

► 计算机与信息技术专业应用教材

Java 语言程序设计

朱福喜 编著

清华大学出版社

北京

内 容 简 介

Java 是一门流行的计算机语言,它已成为 IT 从业人员的必备工具。本书从 Java 的基本概念入手,介绍了 Java 的语言基础与主要功能,系统分析了 Java 语言的编程机制,并从面向对象程序设计的角度,阐述了 Java 语言的编程技巧。本书还介绍了一些 Java 语言的高级特性,如网络编程、数据库连接、多线程以及远程对象调用等。

本书通俗易懂,实用性强,适合计算机及相关专业的本科生作为学习 Java 语言的教材,也适合软件开发人员及其他有关人员阅读参考。

版权所有,翻印必究。举报电话:010-62782989 13901104297 13801310933

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

本书防伪标签采用清华大学核研院专有核径迹膜防伪技术,用户可通过在图案表面涂抹清水,图案消失,水干后图案复现;或将表面膜揭下,放在白纸上用彩笔涂抹,图案在白纸上再现的方法识别真伪。

图书在版编目(CIP)数据

Java 语言程序设计/朱福喜编著. - 北京:清华大学出版社,2005.1

计算机与信息技术专业应用教材

ISBN 7-302-10009-8

.J... .朱... .JAVA 语言—程序设计—高等学校—教材

.TP312

中国版本图书馆 CIP 数据核字(2004)第 124681 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

组稿编辑:夏非彼

文稿编辑:洪英

封面设计:付剑飞

版式设计:科海

印 刷 者:北京市耀华印刷有限公司

发 行 者:新华书店总店北京发行所

开 本:185×260 印张:24.125 字数:587 千字

版 次:2005 年 1 月第 1 版 2005 年 1 月第 1 次印刷

书 号:ISBN 7-302-10009-8/TP·6867

印 数:1~5 000

定 价:35.00 元

本书如存在文字不清、漏印以及缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:(010)82896445

前 言

Java 语言是美国 SUN 公司开发的程序设计语言，由于具有面向对象和网络编程的优点，现已成为网络世界的主流编程语言。自 1995 年发布以来，在 IT 业的应用一直保持强劲的增长势头。高等院校的计算机和非计算机专业相继开设了 Java 程序设计课程。

本书从 Java 的基本概念入手，介绍了面向对象程序设计的主要原理和方法，以及 Java 最主要的核心技术，并作了较深入的讨论。书中强调面向对象的思维方法训练，通过学习、设计及实现，使读者能在掌握理论的同时，具有一定的面向对象设计、开发能力，为大型软件的研究、设计打下基础。在此基础上，本书还系统分析了 Java 语言的编程机制，阐述了 Java 语言的精髓和使用 Java 语言的编程技巧，介绍了一些 Java 语言的高级特性，如多线程、网络编程、数据库连接以及远程对象调用等。书中列举的大量程序实例都经过精心的挑选，既要突出阐明原理和方法，又要保证有一定的实用性，同时也要确保一定的广度和深度，在难易程度上遵循由浅入深、循序渐进的原则。在举例过程中不仅注意到了让读者能从实例运用中举一反三，还尽可能地站在读者的角度去体会 Java 语言的精髓，并可以直接根据这些源程序来快速编写 Java 程序，直接切入相关应用。

