社图书,如有质量问题请与本社发行部联系调换

版权所有 侵权必究

前 言

Java 是一种很优秀的编程语言,具有面向对象、与平台无关、安全、稳定和多线程等特点,是目前软件设计中功能极为强大的编程语言。Java 语言不仅可以用来开发大型的应用程序,而且特别适合于在 Internet 上应用开发。Java 已成为网络时代最重要的编程语言之一。

本书全面地讲解了 Java 的基础内容和编程方法,在内容的深度和广度方面都给予了仔细考虑,在类、对象、继承、接口等重要的基础知识上侧重深度,而在实用类、集合类的讲解上侧重广度。通过对本书的学习,读者可以掌握 Java 面向对象编程的思想和 Java 在网络编程中的一些重要技术。

全书共分 13 章。第 1 章主要介绍了 Java 产生的背景、特点和 Java 开发平台,读者可以了解到 Java 是怎样做到“一次写成,处处运行”的。第 2 章主要介绍了 Java 面向结构化程序设计部分,包括 Java 的基本数据类型、运算符和控制语句。第 3 章是本书的重点内容之一,讲述了类,对象,面向对象的封装性、继承性、多态性,接口,内部类,匿名类以及 Java 语言新增的泛型等重要知识。第 4 章讲述了 Java 的引用数据类型——数组及最常用的字符串在 Java 中如何处理,详细分析了 String 与 StringBuffer 类。第 5 章和第 6 章讲述了常用的实用类,包括 Object 类、包装类、处理日期、数学计算及 Java 中的集合等实用类。第 7 章讲述了如何在 Java 中捕获并处理异常,及在面向对象编程中处理异常与结构化程序设计的不同。第 8 章讲解了 Java 中的输入/输出流技术,特别介绍了怎样使用输入/输出流来克隆对象以及 Java 的文件锁技术。第 9 章是基于 Swing 的 GUI 设计,讲解了容器、布局管理器、常用的组件,对于比较复杂的组件都给出了很实用的例子。第 10 章讲述了多线程技术,这也是很难讲解的一部分内容,在这一章我们通过许多有启发的例子来帮助读者理解多线程编程。第 11 章讲解了 Java 在网络编程中的一些重要技术,涉及到 URL,Socket,InetAddress,DatagramPacket 等重要网络概念。第 12 章主要讲解了 Java 操作数据库的技术,包括如何搭建数据库环境,如何使用 JDBC 查询、更新、插入、删除数据以及预编译技术。第 13 章主要讲解了 Java Applet 的运行原理以及在网络中的角色。

本书注重可读性和实用性,配备了大量的例题,例题全部在 JDK1.5 环境下编译通过。许多例题都经过精心的考虑,既能帮助读者理解知识,又具有启发性。本书通俗易懂,便于自学,针对较难理解的问题,例子都是从简单到复杂,内容逐步深入,便于读者掌握 Java 编程的技巧。本书适合作为大专院校、普通高等学校(本科)Java 语言程序设计的教材,也可以作为自学用书。

编 者
2008 年 8 月

目 录

第1章 Java语言概述	1	3.2.5 方法重载	34
1.1 Java的诞生	1	3.2.6 构造方法	35
1.2 Java的特点	1	3.2.7 类方法和实例方法	35
1.3 Java与C,C++和C#的关系	2	3.2.8 值得注意的问题	35
1.4 Java运行平台	3	3.3 对象	37
1.4.1 三种平台简介	3	3.3.1 创建对象	37
1.4.2 安装Java SE平台	4	3.3.2 使用对象	39
1.5 Java程序的开发过程	5	3.3.3 对象的引用和实体	41
1.6 简单Java应用程序开发示例	6	3.3.4 参数传值	42
1.7 Java程序工作原理	8	3.4 修饰词static	45
小结	9	3.4.1 实例变量和类变量的 区别	45
习题	9	3.4.2 实例方法和类方法的 区别	46
第2章 结构化程序设计	10	3.5 包	48
2.1 标识符和关键字	10	3.5.1 package语句	48
2.2 基本数据类型	10	3.5.2 import语句	48
2.3 基本数据类型的转换	13	3.6 封装性	50
2.4 运算符和表达式	14	3.6.1 私有变量和私有方法	50
2.5 语句概述	21	3.6.2 共有变量和共有方法	51
2.6 控制语句	21	3.6.3 友好变量和友好方法	52
2.7 循环语句	24	3.6.4 受保护的成员变量和 方法	52
2.8 break和continue语句	26	3.6.5 public类与友好类	53
小结	27	3.7 继承性	53
习题	27	3.7.1 创建子类	53
第3章 面向对象程序设计	28	3.7.2 子类的继承性	53
3.1 编程语言的几个发展阶段	28	3.7.3 成员变量的隐藏和方法的 重写	55
3.1.1 机器语言	28	3.7.4 对象的上转型对象	56
3.1.2 过程语言	28	3.8 多态性	58
3.1.3 面向对象编程	29	3.8.1 静态多态性	58
3.2 类	30	3.8.2 动态多态性	59
3.2.1 类声明	30		
3.2.2 类体	31		
3.2.3 成员变量和局部变量	31		
3.2.4 方法	33		

