

21世纪高等学校计算机**基础**实用规划教材

Java程序设计**之** 实验及课程设计教程



李芝兴 主 编
刘 骥 副主编
朱庆生 主 审

清华大学出版社



21世纪高等学校计算机**基础**实用规划教材

Java程序设计**②** 实验及课程设计教程

李芝兴 主 编
刘 骥 副主编
朱庆生 主 审

清华大学出版社
北京

内 容 简 介

本书是《Java 程序设计之网络编程(第2版)》(李芝兴、杨瑞龙主编,清华大学出版社2009年出版,ISBN 978-7-302-20254-7)的配套习题解答与实验指导用书,同时也是“Java 课程设计”的指导用书。

全书分为5部分:第1部分是Java语言集成环境介绍,包括实验教学大纲和课程设计教学大纲;第2部分是配套主教材课后的部分习题解答;第3部分是Java课程设计的实验指导;第4部分是两个桌面应用案例;第5部分是一个关于Web的案例。这些内容对于巩固和深入理解教材的内容,提高独立思考、独立分析问题的能力,是十分有益的。

本书系统独立,可以作为教材的配套实验教材,也可作为各类高等院校、高职高专计算机类专业和各种培训班的辅助教材,还可供广大科技人员和Java技术爱好者参考。

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 程序设计之实验及课程设计教程/李芝兴主编. —北京:清华大学出版社,2011.7

(21世纪高等学校计算机基础实用规划教材)

ISBN 978-7-302-25411-9

I. ①J… II. ①李… 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(2011)第073111号

责任编辑:付弘宇 薛 阳

责任校对:梁 毅

责任印制:李红英

出版发行:清华大学出版社

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

社 总 机:010-62770175

投稿与读者服务:010-62795954,jsjic@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015,zhiliang@tup.tsinghua.edu.cn

地 址:北京清华大学学研大厦A座

邮 编:100084

邮 购:010-62786544

印 刷 者:北京四季青印刷厂

装 订 者:北京市密云县京文制本装订厂

经 销:全国新华书店

开 本:185×260 印 张:14 字 数:348千字

版 次:2011年7月第1版 印 次:2011年7月第1次印刷

印 数:1~3000

定 价:24.00元

产品编号:034091-01

前 言

2006年3月初春,《Java 程序设计之网络编程》第1版由清华大学出版社正式出版,至今已有4年多的时间了。承蒙广大读者的厚爱,在这期间本书已经6次印刷,销售量达到17500册,已达到畅销书的水平,这是一个可喜的成绩。在2008年11月“第八届全国高校出版社优秀畅销书”评审活动中,本书被评为“二等奖”。许多老师和读者给本书提出了相当多的宝贵意见,使我们受益匪浅。2009年8月,《Java 程序设计之网络编程(第2版)》也由清华大学出版社出版。

随着教材的流传和使用,以及课程教学改革的不断深入,配套的实验指导教材的编撰也提上了日程,经过一年多的努力,《Java 程序设计之实验及课程设计教程》得以完成。本书可以作为《Java 程序设计之网络编程(第2版)》(清华大学出版社2009年8月出版)的配套习题解答与实验指导用书(作为第1版的配套实验教材也是可以的),同时也可作为“Java 课程设计”的指导用书。另外,本书拥有独立的知识体系,也可作为Java技术爱好者的参考学习资料。

本书采用案例驱动的方式进行讲解,针对知识点设计了相应的教学案例,这些小案例既相互独立又具有一定的联系。学生在制作这些小案例的过程中可以掌握每个知识点。在每个案例的程序设计之前,还罗列出了案例涉及的知识点。

全书分为5部分:

第1部分的主要内容是介绍Java语言集成环境、Eclipse开发平台、实验教学大纲、课程设计教学大纲。

第2部分主要介绍教材课后的部分习题解答,包括实验报告的模板、部分习题的实验目的、内容、实验过程及分析等内容。

第3部分是Java课程设计的实验指导,包括课程设计的目的和任务,以及各个案例的项目任务、需求分析、技术分析、程序设计难点及源代码。共8个这样的项目,如网络斗地主、坦克大战、连连看、C/S模式的图书馆管理系统、网络五子棋、电子八音盒、C/S模式的网络暗棋和在线电子相册等,读者可以在此基础上进行进一步的开发和学习。

