

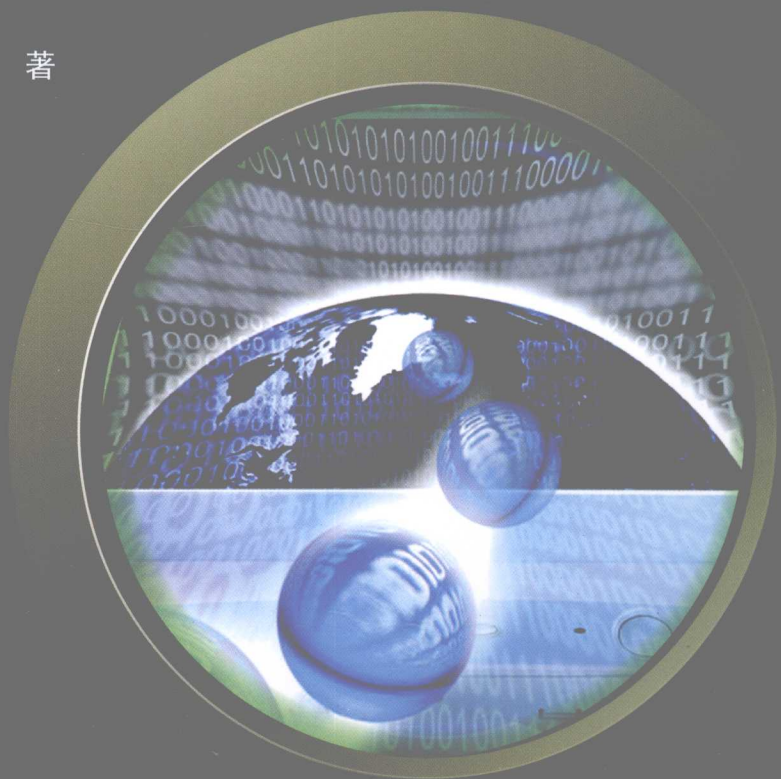


高等院校计算机应用技术系列教材

# Java

# 面向对象程序设计

马迪芳 徐保民 陈旭东 编 著



清华大学出版社 • 北京交通大学出版社

高等院校计算机应用技术系列教材

# Java 面向对象程序设计

马迪芳 徐保民 陈旭东 编著

清华大学出版社

北京交通大学出版社

·北京·

## 内 容 简 介

本书以面向对象程序设计思想为核心,详细讲述了面向对象程序设计的主要技术与编程思路。

本书以 Java SE 6 为基础,全面介绍 Java 语言的功能和技术特点,主要内容包括 Java 语言基础知识、面向对象编程技术、UML、异常、断言与日志、输入输出、泛型、集合框架、多线程、图形用户界面、网络编程等内容。

本书条理清晰,内容丰富,各章配有大量经过调试运行的实例。为了提高读者全面掌握 Java 面向对象编程技术和综合运用所学知识解决问题的能力,在附录中提供了两人对弈的五子棋游戏,包括基本的功能描述和完整的可运行示例代码,并在相关章节中附有配套练习题。

本书适用于高等院校计算机专业和相关专业,作为 Java 程序设计,或面向对象程序设计课程的教材,也非常适合专业技术人员参考和阅读。

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。

版权所有,侵权必究。侵权举报电话:010-62782989 13501256678 13801310933

### 图书在版编目(CIP)数据

Java 面向对象程序设计/马迪芳,徐保民,陈旭东编著. —北京:清华大学出版社;北京交通大学出版社,2009.6

(高等院校计算机应用技术系列教材)

ISBN 978-7-81123-553-1

I. J… II. ①马… ②徐… ③陈… 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(2009)第 050758 号

责任编辑:谭文芳

出版发行:清华大学出版社 邮编:100084 电话:010-62776969 <http://www.tup.com.cn>  
北京交通大学出版社 邮编:100044 电话:010-51686414 <http://press.bjtu.edu.cn>