本书第 1 章主要介绍 Java 的发展历史、语言特点并展示 Java 的独立应用程序和 Applet 程序的小实例，使读者对 Java 语言有一个概括性的了解。第 2 章介绍 Java 编程的基础知识，主要包括数据类型、变量、表达式和流程控制语句。第 3、4 两章介绍 Java 面向对象编程的知识，以类为中心详细地讨论面向对象技术的封装、抽象、继承和多态等特征以及它们在面向对象程序设计中的具体应用。让读者可以掌握 Java 语言和面向对象编程的精髓。第 5 章介绍 Java 图形用户界面的设计和编程实现，通过学习，读者可以编写出丰富多彩的程序界面，使开发的应用程序有“很专业”的外观。第 6 章介绍流和文件，这一章不仅是文件和输入输出操作的基础，也是后续的 Java 高级编程（如网络编程）的基础。第 7 章介绍 Java 的多线程编程和异常处理，掌握这一章的内容可以编写出功能复杂的多线程程序，又能够保证程序有足够的强壮性。第 8 章介绍 Applet 的设计，使 Java 语言在 Web 上充分展示其魅力。第 9 章介绍网络编程，这一章充分显示了 Java 的强大网络编程功能。第 10 章介绍 Java 数据库连接(JDBC)，掌握 JDBC 可以很方便地在 Java 程序中引入数据库应用。第 11 章介绍 Java 远程方法调用(RMI)，充分展示了 Java 的分布计算能力。

本书适合计算机及相关专业的本科生和研究生作为学习 Java 语言的教材，也适合软件开发人员及其他有关人员作为自学的参考书或培训教程。

本书编写过程中，尹为民教授和余振坤硕士给予作者很大帮助，余振坤、唐晓军、邓炜、田明、白圣猛等验证了本书的所有实例。作者还引用了陈劭欣毕业论文中的实例。编排录入工作中得到了王瑛、朱丽达和韩君的大力帮助。同时，本书的编写得到了科海培中夏非彼老师的大力支持，在此谨向他们表示衷心感谢。

由于时间仓促，作者水平有限，书中难免存在疏漏和不足，恳请读者批评指正，使本书得以改进和完善。

作 者

2004年12月于武昌珞珈山

目 录

第 1 章 Java 概述.....	1
1.1 Java 技术的出现与形成.....	1
1.2 Java 语言的特色.....	2
1.3 Java 的开发和执行环境.....	4
1.3.1 JDK 的下载.....	4
1.3.2 JDK 的安装与环境配置.....	4
1.3.3 JDK 环境工具简介.....	5
1.4 一个简单的独立应用程序.....	7
1.4.1 从编辑程序到执行程序的完整过程.....	7
1.4.2 Java 独立应用程序的基本结构.....	10
1.5 一个简单的 Applet 程序.....	11
1.6 一个具有输入功能的程序.....	12
1.7 小结.....	14
习题 1.....	14
第 2 章 Java 语言基础.....	15
2.1 变量和数据类型.....	16
2.1.1 变量的名字.....	16
2.1.2 变量的类型.....	16
2.1.3 变量初始化.....	18
2.1.4 final 变量.....	19
2.2 运算符.....	19
2.2.1 算术运算符.....	20
2.2.2 关系与逻辑运算符.....	23
2.2.3 位运算符.....	25
2.2.4 赋值运算符.....	25
2.2.5 其他运算符.....	26
2.3 表达式与语句.....	27
2.3.1 表达式.....	27
2.3.2 语句.....	28
2.4 控制语句.....	29
2.4.1 if 语句.....	30
2.4.2 switch 语句.....	31

2.4.3 while 和 do-while 语句	32
2.4.4 for 语句	34
2.4.5 跳转语句	34
2.5 数组	37
2.5.1 创建和使用数组	38
2.5.2 对象数组	39
2.5.3 多维数组	41
2.5.4 复制数组	42
2.6 小结	43
习题 2	43
第 3 章 类与对象	45
3.1 类	45
3.1.1 类的概念与定义	45
3.1.2 类声明	46
3.1.3 成员变量的声明	49
3.1.4 方法的声明与实现	51
3.2 创建对象与定义构造方法	61
3.2.1 创建对象	61
3.2.2 定义和使用构造方法	63
3.3 修饰符	64
3.3.1 类的修饰符	64
3.3.2 域修饰符	65
3.3.3 方法修饰符	66
3.3.4 访问控制符	68
3.4 类的进一步说明	75
3.4.1 嵌套类	75
3.4.2 对象的撤销与清理	77
3.5 程序包	77
3.5.1 包的创建	78
3.5.2 包的使用	79
3.6 小结	81
习题 3	81
第 4 章 继承与多态	83
4.1 子类、父类与继承机制	83
4.1.1 继承的概念	83
4.1.2 类的层次	85
4.2 继承的实现	85

