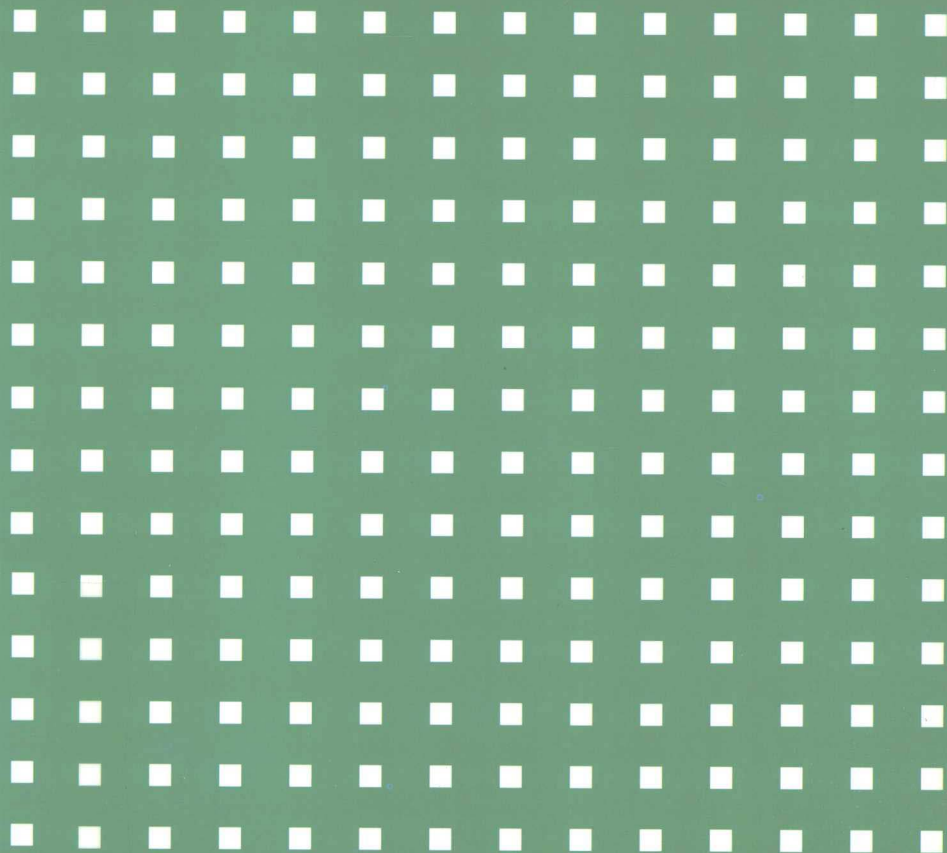


Java应用案例分析

姜国海 赵小薇 马瑞新 编著



高等学校计算机专业教材精选·算法与程序设计

Java应用案例分析

姜国海 赵小微 马瑞新 编著

清华大学出版社
北京

内 容 简 介

本书将 Java 语言的语法知识和项目开发所需的基础知识融合在一起,全面系统地介绍了 Java 语言、Java 程序设计方法以及在 Java SE 平台和 Java EE 平台下开发应用程序的方法和技术。全书由 11 章组成,提供了较为丰富的 Java 应用案例,以加深学生对语法知识的理解,同时对于在实际项目开发过程中采用的数据库编程技术、XML 编程技术以及软件设计的相关知识也给出了基础的案例。同时,本书不仅提供了有关 Hibernate 和 MyBatis 等技术的案例以说明其使用过程,还对其实现原理进行了说明。

本书既可以作为高等学校相关专业 Java 程序设计课程的教材,也可以作为 Java 自学者的学习参考用书。

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 13701121933

图书在版编目(CIP)数据

Java 应用案例分析/姜国海,赵小薇等编著. —北京:清华大学出版社,2012.11

高等学校计算机专业教材精选·算法与程序设计

ISBN 978-7-302-28947-0

I. ①J… II. ①姜… ②赵… III. ①JAVA 语言—程序设计—高等学校—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2012)第 110883 号

责任编辑:白立军 战晓雷

封面设计:傅瑞学

责任校对:时翠兰

责任印制:何 芊

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社 总 机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课件下载: <http://www.tup.com.cn>, 010-62795954

印 刷 者:北京市人民文学印刷厂

装 订 者:三河市溧源装订厂

经 销:全国新华书店

开 本:185mm×260mm 印 张:24

字 数:588 千字

版 次:2012 年 11 月第 1 版

印 次:2012 年 11 月第 1 次印刷

印 数:1~2500

定 价:39.00 元

产品编号:043648-01

前 言

随着 Java 技术的流行,很多大学都开设了 Java 程序设计相关课程。由于受学时限制,很多 Java 程序设计课程以 Java 语言的语法为主,结合少量的例子程序,在例子程序中介绍一些 Java SE 平台的类库使用说明,在实用性方面有所欠缺,本教材希望在 Java 实用性教学方面能够弥补以往同类教材的一些不足,在 Java 语法的基础上,多提供一些应用案例,结合开发工具、设计模式、框架、软件设计和体系结构等相关知识,提高学习者的编程能力。