印刷者:北京东光印刷厂

经 销:全国新华书店

开 本:185×260 印张:21.5 字数:547 千字

版 次:2009年6月第1版 2009年6月第1次印刷

书 号:ISBN 978-7-81123-553-1/TP·472

印 数:1~4 000 册 定价:32.00 元

本书如有质量问题,请向北京交通大学出版社质监组反映。对您的意见和批评,我们表示欢迎和感谢。

投诉电话:010-51686043, 51686008; 传真:010-62225406; E-mail: [press@bjtu.edu.cn](mailto:press@bjtu.edu.cn)。

# 前 言

目前,国内大部分高校计算机相关专业都是采用 C 语言作为第一门程序设计课程,而 Java 语言则是作为面向对象程序设计的后续课程来讲解。尽管国内市面上出版的有关 Java 程序设计方面的教材有很多,但能反映 Java 最新发展技术的教材还较少,大都不太适合已具有一定程序基础的学生使用。

本书定位于已经有一定程序设计经验的读者,全书内容共分 10 章,以 Java SE 6 为基础,全面介绍 Java 语言的功能和技术特点,主要内容包括 Java 语言基础知识、面向对象编程技术、UML、异常、断言与日志、输入输出、泛型、集合框架、多线程、图形用户界面、网络编程等内容。

第 1 章概述了 Java 发展历史、特点和运行机制,并对 Java 语言的基本语法作了较清晰的介绍,包括数据类型、变量和常量、运算符、表达式、控制流程等。

第 2 章主要从面向对象的概念出发,描述了类和对象在 Java 语言中的实现,介绍了类的封装、对象的消息传递、方法的重载、成员的访问权限控制、包的基本概念以及 UML 类图等 Java 基本技术。

第 3 章主要讨论了面向对象程序设计的继承和多态两个重要特征在 Java 语言中的具体实现。对 Java 语言的抽象类、接口、多态以及设计模式进行了描述。

第 4 章介绍了 Java 的泛型编程技术,包括泛型类、泛型方法以及泛型中的通配符的使用等相关技术。

第 5 章讨论了 Java 中集合对象的特点和使用,包括数组的定义与操作以及集合框架的接口、实现和算法等作了详细的描述。

第 6 章描述了 Java 的异常处理机制和进行软件调试的断言及日志机制,为应用程序的出错处理提供强有力的支持。

第 7 章介绍 Java 多线程的基本概念和应用方法,包括线程实现、线程管理、线程同步以及线程组等内容。

第 8 章讲述了 Java 的输入输出的整体架构,重点介绍字节流、字符流、命令行 I/O、格式化 I/O、对象序列化、文件 I/O 以及新 I/O 的特点和具体应用。

第 9 章介绍基于 Swing 的 GUI 编程的基本思想和方法,包括常用 Swing 容器和组件、布局管理器、事件处理和多媒体处理等内容。

第 10 章讨论了 Java 的网络编程技术,主要包括 Java 网络相关类、TCP 通信、UDP 通信、使用 URL 进行网络通信的方法以及与服务器端交互等具体的应用实现。

附录为全书统一练习用的两人对弈的五子棋游戏,包括基本的功能描述和完整的可运行的示例代码。

全书第 1、8、9 章和附录由徐保民编写,第 2、3、7 章由马迪芳编写,第 4、5、6、10 章由陈旭东编写,魏小涛也参与了部分章节的编写与代码的调试。全书由马迪芳负责定稿。

作者结合多年来的教学经验和工程实践基础，力图使本教程具有如下特点：①内容新而全，以 Java SE 6 为基础来讲解 Java 语言的功能和特点；②螺旋式推进，对新概念的引入和讲解循序渐进，逐步展开，确保读者能够掌握每一阶段所讨论内容；③实例式组织，所有知识点都引入实例，便于自主学习；④强调实用性，所有实例程序都在 Java SE 6 环境下调试与运行，读者可以直接选用。

本书的出版得到了北京交通大学出版社谭文芳老师的大力支持，北京交通大学软件学院、计算机与信息技术学院相关课程的老师也为本书的编写提出了不少的建议，在此一并表示深深的谢意。特别感谢魏小涛老师、冯凤娟老师和赵宏老师对编者提供的支持和帮助。