4.2.1 子类的定义	85
4.2.2 域的继承与隐藏	88
4.2.3 方法的继承与覆盖	91
4.3 多态性	93
4.3.1 多态性的概念	93
4.3.2 覆盖实现多态性	94
4.3.3 重载实现多态性	95
4.4 构造方法的继承与重载	98
4.4.1 构造方法的继承	98
4.4.2 构造方法的重载	99
4.4.3 构造方法的综合示例	100
4.5 接口	102
4.5.1 接口与多继承	103
4.5.2 接口的定义	103
4.5.3 接口的实现	104
4.5.4 接口的使用	105
4.5.5 接口的完整实例	106
4.6 小结	108
习题 4	108
第 5 章 Java 的图形用户界面	110
5.1 图形用户界面概述	110
5.2 基本的图形用户界面程序	111
5.3 Container 类	112
5.3.1 一个简单的 GUI 组件与事件的响应	114
5.3.2 简单的 GUI 输入输出	116
5.4 事件处理	119
5.4.1 基本事件	119
5.4.2 Java 中的事件处理	120
5.5 GUI 组件和布局管理器	131
5.5.1 布局管理器、面板和标签	131
5.5.2 List 和 TextArea 组件	136
5.6 菜单和对话框	142
5.6.1 菜单	142
5.6.2 对话框	144
5.7 绘制图形	147
5.7.1 Graphics 类	147
5.7.2 简单绘图	148
5.7.3 Canvas 类	152

5.7.4 Font 类.....	157
5.8 从 AWT 到 Swing 的转换.....	159
5.9 小结.....	161
习题 5.....	161
第 6 章 流和文件.....	163
6.1 I/O 流概述.....	163
6.1.1 从类的层次看 I/O 流.....	164
6.1.2 从处理的对象看 I/O 流.....	171
6.2 保存和读取字节级数据.....	173
6.2.1 保存字节级数据.....	173
6.2.2 读取字节级数据.....	175
6.2.3 运用缓冲流来改善效率.....	178
6.2.4 文件操作.....	182
6.3 保存和读取字符数据.....	187
6.3.1 读取字符数据.....	188
6.3.2 保存字符数据.....	192
6.4 保存和读取对象数据.....	195
6.4.1 保存对象数据.....	195
6.4.2 读取对象数据.....	197
6.5 随机流访问.....	199
6.6 小结.....	201
习题 6.....	202
第 7 章 多线程与异常处理.....	203
7.1 多线程的基本概念.....	203
7.1.1 多任务.....	203
7.1.2 线程与多线程.....	203
7.1.3 Java 对多线程的支持.....	204
7.1.4 线程的状态.....	204
7.1.5 线程的优先级.....	206
7.2 线程的使用方法.....	206
7.2.1 通过继承 Thread 类构造线程.....	207
7.2.2 通过实现 Runnable 接口来构造线程.....	214
7.3 线程的同步.....	216
7.3.1 使用多线程不当造成的数据崩溃.....	216
7.3.2 同步线程.....	222
7.4 异常处理.....	228
7.4.1 Java 的出错类型.....	228