本书作者曾在软件企业从事过多年的软件开发,目前在大连理工大学软件学院讲授 Java 课程,具有多年的教学实践经验,对其他兄弟院校的 Java 教学也有所了解。我们针对大学教育的特点,既不能脱离基础教学,也不能脱离理论教学,同时还要满足软件企业对学动手能力的要求,编写了这本教材,教材内容采用语法知识和应用开发基础相结合的方法,由于 Java 语言的语法非常简单,所以仅仅安排两章的内容来介绍语法知识。在第 3 章中介绍简单的数据结构和算法方面的内容,建议读者在此基础之上学习其他数据结构的编程知识,例如使用赫夫曼编码来压缩一个文本文件。在大致掌握了 Java 语法的基础上,还需要学习 Java SE 平台的类库并扎实掌握 io 包和 util 包中的类库,利用集合框架和文件操作可以完成很多有意义的程序了,Java 语言的优点之一是网络编程能力很强,在本书中介绍了网络编程和多线程编程,读者可以自行设计一些网络应用程序,例如聊天程序、语音通信程序、远程控制程序、文件服务器等程序。本书未介绍 UDP 编程部分,读者可以自行学习,UDP 编程也是网络服务编程中很重要的一部分。Java 对 XML 技术支持得非常好,有很多类库提供对 XML 编程的支援,很多 Java 开源项目提供了各种各样的 XML 数据的处理方式,它们带给我们更多的编程选择。本书另外的重点就是 Java EE 平台编程,实际上是 Java EE 平台下的 Web 应用开发部分占了很大的比重,EJB 技术在本书中并没有介绍,实际开发中的各种框架技术也没有介绍,例如 STRUTS、SPRING 等框架,希望读者在本书的基础上去掌握几种框架技术。

作者本人在大学开设 Java 应用开发这门课程,总学时数是 96 学时,其中讲课学时为 48 学时,上机学时为 48 学时,每学时 45 分钟。上机内容单独布置,在书中部分例子程序的基础上增加了更加实用的内容。

本书的编写离不开大连理工大学软件学院的郭禾教授、周勇教授的指导和支持,正是他们的教学改革建议促使我们改革以前的 Java 程序设计课程,努力地提高学生们的实践能力。以前的 Java 程序设计课程受学时数限制,课程内容安排过少、过于简单,在他们的建议下,我们修改了教学大纲,安排了更多实用性很强的编程内容,在上机环节中强化了开发工具的使用能力。同时,还要感谢软件学院的龙秀敏老师、韩涛老师,她们也积极参与了教学改革,编写了教改调查问卷,使得 Java 教学方法改革获得成功,也要感谢软件学院 09 级本

科生,他们非常积极地参与到本课程的教学活动中来,并对本课程的教学也提出了一些意见和建议。当然,本书也离不开作者的家人的支持和关心。感谢所有提供帮助的人。

感谢您选择本书,欢迎您对本书内容提出意见和建议。作者的电子邮箱是 javateacher@163.com。

目 录

第 1 章 第一个 Java 程序	1
1.1 Java 语言简介	1
1.2 Java 发展历史	1
1.3 历史版本	2
1.4 主要特性	3
1.5 搭建开发环境	4
1.5.1 安装 JDK	4
1.5.2 开发第一个程序	7
1.5.3 使用集成开发环境	7
1.5.4 安装 JDK 文档	10
第 2 章 Java 语法基础	12
2.1 Java 语言数据类型	12
2.1.1 基本数据类型	12
2.1.2 值常量	13
2.1.3 对象数据类型	14
2.1.4 数组类型	16
2.2 标识符和关键字	17
2.3 变量定义	18
2.4 方法及参数传递	19
2.5 表达式和操作符	23
2.6 常用语句	25
2.6.1 if 语句	25
2.6.2 switch 语句	26
2.6.3 循环语句	28
2.6.4 break 和 continue 语句	30
第 3 章 数据结构和算法示例	32
3.1 单链表	32
3.2 排序和查找	37
第 4 章 面向对象编程	41
4.1 面向对象的基本概念	41
4.1.1 什么是对象和类	41

