

普通高等院校计算机专业（本科）实用教程系列

JAVA 2

实用教程（第二版）

耿祥义 张跃平 编著
王克宏 主审

123456789123123

23456789123123



清华大学出版社

普通高等院校计算机专业（本科）实用教程系列

JAVA 2 实用教程

（第二版）

耿祥义 张跃平 编著

王克宏 主审



清华大学出版社

北京

内 容 简 介

Java 语言是一种很优秀的语言, 具有面向对象、与平台无关、安全、稳定和多线程等优良特性, 是目前软件设计中极为健壮的编程语言。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 Internet 的应用开发。尤其是 Java Swing 推出之后, 不仅使 Java 的功能更加强大, 而且使 Java 确实具备了“一旦写成处处可用”的特点, Java 已成为网络时代最重要的语言之一。

本书是一本实用教程, 配备了大量的例子, 叙述详细, 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, 逐步深入, 便于读者掌握 Java 编程技巧。全书分为 26 章, 分别介绍了 Java 的数据类型、运算符与表达式、字符串处理、Java Awt、Java Applet、图形处理、图像处理、多线程机制、输入输出流、网络程序、数据库连接、多媒体、Java Swing、Java 与数据结构等内容。

本书不仅可以作为高等院校教材, 也适合自学者及软件开发人员参考使用。

版权所有, 翻印必究。

本书封面贴有清华大学出版社激光防伪标签, 无标签者不得销售。

图书在版编目(CIP)数据

JAVA 2 实用教程/耿祥义, 张跃平编著. 第二版. —北京: 清华大学出版社, 2004.2

(普通高等院校计算机专业(本科)实用教程系列)

ISBN 7-302-07894-7

I. J… II. ①耿… ②张… 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 (2003) 第 125569 号

出版者: 清华大学出版社

地 址: 北京清华大学学研大厦

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

邮 编: 100084

社总机: 010-62770175

客户服务: 010-62776969

责任编辑: 魏江江

封面设计: 龚正伟

印装者: 北京鑫海金澳胶印有限公司

发行者: 新华书店总店北京发行所

开 本: 185×260 印张: 25.75 字数: 643 千字

版 次: 2004 年 2 月第 2 版 2004 年 2 月第 1 次印刷

书 号: ISBN 7-302-07894-7/TP·5732

印 数: 1~5000

定 价: 36.00 元

序 言

时光更迭、历史嬗递。中国经济带着她足以令世人惊叹的持续高速发展驶入了一个新的世纪，一个新的千年。世纪之初，以微电子、计算机、软件、通信技术为主导的信息技术革命给我们生存的社会所带来的变化令人目不暇接。软件是优化我国产业结构、加速传统产业改造和用信息化带动工业化的基础产业，是体现国家竞争力的战略性产业，是从事知识的提炼、总结、深化和应用的高智型产业；软件关系到国家的安全，是保证我国政治独立、文化不受侵蚀的重要因素；软件也是促进其他学科发展和提升的基础学科；软件作为 20 世纪人类文明进步的最伟大成果之一，代表了先进文化的前进方向。美国政府早在 1992 年“国家关键技术”一文中提出“美国在软件开发和应用上所处的传统领先地位是信息技术及其他重要领域竞争能力的一个关键因素”，“一个成熟的软件制造工业的发展是满足商业与国防对复杂程序日益增长的要求所必需的”，“在很多国家关键技术中，软件是关键起推动作用（或阻碍作用）的因素”。在 1999 年 1 月美国总统信息技术顾问委员会的报告“21 世纪的信息技术”中指出“从台式计算机、电话系统到股市，我们的经济与社会越来越依赖于软件”，“软件研究为基础研究方面最优先发展的领域”。而软件人才的缺乏和激烈竞争是当前国际的共性问题。各国、各企业都对培养、引进软件人才采取了特殊政策与措施。为了满足社会对软件人才的需要，为了让更多的人可以更快地学到实用的软件理论、技术与方法。我们编著了《普通高等院校计算机专业（本科）实用教程系列丛书》。本套丛书面向普通高等院校学生，以培养面向 21 世纪计算机专业应用人才（以软件工程师为主）为目标，以简明实用、便于自学、反映计算机技术最新发展和应用为特色，具体归纳为以下几点：