7.4.2 异常的处理	231
7.4.3 finally 子句	234
7.4.4 异常的抛出	236
7.5 小结	240
习题 7	240
第 8 章 Applet	242
8.1 Applet 基础	242
8.1.1 Applet 类的定义	242
8.1.2 Applet 的生命周期	244
8.1.3 独立应用程序与 Applet 的转换	246
8.1.4 确保 Applet 正常运行	247
8.2 <APPLET>标记	248
8.2.1 <APPLET>标记属性	248
8.2.2 利用标记参数向 Applet 传递信息	251
8.3 Applet 应用	256
8.3.1 Applet 与图形用户界面 (GUI)	256
8.3.2 Applet 与线程	259
8.3.3 Applet 编程技巧	259
8.4 利用 AppletViewer 调试 Applet	267
8.5 小结	270
习题 8	271
第 9 章 网络编程	272
9.1 网络编程基础	272
9.1.1 IP 地址	272
9.1.2 端口	273
9.1.3 套接字	273
9.1.4 Internet 协议	273
9.1.5 URL 类	274
9.2 使用 URL 访问 WWW 资源	275
9.2.1 创建一个 URL	275
9.2.2 直接从 URL 读取内容	276
9.2.3 建立一个 URL 连接并从中读取内容	278
9.3 客户和服务器编程	280
9.3.1 创建客户程序	280
9.3.2 编写服务器应用程序	291
9.4 基于客户/服务器模式的分布计算	297
9.4.1 分布模型	297

9.4.2 并行应用的实现框架	297
9.4.3 并行计算实例	299
9.5 小结	306
习题 9	307
第 10 章 Java 与数据库的连接	308
10.1 JDBC 概述	308
10.1.1 JDBC 的用途	308
10.1.2 从 ODBC 到 JDBC	308
10.1.3 JDBC 的实现及其驱动程序	309
10.2 JDBC 预备知识	310
10.2.1 SQL 基本内容	310
10.2.2 存储过程	313
10.3 JDBC API 介绍	315
10.3.1 连接管理	315
10.3.2 数据库访问	316
10.3.3 数据类型和数据库元数据	316
10.3.4 异常和警告	316
10.4 JDBC API 的使用方法	316
10.4.1 JDBC API 工具	316
10.4.2 连接数据库	318
10.4.3 使用 JDBC 创建基本表和视图	325
10.4.4 使用 JDBC 执行数据库查询	327
10.4.5 使用 JDBC 更新数据库	331
10.5 小结	335
习题 10	335
第 11 章 Java 远程方法调用	337
11.1 概述	337
11.2 一个简单的 RMI 应用实例	338
11.3 RMI 的类和接口及其实现分析	343
11.3.1 远程接口	343
11.3.2 RemoteException 类	343
11.3.3 RemoteObject 类及其子类	344
11.3.4 实现远程接口	345
11.3.5 Stub (存根) 和 Skeleton (框架)	345
11.3.6 配置 RMI 程序	346
11.3.7 启动 RMI 注册表	346
11.3.8 Naming 类	347

11.4 RMI 体系结构与实现机制.....	349
11.4.1 RMI 体系结构.....	349
11.4.2 RMI 实现机制.....	351
11.5 RMI 的进一步探讨.....	353
11.5.1 RMI 的安全性.....	353
11.5.2 传递对象和加载动态类.....	354
11.5.3 远程对象激活.....	360
11.5.4 分布式垃圾收集.....	365
11.6 小结.....	365
习题 11.....	367
附录 JDK 环境工具及其参数补充说明.....	368

P197

该类用于读出一个 `ObjectOutputStream` 流写入的初始类型数据以及整个对象。只有支持 `Serializable` 接口的对象才可以从这个流中读出。在恢复对象的类型过程中还应使用 Java 的类型转换 (type cast) 功能。Java API 对 `ObjectInputStream` 类的定义为：

```
public class ObjectInputStream extends InputStream implements
    ObjectInput, ObjectStreamConstants{
    public ObjectInputStream(InputStream in) throws IOException,
        StreamCorruptedException;
    public final Object readObject() throws OptionalDataException,
        ClassNotFoundException, IOException;
    public void close() throws IOException;
    public boolean readBoolean() throws IOException;
    public byte readByte() throws IOException;
    public char readChar()throws IOException;
    public int readInt() throws IOException;
    public double readDouble() throws IOException;
    public String readUTF() throws IOException;
}
```