4.1.2	类的抽象	42
4.1.3	类的封装	43
4.1.4	静态成员	43
4.2	类的继承	44
4.3	包及归档文件	45
4.4	成员的访问控制	49
4.5	对象的初始化和回收	51
4.5.1	垃圾对象和对象的创建	51
4.5.2	构造方法	52
4.5.3	初始化代码	54
4.5.4	属性的读写方法	54
4.6	异常处理	56
4.6.1	错误的定义及分类	56
4.6.2	两种异常处理方法	58
4.6.3	常见的异常处理	61
4.7	抽象类及接口	64
4.7.1	抽象类	64
4.7.2	接口	65
4.8	多态	67
4.8.1	重载	67
4.8.2	覆盖	68
4.9	final 修饰符	71
4.10	内部类	72
4.11	强制转换和提升	75
第 5 章	XML 编程	80
5.1	集合编程	80
5.1.1	对象的关系操作	80
5.1.2	包装类	82
5.1.3	字符串类	83
5.1.4	集合框架	86
5.1.5	泛型支持	91
5.2	文件和流	94
5.2.1	随机访问文件	94
5.2.2	文件流	98
5.2.3	流链接和文本文件处理	101
5.2.4	其他流	106
5.3	XML 解析	107
5.3.1	XML 格式控制	111

5.3.2	XPATH	115
5.3.3	XML 解析	118
第 6 章	多线程编程及网络应用	125
6.1	多线程编程	125
6.1.1	线程的创建	125
6.1.2	线程的控制	127
6.1.3	多线程的同步	128
6.1.4	阻塞机制	132
6.2	网络编程	133
6.2.1	Socket 及 ServerSocket	134
6.2.2	简单的网络计算器	136
6.2.3	线程池	139
6.3	RMI 编程	141
第 7 章	GUI 编程	146
7.1	事件处理模型	146
7.2	AWT 编程	148
7.3	布局管理器	151
7.4	组件绘图	155
7.5	MVC 模式	158
7.6	Swing 编程	161
7.6.1	Swing 容器类	164
7.6.2	Swing 组件类	170
7.7	Swing 综合案例	182
第 8 章	数据库编程	205
8.1	JDBC 接口	206
8.2	JDBC 编程	209
8.3	数据源	219
8.4	ORM 工具	225
第 9 章	综合数据处理及图表案例	243
9.1	数据加载过程	244
9.1.1	加载文本文件	245
9.1.2	加载 Excel 文件	246
9.1.3	加载 XML 文件	248
9.1.4	数据库读取方式	257
9.1.5	网络服务器读取方式	259

9.2	数据处理过程	263
第 10 章	Web 应用编程	273
10.1	Java EE 平台简介	273
10.2	第一个 Web 应用	277
10.3	Servlet 组件编程	285
10.3.1	Servlet 组件规范	285
10.3.2	Java Annotation	293
10.3.3	Session 编程	297
10.3.4	Filter 组件编程	306
10.3.5	Listener 组件编程	313
10.4	JSP 编程	315
10.4.1	JSP 基本概念	315
10.4.2	JSP 语法简介	318
10.4.3	JSP 预置隐含对象	321
10.4.4	JSP 实现猜数游戏	322
10.4.5	JSP EL 表达式语言	324
10.5	Tag 编程	325
10.5.1	标记处理器类	326
10.5.2	标记库描述符	327
10.5.3	JSP 和标记处理器通信	329
10.5.4	迭代标记和标记体处理	337
10.5.5	简单标记	342
10.5.6	标记文件	344
10.5.7	DisplayTag 表格标记	348
10.5.8	国际化问题	351
第 11 章	毕业论文格式检查案例	355
11.1	JavaMail 编程	356
11.1.1	邮件注册激活	360
11.1.2	接收论文邮件	369
11.2	论文上传	373

第 1 章 第一个 Java 程序

1.1 Java 语言简介

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,是由 Sun Microsystems 公司于 1995 年 5 月推出的一种编程语言,Java 不仅是一种编程语言,它还是一个开发和运行平台。Java 平台按照规模和目标定位分为 Java Standard Edition(Java SE)、Java Micro Edition(Java ME)和 Java Enterprise Edition(Java EE)。Java 技术具有卓越的通用性、高效性、平台移植性和安全性,广泛应用于 PC、数据中心、游戏控制台、科学超级计算机、移动电话和互联网,同时拥有全球最大的开发者专业群体。在全球云计算和移动互联网的产业环境下,Java 更具备了显著优势和广阔前景。这些优势都是由其跨平台移植性和简单易用的特点决定的。Java 语言在语法上和 C++ 语言类似,在语义上类似于 SmallTalk 语言。