由于作者水平所限，难免会存在很多不足和错误，恳请各位读者不吝赐教。作者联系的电子邮箱分别为：dfma@bjtu.edu.cn（马迪芳）、xubaomin@yahoo.com.cn（徐保民）和 xudong\_chen@tom.com（陈旭东）。

编 者  
2009 年 4 月

# 目 录

第 1 章 Java 语言基础 .....	1
1.1 Java 概述 .....	1
1.1.1 Java 发展历史 .....	1
1.1.2 Java 语言的特点 .....	2
1.1.3 简单的 Java 程序 .....	3
1.1.4 Java 运行平台 .....	10
1.2 Java 语言基础 .....	10
1.2.1 基本元素 .....	10
1.2.2 简单数据类型 .....	12
1.2.3 枚举类型 .....	13
1.2.4 常量与变量 .....	14
1.2.5 运算符与表达式 .....	14
1.2.6 类型转换 .....	17
1.2.7 流程控制 .....	18
习题 .....	25
第 2 章 类和对象 .....	26
2.1 类和对象的概念 .....	26
2.2 类的实现 .....	27
2.2.1 类的定义 .....	27
2.2.2 成员变量 .....	28
2.2.3 定义方法 .....	29
2.2.4 构造方法 .....	32
2.3 对象的实现 .....	33
2.3.1 创建对象 .....	33
2.3.2 对象的使用 .....	35
2.3.3 清除对象 .....	36
2.4 封装与信息隐藏 .....	37
2.5 对象间的消息传递 .....	38
2.6 Get 和 Set 方法 .....	39
2.7 方法重载 .....	40
2.8 关键字 this .....	41
2.8.1 对字段使用 this 关键字 .....	42
2.8.2 构造方法中使用 this 关键字 .....	42

2.8.3 将当前对象的引用作为参数传给其他对象	43
2.9 类成员和实例成员	44
2.9.1 类变量和实例变量	44
2.9.2 类方法和实例方法	44
2.9.3 类变量和实例变量的初始化	46
2.10 包	49
2.10.1 包的概念	49
2.10.2 创建包	49
2.10.3 引用包	50
2.10.4 包对文件的管理	51
2.11 访问权限控制	52
2.11.1 private	53
2.11.2 包访问控制	54
2.11.3 protected	55
2.11.4 public	55
2.12 嵌套类	56
2.12.1 静态嵌套类	57
2.12.2 内部类	58
2.13 Java 程序结构	62
习题	62
<b>第3章 继承和多态</b>	<b>64</b>
3.1 类的继承	64
3.1.1 继承概念	64
3.1.2 继承实现	65
3.1.3 继承的优点	66
3.1.4 成员变量的隐藏和方法重写	66
3.1.5 super 关键字	67
3.1.6 instanceof 操作符	71
3.1.7 java.lang.Object 类	71
3.1.8 final 数据、方法和类	74
3.2 抽象类和抽象方法	76
3.2.1 抽象类	76
3.2.2 抽象方法	77
3.3 接口	78
3.3.1 接口概念	78
3.3.2 定义接口	79
3.3.3 实现接口	81
3.3.4 接口类型的使用	82
3.3.5 接口的类型转换	83

3.4	多态	84
3.4.1	继承与多态	85
3.4.2	接口与多态	87
3.4.3	多态的优点	88
3.5	设计模式	89
3.5.1	单实例模式	90
3.5.2	策略模式	92
3.6	UML 类图	96
3.6.1	UML 简介	96
3.6.2	类图	97
3.6.3	类之间的关系	98
3.6.4	多重性	100
	习题	102
<b>第4章</b>	<b>泛型</b>	<b>103</b>
4.1	问题的提出	103
4.2	泛型类	108
4.2.1	定义泛型类	108
4.2.2	使用泛型类	109
4.2.3	泛型类实例	110
4.3	泛型方法	113
4.3.1	使用泛型方法	113
4.3.2	受限类型参数	114
4.3.3	使用子类型	116
4.4	通配符	117
4.4.1	使用通配符	118
4.4.2	通配符捕获	120
4.5	类型擦除	122
	习题	125
<b>第5章</b>	<b>集合</b>	<b>126</b>
5.1	数组	126
5.1.1	创建数组	126
5.1.2	访问数组元素	127
5.1.3	数组初始化	129
5.1.4	返回数组的方法	130
5.2	数组的基本操作	131
5.2.1	数组复制	131
5.2.2	数组比较	133
5.2.3	数组排序	134
5.2.4	数组查找	139