第 1 章

Java 概述

Java 是 Sun Microsystem 公司研制的一种新型的程序设计语言。在高级语言已经非常丰富的背景下，Java 语言脱颖而出，不仅成为一门最为流行的计算机语言，而且形成一种专门的技术，有其独特的历史背景和独树一帜的品质。

1.1 Java 技术的出现与形成

Java 技术的出现与形成可追溯到十多年前，一些敢于创新的软件工程师试图开发出一种可移植软件，用于控制诸如烤箱、电视、冰箱、录像机、灯光设备、电话、呼机、传真机等家用电器。

1994 年 4 月，由美国 Sun Microsystem 的 Patrick Nawghton、Jame Gosling 和 Mike Sheridan 等人组成的开发小组，开始了代号为 Green 的项目的研制，其目标是研制一种开发家用电器的逻辑控制系统，产品名称为 Oak。但是，这个在技术上非常成功的产品，当时并未获得商业上的成功。

1994 年末，由于 Internet 的迅猛发展，WWW 以极快的速度风靡全球。Green 项目小组发现他们的新型编程语言 Oak 比较适合于 Internet 程序的编写，于是他们结合 WWW 的需要，对 Oak 进行改进和完善，并获得了极大的成功。

1995 年 1 月，Oak 被更名为 Java。这个名字的产生，既不是根据语言本身的特色来命名，也不是由几个英文单词的首字母拼成，更不是由人名或典故而来，而是来自于印度尼西亚的一个盛产咖啡的小岛的名字，小岛的中文名叫爪哇。正是因为许多程序设计师从所钟爱的热腾腾的香浓咖啡中得到灵感，因而热腾腾的香浓咖啡也就成为 Java 语言的标志。

1995 年 5 月 23 日 Java 正式公布，以后人们对 Java 的兴趣和重视证明了这项技术将是未来网络计算的主流技术。

1995 年一些著名的公司，如 IBM，Microsoft，Netscape，Novell，Apple，DEC，SGI 等都纷纷购买语言使用权。

1996 年 Java 应用软件纷纷问世，赢得软件工业界的广泛支持。

1997 年 Java 技术用于网络计算，从业界的事实标准走向法律标准。

今天，Java 已经形成一门技术，主要体现在：

- Java 软件技术 Java JDK、Java Runtime Environment、Hotjava、Java OS、JDBC、Java Beans。
- Java 虚拟机及规范 JRE 等。
- Java 嵌入技术 Java 芯片 (如 MicroJava701), 基于 Java 技术的 NC、Java Station、WebTV (机顶盒)、Java 汽车、Java 手机、Java 掌上机等。
- Java Computing 基于处理功能的综合处理系统, 如 J2EE Sever 或 Web Application Sever, 其典型产品有 Weblogic、Webspher 等。

1.2 Java 语言的特色

在 Sun 公司的白皮书中,对 Java 的定义是“Java :A Simple ,object-oriented ,distributed ,robust ,secure ,architecture-neutral ,portable ,high-performance ,multi-threaded ,and dynamic language.”即 Java 是一种简单的、面向对象的、分布式的、强壮的、安全的、体系结构中立的、可移植的、高性能的、多线程的动态语言。

这个定义充分说明了 Java 语言的特点。

1. 简单性

Java 是一种简单的语言。语言的设计者尽量把语言的规模变小。Java 取消了许多语言中十分繁琐和难以理解的内容,例如 C++的指针、运算符重载、类的多继承等,并且通过实现自动垃圾收集功能大大减化了程序设计者的内存管理工作。Java 在外观上让大多数程序员感到很熟悉,便于学习。同时 Java 的编译器也很小,便于在各种机型上实现。

2. 面向对象

Java 是一种面向对象的语言。这里的对象是指封装数据及其操作方法的程序实体。Java 的程序设计集中于对象及其接口,Java 提供了简单的类机制以及动态的接口模型,实现了模块化和信息封装。Java 类可提供一类对象的原型,再通过继承机制,实现了代码的重用。