第4部分是两个桌面应用案例,一个是MP3在线搜索程序,输入歌曲的名字,就可以在互联网上搜索和下载歌曲,支持多线程并发下载。另一个是十字路口车辆运行的简单状况,用Java多线程技术和IBM提供的并发组件来实现在有交通信号灯控制情况下多车辆运行的简单控制。这两个案例涉及多线程、图形技术和桌面程序设计技术。

第5部分是一个关于Web的案例,利用Hibernate等技术开发一个B/S模式下的实验教学管理系统,实现网上布置实验作业、控制实验进度、学生作业电子文档上传下载,以及作业批改、统计实验分数等功能。采用了多层架构模型和面向接口的编程技术,有兴趣的读者

可以将本系统应用到实际工作中,还可做更深入的扩展。

参与本书编写工作的人员除封面署名外,还有杨瑞龙、王飞鹏、唐杰、谢海泉、王中亚、张乘、马帅军、王杰、陈维等。除王飞鹏老师为九江学院的老师外,其余均为重庆大学的老师和工作人员。朱庆生教授为本书的主审,对本书做了详细的审查,并提出了宝贵的意见。李芝兴老师对全书进行了认真和反复的修改,王飞鹏老师对本书提出了许多宝贵的意见。本书的最终出版得到了许多老师和同学的帮助。清华大学出版社为本书的编写和出版尽职尽责。在本书完成之际,一并向他们表示诚挚的感谢。

在清华大学出版社的网站 www.tup.tsinghua.edu.cn 上提供了本书的所有案例源代码的下载,供读者参考,有关下载问题请联系 fuhy@tup.tsinghua.edu.cn。尽管作者在写作过程中投入了大量的时间和精力,但由于水平有限,错误和不足之处仍在所难免,敬请读者批评指正(任何建议可以发至邮箱 cqlizhx@163.com),我们会在适当时间进行修订和补充,并发布在重庆大学计算机学院 Java 精品课程网站 <http://www.cs.cqu.edu.cn> 上。

编 者

2010 年 10 月

于重庆大学

目 录

第 1 章 Java 开发环境及实验要求	1
1.1 Java 运行环境与开发环境	2
1.2 JDK 开发工具	6
1.3 Eclipse 平台	10
1.4 实验教学大纲	15
1.5 课程设计教学大纲	18
第 2 章 上机实验分析及参考答案	21
2.1 Java 程序基础实验	21
2.2 面向对象程序设计基础	25
2.2.1 实验目的	25
2.2.2 实验要求	26
2.2.3 实验过程、步骤及原始记录(算法、源程序、测试结果、分析等)	26
2.3 掌握继承、接口、内部类和包	30
2.3.1 实验目的	30
2.3.2 实验要求	31
2.3.3 实验过程、步骤及原始记录(算法、源程序、测试结果、分析等)	31
2.4 熟练掌握字符串处理技术	35
2.4.1 实验目的	35
2.4.2 实验要求	35
2.4.3 实验过程、步骤及原始记录(算法、源程序、测试结果、分析等)	36
2.5 掌握日期处理技术	38
2.5.1 实验目的	38
2.5.2 实验要求	38
2.5.3 实验过程、步骤及原始记录(算法、源程序、测试结果、分析等)	38
2.5.4 结论及提高	44
2.6 掌握线程编程技术	45
2.6.1 实验目的	45
2.6.2 实验要求	45
2.6.3 实验过程、步骤及原始记录(算法、源程序、测试结果、分析等)	46