5.3	多维数组 .....	141
5.4	集合框架简介 .....	143
5.4.1	集合接口 .....	143
5.4.2	集合实现 .....	144
5.4.3	集合算法 .....	145
5.5	集合实现 .....	146
5.5.1	ArrayList 类 .....	146
5.5.2	HashSet 类 .....	148
5.5.3	HashMap 类 .....	151
5.6	集合算法 .....	152
5.6.1	排序算法 .....	153
5.6.2	查找算法 .....	154
5.7	自定义集合实现类 .....	156
	习题 .....	159
<b>第6章</b>	<b>异常处理</b> .....	<b>160</b>
6.1	异常 .....	160
6.1.1	基本概念 .....	160
6.1.2	异常分类 .....	160
6.1.3	常用标准异常 .....	161
6.2	异常处理 .....	162
6.2.1	捕获异常 .....	162
6.2.2	声明异常 .....	166
6.2.3	抛出异常 .....	167
6.2.4	异常链 .....	168
6.2.5	覆盖抛出异常的方法 .....	169
6.3	自定义异常类 .....	171
6.3.1	创建自定义异常类 .....	171
6.3.2	使用自定义异常类 .....	172
6.4	日志 .....	173
6.4.1	日志记录器 .....	173
6.4.2	使用全局日志记录器 .....	173
6.4.3	使用自定义日志记录器 .....	174
6.5	断言 .....	176
6.5.1	断言编译 .....	177
6.5.2	打开与关闭断言 .....	177
6.5.3	状态断言 .....	178
6.5.4	控制流断言 .....	179
	习题 .....	181
<b>第7章</b>	<b>线程</b> .....	<b>182</b>

7.1	线程概念 .....	182
7.2	线程的实现 .....	183
7.2.1	通过继承 Thread 类构造线程 .....	183
7.2.2	通过 Runnable 接口构造线程 .....	185
7.3	线程的生命周期 .....	188
7.4	线程优先级和线程调度 .....	190
7.5	多线程资源共享与同步 .....	196
7.5.1	多线程访问共享资源的加锁机制 .....	196
7.5.2	多线程的协作同步运行机制 .....	202
7.5.3	死锁问题 .....	210
7.6	Daemon 线程 .....	211
7.7	线程组 .....	211
	习题 .....	213
<b>第8章</b>	<b>输入输出 .....</b>	<b>215</b>
8.1	流的概念 .....	215
8.2	字节流 .....	215
8.2.1	InputStream 类 .....	216
8.2.2	OutputStream 类 .....	216
8.2.3	示例 .....	217
8.3	字符流 .....	219
8.3.1	Reader 类 .....	219
8.3.2	Writer 类 .....	220
8.3.3	示例 .....	221
8.4	装饰模式 .....	221
8.5	命令行 I/O .....	224
8.5.1	标准流 .....	224
8.5.2	控制台 .....	226
8.6	格式化 I/O .....	227
8.6.1	格式化输入 .....	227
8.6.2	格式化输出 .....	227
8.6.3	示例 .....	228
8.7	对象序列化 .....	229
8.8	文件操作 .....	230
8.8.1	File 类 .....	230
8.8.2	随机访问文件 .....	232
8.9	新 I/O .....	233
	习题 .....	235
<b>第9章</b>	<b>图形用户界面 .....</b>	<b>236</b>
9.1	Swing 概述 .....	236

