

21 世纪计算机应用技术系列规划教材

Java 程序设计教程

丁新民 主 编

石林祥 副主编

人民邮电出版社

图书在版编目 (CIP) 数据

Java 程序设计教程 / 丁新民主编. —北京: 人民邮电出版社, 2006.2

(21 世纪计算机应用技术系列规划教材)

ISBN 7-115-14168-1

I. J... II. 丁... III. JAVA 语言—程序设计—教材 IV. TP312

中国版本图书馆 CIP 数据核字 (2006) 第 006675 号

内 容 提 要

本书从计算机软件设计员的角度出发, 结合软件设计语言的发展趋势, 就 Java 语言的应用与面向对象的程序设计, 详尽介绍传统结构化语言与面向对象语言的区别, 由浅入深地叙述 C、C++语言的基本结构, 从而引出 Java 面向对象程序设计思想。本书从 Java 语言和面向对象程序设计方法相结合入手, 以大量实例详细介绍 Java 的编程思想和编程方法。全书共分 10 章, 主要内容包括 Java 语言基础、面向对象程序结构、异常处理、Java 资源及其利用、图形用户界面、小应用程序、多线程机制、网络编程、Java 与数据库的连接。每章后都设有大量的基本概念题和程序设计题。

本书内容丰富、实用、可操作性强, 概念叙述深入浅出, 知识点结合实例讨论, 既可作为高等院校工程应用型本科专业的“Java 语言程序设计”课程的教材, 也可以作为软件程序员的培训教材, 以及从事软件设计的工程技术人员的技术参考书。

21 世纪计算机应用技术系列规划教材

Java 程序设计教程

-
- ◆ 主 编 丁新民
副 主 编 石林祥
责任编辑 邹文波
 - ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京通州大中印刷厂印刷
新华书店总店北京发行所经销
 - ◆ 开本: 787×1092 1/16
印张: 19.5
字数: 462 千字 2006 年 2 月第 1 版
印数: 1—3 000 册 2006 年 2 月北京第 1 次印刷

ISBN 7-115-14168-1/TP · 5071

定价: 26.00 元

读者服务热线: (010)67170985 印装质量热线: (010)67129223



Java 语言是网络时代广泛使用的面向对象的编程语言，近年来，我国计算机软件业被广泛地运用于各类信息管理和控制工程等领域。Java 语言成为程序设计语言的一个重要部分，目前，许多高等院校都将它作为计算机基础必修课程。《Java 程序设计教程》教材旨在满足高等院校计算机基础课程的教学与实践需求，以及计算机编程者入门和提高编程能力，同时又兼顾 IT 国际认证对 Java 编程考核的要求。

“Java 程序设计教程”课程作为计算机类专业的专业基础课程，要让学生理解程序设计的基本概念，学会使用 Java 语言创建 Java 程序，掌握一种开发平台，会用 Java 语言面向对象的特点；会用 Java 语言的事件处理模式，实现程序的输入/输出以及文件操作，会用 Swing 类库开发图形用户界面，会用 Java 语言的异常处理机制控制程序的执行和定义用户自己的异常处理，掌握 Java 中多线程的设计方法，会用 JDBC 访问数据库。通过学习掌握一个利用计算机解决本学科问题的基本方法，为今后进一步学习打下坚实的基础。

为此，本书各部分内容的选择，突出面向对象程序设计和计算机网络环境下的程序设计，以适应软件程序员的技术特征为着手点。本书内容由浅入深，增加基本算法的介绍，以培养学习者的编程思想，逐步提高编程能力，内容的叙述以贴近实际并适合初学者为切入点。在内容的选取和安排上，适当地介绍了 C++ 语言的基本结构和 Java 语言程序的区别，着重阐述 Java 面向对象语言结构及其应用技术，同时也介绍了 TCP 和 UDP 编程、URL 类编程、JDBC 应用程序接口和动态数据库访问等编程技术。