3.9 修饰词 final 和 abstract	60	5.2.2 包装类的常用方法	101
3.9.1 修饰词 final	60	5.2.3 包装类的特点	102
3.9.2 修饰词 abstract	60	5.3 Math 类	102
3.10 this 关键字	61	5.4 Random 类	103
3.11 super 关键字	63	5.5 处理日期的类	104
3.12 接口	64	5.5.1 Date 类	104
3.12.1 接口的声明与使用	65	5.5.2 DateFormat 类	104
3.12.2 接口回调	67	5.5.3 Calendar 类	105
3.12.3 接口做参数	69	5.6 BigDecimal 类	107
3.12.4 抽象类与接口的比较	70	小结	108
3.13 内部类	70	习题	108
3.14 匿名类	71	第6章 Java 集合	109
3.14.1 和类有关的匿名类	71	6.1 Collection 和 Iterator 接口	110
3.14.2 和接口有关的匿名类	73	6.2 Set(集)	112
3.15 泛型类	74	6.2.1 Set 的一般用法	112
3.16 Class 类	75	6.2.2 HashSet 类	113
3.17 jar 文件	76	6.2.3 TreeSet 类	113
小结	77	6.3 List(列表)	114
习题	77	6.3.1 访问列表的元素	114
第4章 数组、字符串	79	6.3.2 为列表排序	114
4.1 数组	79	6.3.3 ListIterator 接口	115
4.1.1 一维数组	79	6.3.4 获得固定长度的 List 对象	116
4.1.2 多维数组	82	6.3.5 比较 Java 数组和各种 List 的性能	116
4.2 String 类和 StringBuffer 类	84	6.4 Map(映射)	116
4.2.1 String 类	84	6.5 HashSet 和 HashMap 的 负载因子	118
4.2.2 String 类常用方法	85	6.6 集合的编译时类型检查	119
4.2.3 “hello”与 new String (“hello”)的区别	89	6.7 用 for 循环遍历集合	120
4.2.4 字符串与基本数据的 相互转化	91	6.8 集合实用类	121
4.2.5 StringBuffer 类	92	6.9 历史集合类	123
4.2.6 String 类与 StringBuffer 类的 区别	93	小结	123
4.2.7 正则表达式	94	习题	124
小结	96	第7章 异常处理	125
习题	97	7.1 异常及其种类	125
第5章 常用实用类	99	7.2 异常产生	126
5.1 Object 类	99	7.3 异常处理	127
5.2 包装类	100	7.3.1 捕捉异常	127
5.2.1 包装类的构造方法	100	7.3.2 转移异常	129