2.6.4	实验结论和分析	49
2.7	掌握 GUI 编程技术	49
2.7.1	实验目的	49
2.7.2	实验要求	49
2.7.3	实验过程、步骤及原始记录(算法、源程序、测试结果、分析等)	50
2.7.4	实验结论和分析	53
2.8	掌握简单的网络通信技术	53
2.8.1	实验目的	53
2.8.2	实验要求	53
2.8.3	实验过程、步骤及原始记录(算法、源程序、测试结果、分析等)	54
2.8.4	实验分析	60
第 3 章	课程设计项目分析及参考答案	62
3.1	课程设计的目的和任务	62
3.2	项目 1: 网络斗地主	62
3.2.1	项目内容	62
3.2.2	程序结构	63
3.2.3	出牌规则	64
3.2.4	游戏界面	66
3.2.5	网络通信	69
3.3	项目 2: 坦克大战	79
3.3.1	项目内容	79
3.3.2	程序结构	79
3.3.3	游戏实体的表示	79
3.3.4	游戏界面的绘制	81
3.3.5	己方坦克的控制	82
3.3.6	敌方坦克和子弹的控制	84
3.4	项目 3: 连连看	92
3.4.1	项目内容	92
3.4.2	程序结构	92
3.4.3	游戏区域的绘制	93
3.4.4	标记图片和图片消除	93
3.5	项目 4: C/S 模式的图书馆管理系统	98
3.5.1	项目内容	98
3.5.2	程序结构	98
3.5.3	数据库的基本操作	98
3.5.4	登录功能	103
3.5.5	程序主窗口	103
3.5.6	图书检索	104

3.5.7	添加图书	104
3.5.8	修改图书信息	106
3.5.9	删除图书	106
3.5.10	修改密码	106
3.6	项目 5: 网络五子棋	107
3.6.1	项目内容	107
3.6.2	程序结构	107
3.6.3	五子棋游戏规则	108
3.6.4	界面绘制	111
3.6.5	网络功能的实现	116
3.7	项目 6: 电子八音盒	121
3.7.1	项目内容	121
3.7.2	程序结构	121
3.7.3	音乐播放列表	121
3.7.4	程序界面	122
3.7.5	添加音乐	122
3.7.6	播放音乐	123
3.7.7	暂停和继续音乐播放	124
3.7.8	删除音乐	124
3.8	项目 7: C/S 模式的网络暗棋	125
3.8.1	项目内容	125
3.8.2	程序结构	125
3.8.3	网络连接	126
3.8.4	棋子和棋盘	126
3.8.5	游戏界面	128
3.8.6	建立网络连接功能	129
3.8.7	走子功能	129
3.8.8	认输与求和功能	131
3.9	项目 8: 在线电子相册	132
3.9.1	项目内容	132
3.9.2	程序结构	132
3.9.3	数据操作	133
3.9.4	Servlet 和 Filter	139
3.9.5	页面功能	140
3.10	项目 9: 网络白板	144
3.10.1	项目简介	144
3.10.2	程序结构	144
3.10.3	网络通信	144
3.10.4	界面功能	145

第 4 章 Java 应用案例分析	151
4.1 MP3 在线搜索下载程序	151
4.1.1 目的和意义	151
4.1.2 主要界面	151
4.1.3 主要功能	151
4.1.4 主要操作流程	154
4.1.5 安装运行	154
4.1.6 程序分析	155
4.1.7 总结	166
4.2 十字路口交通简单模拟程序	166
4.2.1 目的和意义	166
4.2.2 主要界面	167
4.2.3 主要功能	167
4.2.4 主要操作流程	168
4.2.5 程序分析	168
4.2.6 总结	177
第 5 章 实验教学管理系统案例	178
5.1 开发环境	178
5.1.1 基本配置	178
5.1.2 运行网上源代码附件中的实例	178
5.2 系统需求	179
5.3 系统功能描述	180
5.3.1 学生模块的功能	180
5.3.2 教师模块的功能	182
5.3.3 管理员模块的功能	184
5.4 系统设计	185
5.5 系统框架搭建	188
5.5.1 编写 Hibernate 配置文件	190
5.5.2 编写 POJO 文件	191
5.5.3 用户登录及验证技术	194
5.5.4 学生模块	196
5.5.5 教师模块	202
5.5.6 管理员模块	206
附录 源代码资源使用说明	207