本书注重对基本概念、基本编程技术的深度介绍，强调内容覆盖面的广度与应用技术的深度。同时，在重要章节都增加了应用示例、程序设计示例和实例，灵活地应用本章节的相关知识点，通过示例把相关知识串联起来，使得学习者能对本章节的知识运用自如，能利用 Java 编写实际的应用程序。注意全书的系统性和实用性、循序渐进、突出理论与实践的紧密结合。书中引用的所有示例，都上机通过验证；并在每章的结束都配有相应的习题和思考题。

本书主要内容共包括 10 章，各章主要内容如下。

第 1 章介绍程序设计方法的基本知识和基本概念，描述 Java 语言的特点、运行环境和开发步骤。

第 2 章结合简单的 Java 程序，介绍了 Java 语言中的基本数据类型、保留字、算术运算

符、算术表达式、运算优先权、Java 程序设计中的流程控制语句，以及穷举、叠代和递归算法在 Java 中的实现。

第 3 章详细讲述 Java 基于类面向对象的程序设计所需的知识，主要介绍类、接口和包的概念。最后以数据结构中的单向链表、队列和堆栈在 Java 中的实现为例，综合练习 Java 的面向对象的程序设计技术。

第 4 章介绍 Java 程序中异常产生的原因，异常捕获处理的语法结构 try-catch-finally，同时通过例子强调了除了系统已定义的异常外，用户自己也可定义和抛出异常。

第 5 章首先介绍系统类、包装类(数据类型类)、数组、字符串类、缓冲串类和数学类的使用。然后详细介绍了 Java 中用于输入、输出以及文件操作的 I/O 包，同时通过具体例子，给出了包中各个类的使用方法以及如何在编程中灵活正确地应用 Java 语言的异常处理机制。最后简单介绍了 Java.util 包中的日期时间类、向量类和堆栈类。

第 6 章结合 Java 语言强大的 Swing 类库，介绍了图形用户界面的设计，内容包括按钮、菜单、文本框、列表框、滚动条等基本组件，以及界面布局管理、事件处理机制等。最后列举了编写一个文本编辑器的综合实例。

第 7 章讲述 Java 小应用程序设计的基本方法，使得学习者能在网页中调用 Applet 程序。通过 Applet 显示图像、播放声音和处理鼠标、键盘的事件。

第 8 章简单介绍 Java 中多线程的思路和方法，使学习者初步掌握 Java 多线程的设计方法，了解线程的状态和控制方法；以一个线程控制的综合实例详细讲解了 Java 中多线程的控制方法。

第 9 章介绍在 Java 中进行网络编程的方法和相应的类的使用。重点介绍 Socket 类和 ServerSocket 类，了解 TCP 和 UDP 类的使用，掌握 URL 类的使用，了解 URLConnection 类的使用。使学习者能进行简单的网络编程。

第 10 章介绍在 Java 中如何与数据库进行连接、访问数据库中的表和记录等信息，从而学会通过 JDBC 进行数据库类程序的开发。

本书的教学总学时数，一般可以安排 80 学时，其中课堂讲授学时 50 学时，上机实践学时 30，各章学时分配如下。

序 号	教 学 内 容	总 学 时	讲 课	实验实训上机
1	Java 语言概述	3	2	1
2	Java 语言基础	8	6	2
3	Java 面向对象程序结构	10	6	4
4	异常处理	3	2	1
5	Java 资源及其利用	12	8	4
6	图形用户界面 GUI	12	8	4
7	Java 小应用程序 Applet	8	4	4
8	Java 的多线程机制	6	4	2
9	Java 网络编程	10	6	4
10	Java 与数据库的连接	8	4	4
	小 计	80	50	30