7.4 自定义异常类型	131	9.4.5 GridBagLayout 网格包布局 管理器	172
小结	132	9.4.6 BoxLayout 布局管理器	175
习题	132	9.5 事件处理	178
第 8 章 输入与输出流	133	9.5.1 事件处理的软件实现	178
8.1 File 类	133	9.5.2 事件源事件和监听器的类 层次和关系	185
8.1.1 文件的属性	133	9.6 中间容器	191
8.1.2 目录	134	9.7 按钮(JButton)	193
8.1.3 文件的创建与删除	135	9.8 标签(JLabel)	194
8.1.4 运行可执行文件	136	9.9 文本框(JTextField)、密码框 (JPasswordField)、文本区 (JTextArea)	196
8.2 FileInputStream 类	136	9.10 复选框(JCheckBox)和单选按钮 (JRadioButton)	200
8.2.1 使用文件输入流读取 文件	137	9.11 下拉列表	203
8.2.2 处理 I/O 异常	137	9.12 列表框	204
8.2.3 从输入流中读取字节	137	9.13 页签面板	206
8.2.4 关闭流	138	9.14 菜单	207
8.3 FileOutputStream 类	138	9.15 对话框(JDialog)	212
8.4 FileReader 类和 FileWriter 类 ..	140	9.16 文件对话框	214
8.5 RandomAccessFile 类	142	9.17 消息框	217
8.6 从控制台窗口读入数据	145	小结	219
8.7 数据流	146	习题	219
8.8 数组流	148	第 10 章 多线程程序设计	221
8.9 对象流	150	10.1 多线程基本原理	221
8.10 序列化与对象克隆	151	10.1.1 线程的状态与生命 周期	221
8.11 文件锁 FileLock	155	10.1.2 线程调度与优先级	222
小结	156	10.2 Thread 类和 Runnable 接口	223
习题	156	10.2.1 Thread 类	223
第 9 章 Swing 图形用户界面程序设计	158	10.2.2 Runnable 接口	224
9.1 Swing 组件概述	158	10.3 创建一个线程	224
9.2 组件和容器总体介绍	159	10.3.1 用 Thread 类的子类创建 线程	224
9.3 JFrame 和 JPanel	160	10.3.2 实现 Runnable 接口	226
9.4 布局管理器	162	10.4 创建多重线程	227
9.4.1 FlowLayout 流式布局 管理器	163	10.5 确定线程何时结束	229
9.4.2 BorderLayout 边界布局 管理器	165	10.6 线程的调度	231
9.4.3 GridLayout 网格布局 管理器	167		
9.4.4 CardLayout 卡片布局 管理器	170		

10.7 同步化	233	第 12 章 Java 与数据库操作	264
10.8 使用同步化方法	233	12.1 JDBC 简介	264
10.9 同步化语句	236	12.2 数据库环境建立	265
10.10 使用 notify(),wait()和 notifyAll()的线程通信	237	12.2.1 Microsoft Access 数据库 ...	265
小结	240	12.2.2 JDBC-ODBC 桥接器	265
习题	241	12.2.3 Microsoft SQL Server 2000 数据库	267
第 11 章 Java 中的网络编程	242	12.2.4 使用纯 Java 数据库驱动 程序	267
11.1 网络通信基础	242	12.3 查询操作	267
11.1.1 套接字	242	12.4 可滚动结果集	270
11.1.2 TCP 和 UDP 传输协议 ...	242	12.5 更新、插入与删除操作	272
11.2 基于 URL 的 Java 网络编程	243	12.6 预编译语句	274
11.2.1 使用 URL	243	小结	275
11.2.2 读取 URL 中的资源	244	习题	276
11.2.3 显示 URL 资源中的 HTML 文件	246	第 13 章 小应用程序	277
11.3 InetAddress 类	248	13.1 源程序	277
11.4 基于 TCP 的网络编程	249	13.1.1 生命周期	277
11.4.1 套接字 Socket	249	13.1.2 图形用户界面	280
11.4.2 客户建立连接到服务器的 套接字对象	250	13.2 网页标记	281
11.4.3 ServerSocket 类	250	13.2.1 采用 APPLET 网页 标记	282
11.4.4 把套接字连接放在一个 线程中	253	13.2.2 归档文件	282
11.5 基于 UDP 的网络编程	257	13.2.3 小应用程序参数	282
小结	263	13.3 应用程序与小应用程序	282
习题	263	小结	283
		习题	283

第1章 Java 语言概述

Java 语言是一门很优秀的语言,具有面向对象、与平台无关、安全、稳定和多线程等优良特性,是目前软件设计中功能极为强大的编程语言。Java 语言不仅可以用来开发大型的应用程序,而且特别适合于在 Internet 上应用开发。Java 具备了“一次写成,处处运行”的特点。Java 已成为网络时代最重要的语言之一。

1.1 Java 的诞生

Java 是 1995 年 6 月由 Sun 公司发布的革命性的编程语言。之所以称 Java 为革命性编程语言,是因为传统的软件往往与具体的实现环境有关,一旦环境有所变化就需要对软件进行一番改动,耗时费力,而 Java 编写的软件能在执行码上兼容。这样,只要计算机提供了 Java 解释器,Java 编写的软件就能在其上运行。