1.2 Java 发展历史

Java 平台和语言最开始只是 Sun 公司在 1991 年开始研究的一个内部项目,当时的项目目标是开发消费类电子产品市场,研发小组的名称为 Green,目标之一就是解决诸如电视机、电话、闹钟、烤面包机等家用电器的控制和通信问题。项目最初采用的开发语言是 C 语言,后来由于 C 语言编译器复杂,C 程序健壮性不高,C 语言的内存管理不方便等诸多原因放弃了 C 语言。项目组成员之一,James Gosling(Java 之父,出生于加拿大,在卡内基·梅隆大学攻读计算机博士学位时编写了多处理器版本的 UNIX 操作系统),决定开发一个新的计算机语言,该语言被他命名为 Oak,以他的办公室外的橡树来命名。但 Oak 是另外一个注册公司的名字,不可能再用了。在命名征集会上,大家提出了很多名字。最后按大家的评选次序,将十几个名字排列成表,上报给商标律师。排在第一位的是 Silk(丝绸)。尽管大家都喜欢这个名字,但遭到 James Gosling 的坚决反对。排在第二和第三的都没有通过律师这一关。只有排在第四位的名字得到了所有人的认可和律师的通过,这个名字就是 Java。Java 是印度尼西亚爪哇岛的英文名称,该岛因盛产咖啡而闻名。Java 语言中的许多类库名称与咖啡有关,如 JavaBeans(咖啡豆)、NetBeans(网络豆)以及 ObjectBeans(对象豆)等。Java 的标识也正是一杯正冒着热气的咖啡。10 多年来,Java 就像爪哇咖啡一样誉满全球,成为实至名归的企业级应用平台的霸主。

就像很多开发新技术的秘密工程一样,Green 小组没日没夜地工作,到了 1992 年的夏天,他们能够演示新平台的一部分了,包括 Green 操作系统、Oak 程序设计语言、类库和硬件。最初的尝试是一种类 PDA 设备,被命名为 Star7,这种设备有鲜艳的图形界面和被称为 Duke 的智能代理来帮助用户。1992 年 12 月 3 日,这台设备进行了展示。同年 11 月,Green 计划被转化成了 FirstPerson 有限公司——Sun 公司的一个全资子公司,团队也被重

新安排到了帕洛阿尔托。FirstPerson 公司对建造一种高度互动的设备感兴趣,当时代华纳公司发布了一个关于电视机顶盒的征求意见书时(Request for proposal),FirstPerson 公司改变了他们的目标,作为对征求意见书的响应,提出了一个机顶盒平台的提议。但是有线电视业界觉得 FirstPerson 公司的平台给予用户过多的控制权,因此 FirstPerson 公司的投标败给了 SGI 公司。与 3DO 公司的另外一笔关于机顶盒的交易也没有成功,由于他们的平台不能在电视工业产生任何效益,FirstPerson 公司再并回 Sun 公司。随后 Sun 放弃了该项计划。就在 Oak 几近失败之时,在 1993 年和 1994 年随着互联网的发展,Sun 公司看到了 Oak 在计算机网络上的广阔应用前景,于是改造了 Oak,以 Java 的名称正式发布。同期,以 Java 语言开发的浏览器软件 HotJava 也面世了,尽管它比 Mosaic 浏览器和 Netscape 浏览器在超文本解析方面差很多,但是其跨平台特性和安全性受到业界的很大关注,软件开发商们也普遍看好 Java。可以这么说,Java 得益于互联网的发展,也推动了互联网的发展。

1.3 历史版本

1995 年 5 月 23 日,Java 语言诞生。

1996 年 1 月,第一个 JDK——JDK 1.0 诞生。

1996 年 4 月,10 个最主要的操作系统供应商声明将在其产品中嵌入 Java 技术。

1996 年 9 月,约 8.3 万个网页应用了 Java 技术来制作。

1997 年 2 月 18 日,JDK 1.1 发布。在这个平台上诞生了许多商业软件。

1997 年 4 月 2 日,JavaOne 会议召开,与会者逾一万人,创当时全球同类会议规模的纪录。

1997 年 9 月,JavaDeveloperConnection 社区成员超过十万人。

1998 年 2 月,JDK 1.1 被下载超过两百万次。

1998 年 12 月 8 日,Java 2 企业平台 J2EE 发布。

1999 年 6 月,Sun 公司发布 Java 的 3 个版本:标准版(J2SE)、企业版(J2EE)和微型版(J2ME)。

2000 年 5 月 8 日,JDK 1.3 发布。

2001 年 6 月 5 日,Nokia 宣布,到 2003 年将出售 1 亿部支持 Java 的手机。

2001 年 9 月 24 日,J2EE 1.3 发布。

2002 年 2 月 26 日,JDK 1.4 发布,J2SE 1.4 发布,自此 Java 的计算能力有了大幅提升。

2004 年 9 月 30 日,J2SE 1.5 发布,成为 Java 语言发展史上的又一里程碑。为了表示该版本的重要性,J2SE 1.5 更名为 Java SE 5.0。