使用本书的读者，可不必先修“程序设计基础”、“C 语言”等基础课程。对已修过上述课程的读者，对第 1 章和第 2 章的内容可适当做调整。

本书的第 1 章和第 2 章由丁新民编写；第 3、5、10 章由石林祥编写；第 4、6、7、8 章由贺海晖编写；第 9 章由李木杰编写。

华东师范大学张汝杰教授审阅了本书的编写大纲和书稿，提出了许多指导性意见。另外，刘中原老师、贺海治老师为本书的出版做了许多工作，在此，对他们表示深深的谢意。

由于作者水平有限，书中难免存在错误之处，希望读者批评指正。

编者 于上海第二工业大学
2005 年 12 月

目 录



第 1 章 Java 语言概述	1
1.1 Java 简介	1
1.1.1 Java 的崛起	1
1.1.2 面向对象的程序设计语言	1
1.1.3 Java 平台和虚拟机	2
1.1.4 两类 Java 程序	3
1.1.5 Java 程序开发环境——SDK	3
1.2 Java 程序开发过程	4
1.2.1 Java 应用程序开发过程举例	4
1.2.2 Java Applet 开发过程举例	6
1.3 NetBeans 3.6 集成开发环境简介	7
1.3.1 主窗口	8
1.3.2 主菜单栏	9
1.3.3 工具栏	10
1.3.4 组件选项面板	10
1.3.5 NetBeans 下的 Java 编程	10
小结	12
习题	12
第 2 章 Java 语言基础	14
2.1 一个简单的 Java 程序实例	14
2.1.1 类首部	15
2.1.2 类体	15
2.2 标识符、注释和分隔符	15
2.2.1 标识符	15
2.2.2 注释	16

2.2.3 分隔符	16
2.3 基本数据类型	16
2.4 变量和常量	17
2.4.1 常量	17
2.4.2 变量	19
2.4.3 数据类型转换	22
2.5 运算符及表达式	23
2.5.1 表达式	23
2.5.2 运算符	23
2.5.3 算术运算符	24
2.5.4 关系运算符	24
2.5.5 逻辑运算符	25
2.5.6 位运算符	25
2.5.7 赋值运算符	27
2.5.8 条件运算符	27
2.6 运算符的优先级	28
2.7 流程控制语句	28
2.7.1 if 语句和 if-else 语句	29
2.7.2 switch-case 语句	31
2.7.3 循环语句	33
2.7.4 for 语句	35
2.7.5 循环嵌套语句	36
2.7.6 break 和 continue 语句	37
2.8 Java 语言与 C++ 语言的异同	38
2.9 算法设计实例	40
2.9.1 穷举	40
2.9.2 迭代	41
2.9.3 递归	42
小结	42
习题	42
第 3 章 Java 面向对象程序结构	47
3.1 面向对象的理论基础	47
3.2 类与对象的基本概念	48
3.3 Java 的类	49
3.3.1 类首说明	49
3.3.2 类体	50
3.3.3 方法说明	54
3.4 类的层次结构	56

3.4.1	类继承的语法形式	56
3.4.2	类的继承规则	58
3.4.3	抽象类和方法	58
3.4.4	最终类 (final 类)	59
3.5	对象的创建与使用	60
3.5.1	对象说明	60
3.5.2	对象的实例化和初始化	60
3.5.3	对象的成员变量及方法的访问	60
3.6	多态性	61
3.6.1	方法的重构与重载	61
3.6.2	方法的动态调用	63
3.7	接口	65
3.7.1	接口的概念	65
3.7.2	接口的定义	66
3.7.3	接口的实现	67
3.8	包	68
3.8.1	包说明	68
3.8.2	包的层次结构	68
3.8.3	包的使用	68
3.8.4	Java 包	70
3.9	访问控制与修饰符	70
3.9.1	类成员的访问权限	70
3.9.2	类的修饰符号	74
3.9.3	null、this、super	74
3.10	综合应用示例	76
3.10.1	单向链表	76
3.10.2	队列实例	81
3.10.3	堆栈	81
3.10.4	单向链表、队列和堆栈应用实例	81
	小结	83
	习题	84
第 4 章	异常处理	87
4.1	异常类的层次	87
4.2	间接抛出异常 (throws)	89
4.3	直接抛出异常 (throw)	89
4.4	try-catch 异常处理结构	90
4.5	自定义异常	94
4.6	正确使用异常	96