1. 讲透基本理论、基本原理、方法和技术，在写法上力求叙述详细，算法具体，通俗易懂，便于自学。

2. 理论结合实际。计算机是一门实践性很强的科学，丛书贯彻从实践中来到实践中去的原则，许多技术理论结合实例讲，以便于学习和理解。

3. 本丛书形成完整的体系，每本教材既有相对独立性，又有相互衔接和呼应，为总的培养目标服务。

4. 每本教材都配以习题和实验，在各教学阶段安排课程设计或大作业，培养学生的实战能力与创新精神。习题和实验可以制作成光盘。

新世纪曙光激人向上，催人奋进。江总书记在十五届五中全会上的讲话：“大力推进国民经济和社会信息化，是覆盖现代化建设全局的战略举措。以信息化带动工业化，发挥优势，实现社会生产力的跨越式发展。”指明了我国信息界前进的方向。21 世纪日趋开放的国策与更加迅速发展的科技会托起祖国更加辉煌灿烂的明天。

孙家广

2001 年 3 月

普通高等院校计算机专业（本科）实用教程系列

编 委 会

主 任 孙家广（清华大学教授，中国工程院院士）

成 员（按姓氏笔划排序）

王玉龙（北方工业大学教授）

艾德才（天津大学教授）

刘 云（北方交通大学教授）

任爱华（北京航空航天大学教授）

幸云辉（北京邮电大学教授）

张海藩（北京信息工程学院教授）

徐孝凯（中国广播电视大学教授）

徐培忠（清华大学出版社编审）

樊孝忠（北京理工大学教授）

丛书策划 徐培忠 徐孝凯

目 录

第 1 章 Java 语言入门	1
1.1 Java 的诞生	1
1.2 Java 的特点	1
1.3 安装 Sun 公司的 SDK	3
1.4 一个 Java 程序的开发过程.....	5
1.5 一个简单的 Java 应用程序.....	5
1.6 一个简单的 Java 小应用程序 (Java Applet)	8
1.7 什么是 JSP	9
习题	10
第 2 章 标识符、关键字和数据类型	11
2.1 标识符和关键字.....	11
2.2 Java 语言基本数据类型.....	11
习题	16
第 3 章 运算符、表达式和语句	18
3.1 运算符与表达式.....	18
3.2 语句	23
习题	30
第 4 章 类、对象和接口	31
4.1 编程语言的几个发展阶段.....	31
4.2 类	33
4.3 对象	41
4.4 static 关键字	49
4.5 this 关键字.....	52
4.6 包	54
4.7 访问权限	59
4.8 类的继承	63
4.9 对象的上转型对象.....	67
4.10 多态性	69
4.11 abstract 类和 abstract 方法.....	70
4.12 super 关键字.....	72
4.13 接口	74

4.14 jar 文件	80
习题	82
第 5 章 数组与字符串	83
5.1 声明数组	83
5.2 创建数组	83
5.3 数组元素的使用	84
5.4 数组的初始化	85
5.5 字符串	85
5.6 对象的字符串表示	93
5.7 使用 StringTokenizer 类分析字符串	93
5.8 Character 类	94
5.9 字符串与字符、字节数组	95
习题	97
第 6 章 时间、日期和数字	99
6.1 Data 类	99
6.2 Calendar 类	101
6.3 Math 类	103
习题	104
第 7 章 AWT 工具集简介	105
习题	106
第 8 章 Java Applet 基础	107
习题	110
第 9 章 文本框和文本区	112
9.1 文本框	112
9.2 文本框上的 ActionEvent 事件	113
9.3 文本区	118
9.4 文本区上的 TextEvent 事件	120
习题	121
第 10 章 按钮与标签	124
10.1 按钮	124
10.2 扩展按钮	125
10.3 标签	128
10.4 扩展标签	128