2005 年 6 月,JavaOne 大会召开,Sun 公司公开 Java SE 6。此时,Java 的各种版本已经更名,以取消其中的数字 2,J2EE 更名为 Java EE,J2SE 更名为 Java SE,J2ME 更名为 Java ME。

2006 年 12 月,Sun 公司发布 JRE 6.0。

2009 年 4 月 20 日,甲骨文(Oracle)公司以 74 亿美元收购 Sun,取得了 Java 的版权。

2011 年 7 月,甲骨文公司发布 Java 7 的正式版。

1.4 主要特性

1. 简洁性

Java 语言的语法与 C 语言和 C++ 语言很接近,使得大多数程序员很容易学习和使用 Java。另一方面,Java 丢弃了 C++ 中很少使用的、很难理解的、令人迷惑的那些特性,如操作符重载、多继承和自动的强制类型转换。特别地,Java 语言不使用指针,并提供了自动的垃圾内存收集,使得程序员不必为内存管理而担忧。

2. 面向对象

Java 语言提供类、接口和继承等原语。为了简单起见,只支持类之间的单一继承,但支持接口之间的多继承,并支持类与接口之间的实现机制(关键字为 implements)。Java 语言全面支持动态绑定,而 C++ 语言只对虚函数使用动态绑定。总之,Java 语言是一个纯粹的面向对象程序设计语言。

3. 分布式

Java 语言支持 Internet 应用的开发,在基本的 Java 应用编程接口中有一个网络应用编程接口(java.net),它提供了用于网络应用编程的类库,包括 URL、URLConnection、Socket 和 ServerSocket 等。Java 的 RMI(远程方法激活)机制也是开发分布式应用的重要手段。它还提供对 CORBA 技术的实现。

4. 健壮性

Java 语言的强类型机制、异常处理和垃圾对象的自动收集等是 Java 程序健壮性的重要保证。对指针的丢弃是 Java 的明智选择。Java 的安全检查机制使得 Java 程序更具健壮性。

5. 安全性

Java 通常被用在网络环境中,为此,Java 提供了一个安全机制以防恶意代码的攻击。除了 Java 语言具有的许多安全特性以外,Java 对通过网络下载类具有一个安全防范机制(ClassLoader 类),如分配不同的名字空间以防替代本地的同名类,以及字节代码检查,并提供安全管理机制(SecurityManager 类)让 Java 应用设置安全哨兵。

6. 体系结构中立

Java 程序(后缀为 java 的文件)在 Java 平台上被编译为体系结构中立的字节码格式(后缀为 class 的文件),然后可以在实现 Java 平台的任何系统中运行。这种途径适合于异构的网络环境和软件的分发。

7. 可移植性

这种可移植性来源于体系结构中立性,另外,Java 还严格规定了各个基本数据类型的长度。Java 系统本身 also 具有很强的可移植性,Java 编译器是用 Java 实现的,Java 的运行环境是用 ANSI C 实现的。使用 Java 语言编写的程序可以实现“编译一次,随处运行”。

8. 解释型

如前所述,Java 程序在 Java 平台上被编译为字节码格式(byte code),然后可以在实现这个 Java 平台的任何系统中运行。在运行时,Java 平台中的 Java 解释器对这些字节码进行解释执行,执行过程中需要的类在连接阶段被载入到运行环境中。

9. 高性能

与那些解释型的高级脚本语言相比,Java 的确是高性能的。事实上,Java 的运行速度随着 JIT(Just-In-Time)编译器技术的发展越来越接近于 C++。

10. 多线程

在 Java 语言中支持多个线程的同时执行,并提供多线程之间的同步机制。通过多线程技术,Java 程序可以实现“多任务”并发执行,其多线程编程实现简单且功能强大。

11. 动态性

Java 语言的设计目标之一是适应动态变化的环境。Java 程序需要的类能够动态地载入运行环境,也可以通过网络来载入所需要的类,这也有利于软件的升级。另外,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。

12. 多平台支持

Java 不仅仅是编程语言,它还是一个开发和运行平台,针对不同的应用目标,Java 提供不同的平台,有针对计算平台规模比较小的软件应用,有针对桌面计算机平台的应用,也有针对分布式计算平台的企业应用。使用 Java 语言可以开发手机应用和桌面应用,也可以开发大规模的企业应用。

1.5 搭建开发环境

1.5.1 安装 JDK

只要在计算机中有 JRE(Java Runtime Environment, Java 运行环境,包括 Java 虚拟机和平台 API 类库),就可以运行 Java 程序。但是要想开发 Java 程序,还需要 Java 编译器和打包程序等诸多工具,这些都包括在 Java 程序开发包(JDK)中。