小结	96
习题	97
第 5 章 Java 资源及其利用	99
5.1 Java 的标准库结构	99
5.2 语言包 (Java.lang)	100
5.2.1 基本对象类	100
5.2.2 系统类	101
5.2.3 包装类 (数据类型类)	103
5.2.4 数组	107
5.2.5 字符串类和缓冲串类	110
5.2.6 数学类	116
5.3 输入/输出包 (java.io)	118
5.3.1 流的综述	118
5.3.2 基本输入输出流	119
5.3.3 File 类	121
5.3.4 文件输入输出流	123
5.4 java.util 包	128
5.4.1 日期时间类	128
5.4.2 向量类及其使用	130
5.4.3 堆栈类	133
5.5 文件读写应用实例	134
小结	138
习题	139
第 6 章 图形用户界面	142
6.1 GUI 组件包简介——Swing 组件包	142
6.2 基本组件	143
6.2.1 按钮类 (JButton 类)	143
6.2.2 标签类 (JLabel 类)	144
6.2.3 文本框 (JTextField 类)、多行文本框 (JTextArea 类)、JPasswordField 类	145
6.2.4 复选框类 (JCheckBox 类)、单选框类 (JRadioButton 类)	148
6.2.5 组合列表类 (JComboBox 类)、列表类 (JList 类)	153
6.2.6 滚动条类 (JScrollBar 类)	154
6.2.7 JMenuBar 类、JMenuItem 类、JPopupMenu 类	156
6.2.8 JFrame 类和 JPanel 类	160
6.3 Java 的事件处理	162
6.3.1 事件处理模型	162
6.3.2 鼠标事件处理	164

6.3.3 键盘事件处理	166
6.4 Java 布局管理器	168
6.4.1 FlowLayout 类	168
6.4.2 BorderLayout 类	169
6.4.3 GridLayout 类	171
6.4.4 CardLayout 类	172
6.4.5 GridBagLayout 类	174
6.5 NetBeans 中可视化程序设计实例	175
小结	187
习题	187
第 7 章 Java 小应用程序 Applet	191
7.1 Applet 程序的设计基础	191
7.2 Applet 类	192
7.3 Applet 类的 5 种常用方法	193
7.4 Applet 的应用	194
7.4.1 利用 Applet 接收 HTML 中传递过来的参数	194
7.4.2 利用 Applet 显示图像	196
7.4.3 利用 Applet 播放声音	199
7.4.4 Applet 中的鼠标、键盘操作	199
7.4.5 Applet 之间进行通信	204
7.5 Applet 的执行	207
7.5.1 利用 Appletviewer 执行	207
7.5.2 通过浏览器执行	208
7.6 Applet 程序开发实例	209
小结	211
习题	212
第 8 章 Java 的多线程机制	213
8.1 多线程的基本概念	213
8.2 多线程的实现机制	214
8.2.1 继承 Thread 类	214
8.2.2 实现 Runnable 接口	216
8.3 Thread 类	217
8.3.1 Thread 类成员变量及方法	217
8.3.2 线程状态	218
8.3.3 线程的优先级	219
8.3.4 线程同步	220
8.4 线程通信与死锁	224