3. 分布性

Java 是一个分布式语言。它有一套很齐全的通信及相关功能的程序库,可以处理 TCP/IP 协议及其他协议,用户可用 URL 地址在网络上很方便地访问其他对象。利用 Java 来开发分布式的网络程序是 Java 的一个主要应用之一。

4. 健壮性

用 Java 编写的程序能够在多种情况下稳定执行,这主要是因为编译和运行时都要对可能出现的问题进行检查。Java 有一个专门的指针模型,它的作用是排除内存中的数据被覆盖或毁损的可能性。Java 还通过集成面向对象的异常处理机制,在编译时提示可能出现但未被处理的异常,以防止系统的崩溃。

5. 安全性

Java 是一种安全的网络编程语言，不支持指针类型，一切对内存的访问都必须通过对象的实例来实现。这样能够防止他人使用欺骗手段访问对象的私有成员，也能够避免在指针操作中易产生的错误。此外，Java 有多个阶层的互锁保护措施，能有效地防止病毒侵入和蓄意的破坏行为。

6. 体系结构中立

Java 编译器能够产生一种与计算机体系结构无关的字节码 (Byte Code)，只要安装了 Java 虚拟机 (Java Virtual Machine)，Java 源代码 (Java Source Code) 就可以在相应的处理机上执行。这个过程可以用图 1.1 说明。

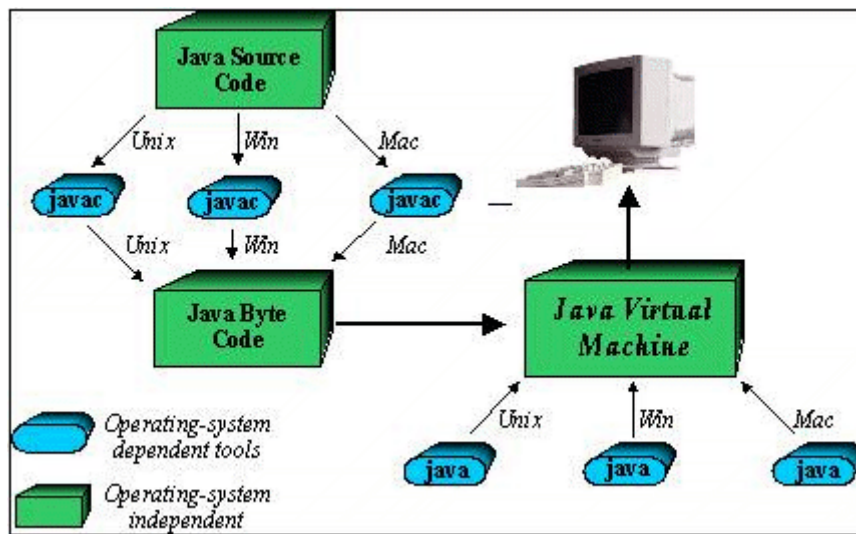


图 1.1 Java 程序与虚拟机

7. 可移植性

Java 具备有很好的移植性，这主要得益于它与平台无关的特性。同时，Java 的类库中也实现了与平台无关的接口，这使得这些类库也能移植。同时，Java 编译器主要是由 Java 本身来实现的，Java 的运行系统（虚拟机）由标准 C 实现，因而整个 Java 系统都具有可移植性。

8. 解释型语言

Java 虚拟机能直接在任意机器上解释执行 Java 字节码，字节码本身带有许多编译信息，使得连接过程更加简单。

9. 高效性

Java 的字节码能够迅速地转换成机器码，充分地利用硬件平台资源，从而可以得到较高的整体性能。

10. 多线程机制

Java 具有多线程机制，这使得应用程序能够并行地执行。它的同步机制也保证了对共享数据的共享操作，而且线程具有优先级的机制，有助于分别使用不同线程完成特定行为，也提高了交互的实时响应能力。