第 1 章

Java 开发环境及实验要求

Java 是一种简单的、跨平台的、面向对象的、分布式的、解释的、健壮的、安全的、结构中立的、可移植的、性能优异的多线程的动态计算机语言。

Java 不仅提供了一个丰富的语言和运行环境,而且还提供了一个免费的 Java 开发工具集(Java Developers Kits, JDK)。编程人员和最终用户可以利用这些工具来开发 Java 程序或调用 Java 内容。通常以 JDK 的版本来定义 Java 的版本。JDK 1.0 版于 1996 年初公开, JDK 1.1 版于 1997 年初公开, JDK 1.2 版于 1998 年底公开。基于市场行销的考量, Sun 公司在 JDK 1.2 版公开后随即将 Java 改名为“Java 2”, 将 JDK 改名为 J2SDK。J2SDK 1.3 于 2000 年 4 月公开, 此版本仍被称做“Java 2”, 不过按照习惯也把 J2SDK 叫做 JDK。目前 Java 语言的版本可以采用 Java SE 的最新版本, 从 Oracle 公司的网站上可以看到最新版本为 JDK 6 Update 24(作者撰稿时)。读者可以从 Oracle 公司的官方网站 <http://www.oracle.com/technetwork/java/> 上查看最新的信息(2009 年 Oracle 公司收购了 Sun 公司)。

Java 2 出现后, Sun 公司将 Java 分成 Java 2 Platform Enterprise Edition(简称 J2EE)、Java 2 Platform Standard Edition(简称 J2SE)、Java 2 Platform Micro Edition(简称 J2ME)三大领域, 目前流行的叫法为 Java EE、Java SE 和 Java ME。Java EE 适用于服务器, 目前已经成为企业运算、电子商务等领域中相当热门的技术; Java SE 适用于一般的计算机; Java ME 适用于消费性电子产品。除了这三者之外, JavaCard 依然是独立的一套标准。

随着应用领域的不同, Java 有许多 API(Application Programming Interface), 这些 API 可分成 3 大类。

(1) Java Core API: 由 Sun 制定的基本 API, 任何 Java 平台都必须提供。

(2) Java Standard Extension API(javax): 由 Sun 制定的扩充 API, Java 平台可以选择性地提供或加装。

(3) 厂商或组织所提供的 API: 由各家公司或组织所提供。

其中 Core API 和 Standard Extension API 已经逐渐涵盖了大部分信息应用领域, 例如多媒体、数据库、Web、企业运算、语音、实时系统、网络、电话、影像处理、加解密、GUI、分布式运算等。如果你有某项需求尚未有标准的 Java API 可遵循, 你可以向 Sun 提出制定新 API 的请求。经过审核之后, 你的要求可能会通过或驳回。如果通过, 就会开始进入制定 API 的程序。

1.1 Java 运行环境与开发环境

1. 安装 Java Development Kit(JDK)

如果只想运行别人的 Java 程序,可以只安装 Java 运行环境(Java Runtime Environment, 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 公司的网站 <http://java.sun.com/j2se> 免费下载 Java 的 JRE,例如可以根据提示下载支持 Microsoft Windows 操作系统的 JRE 文件 `jre-6u7-windows-i586-p.exe`。安装时可以选择默认的安装路径,也可以更改路径。

Sun 公司为所有的 Java 程序员提供了一套免费的 Java 开发和运行环境。本书将使用 JDK 目前最新的版本 JDK 6.0 版(也就是 J2SE 6.0)。可以通过 IE 或 Netscape 浏览器浏览网址 <http://java.sun.com/j2se>,根据提示可以下载支持 Microsoft Windows 操作系统的 `jdk-6u7-windows-i586-p.exe` 文件到本地硬盘。

安装时可以选择安装到任意的硬盘驱动器上,例如安装到 `d:\java\jdk` 目录下,如图 1-1 所示。正确安装后,在 JDK 目录下有 `bin`,`demo`,`lib`,`jre` 等子目录,如图 1-2 所示。其中 `bin` 目录保存了 `javac`,`java`,`appletviewer` 等命令文件,`demo` 目录保存了许多 Java 的例子,`lib` 目录保存了 Java 的类库文件,`jre` 目录保存的是 Java 运行时的环境(JRE)。