8.4.1 线程通信	224
8.4.2 死锁	226
8.5 线程控制综合举例	228
小结	235
习题	235
第 9 章 Java 网络编程	238
9.1 Java 网络包 (java.net)	238
9.1.1 套接口类 (Socket 类)	238
9.1.2 服务器套接口 (ServerSocket 类)	240
9.2 使用 TCP 和 UDP 编程	240
9.2.1 基于 TCP 的网络程序设计	240
9.2.2 基于 UDP 的网络程序设计	246
9.3 URL 类编程	250
9.3.1 URL 简介	250
9.3.2 构造 URL 类对象	250
9.3.3 URL 类中的异常	251
9.3.4 URL 类对象获取 URL 特征	251
9.3.5 利用 URL 类对象获取 URL 内容	252
9.4 URLConnection 类的使用	253
9.4.1 利用 URLConnection 类对象获取 URL 内容	253
9.4.2 利用 URLConnection 类对象向 URL 对象发送服务请求	254
9.5 网络编程综合应用实例	256
小结	270
习题	271
第 10 章 Java 与数据库的连接	272
10.1 JDBC 数据库连接简介	272
10.1.1 JDBC 的基本功能	272
10.1.2 JDBC 在数据库访问的两层与三层模型上的应用	272
10.1.3 JDBC 驱动程序类型	273
10.2 JDBC 应用程序接口	274
10.3 JDBC 的 API 简介	275
10.4 JDBC 应用	276
10.4.1 与数据库建立连接	277
10.4.2 执行查询语句	278
10.4.3 检索结果集	279
10.4.4 更新数据库操作	280
10.4.5 参数的输入和输出	281

目 录

10.5 动态数据库访问	282
10.6 Java 数据库编程综合实例	283
小结	293
习题	293
参考文献	296

第 1 章

Java 语言概述



在计算机技术的迅速发展过程中，计算机程序设计语言也在飞速发展。其发展的方向之一就是更接近于人们的思维习惯，使得计算机程序的编写、阅读、维护、复用和移植越来越容易。Java 语言就是这样发展起来的一种高级语言，它具有可移植、稳定、简单、高性能和可动态执行等特性。目前，它已经成为最常用的计算机语言之一。本章主要介绍 Java 语言的历史背景和特点，以及两类 Java 程序及其开发环境。

1.1 Java 简介

1.1.1 Java 的崛起

计算机语言种类很多，但要做到编写的程序一次就可在不同的平台上执行是很难达到的，Java 却做到了。Java 语言的前身是 Oak 语言。1991 年，Sun 公司为了占领智能消费型电子产品的市场，资助了一个“绿色项目”，这个项目主要是开发用于智能型家电产品的语言，即 Oak 语言。Oak 语言是在 C 和 C++ 语言的基础上进行简化和改进的一种语言，后来发现另有程序语言已经命名为 Oak，于是，Sun 公司重新给这种语言命名为 Java 语言。

但 Java 语言很快就遇到了一些困难，因为 Sun 公司发现市场对于智能型家电产品的需求并不如预期的那样快速，而且当时 Sun 公司在竞争一个大项目时失败了，Sun 公司差一点就要取消这个“绿色项目”。1993 年，Sun 公司重新分析市场需求，认为网络具有很好的发展前景，而且 Java 语言似乎非常适合网络编程。于是 Sun 公司将 Java 语言的应用背景转向网络市场，为网页增加了“动态的内容”。

Sun 公司于 1995 年 5 月 23 日推出 Java 语言，一经公布，立即引起了巨大的轰动。Java 语言为网络的发展开辟了一个新纪元。1996 年，Sun 公司专门成立 Javasoft 公司来发展 Java。Java 语言从此得到了迅猛的发展和广泛的应用，这种速度也是前所未有的。目前，Java 语言拥有“互联网上的世界语”的美誉。

1.1.2 面向对象的程序设计语言

Java 语言非常像 C++ 语言，但没有 C++ 语言那么复杂。它在面向对象的特性上就比 C++

语言更为彻底。面向对象模型是一种模拟人类社会和人解决实际问题的模型，因此更符合人们的思维习惯。与其他程序设计方式相比，采用这种方式进行程序设计似乎显得更为合理。Java 语言是一种面向对象的程序设计语言，它通过描述对象本身以及对象间的相互关系，并通过激发对象间的相互作用，来求解问题。

面向对象程序设计的 3 个特性如下。

1. 数据封装 (Encapsulation)