习题	130
第 11 章 面板和画布	131
11.1 面板	131
11.2 画布	132
习题	135
第 12 章 布局设计	136
12.1 FlowLayout 布局	136
12.2 BorderLayout 布局	137
12.3 CardLayout 布局	138
12.4 GridLayout 布局	140
12.5 BoxLayout 布局	141
12.6 null 布局	144
习题	145
第 13 章 选择型组件	146
13.1 选择框	146
13.2 下拉列表	152
13.3 滚动列表	154
习题	156
第 14 章 Component 类的常用方法	157
14.1 组件的颜色	157
14.2 组件的字体	157
14.3 组件的大小与位置	159
14.4 组件的激活与可见性	160
14.5 组件上的光标	160
14.6 paint 方法与 repaint 方法	162
习题	163
第 15 章 建立窗口和菜单	164
15.1 Java 窗口	164
15.2 窗口与屏幕	165
15.3 菜单条、菜单、菜单项	167
15.4 有关菜单的几个技巧	170
15.5 窗口事件	171
15.6 WindowAdapter 适配器	173
15.7 打印	175

15.8 使用剪贴板.....	177
习题	180
第 16 章 建立对话框.....	181
16.1 Dialog 类.....	181
16.2 文件对话框.....	183
16.3 消息对话框.....	185
16.4 确认对话框.....	187
16.5 颜色对话框.....	188
习题	190
第 17 章 Java 与图形.....	191
17.1 绘制文本	191
17.2 绘制基本图形.....	192
17.3 建立字体	194
17.4 清除	195
17.5 Java 2D	196
17.6 图形的布尔运算.....	206
17.7 XOR 绘图模式	207
17.8 打印图形	209
习题	210
第 18 章 Java 中的鼠标事件和键盘事件.....	211
18.1 使用 MouseListener 接口处理鼠标事件.....	211
18.2 使用 MouseMotionListener 接口处理鼠标事件.....	215
18.3 鼠标事件的转移.....	217
18.4 键盘事件	219
18.5 围棋对弈、迷宫程序及华容道.....	221
习题	230
第 19 章 Java 多线程机制.....	233
19.1 Java 中的线程	233
19.2 Thread 类与 Runnable 接口	234
19.3 如何在程序中实现多线程.....	236
19.4 Thread 类的静态方法 sleep()	241
19.5 线程同步	241
19.6 在同步方法中使用 wait()、notify 和 notifyAll()方法.....	244
19.7 线程的 interrupt()方法	245
19.8 用线程显示本地时间.....	248

习题	252
第 20 章 输入输出流	254
20.1 File 类	254
20.2 FileInputStream 类	256
20.3 FileOutputStream 类	259
20.4 FileReader 类和 FileWriter 类	260
20.5 使用文件对话框打开和保存文件	266
20.6 运行可执行文件	268
20.7 RandomAccessFile 类	269
20.8 数据流	274
20.9 对象流	275
20.10 Process 类中的流	278
习题	281
第 21 章 Java 网络的基本知识	283
21.1 使用 URL	283
21.2 套接字	285
21.3 InetAddress 类	292
21.4 UDP 数据报	294
21.5 广播数据包	299
习题	303
第 22 章 Java 与图像	304
22.1 图像类型	304
22.2 Image 类	304
22.3 播放幻灯片和动画	306
22.4 怎样在应用程序中绘制图像	308
22.5 怎样设置 Java 窗口的图标	310
习题	311
第 23 章 Java 数据库连接(JDBC)	312
23.1 设置数据源	312
23.2 JDBC-ODBC 桥接器	314
23.3 顺序查询	316
23.4 可滚动结果集	319
23.5 排序查询	321
23.6 模糊查询	322
23.7 更新、添加、删除记录	323