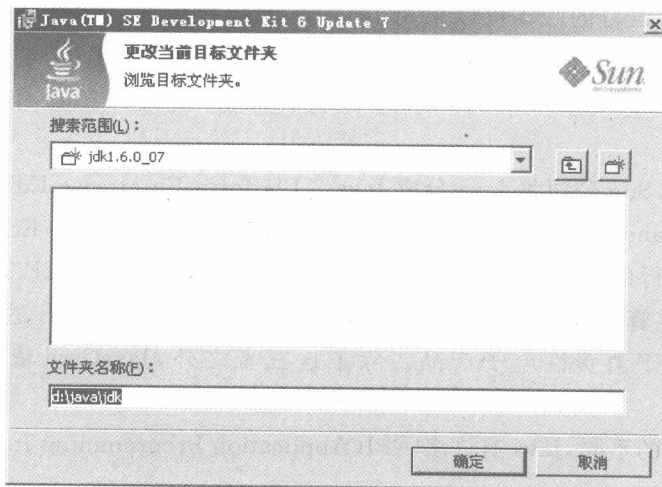


图 1-1 JDK 6.0 安装界面

JDK 简单易学,可以通过任何文本编辑器(例如 notepad, UltraEdit, Editplus, FrontPage 以及 Dreamweaver 等)编写 Java 源文件,然后在 DOS 命令行窗口下通过 `javac` 命令将 Java 源程序编译成字节码,通过 `java` 命令来执行编译后的 Java 文件。Java 初学者一般都采用这种开发工具。

从初学者的角度来看,采用 JDK 开发 Java 程序能够很快理解程序中各部分代码之间的关系,有利于理解 Java 面向对象的设计思想。JDK 的另一个显著特点是随着 Java (J2EE, J2SE 以及 J2ME)版本的升级而升级。但它的缺点也是非常明显的,就是从事大规模企业级 Java 应用开发非常困难,不能进行复杂的 Java 软件开发,也不利于团体协同

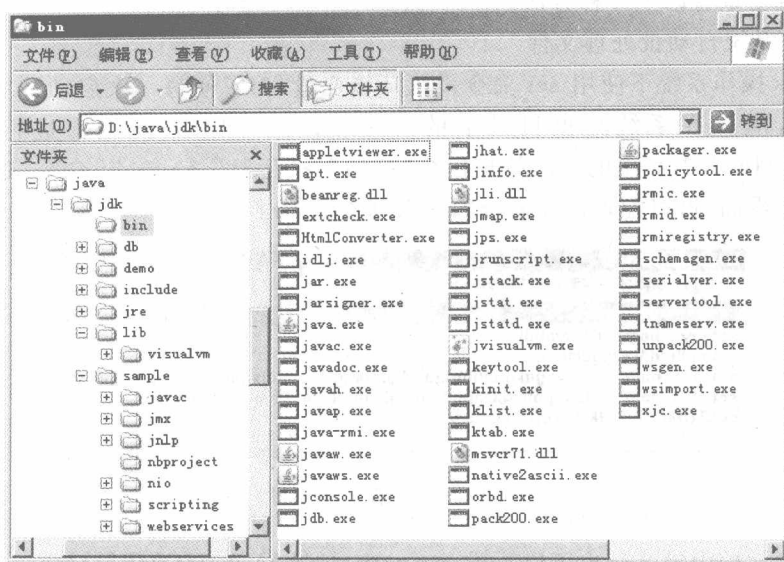


图 1-2 JDK 6.0 目录结构及文件

开发。

2. 环境变量的设置

设置环境变量是为了能够正常使用所安装的 JDK 开发包。通常需要设置 3 个环境变量：JAVA_HOME, PATH 和 CLASSPATH。

(1) JAVA_HOME

该环境变量的值就是 Java 所在的目录，一些 Java 版的软件和一些 Java 的工具需要用到该变量，设置 PATH 和 CLASSPATH 时，也可以使用该变量，以方便设置。