数据封装是指将数据分成私用 (Private)、保护 (Protected) 和公用 (Public) 等，实现数据保密的目的。

2. 继承 (Inheritance)

有了继承的关系后，父类 (Super Class) 中的数据 (Data) 或方法 (Method) 的子类 (Subclass) 就可继承使用，子类的子类也可以继承使用，实现数据重复使用 (Reuse) 的目的。

3. 多态 (Polymorphism)

同样名称的方法，其程序代码不同。假如，父类中有一个名叫 show() 的方法，用于显示家庭地址数据，子类中也有一个名叫 show() 的方法，用于显示电话号码数据，这两个方法都称为 show()，但做法不同，程序代码也不同，子类的 show() 方法将覆盖掉父类的 show() 方法，称为多态。

1.1.3 Java 平台和虚拟机

平台是支持程序运行的软硬件环境。Java 程序必须有自己的运行环境，即 Java 平台。一个 Java 程序运行环境包括实际计算机、适配器、Java 虚拟机 (JVM)、Java 基本软件和 Java 应用程序接口 (API)。Java 虚拟机是 Java 运行环境的核心，它易于被移植到基于不同硬件的平台上；Java 的基本软件也称为基本类，而 Java 应用程序接口是大量已编译好的程序代码库，可以直接使用它们，这个接口提供了丰富的 Java 资源。实际上，Java 的基本类和 API 的规模并不固定，许多 Java 平台还补充了扩展类。

Java 虚拟机从结构上看与实际的计算机相似，如图 1.1 所示。它的作用是使得一台实际的机器能够运行 Java 字节码程序。一台 Java 虚拟机就是一台 Java 解释器。经过 Java 解释器的解释，将 Java 源程序转换成 Java 字节码程序。Java 字节码不能被计算机直接执行，但它可以被所有的 Java 虚拟机执行，这就是 Java 字节码程序可以在网络上移植的原因。

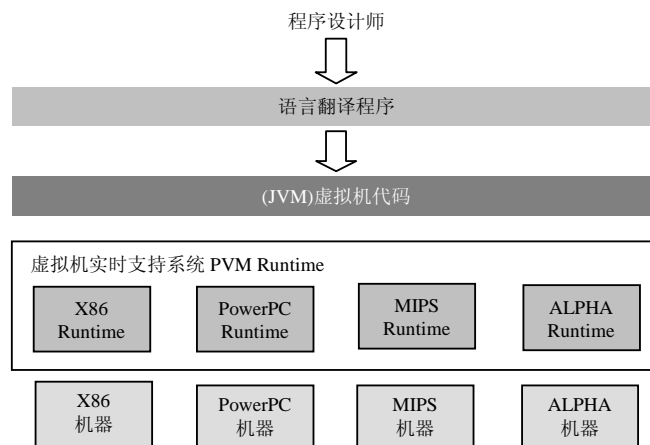


图 1.1 虚拟机原理示意图

由此可见, Java 虚拟机与操作系统和计算机硬件无关。经过编译后的 Java 字节码程序可以在网络上传送到任意支持 Java 虚拟机的网点机上运行。

利用 Java 语言可以开发两种形式的应用程序: Java 应用程序 (Java Application)和 Java 小应用程序 (Java Applet)。Java 小应用程序是指借助浏览器运行的程序。

1.1.4 两类 Java 程序

Java 应用程序和 Java 小应用程序。都是在 Java 虚拟机上执行, Java 应用程序在本机上由 Java 解释程序来激活 Java 虚拟机, Java Applet 是通过浏览器来激活 Java 虚拟机的。

1. Java 应用程序

Java 应用程序是指能够独立执行的程序,可以用 Java 解释器直接执行编译后的应用程序。在 Java 语言中,每个应用程序都是以类的形式出现的,一个程序中可以包含一个或多个类。Java 提供了一个特殊的方法—main ()方法,每个应用程序的执行都是从主方法 main ()开始的,没有 main () 的应用程序不能执行。包含了 main ()方法的类叫主类,程序的主文件名必须与主类相同。