23.8 数据库访问中的套接字技术.....	326
习题.....	329
第 24 章 Java 与多媒体.....	330
24.1 在小程序中播放声音.....	330
24.2 在另一个线程中创建音频对象.....	331
24.3 Java 媒体框架(JMF).....	333
习题.....	336
第 25 章 Java Swing 基础.....	337
25.1 几个重要的类.....	338
25.2 中间容器.....	342
25.3 各种组件.....	346
习题.....	376
第 26 章 常见数据结构的 Java 实现.....	377
26.1 链表.....	377
26.2 堆栈.....	383
26.3 树集.....	385
26.4 散列表.....	391
26.5 向量.....	395
习题.....	398
附录 A JDK 常用命令.....	399

第 1 章 Java 语言入门

Java 语言是一种很优秀的语言，具有面向对象、与平台无关、安全、稳定和多线程等优良特性，是目前软件设计中极为健壮的编程语言。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 Internet 的应用开发。Java 确实具备了“一旦写成处处可用”的特点，Java 已成为网络时代最重要的语言之一。本章将对 Java 语言做一个简单的介绍，读者可以初步了解什么是 Java 应用程序，什么是 Java 小应用程序，有关的细节会在后续的章节中讨论。

1.1 Java 的诞生

Java 是 1995 年 6 月由 Sun 公司引进到我们这个世界的革命性编程语言，它被美国的著名杂志《PC Magazine》评为 1995 年十大优秀科技产品。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作一番改动，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 语言的出现是源于对独立于平台语言的需要，希望这种语言能编写出嵌入各种家用电器等设备的芯片上、且易于维护的程序。但是，人们发现当时的编程语言，比如 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，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。他们的精心专研和努力促成了 Java 语言的诞生。Java 的快速发展得益于 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到来。

注：印度尼西亚有一个重要的盛产咖啡的岛屿叫 Java，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。

简单: Java 语言简单是指这门语言既易学又好用, 不要简单误解为这门语言很干瘪。你可能很赞同这样的观点: 英语要比阿拉伯语言容易学。但这并不意味着英语就不能表达丰富的内容和深刻的思想, 许多诺贝尔文学奖的作品都是用英文写的。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。

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。我们将在第 4 章详细地讨论类、对象等概念。

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 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) 的计算机上正确运行, Sun 公司实现了自己的目标——一次写成, 处处运行。

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, 该代码的运行就和特定的 CUP 有关。例如, 在 C 语言中, 我们都碰到过类似下面的问题:

int 型变量的值是 10, 那么下面代码的输出结果是什么呢?

```
printf("%d,%d",x,x=x+1);
```

如果上述语句的计算顺序是从左到右, 结果是 10, 11; 但是, 有些机器会从右到左计算, 那么结果就是 11, 11。

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种“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”相关, 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。

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。多线程易使人产生多个任务在同时执行的错觉, 其实, 并非如此。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。

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 小应用程序将被限制在 Java 运行环境中，不允许它访问计算机的其他部分。我们将在第 8 章详细讲述 Java 小应用程序。

动态：在学习了第 4 章之后，我们就会知道，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 安装 Sun 公司的 SDK

学习 Java 需要有一个程序开发环境。目前有许多很好的 Java 程序开发环境可用，包括来自 Sun、Borland、Sysmantech 公司的产品，例如 JBuild 等。但学习 Java 最好选用 Sun 公司推出的软件开发工具箱 SDK（以前称为 Java 开发工具箱 JDK）。目前 Sun 公司发布了 SDK 的 1.4 版本，可以登录到 Sun 公司的网站 <http://java.sun.com>，免费下载 SDK1.4（例如 j2sdk-1_4_1_02-windows-i586.exe）。如果你购买某些 Java 书，也可能得到一个含有 SDK 的光盘。假设 SDK 安装到 F:\jdk1.4 目录下，则会生成如图 1.1 所示的目录结构。