Java 语言的出现是源于人们对独立于平台语言的需要,希望这种语言能编写出可以嵌入各种家用电器等设备的芯片,且易于维护的程序。但是,人们发现当时的编程语言,如 C、C++ 等都有一个共同的缺点,那就是只能对特定的 CPU 芯片进行编译。这样,一旦电器设备更换了芯片就不能保证程序正确运行,就可能需要修改程序并针对新的芯片重新进行编译。1990 年 Sun 公司成立了由 James Gosling 领导的开发小组,开始致力于开发一种可移植的、跨平台的语言,该语言能生成正确运行于各种操作系统、各种 CPU 芯片上的代码。他们的精心钻研和努力促成了 Java 语言的诞生。Java 的快速发展得利于 Internet 和 Web 的出现,Internet 上有各种不同的计算机,它们可能使用完全不同的操作系统和 CPU 芯片,但仍希望运行相同的程序,Java 的出现标志着真正的分布式系统的出现。

注:印度尼西亚有一个重要的盛产咖啡的岛屿,中文名叫爪哇,开发人员为这种新的语言起名为 Java,其寓意是为世人端上一杯热咖啡。

1.2 Java 的特点

Java 是目前使用最为广泛的网络编程语言之一。它具有简单、面向对象、稳定、与平台无关、解释型、多线程、动态等特点。

1. 简单

Java 语言简单是指这门语言既易学又好用。但不要将简单误解为这门语言很干瘪。如果你学习过 C++ 语言,你会感觉 Java 很眼熟,因为 Java 中许多基本语句的语法和 C++ 一样,像常用的循环语句、控制语句等和 C++ 几乎一样。但不要误解为 Java 是 C++ 的增强版,Java 和 C++ 是两种完全不同的语言,它们各有各的优势,将会长期并存下去。Java 语言和 C++ 语言已成为软件开发者应当掌握的语言。如果从语言的简单性方面看,Java 要比 C++

简单,C++中许多容易混淆的概念,或者被Java弃之不用了,或者以一种更清楚、更容易理解的方式实现,例如,Java不再有指针的概念。

2. 面向对象

基于对象的编程更符合人的思维模式,使人们更容易编写程序。对此,将在第3章详细地讨论类、对象等概念。

3. 与平台无关

与平台无关是Java语言最大的优势。其他语言编写的程序面临的一个主要问题是:操作系统的变化、处理器升级以及核心系统资源的变化,都可能导致程序出现错误或无法运行。Java的虚拟机成功地解决了这个问题,Java编写的程序可以在任何安装了Java虚拟机(JVM)的计算机上正确地运行,Sun公司实现了自己的目标“一次写成,处处运行”。

4. 解释型

我们知道C、C++等语言都是只能对特定的CPU芯片进行编译,生成机器代码,该代码的运行和特定的CPU有关。Java不像C++,它不针对特定的CPU芯片进行编译,而是把程序编译为称做“字节码”的一个中间代码。字节码是很接近机器码的文件,可以在提供了Java虚拟机JVM的任何系统上被解释执行。

如果把Java的程序比做“汉语”的话,字节码就相当于“世界语”,世界语不和具体的“国家”有关,只要这个“国家”提供了“翻译”,就可以快速地把世界语翻译成本地语言。

5. 多线程

Java的特点之一就是内置对多线程的支持。多线程允许同时完成多个任务。实际上多线程只是使人产生多个任务在同时执行的错觉,因为,目前的计算机的处理器在同一时刻只能执行一个线程,但处理器可以在不同的线程之间快速地切换,由于处理器运行速度非常快,远远超过了人接收信息的速度,所以给人的感觉好像多个任务在同时执行。C++没有内置的多线程机制,因此必须调用操作系统的多线程功能来进行多线程程序的设计。

6. 安全

当你准备从网络上下载一个程序时,你最大的担心是程序中含有恶意的代码,比如试图读取或删除本地机上的一些重要文件,甚至该程序是一个病毒程序等。当你使用支持Java的浏览器时,你可以放心地运行Java的小应用程序(Java Applet),不必担心病毒的感染和恶意的企图。Java小应用程序将限制在Java运行环境中,不允许它访问计算机的其他部分。

7. 动态

Java程序的基本组成单元就是类,有些类是自己编写的,有一些是从类库中引入的,而类又是运行时动态装载的,这就使得Java可以在分布环境中动态地维护程序及类库,而不像C++那样,每当其类库升级之后,相应的程序都必须重新修改、编译。

1.3 Java与C、C++和C#的关系

虽然它们不是一种语言,但它们却也有所联系。广泛地说,C语言可以看做是其他三种语言的源语言,因为其他三种语言几乎都是从C语言发展而来的,无论从数据类型还是控制语句看,其他三种语言都有从C语言得来的迹象。