9.2	Swing 容器和组件 .....	236
9.2.1	Swing 容器 .....	237
9.2.2	Swing 组件 .....	242
9.3	布局管理器 .....	262
9.3.1	BorderLayout .....	262
9.3.2	FlowLayout .....	263
9.3.3	BoxLayout .....	264
9.3.4	GridLayout .....	265
9.3.5	CardLayout .....	266
9.3.6	GridBagLayout .....	267
9.4	事件处理 .....	269
9.4.1	事件处理机制 .....	269
9.4.2	事件类 .....	270
9.4.3	事件处理示例 .....	273
9.4.4	适配器类 .....	276
9.5	多媒体 .....	277
9.5.1	颜色和字体 .....	277
9.5.2	绘图 .....	278
9.5.3	基本图形 .....	278
9.5.4	图像 .....	281
9.5.5	动画 .....	284
	习题 .....	285
<b>第 10 章</b>	<b>网络通信 .....</b>	<b>286</b>
10.1	网络基本概念 .....	286
10.1.1	TCP 协议 .....	286
10.1.2	UDP 协议 .....	286
10.1.3	IP 地址和端口 .....	287
10.2	Java 网络功能 .....	288
10.2.1	面向网络接口层的类 .....	288
10.2.2	面向网络层的类 .....	290
10.2.3	面向传输层的类 .....	291
10.2.4	面向应用层的类 .....	291
10.3	TCP 通信 .....	292
10.3.1	TCP 服务器的实现 .....	292
10.3.2	TCP 客户的实现 .....	294
10.3.3	处理多客户请求的 TCP 服务器 .....	296
10.4	UDP 通信 .....	299
10.4.1	UDP 服务器的实现 .....	300
10.4.2	客户端编程 .....	301

10.4.3 多播通信 .....	303
10.5 使用 URL .....	307
10.5.1 创建 URL 对象 .....	308
10.5.2 解析 URL .....	309
10.5.3 读取 URL 资源内容 .....	310
10.5.4 使用 URL 连接 .....	311
10.5.5 与 Servlet 交互 .....	311
习题 .....	315
<b>附录 A 双人联机五子棋游戏</b> .....	<b>316</b>
A.1 双人联机五子棋游戏实例说明 .....	316
A.2 完整的源代码 .....	317
A.2.1 配置文件 server.properties .....	317
A.2.2 ReadConfig 类 .....	317
A.2.3 FiveInARowServer 类 .....	318
A.2.4 ServerThread 类 .....	319
A.2.5 FiveInARowMain 类 .....	321
A.2.6 ClientThread 类 .....	322
A.2.7 ClientGUI 类 .....	324
A.2.8 Logic 类 .....	328

# 第 1 章 Java 语言基础

在计算机编程技术发展史上, Java 技术是最易被接受、普及最快的一种技术。目前, Java 技术以其独特的魅力征服了几乎所有的程序员。

本章首先介绍 Java 的一些背景知识, 然后重点讲解 Java 语言编程基础知识, 主要包括常量与变量、数据类型、运算符、流程控制及构成 Java 应用程序的基本结构等。

## 1.1 Java 概述

### 1.1.1 Java 发展历史

Java 是美国 Sun 公司所开发的一种程序语言。1991 年, Sun 公司成立了一个称为“绿色”的项目小组, 其主要目的是开发消费性电子产品的控制软件。由于当时所使用的 C++ 程序语言过于复杂且缺乏安全性, 所以, 那时的“绿色”项目主持人 James Gosling 博士便以 C++ 为基础, 重新开发了一套新的程序语言, 命名为“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, “绿色”项目小组将它们的开发转向了 Internet, 用 Oak 编写了一系列网络应用程序, 其中比较有名的是网络浏览器 Web Runner 等。随后“Web Runner”改名为“Hot-Java”, 并于 1995 年 5 月 23 日发表后, 在产业界引起了巨大的轰动。

1996 年, Sun 公司成立 Javasoft 分公司, 专门负责 Java 事宜。同时, 正式发表 Java 开发者版本 JDK 1.0。这个版本包括运行 Java 程序的运行环境 JRE (Java Runtime Environment) 和开发 Java 程序的开发环境 JDK (Java Development Kits) 两部分。在 JDK 1.0 时代, 除了 AWT (用于开发图形用户界面的 API) 类库外, 其他类库并不完整, 即 JDK 1.0 并不能完全适合应用程序的开发。