2. Java 小应用程序

Java 小应用程序 (Java Applet) 与 Java 应用程序主要的区别在于它不能被直接执行,而必须通过支持 Java 的网络浏览器,如 Navigator、IE 等能看到执行的结果。我们也可以把它看作是网络浏览器执行的应用程序。Java (Applet) 可以使原本静态的网页变成一个充满活力的页面。

1.1.5 Java 程序开发环境——SDK

一个 Java 程序的开发过程要经过源程序文件的编辑、源程序文件的半编译和字节码文件的半解释 3 个步骤。Java 的源程序可以用编辑程序 (如记事本) 进行编辑,然后以纯文本方式存储在文本中,文件必须以 .java 作为后缀。Java 源程序由 Java 编译器编译成字节码,存储在以 .class 为后缀的文件中,最后由运行系统解释和执行,如图 1.2 所示。

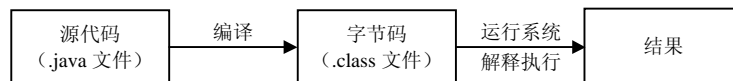


图 1.2 Java 程序的开发、执行流程

目前流行的 Java 开发环境有多种,如 Borland JBuilder、Visual J++和 JCreate 等,但从教学角度看, Sun 公司开发的 Java 2 SDK 环境有利于学习者理解整个程序的结构以及各部分的关系,有利于培养学习者的编程设计思想。

SDK 集成有 Java 编译器、Java 解释器和简单的排错工具等功能。目前 SDK 所支持的操作平台包括 Linux、Windows 等。

SDK 可以从 Sun 公司的站点 <http://java.sun.com/j2se/1.4.2/download.html> 下载,然后根据安装指南进行安装。以下把 Java 2SDK 简称为 JDK。

以 Windows 2000 系统为例, JDK 的安装与环境配置如下。

在 Windows 2000 下直接运行 exe 文件,安装到一个目录,例如,将其安装到 C:\JDK1.4 目录下面。安装完毕后查看该目录内容应具有:

```

C:\jdk1.4 的目录
2004-09-10  22:52      <DIR>      .
2004-09-10  22:52      <DIR>      ..
2003-06-20  07:00                4,519 COPYRIGHT
2004-06-08  20:22                17,932 LICENSE
2004-06-08  20:22                28,905 LICENSE.RTF
2004-06-08  20:22                16,703 README.html
2004-06-08  20:22                9,215 README.TXT
2004-06-08  20:22                10,367 THIRDPARTYLICENSEREADME.txt
2003-06-20  07:00            11,849,754 SRC.ZIP
2004-09-10  22:52      <DIR>      DEMO
2004-09-10  22:52      <DIR>      LIB
2004-09-10  22:52      <DIR>      JRE
2004-09-10  22:53      <DIR>      INCLUDE
2004-09-10  22:53      <DIR>      BIN
2004-10-09  22:13      <DIR>      SRC
          7 个文件      11,937,395 字节
          8 个目录      742,150,144 可用字节

```

其中 LIB 目录下保存的是库文件，BIN 目录下是一些执行文件和动态链接文件，DEMO 目录下是一些 Applet 的示例。

配置环境变量过程如下：在桌面上用鼠标右键单击“我的电脑”图标，依次选择“属性”→“高级”→“环境变量”，在“系统变量”栏中单击“新建”按钮；在“变量名”文本框中输入 CLASSPATH，在“变量值”文本框中输入类库名 C:\JDK1.4\LIB\DT.JAR；C:\JDK1.4\LIB\TOOLS.JAR；然后单击“确定”按钮。再次单击“系统变量”栏中的“新建”按钮，在“变量名”文本框中输入 PATH，在“变量值”文本框中输入 C:\JDK1.4\BIN，然后单击“确定”按钮。或者直接修改 Autoexec.bat 文件，利用任何编辑器编辑该文件，然后在其中加入如下的设置语句：

```

set path=c:\jdk1.4\bin;
set classpath=c:\jdk1.4\lib\dt.jar;c:\jdk1.4\lib\tools.jar;