其中C++一般看做是对C语言的扩展。因为C语言没有面向对象的语法结构,而当时业界又迫切需要面向对象的编程特性,所以贝尔实验室的开发者就为C语言添加了面向对

象的结构。现在C++已经不只是C语言的扩展了,它已经完全可以被看做一种新的编程语言。虽然C语言的特性以及库函数仍然被C++支持,不过C++仍拥有自己独立的类库体系,功能相当强大。

Java是一种可以编写跨平台应用程序的面向对象的程序设计语言,Java编程语言的风格十分接近C++语言。Java继承了C++语言面向对象技术的核心,舍弃了C++语言中的指针(以引用取代)、运算符重载、多重继承(以接口取代)等成分,增加了自动垃圾收集功能,用于回收不再被引用的对象所占据的内存空间。它的跨平台的特性尤其受开发者欢迎,你只需要开发一次就能在所有安装了Java运行平台的系统上运行。

C#是微软开发的一种编程语言,语法类似Java,几乎就是Java的翻版,运行原理和Java也类似,也是在运行时库的支持下运行,不过支持的平台还很有限,而Java几乎被所有平台支持。

虽然Java和C#在某种程度上要归结于它们共同的来源——C和C++,但是,当Gosling和他的同事们坐下来创造Java的时候,他们不仅吸取了C++的优点,更重要的是,他们去掉了C++的一些无用的特性。C#的设计者加入了很多C++的特性,但是C#却没有去掉C++的最糟糕的一些特性。

1.4 Java运行平台

1.4.1 三种平台简介

Java语言要实现“一处写成,处处运行”的目标,就需要相应的运行平台。目前Java运行平台主要分为以下三个版本。

(1)Java SE(Java Platform, Standard Edition):以前称为J2SE,是Java平台标准版的简称,适用于桌面系统,用于开发和部署桌面、服务器以及嵌入设备和实时环境中的Java应用程序。Java SE包括用于开发Java Web服务的类库,同时,Java SE为Java EE提供了基础。

(2)Java EE(Java Platform, Enterprise Edition):以前称为J2EE,是Java平台企业版的简称,适用于创建服务器应用程序和服务,用于开发便于组装、功能强大、可扩展、安全的服务器端Java应用。Java EE建立于Java SE之上,具有Web服务、组件模型以及通信API等特性,这些为面向服务的架构(SOA)以及开发Web2.0应用提供了支持。Java EE基于Java SE建立,此外新增加了企业应用所需的类库。

(3)Java ME(Java Platform, Micro Edition):以前称为J2ME,是Java平台微型版的简称,适用于小型设备和智能卡。它为移动设备(包括消费类产品、嵌入式设备、高级移动设备等)提供了基于Java环境的开发与应用平台。Java ME目前分为两类配置,一类面向小型移动设备,一类面向功能更强大的移动设备,如智能手机和顶盒。

登录Sun公司的网站<http://java.sun.com>,就能看到有关Java SE,Java EE,Java ME的介绍。上述运行平台都包括了相应的Java虚拟机(Java Virtual Machine,JVM),虚拟机负责将字节码文件(包括程序使用的类库中的字节码)加载到内存,然后采用解释方式来执行字节码文件,即根据相应硬件的机器指令解释一句执行一句。

1.4.2 安装 Java SE 平台

学习 Java 必须从 Java SE 开始。Sun 公司提供了 JDK(Java SE Development Kit),即 Java 标准版(Standard Edition)开发工具包,这是 Java 开发和运行的基本平台,换句话说所有用 Java 语言编写的程序的运行都离不开它,而用它可以把 Java 代码编译为字节码文件。可以到 Sun 公司网站 <http://java.sun.com> 免费下载多种操作系统下的 JDK,目前最新版本是 JDK 6(习惯称为 JDK1.6)。本书使用 Windows 平台的 JDK 5(JDK1.5)版本。

注意:不要下载 JRE(Java Runtime Environment,Java 运行时环境),因为 JRE 不包含 Java 编译器。

下载 JDK 安装文件 jdk-1_5_0_16-windows-i586-p.exe,双击执行安装。如果将 JDK 安装到 D:\jdk1.5 目录下,目录结构如图 1-1 所示。

1. 系统环境变量 Path 值的设置

JDK 的安装目录的 bin 文件夹中含有 Java 编译器(javac.exe)、Java 解释器(java.exe)和一些其他的可执行文件。为了能在任何目录中使用编译器和解释器,应在系统环境变量中设置 Path 值。