1997 年, Sun 公司发布了 JDK 1.1。相对于旧版本, JDK 1.1 的最大改进是推出 Java 即时编译器 (Just-In-Time, JIT)。它使得 Java 程序的执行速度有了很大的提高。

1998 年标志着 Java 2 平台诞生的 JDK 1.2 版本发布。为了方便市场推广, Sun 公司将 Java 改名为 Java 2, 同时 Sun 公司将 Java 2 平台分成 Java 企业版 J2EE (Java 2 Platform Enterprise Edition, 主要用于开发分布式及基于 Web 的应用程序)、Java 标准版 J2SE (Java 2 Platform Standard Edition, 主要目的是为台式机和 workstation 提供一个开发和运行常见的桌面应用程序的平台) 和 Java 微缩版本 J2ME (Java 2 Platform 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 的运行平台, 使得 Java 程序能够在手机、机顶盒、PDA 等产品上运行) 3 大块。这表明 Java 开始向企业、桌面应用和移动设备应用 3 大领域挺进。此时, Java 才真正成为现代开发工具中的利器。

在随后的几年内, Sun 公司于 2000 年发布了 JDK 1.3。该版本的主要改进表现在一些类

库上（如数学运算、新的 Timer API 等）、在 JNDI 接口方面增加了一些 DNS 的支持、增加了 JNI 的支持，这使得 Java 可以访问本地资源、支持 XML 以及使用新的 Hotspot 虚拟机代替了传统的 Java 虚拟机（Java Virtual Machine, JVM）。

2002 年发布 JDK 1.4。该版本主要对 Hotspot 虚拟机的锁机制进行了改进，使 JDK 1.4 的性能有了质的飞跃，同时增加了断言功能。

2004 年发布 JDK 5.0。该版本的主题是易用，它不仅增加了诸如泛型、增强的 for 语句、可变数目参数、注释、自动拆箱和装箱等功能，同时推出了 EJB 3.0 规范和 Java 服务器界面编程规范 JSF（Java Server Faces）。JSF 类似于 ASP.NET 的服务端控件，通过它可以快速地建立复杂的 JSP 界面。为了暗示该版本较以前版本有较大的改动，Sun 公司将版本号 1.5 改为 5.0。同时改名为 J2SE 5.0（Java 2 Platform Standard Edition 5.0）。

2006 年发布 JSE 6.0（不再带有数字 2）。该版本主要提供了许多实用和方便的功能如脚本、Web Service、XML、编译器 API 数据库等。JSE 6.0 是专为 Vista 而设计的，它在 Vista 上将会拥有更好的性能。

目前，Java 语言的测试版是 JSE 7.0，预计 2009 年正式发布。

## 1.1.2 Java 语言的特点

Sun 公司在“Java 白皮书”中对 Java 的定义是：Java 是一种简单的、面向对象的、分布式的、解释执行的、健壮的、安全的、结构中立的、可移植的、高效率的、多线程的和动态的语言。从此定义可以看出 Java 语言的主要特点有以下几方面。

### （1）结构简单

Java 语言的程序构成与 C 语言和 C++ 语言类似，但是 Java 语言摒弃了 C 语言和 C++ 语言的复杂、不安全特性。例如，指针的操作、内存的管理、运算符多载、多重继承。同时，Java 语言也增加一些它本身的特殊功能，例如，垃圾收集机制，让系统本身可以自行回收程序中不再使用到的资源，使得程序设计的工作更加容易。

此外，Java 语言提供了种类丰富、功能强大的类库，提高了编程效率。

### （2）面向对象

在 Java 语言中，没有采用传统的、以过程为中心的编程方法，而是采用以对象为中心，通过对象之间的调用来解决问题的编程方法。与 C++ 语言不同，Java 语言是完全面向对象的。

### （3）网络功能

Java 语言支持 Internet 应用的开发。针对 TCP/IP 协议的不同层次，Java 提供有不同的类库。利用这些类库，设计者可以轻松地开发网络程序。

### （4）跨平台

既然 Java 语言被设计成网络程序的开发语言，Java 程序必须能运行于不同类型的 CPU 和操作系统中。Java 的编译程序将 Java 源程序编译成架构中立的字节码，而不同机器上的 Java 虚拟机均能运行该字节码格式文件。