现在，就可以编写 Java 程序，并进行编译、运行程序了，因为安装 SDK 的同时，我们的计算机中就安装上了能运行 Java 程序的 Java 虚拟机。

SDK 安装目录的 bin 文件夹中含有编译器 (javac.exe)、解释器 (java.exe) 和一些其他的可执行文件，为了方便地使用编译器和解释器，应当将 bin 包含在 Path 的设置中，如果使用 Windows 操作系统，可以在 Ms-dos 命令行键入下列命令后回车确定即可：

```
Path F:\jdk1.4\bin
```

你也可以在系统特性中设置 Path。对于 Windows 2000，用鼠标右键单击“我的电脑”，弹出快捷菜单，然后选择“属性”命令，弹出“系统特性”对话框，再单击该对话框中的高级选项，然后单击“环境变量”按钮，添加如下的系统环境变量：

```
变量名: Path, 变量值: F:\jdk1.4\bin
```

如果曾经设置过环境变量 Path，可单击该变量进行编辑操作，将需要的值加入即可。对于 Win 9x，用记事本编辑 Autoexec.bat 文件，将如下的设置语句加入即可：

```
Path = C:\jdk1.4\bin;
```

但是，如果你的机器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，

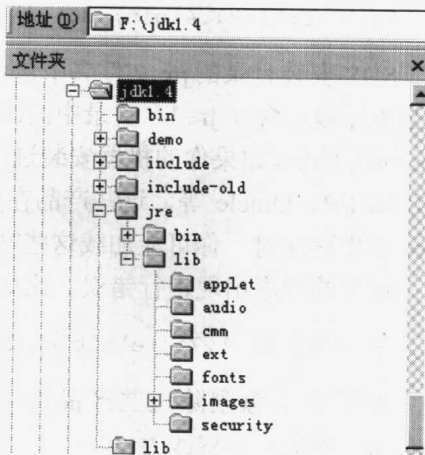


图 1.1 SDK 目录结构

如 PB、Oracle 等, 这些产品在安装后, 可能会修改了 Path 的值, 那么当你运行编译器或者解释器时, 你可能运行的是这些产品所带的老版本的编译器或者解释器。因此如果想使用 SDK 的编译器或解释器, 在设置 Path 的值时, 将你需要的值写在前面。比如, 某个软件用到的 Path 值是

```
d:\PB\jdk1.1.7\bin;
```

如果你不想删除这些产品对 Path 的设置, 那么你可以重新编辑系统环境变量 Path 的值, 并修改为:

```
F:\jdk1.4\bin; d:\PB\jdk1.1.7\bin;
```

如果是 Win 9x 操作系统, 就将 Autoexec.bat 文件的 Path 修改为:

```
Path=F:\jdk1.4\bin; d:\PB\jdk1.1.7\bin;
```

也可以在 Ms-dos 命令行键入下列命令后, 回车确认。

```
Path F:\jdk1.4\bin; d:\PB\jdk1.1.7\bin;
```

SDK 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 jre\lib 目录中的压缩文件 rt.jar 中。安装 SDK 一般不需要设置环境变量 classpath 的值, 如果你的机器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, 这些产品在安装后, 也可能会修改 classpath 的值, 那么当你运行 Java 应用程序时, 你可能加载这些产品所带的老版本的类库而导致程序要加载的类无法找到, 使你的程序出现运行错误。比如, 某个软件用到的 classpath 值为:

```
d:\PB\jdk1.1.7\jre\lib\classes.zip; .;
```

如果你不想删除这些产品的 classpath 设置, 那么你可以重新编辑系统环境变量 classpath 的值, 并修改为:

```
F:\jdk1.4\jre\lib\rt.jar; .;d:\PB\jdk1.1.7\jre\lib\rt.jar;
```

如果是 Win 9x 操作系统, 就将 Autoexec.bat 文件的 classpath 修改为:

```
set classpath=F:\jdk1.4\jre\lib\rt.jar; .;d:\PB\jdk1.1.7\ jre\lib\classes.zip
```