在 Windows 系统中,右击“我的电脑”,在“我的电脑→属性→高级→环境变量→系统变量”中添加以下系统环境变量:

变量名:Path,变量值:D:\jdk1.5\bin

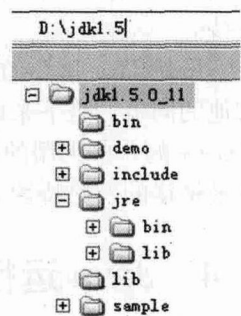


图 1-1 jdk 目录结构

如果曾经设置过环境变量 Path,可双击该变量进行编辑操作,将需要的值加入即可,如图 1-2 所示。也可以在命令窗口(MS-DOS 窗口)中输入命令:

Path = D:\jdk1.5\bin

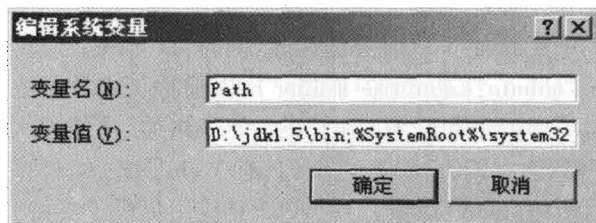


图 1-2 设置系统环境变量 Path

注意:在命令行窗口下的设置只对当前这个窗口有效,并不会影响到其他命令行窗口和程序。

2. 系统环境变量 ClassPath 值的设置

JDK 安装目录的 jre 文件夹中包含着 Java 应用程序运行时所需要的 Java 类库和虚拟机,这些类库被包含在一个 jre\lib 中的压缩文件 rt.jar 中。安装 JDK 一般不需要设置环境变量 ClassPath 的值,但如果计算机安装过一些商业化的 Java 开发产品或带有 Java 技术的一些产品,如 PB, Oracle 等,那么这些产品在安装后,也可能会修改 ClassPath 的值,那么当运行 Java 应用程序时,可能会加载这些产品所带的旧版本的类库,而导致程序要加载的类无法找到,使程序出现运行错误。这时需要重新编辑系统环境变量 ClassPath 的值。

在 Windows 系统中,右击“我的电脑”,在“我的电脑→属性→高级→环境变量→系统变量”中添加以下系统环境变量。

变量名:ClassPath,变量值:D:\jdk1.5\jre\lib\rt.jar;.;

如果曾经设置过环境变量 ClassPath,可双击该变量进行编辑操作,将需要的值加入即可,如图 1-3 所示。也可以在命令窗口(MS-DOS 窗口)中输入命令:

```
set classpath = D:\jdk1.5\jre\lib\rt.jar;.;
```

注:ClassPath 环境变量中的“.”是指可以加载应用程序当前目录及其子目录中的类。在命令行窗口下的设置只对当前这个窗口有效,并不会影响到其他命令行窗口和程序。

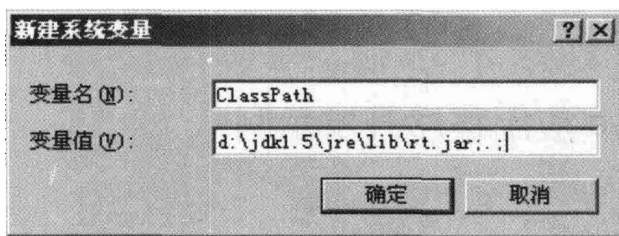


图 1-3 设置系统环境变量 ClassPath

当这些变量设置完毕后,就可以在命令行里面检查是否设置成功。点击 Windows 的开始按钮,选择“运行(R)”,输入 cmd 后按下回车,进入命令行窗口。输入 javac 并按下回车键,如果能看到 javac 命令用法、选项列表,则环境变量已经配置成功;如果发现输出“javac 不是内部或外部命令,也不是可运行的程序或批处理文件”这样的提示信息,则说明环境变量配置失败,请仔细检查设置的步骤。

3. 仅安装 JRE

如果只想运行别人的 Java 程序可以只安装 Java 运行环境 JRE,JRE 由 Java 虚拟机、Java 的核心类,以及一些支持文件组成。可以登陆 Sun 的网站免费下载 Java 的 JRE。