### （5）健壮性

因为 Java 最初的设计目的是应用于电子类家庭消费产品，所以它对程序的健壮性要求

较高，并期望利用 Java 开发的应用程序在各种情形下都能稳定地执行。为此，Java 采取了一系列的技术措施。比如，Java 程序在编译时，编译程序就会严格地检查程序代码，并尝试找出程序执行时，可能产生的错误；Java 语言提供了异常处理机制，有效地避免了因程序编写错误而导致的死机现象。

#### (6) 安全性

Java 被设计的本意是希望能用于分布式网络的开发环境中，因此它对安全性有很高的要求。如果没有安全保证，用户运行从网络下载的 Java 语言应用程序是十分危险的。例如，在编译期间，Java 编译器并不分配内存，而是推迟到运行时由解释器决定，这样编程人员就无法通过指针访问内存；在运行期间，Java 的运行环境提供有字节码校验器、类装载器、运行时内存布局和文件的访问限制等安全保障机制。

#### (7) 多线程

多线程是指在一个程序中可以同时执行多个简单任务。C 语言和 C++ 语言采用单线程体系结构，而 Java 语言支持多线程技术。因此利用 Java 语言可以很容易地使应用程序达成同时执行多项工作的任务。

#### (8) 高效率

与常见的解释执行语言比如 Python、TCL 不同，Java 语言既是一种编译型的语言，同时也是一种解释型的语言。编译就是把 Java 源程序转化为 .class 文件。解释指的是 Java 虚拟机通过解释 .class 文件完成对 .class 文件的执行；Java 的字节码格式是针对机器码的转换设计的，因此，从字节码到机器码的转换非常简便和简单。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。随着对 Java 虚拟机的改进和 Java 即时编译技术的出现使得 Java 的执行速度有了更大的提高。

#### (9) 可移植

Java 开发的程序是可移植的。跨平台是确保程序可移植的必要条件，此外还需很多其他条件的配合。比如 Java 严格规定了其简单数据类型的长度；Java 编译器自身是用 Java 语言实现等。

#### (10) 动态性

Java 语言的设计目标之一是适应于动态变化的环境。这主要表现在 Java 程序所需要的类能动态地被载入到运行环境，也可以通过网络来载入所需要的类；Java 中的类有一个运行时刻的表示，能进行运行时刻的类型检查；Java 类库中可以自由地加入新的方法和数据成员而不会影响用户程序的执行等。

### 1.1.3 简单的 Java 程序

Java 的用户程序分为 Java 应用程序 (Java Application) 和 Java 小程序 (Java Applet) 两大类。这两类程序在组成结构和执行机制上都存在一定的差异。

#### 1. 第一个 Java 应用程序

像使用 C、C++ 等编程语言编写的程序一样，Java 应用程序是一个独立的由 Java 解释器来解释运行的程序。编写 Java 应用程序的步骤如下：

- ① 利用某一种文本编辑器建立 Java 源程序文件，扩展名为 java；

- ② 利用 Java 编译器 javac 产生 .class 字节码文件;
- ③ 利用 Java 解释器解释字节码文件, 完成该程序的运行过程。

下面以最简单的“Hello, World!”程序为例, 说明 Java 应用程序的建立及运行步骤。

#### (1) 建立 Java 源程序文件

要建立一个 Java 程序, 首先要创建 Java 程序的源代码, 即一个包含有符合 Java 规范的各种语句构成的文本文件。Java 源程序的编写可以使用任何一种文本编辑器, 比如 Ultra-Edit、Notepad 等, 然后只要把编辑好的文件存成 .java 文件即可。当然也可以用集成开发环境比如 Eclipse、NetBeans 等编写 Java 源程序。

例 1-1 是本书的第一个 Java 应用程序的源程序。打开记事本, 在记事本中输入例 1-1 所示文本内容, 完成后将其保存于目录 D:\java\_book\chap01 下, 并命名为 HelloWorld.java。该程序的功能是在屏幕上显示“Hello, World!”字符串信息。