11. 动态性

Java 比 C++ 语言更有动态性，更能适应不断变化的环境。Java 不会因程序库某些部分的更新而需要全部重新编译程序，所以，在类库中可以自由地加入新的方法和实例变量，不会影响用户程序的执行。并且 Java 通过接口（interface）机制支持多重继承，使之比严格的类继承更具有灵活性和扩展性。

1.3 Java 的开发和执行环境

Sun 公司是 Java 的开创者，它的开发工具和运行环境都是免费的。只要可以进入 Internet，就可以免费下载开发工具集 JDK，其中包括了 Java 的全套文档资料。

除了 JDK 之外，目前有许多公司也成功地开发出了集成化的 Java 开发环境，其中比较流行的是 Inprise 公司的 Jbuilder 和 MicroSoft 的 Visual J++，它们都属于商业版的 Java 开发环境。

1.3.1 JDK 的下载

通过浏览器上网后，可以在 Sun 公司的 Internet 站点找到 JDK 的最新版本，自由下载。如果从其他镜像站点下载 JDK，则要注意这些镜像站点保存的是否是 JDK 的最新版本。

当前，JDK 的最高版本是 JDK1.5，它的下载网址是：<http://java.sun.com/j2se/1.5.0/download.jsp>。

Sun 公司提供支持 Windows XP/2000/NT 操作系统的版本，也提供支持类似 Solaris SPARC 和 Macintosh 操作系统的版本，用户可以根据操作系统平台选取合适的 JDK 版本进行下载。

1.3.2 JDK 的安装与环境配置

下面以 Windows XP/2000/NT 环境为背景，讨论 JDK 的安装与环境配置。

在 Windows 环境下，直接单击所下载的 JDK 压缩文件的图标，即可运行该文件，因为它是一个自解压的 EXE 文件。解压后，自动进入安装过程，此时可以按照提示过程，逐步完成安装。

安装时需要选择一个目录，可以选择比较简单的目录，例如，C:\JDK15，因为如果使用拼写复杂的目录会给 DOS 界面下的操作带来很多不便。安装完毕后，用 DIR 命令可以显示该目录。该目录的子目录有 BIN、LIB、DEMO、CLASSES、INCLUDE、JRE 等。

其中，BIN 目录下是一些执行文件，Java 的编译程序 Javac 就在该目录下。LIB 目录下保存库文件。DEMO 目录下是一些 Applet 的实例。CLASSES 目录下保存的是 Java 类库，在该目录下的文件有 COPYRIGHT、LICENSE、README.HTML、README.TXT 等。

在 Windows 2000/XP 下配置环境变量的方法是：右击桌面上的“我的电脑”图标，从弹出的快捷菜单中选择“属性”命令，在随后打开的对话框中单击“高级”标签，单击“环境变量”按钮，打开“环境变量”对话框，在“系统变量”区中单击“新建”按钮，在“新建系统变量”对话框的“变量名”文本框中输入 `classpath`，在“变量值”文本框中输入“`C:\JDK15\BIN; C:\JDK15\LIB\dt.jar; C:\JDK15\LIB\TOOLS.jar`”，最后单击“确定”按钮。

如果在低版本的 Windows 操作系统下，可直接编辑自动批处理文件 `Autoexec.bat`，在该文件中添加设置语句：

```
set classpath = C:\JDK15\BIN; C:\JDK15\LIB\dt.jar;
                C:\JDK15\LIB\TOOLS.jar;
```

配置完成后，需重新启动计算机，环境变量方能生效。

1.3.3 JDK 环境工具简介

本小节将简单介绍 JDK 环境工具，包括 Java 编译器、Java 虚拟机、Applet 程序观察器、Java 文档生成器、Java 调试器、C 文件生成器等。

首先，我们要了解 Java 程序主要分为两种类型：独立应用程序（stand-alone application）和小应用程序（Applet）。独立应用程序编译后由虚拟机直接解释执行，Java JDK 就是源代码编写、编译、预执行的集成环境；对于 Applet，需用同样的 Java 编译器先将它编译成 `.class` 的代码，然后再嵌入到某个 HTML 文件中，由某个浏览器解释执行。Applet 的嵌入可以增加 WWW 页面的交互性和动态性。