注:建议下载 Sun 公司的 Java 类库文档,例如,jdk-6-doc.zip。

1.5 Java 程序的开发过程

Java 程序的开发过程如图 1-4 所示。

注:字节码文件是与平台无关的二进制文件,执行时由解释器解释成本地机器码,解释一句执行一句。

(1)编写源文件:使用一个文字编辑器,如 EditPlus,UltraEdit 或记事本来编写源文件。不可使用 Word 编辑器,因为它含有不可见字符。将编好的源文件保存起来,源文件的扩展名必须是.java。

(2)编译 Java 源程序:使用 Java 编译器(javac.exe)编译源文件得到字节码文件。

(3)运行 Java 程序:Java 程序分为两类——Java 应用程序和 Java 小应用程序。Java 应用程序必须通过 Java 解释器(java.exe)来解释执行其字节码文件,Java 小应用程序必须通过支持 Java 标准的浏览器来解释执行。

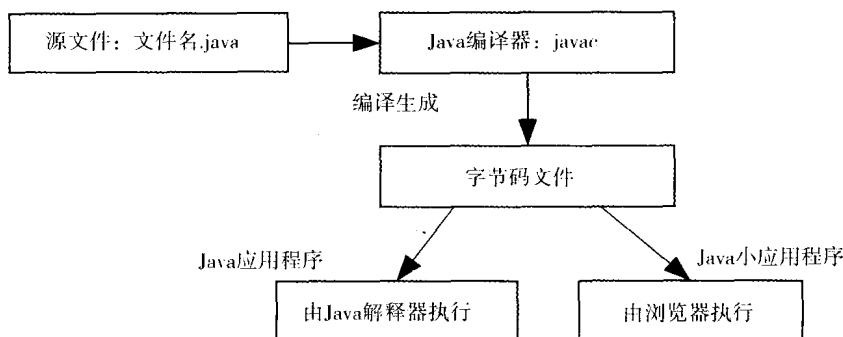


图 1-4 Java 程序的开发过程

1.6 简单 Java 应用程序开发示例

1. 编写源文件

```

public class Hello{

    public static void main( String args[ ] ){

        System. out. println( " 你好,很高兴学习 Java" );

    }

}

```

注:Java 源程序中语句所涉及到的括号及标点符号都是英文状态下输入的括号和标点符号,比如"你好,很高兴学习 Java"中的引号必须是英文状态下的引号,而字符串里面的符号不受汉语状态或英语状态的限制。

(1)一个 Java 源程序是由若干个类组成的。上面的这个 Java 应用程序简单到只有一个类,类的名字是 Hello。

(2)class 是 Java 的关键字,是用来定义类的。public 也是关键字,说明 Hello 是一个 public 类。第一个大括号和最后一个大括号以及它们之间的内容叫做类体。

(3)public static void main(String args[])是类体中的一个方法,之后的两个大括号以及之间的内容叫做方法体。一个 Java 应用程序必须有一个类且只有一个类含有 main 方法,这个类称为应用程序的主类。public,static 和 void 分别是对 main 方法的说明。在一个 Java 应用程序中 main 方法必须被说明为 public static void。

(4)String args[]声明一个字符串类型的数组 args[](注意:String 的第一个字母是大写的),它是 main 方法的参数。main 方法是程序开始执行的位置。

现在将源文件保存到 D:\JavaLesson\中,并命名为 Hello. java。注意不可写成 hello. java,因为 Java 语言是区分大小写的。

源文件的命名规则是这样的:如果源文件中有多类,那么只能有一个类是 public 类;如果有一个类是 public 类,那么源文件的名称必须与这个类的名称完全相同,扩展名是 . java;如果源文件没有 public 类,那么源文件的名称只要和某个类的名称相同,并且扩展名是 . java 就可以了。

2. 编译

创建了 Hello.java 这个源文件后,就要使用 Java 编译器 javac.exe 对其进行编译。

```
D:\JavaLesson\ > javac Hello.java
```

编译完成后生成一个 Hello.class 文件,该文件称为字节码文件。这个字节码文件 Hello.class 将被存放在与源文件相同的目录中。

如果 Java 源程序中包含了多个类,那么用编译器 javac.exe 编译完源文件后将生成多个扩展名为.class 的文件,每个扩展名是.class 的文件中只存放一个类的字节码,其文件名与该类的名字相同。这些字节码文件将被存放在与源文件相同的目录中。

如果对源文件进行了修改,那么必须重新编译,再生成新的字节码文件。

3. 运行

Java 应用程序必须通过 Java 虚拟机中的 Java 解释器(java.exe)来解释执行其字节码文件。Java 应用程序总是从主类的 main 方法开始执行。使用如下命令运行 Java 应用程序。

```
D:\JavaLesson\ > java Hello
```