进入网址 <http://www.oracle.com/technetwork/java/javase/downloads/index.html>, 选择 Java SE 7 的 JDK Download 进行 JDK7 的下载,进入图 1-1 所示的界面,单击 Accept License Agreement 单选按钮接受许可协议(Oracle 二进制文件许可协议)。选择对应操作系统的 JDK 文件进行下载。例如,安装 Windows x64 位系统下的 jdk-7-windows-x64.exe (假设使用的操作系统是 Windows x64 位系统)。

下载完毕后,运行该程序,进入安装向导,单击“下一步”按钮,选中所有功能(包括演示程序及示例),再单击“下一步”按钮,就开始安装 JDK 了。如果要更改安装路径,单击“更改”按钮,把安装路径更改为 D:\Java\jdk1.7.0\,然后单击“下一步”按钮进行安装,如图 1-2 所示。最后会询问是否安装本机独立的 JRE 运行环境,它为浏览器提供运行 Java 小程序(Applet)的运行环境,这个可以暂不考虑。不同版本的 JDK 安装可能会不一样,安装完 JDK 后可能有的版本会提示是否安装其他开发工具,可同样暂时不考虑安装这些工具。

安装完毕后需要配置 JDK,选择“控制面板”→“系统”→“高级系统设置”→“环境变量”,新建一个环境变量,名称为 JAVA_HOME,值为 D:\Java\jdk1.7.0,这是配置 JDK 安装路径。修改 PATH 环境变量的值,在原有的值前面加上 D:\Java\jdk1.7.0\bin; 请注意,Windows 操作系统下路径分隔符是分号,而 Linux 系统下路径分隔符是冒号,配置界面如图 1-3 所示。如果 JDK 安装在其他路径,把上面环境变量的值相应修改即可。至于 CLASSPATH 环境变量并不建议进行配置。