下面介绍的 JDK 工具有些是针对独立应用程序，有些是针对 Applet 程序的，这些工具的使用参数的详细介绍可参考附录。

1. 编译器——javac

`javac` 的作用是将源程序（`.java` 文件）编译成字节码（`.class` 文件）。Java 源程序的后缀名必须是 `.java`。`javac` 一次可以编译一个或多个源程序，对于源程序中定义的每个类，都会生成一个单独的类文件。因此，Java 源文件与生成的 `.class` 文件之间并不存在一一对应的关系。例如，如果在 `test.java` 中定义了 A、B、C 3 个类，则经过 `javac` 编译后要生成 `A.class`、`B.class`、`C.class` 3 个类文件。

`javac` 的调用格式为：

```
javac [选项] 源文件名表
```

其中，源文件名表是多个带 `.java` 后缀的源文件名；选项参见附录中的 `javac` 选项表。

2. Java 的语言解释器——java

`java` 命令解释执行 Java 字节码。其格式为：

```
java [选项] 类名 参数表
```

这里的类名代表要执行的程序名，即由编译后生成的带 `.class` 后缀的类文件名，但在上述命令中不需要带后缀。这个类名必须是一个独立程序（不能是 Applet），其中必须带有一个按如下格式声明的 `main()` 方法。

```
public static void main(String [ ] args ) {...}
```

并且包含 main()方法的类名必须与类文件名相同，即与现在命令行中的类名相同。

在执行 java 命令时，若类名后带有参数表，则参数表中的参数依次直接传递给该类中的 main()方法的 args 数组，这样在 main()方法中就可以使用这些数组元素。

java 命令所使用的选项参见附录中的 java 选项表。

3. Java 语言调试工具——jdb

jdb 调试用 Java 语言编写的程序。其格式为：

```
jdb [选项] 类名
```

或

```
jdb [-host 主机名] -password 口令
```

jdb 装载指定的类，启动内嵌的 Java 虚拟机，然后等待用户发出相应的调试命令，通过使用 Java debugger API 能够对本地或远程的 Java 虚拟机进行调试。

如果使用第一种命令格式，那么是由 jdb 解释执行被调试的类。若使用第二种格式，jdb 将被嵌入到一个正在运行的 Java 虚拟机之中，这个虚拟机必须事先用 -debug 选项启动，而且要求用户输入一个口令，这个口令也就是出现在命令行中的口令。如果使用了 -host 选项，jdb 就可以嵌入到网络中名为命令行中的“主机名”所指出的主机上正在运行的虚拟机之中。

所有的调试命令参数见附录中的调试命令表。

4. Java 文档生成器——javadoc

javadoc 从 java 源文件生成 HTML 格式的 API 文档，内容包括类和接口的描述、类的继承层次以及类中任何非私有域的索引和介绍。其格式为：

```
javadoc [选项] [包|文件名]
```

其中，选项参见附录中的 javadoc 参数表。

用户可以用包名或一系列的 Java 源程序名作参数。调用时，javadoc 可以自动对类、界面、方法和变量进行分析，然后为每个类生成一份 HTML 文档，并为类库中的类生成一份 HTML 索引。

5. C 头文件和源文件生成器——javah

javah 命令从一个 Java 类中生成实现 native()方法所需的 C 头文件和存根文件（.h 文件和.c 文件），利用这些文件可以把 C 语言的源代码装到 Java 应用程序中，使 C 可以访问一个 Java 对象的实例变量。其格式为：

```
javah [选项] 类名
```

在默认的情况下，javah 为每一个类生成一个文件，保存在当前目录中。若使用 -stub 选项则生成源文件；若使用 -o 选项则把所有类的结果存于一个文件之中。