```

在配置完成后，要重新启动计算机，环境变量配置才能生效。

1.2 Java 程序开发过程

本节通过两个例子，分别介绍 Java 应用程序和 Java 小应用程序（Java Applet）的开发过程。

1.2.1 Java 应用程序开发过程举例

用 Windows 文字编辑工具和 SDK 开发一个 Java 应用程序的过程，分以下 3 步。

(1) 编辑：创建 Java 源程序。源程序是用 Java 编程语言直接编写的程序，通常是由一

个或多个类组成的源文件，后缀为.java。创建、修改和保存源程序文件的过程称为源程序文件的编辑，这项工作可以在任何一种文本编辑器上进行。

(2) 半编译：是用 Java 编译器 (Javac) 将 Java 源程序翻译成 Java 虚拟机能理解的指令，再将这些指令组织为字节码文件 (后缀为.class) 的过程。如果在编译中出错，一般要返回第 (1) 步改正源程序中的错误。

(3) 半解释 (运行)：Java 虚拟机解释、运行包含在字节码文件中的程序。运行出错，应返回第 (1) 步改正源程序中的错误，再重新编译、运行。

1. 创建源文件

打开文本编辑器 (如记事本)，在一个新的文件中，输入程序源代码，如图 1.3 所示。

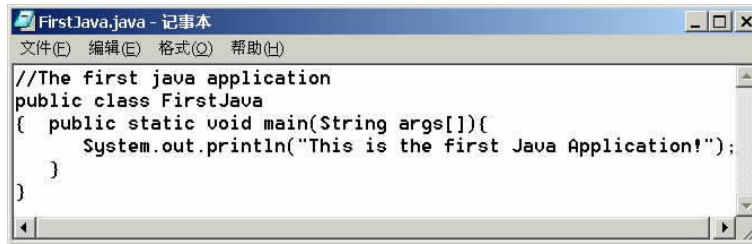


图 1.3 FirstJava.java 源程序

可以看出 Java 程序实际上由一组关键字、标识符、变量、分隔符和注释等“单词”构成。在输入源代码时，采用按层次对齐和缩进书写格式，能使程序结构层次清晰、易懂，便于修改。

程序源代码输入完毕，检查无错后，应将这些代码保存到一个文件中。Java 规定，Java 应用程序源代码的主文件名必须与主类或公共 (public) 类名相同，且大小写一致，后缀名必须为.java，因此上述程序必须用 FirstJava.java 作为文件名并以纯文本格式存储。在保存源程序文件时，一是选择好源文件的保存位置，建议在 D 盘上建立一个文件夹 javasample，然后将所有的源文件都保存在这里；二是最好在文件名前后加上双引号。这样就可以确保系统不会在文件名后再自动加上“.txt”扩展名 (因为我们选择的保存类型是“文本文档”)，而是完全按双引号内的形式来保存。

2. 半编译

在 JDK 中，Java 的编译器是 javac.exe。使用 javac 命令，就可以将 Java 源文件编译成字节码。对应于 Java 源文件里的每一个类，javac 命令都将生成一个与此类类名相同但后缀为.class 的字节码文件，并将它们存放在.java 源文件所在的文件夹中。一般情况下，为了操作的方便和有效，建议将源文件所在的文件夹设为当前文件夹。

javac 命令的一般格式为

```
javac <文件名>.java
```

在本例中，编译源文件时使用如下命令 (注意大小写)：

```
javac FirstJava.java
```

结果将在同一文件夹中生成字节码文件 FirstJava.class。注意：一定要按上一节设置好 path 环境变量。

3. 半解释——运行

编译生成的字节码文件可以在本机中由 Java 解释器 java.exe 运行。java 命令的一般格式为