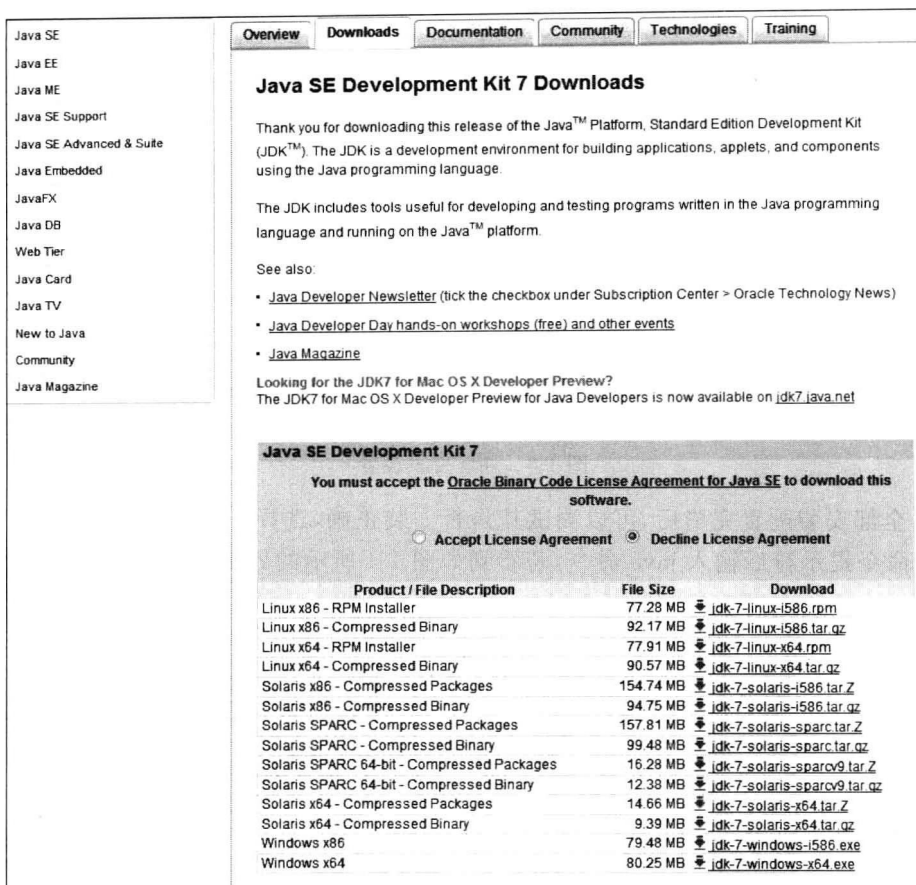


图 1-1 JDK 下载页面

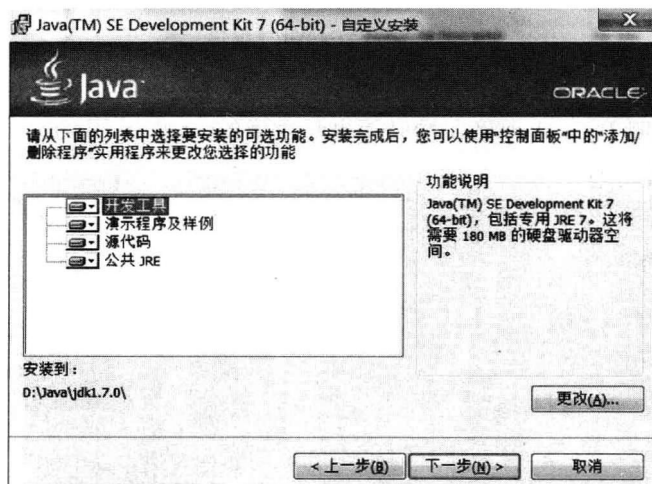


图 1-2 JDK 安装界面

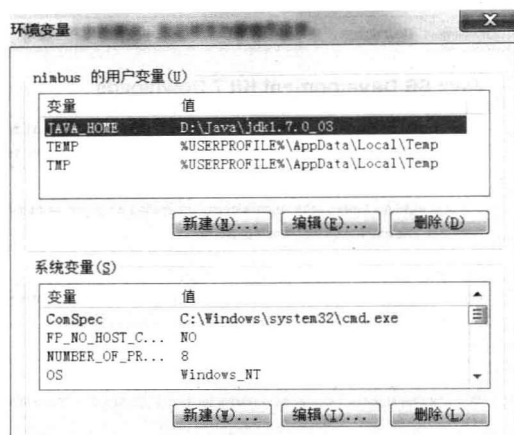


图 1-3 环境变量设置

JDK 全部安装配置完毕后,可以测试其是否安装正确,选择“开始”→“附件”→“命令提示符”,在命令提示符后输入 java 命令,若看到如图 1-4 所示的界面,表明 JDK 安装成功。

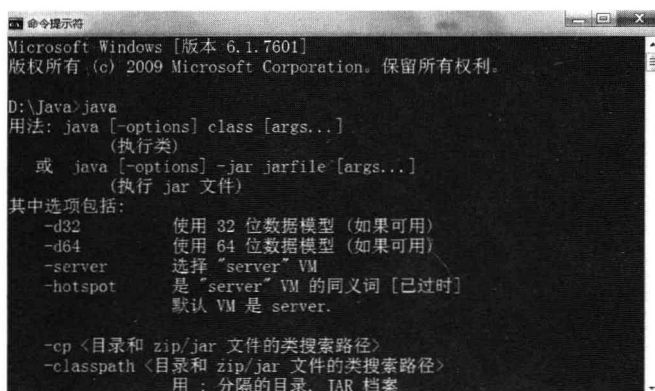


图 1-4 运行 Java 虚拟机

然后运行 javac 命令执行 Java 编译器程序,检查编译程序是否能够正常运行,运行结果如图 1-5 所示,这表明 PATH 环境变量设置也是正确的。

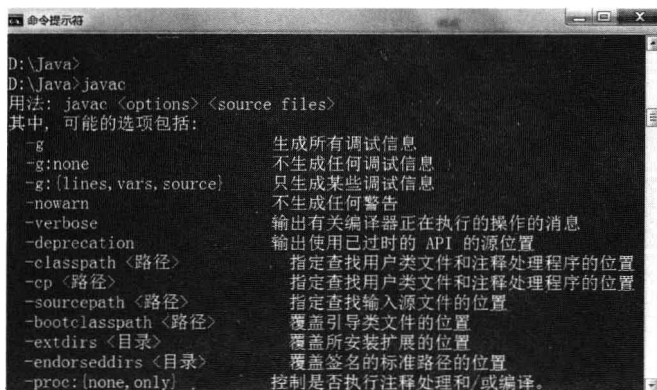


图 1-5 执行 Java 编译程序

1.5.2 开发第一个程序

JDK 安装完毕后,可以编写 Java 源程序了,可以选择一个较好的编辑器,例如 UltraEdit、Editplus 或 Notepad++ 等。本书编写第一个 Java 程序使用的是免费开源的 Notepad++ 5.9.3,打开 Notepad++ 程序后,选择“文件”→“新建”,输入如下代码:

```
public class HelloWorld {  
    public static void main(String[] args) {  
        System.out.println("Hello World!");  
    }  
}
```

即使不安装上述编辑器,使用 Windows 自带的记事本程序也可以进行文件编辑,只要注意文件名一定是 HelloWorld.java 就可以了,例如保存为 d:\Java\HelloWorld.java,文件名要注意大小写。进入命令提示符窗口,更改盘符,输入 d:,然后输入 cd Java 命令,进入 d:\Java 目录,输入命令 javac HelloWorld.java,如果没有输出任何提示,则编译成功,生成了 Java 的可执行程序文件 HelloWorld.class,如图 1-6 所示。如果生成可执行文件有错误,则有可能是文件内容输入错误。Java 语言区分大小写字符,这一点请注意。