MS-DOS 命令行窗口将显示如下信息:

你好,很高兴学习 Java

注:当 Java 应用程序中有多个类时,Java 命令后的类名必须是包含了 main 方法的那个类的名字,即主类的名字。

再看一个简单的 Java 应用程序。学习如何命名、保存源程序,怎样使用编译器编译源程序,怎样使用解释器运行程序。

(1) 源文件

```
class Car {  
    float carCc; // 汽车车长  
    String carYs, carPp; // 汽车颜色、汽车品牌  
  
    void run( String s ) {  
        System.out.println( s );  
    }  
}  
  
public class A {  
    public static void main( String args[] ) {  
        Car car;  
        car = new Car();  
        car.carCc = 10f;  
        car.carYs = "蓝色";  
        car.carPp = "上海大众 Polo";  
        //输出: 颜色:蓝色 品牌:上海大众 Polo  
        System.out.println("颜色:" + car.carYs + " 品牌:" + car.carPp);  
        //输出: 10.0  
        System.out.println( car.carCc );  
    }  
}
```

```
//输出: 奔跑
car.run("奔跑");
```

必须把源文件保存起来并命名为 A.java。假设保存 A.java 在 D:\JavaLesson 下。

(2) 编译源文件

```
D:\JavaLesson\>javac A.java
```

如果编译成功,目录下就会有 A.class 和 Car.class 这两个字节码文件。

(3) 执行

```
D:\JavaLesson\>java A
```

Java 命令后的名字必须是含有 main 方法的那个类的名字(注意:运行时,类名 A 不带扩展名.class)。

1.7 Java 程序工作原理

Sun 公司设计 Java 语言的目标是让 Java 程序不必经过修改就可以在各种各样的计算机上运行。为了实现这一目标,Sun 公司提出了 Java 虚拟机(JVM)的机制。其工作流程和原理如图 1-5 所示。

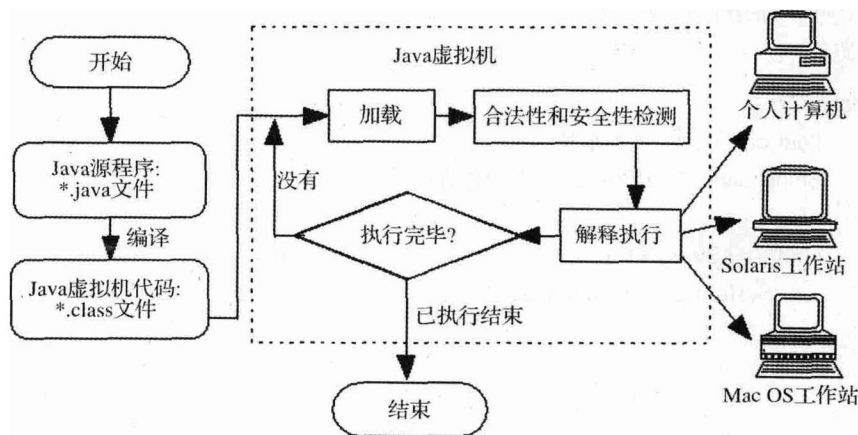


图 1-5 Java 虚拟机的工作原理及其工作流程

Java 虚拟机是编译和运行 Java 程序的各种命令及运行环境的总称。Java 源程序在编译之后生成后缀为“.class”的文件,该文件以字节码(bytecode)的方式进行编码。这种字节码文件是一种伪代码,它包含各种指令,这些指令是与平台无关的指令。Java 虚拟机在字节码文件的基础上解释这些字节码,即将这些字节码转换成本地计算机平台的机器代码,并交给本地计算机执行。

小 结

Java 语言的发展史是简短而曲折的。由于它与网络的良好结合,使得它在刚发布时就引起了轰动。本章讲解了 Java 的起源、基本特点和运行环境,编写 Java 应用程序的过程及工作原理。Java 程序是面向对象的;Java 源代码以文件的形式存储,其扩展名为 .java;Java 程序被编译为 .class 字节码文件,由 Java 虚拟机负责解释执行。

习 题

1. Java 语言的基本特点有什么?
2. 开发与运行 Java 程序需要经过哪些主要步骤和过程?
3. Java 程序是由什么组成的? 一个程序中必须要有 public 类吗? Java 源文件的命名规则是怎样的?
4. Java 程序可以分成哪几种? 分别是什么?