#### 例 1-1 第一个 Java 应用程序

文件名: HelloWorld.java

```
/* 本程序功能是在屏幕上输出"Hello, World!" */
public class HelloWorld {
    /**
     * 方法 main() 是 Java 应用序的唯一入口
     * @param args 输入参数
     * @exception 没有异常抛出
     */
    public static void main( String[] args ) {
        System.out.println( "Hello, World!" ); //输出"Hello, World!"
    }
}
```

在编辑程序时必须注意区分大小写, Java 语言是严格区分大小写的; 空白符只能是半角空格符或是 Tab 字符。

#### (2) 编译源程序

在 Windows XP 的“开始”菜单中执行“运行”命令, 在如图 1-1 所示的运行对话框中键入“cmd”命令, 可进入命令行界面。

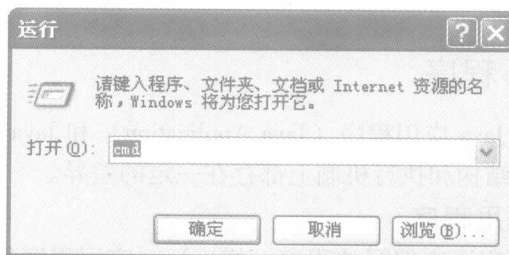


图 1-1 “运行”命令对话框



使用命令“cd”，进入待编译文件所在目录 D:\java\_book\chap01。在 DOS 命令提示符下输入如下命令，并按回车键（键盘上的“Enter”键），如图 1-2 所示。

```
D:\java_book\chap01 > javac HelloWorld.java
```

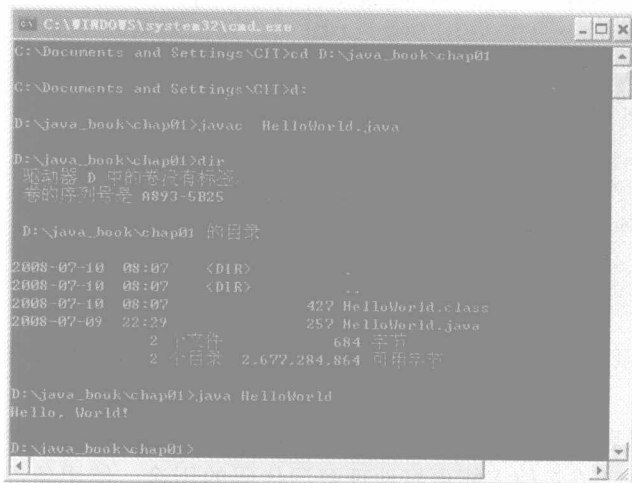


图 1-2 命令行界面

如果命令提示符窗口没有出现任何提示信息，则说明 Java 源程序已经编译成功，并在当前目录 D:\java\_book\chap01 下产生一个扩展名为 class 的文件，该文件即为 Java 源程序编译后所产生的 HelloWorld.class 文件。由于.class 文件不是可以直接运行的机器代码，因此它不能直接在操作系统环境下执行。但是它可以在 Java 虚拟机内执行。

命令 javac 是 Sun 公司发布的 JDK 中所提供的 Java 编译器，它的作用是将 Java 源程序编译成 Java 虚拟机能够解释执行的.class 文件。

### (3) 运行 Java 应用程序

在 DOS 命令提示符下输入如下命令，并按回车键。

```
D:\java_book\chap01 > java HelloWorld
```

如果程序没有错误的话，屏幕上则显示如下信息。

```
Hello, World!
```

命令 java 是 Sun 公司发布的 JDK 中所提供的 Java 解释器，其作用是调用 Java 虚拟机将.class 文件解释成它所在计算机能识别的二进制文件，并加以执行。

### (4) “Hello World” 应用程序分析

通过例 1-1 可以看出，Java 应用程序的程序结构并不复杂。其基本结构是：

```
public class 类名 {
    public static void main ( String [ ] args ) { //main 方法
        ..... //各种功能语句
    }
}
```