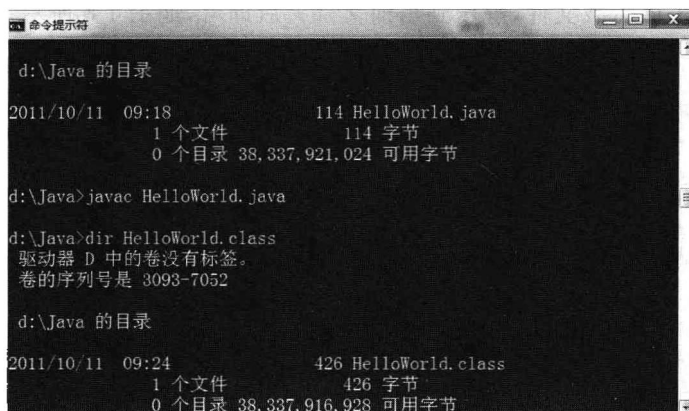


图 1-6 编译 HelloWorld 程序

最后,运行生成的 Java 程序,输入 java HelloWorld,启动 Java 虚拟机,加载 Java 类,运行该 Java 应用程序。注意,命令行执行的是 java HelloWorld,而不是 java HelloWorld.class。输出结果如图 1-7 所示。

如果需要修改程序,例如将输出的字符串修改为“你好,程序员!”,则应在编辑器中修改,然后保存文件,重新编译,再次运行。这个过程通过键盘命令来完成显得很麻烦,下面介绍更好的开发方式。

1.5.3 使用集成开发环境

当前的计算机使用环境早已图形化了,文本命令的不便利被鼠标的图形操作代替了,可以将这些命令“图形化”,使用一个带有图形用户界面的程序来进行代码编辑、编译和运行程序,会使开发过程简单直观。这个图形化的开发环境还集成了调试和发布等很多功能,因此



图 1-7 执行 HelloWorld 程序

被称为集成开发环境 (IDE)，现在有很多集成开发环境可供 Java 程序员选择，有开源的 Eclipse 工具，有 Oracle 免费提供的 NetBeans 工具，也有 JBuilder、Rational Application Developer 等商业软件，本书采用的是 Eclipse 工具，可以从网址 <http://www.eclipse.org/downloads> 下载。选择相应的平台版本下载并解压缩后就可以使用了，如果安装了 WinRAR 软件，右击该下载文件，解压到当前文件夹后即可，如图 1-8 所示。

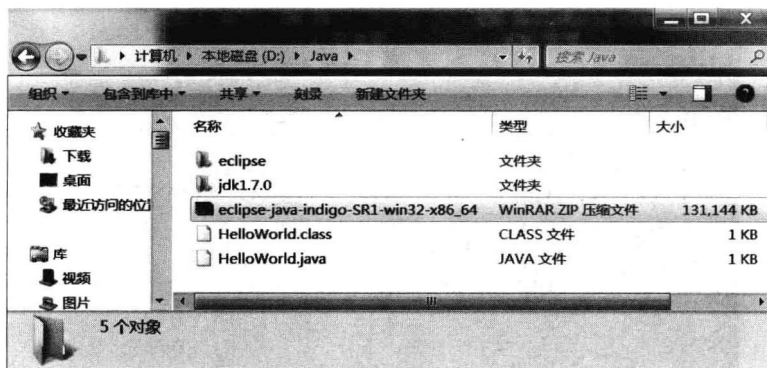


图 1-8 安装 Eclipse

进入 eclipse 目录，执行 eclipse.exe 程序。第一次进入 Eclipse，它会询问将工作空间放在哪个目录下，如图 1-9 所示，以后开发的 Java 项目文件都放在这个工作空间目录下。工作空间 (workspace) 是对项目 (project) 的目录组织。当然，以后也可以更改工作空间目录，在提示窗口的下面还有一个复选框，用来设置把这个工作空间目录作为默认工作空间目录并且以后不再提示。选中这个按钮，单击 OK 按钮确认后即可以进入 eclipse 欢迎窗口页面了。把这个视图关掉后，可以看到主窗口，如图 1-10 所示。如果感觉这个主窗口比较凌乱，可以关闭 Task List 等视图，还可以拖动 Outline 视图到窗口左侧下方，看到向下的箭头后，放置这个视图。然后选择 Window → Show View 命令，选择打开 Navigator、Type Hierarchy 和 Console 视图，这些视图可以随意调整位置。调整好视图位置后的开发环境界面如图 1-11 所示。