也可以在 Ms-dos 命令行键入下列命令后, 回车确认。

```
set classpath=F:\jdk1.4\jre\lib\rt.jar; .;
```

注: ① classpath 环境变量设置中的 “.” 是指可以加载应用程序当前目录中的类。

② 如果你的计算机安装过两个版本的 SDK, 比如 SDK1.3 和 SDK1.4, 若运行出现问题, 可以在 Ms-dos 命令行分别键入下列命令回车确认:

```
Path f:\jdk1.4\bin
```

```
Set Classpath=f:\jdk1.4\jre\lib\rt.jar; .;
```

如果你只想运行别人的 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。可以登录 Sun 的网站免费下载 Java 的 JRE。如果安装时选择了默认的安装路径就会形成如图 1.2 所示的目录结构。

注：建议下载 Sun 公司的 Java 类库文档，例如 j2sdk-1_4_1-doc。

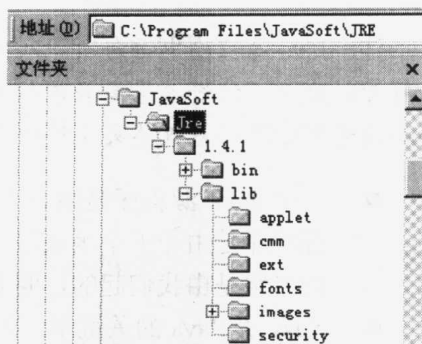


图 1.2 JRE 目录结构

1.4 一个 Java 程序的开发过程

Java 程序的开发过程如图 1.3 所示。

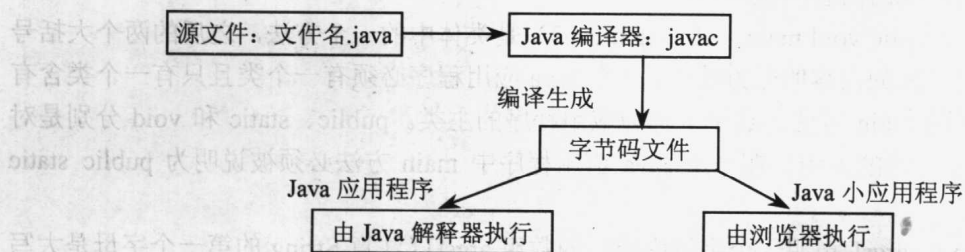


图 1.3 Java 程序的开发过程

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

编写源文件：使用一个文本编辑器，如 Edit 或记事本来编写源文件。不可使用 Word 编辑器，因它含有不可见字符。将编好的源文件保存起来，源文件的扩展名必须是 java。

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。

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。你马上就会知道怎样使用解释器和浏览器来运行程序。通常使用的 Netscape Navigator 和 Microsoft Explorer 浏览器都完全支持 Java。

1.5 一个简单的 Java 应用程序

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字或英文字符的限制。

- 一个 Java 源程序是由若干个类组成的。如果你学过 C 语言, 你就会知道一个 C 源程序是由若干个函数组成的。上面的这个 Java 应用程序简单到只有一个类, 类的名字是由我们起的, 叫 Hello。
- `class` 是 Java 的关键字, 用来定义类。`public` 也是关键字, 说明 Hello 是一个 `public` 类(后面将会系统地学习类的定义和使用)。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。
- `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这样的 `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`。
- `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 (以后会学习怎样使用这个参数, 见 5.9 节)。`main` 方法是程序开始执行的位置。

现在将源文件保存到 `C:\1000\` 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: 如果源文件中有多个类, 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

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

```
C:\1000\>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` 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称相同。这些字节码文件将被存放在与源文件相同的目录中。

如果你对源文件进行了修改, 那么你必须重新编译, 再生成新的字节码文件。

注: 如果你在安装 SDK1.4 时没有另外指定目录, `javac.exe` 和 `java.exe` 被存放在 `C:\jdk1.4\bin` 目录下, 如果你在任何目录下能使用编译器和解释器, 应在 DOS 提示符下键