(2) PATH

该环境变量指定一个路径列表，用于搜索可执行文件。执行一个可执行文件时，如果该文件不能在当前路径下找到，则依次寻找 PATH 中的每一个路径，直至找到。如果找完 PATH 中的路径也不能找到，则报错。Java 的编译命令 (javac)、执行命令 (java) 和一些工具命令 (javadoc, jdb 等) 都在其安装路径下的 bin 目录中。因此应该将该路径添加到 PATH 变量中。

(3) CLASSPATH

该环境变量也指定一个路径列表，用于搜索 Java 编译或者运行时需要用到的类。在 CLASSPATH 列表中除了可以包含路径外，还可以包含 .jar 文件。Java 查找类时会把这个 .jar 文件当做一个目录来进行查找。通常需要把 JDK 安装路径下的 jre\lib\rt.jar 文件包含在 CLASSPATH 中。

PATH 和 CLASSPATH 都指定路径列表，列表中的各项（即各个路径）之间使用分隔符分隔。在 Windows 下，分隔符是分号（;）。

下面说明 3 个环境变量在 Windows 下如何设置，假设 JDK 在 Windows 下的安装路径是 d:\java\jdk，那么安装后的 JDK 至少会包括 bin, demo, include, jre, lib 等目录，如图 1-2 所示。

设置环境变量有以下 3 种方法。

(1) 修改系统自动批处理文件

Windows 操作系统下使用 set 命令设置环境变量,为了使每一次启动计算机都设置这些环境变量,应该在系统盘根目录下的 AUTOEXEC. BAT 文件中进行设置。对于 Windows 98/ME/XP,简单的方法就是执行“开始”→“运行”→sysedit 命令,这时会显示一个实用程序的界面,可以配置操作系统的一系列参数,如图 1-3 所示。

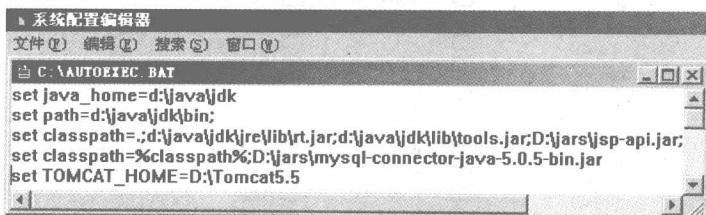


图 1-3 系统配置编辑器

上述配置文档中第 2 行也可以写成 `set path=%JAVA_HOME%\bin`。

有些版本的 Windows 不能用 %变量名% 来替换环境变量的内容,那么就只好直接写 `d:\java\jdk` 而不是 `%JAVA_HOME%` 了。另外, `c:\Windows` 和 `c:\Windows\Command` 是 Windows 操作系统自动加入到路径的,所以可以从设置中去掉。如果在 AUTOEXEC. BAT 中已经设置了 PATH,那只需要将 `%JAVA_HOME%\bin` 加到原来设置 PATH 的那条语句中。

CLASSPATH 也可以根据需要设置或者加入其他路径,例如你想把自己写的一些类放在 `d:\user\chap01` 中,就可以把 `d:\user\chap01` 也添加到 CLASSPATH 中,如下行语句。

```
set CLASSPATH=.;%JAVA_HOME%\jre\lib\rt.jar;d:\user\chap01;
```

注意,在 CLASSPATH 中包含了一个“当前目录(.)”。包含了该目录后,就可以到任意目录下执行需要用到该目录下某个类的 Java 程序,即使该路径并未包含在 CLASSPATH 中也可以。原因很简单:虽然没有明确地把该路径包含在 CLASSPATH 中,但 CLASSPATH 中的“.”在此时就代表了该路径。

如果读者像本书一样使用 JDK 6.0,那么可以使用 JDK 6.0 的新功能来设置 CLASSPATH (而 JDK 5.0 及其以前的版本没有该功能),即可以用“*”代替某目录下一系列的 *.jar 文件。格式如下:

```
set CLASSPATH=.;d:\jars\*;
```

(2) 在系统特性中设置 PATH 和 CLASSPATH

对于 Windows 2000/NT/XP 操作系统,右击“我的电脑”,弹出菜单,然后选择“属性”,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,添加如下系统环境变量。

- 变量名: JAVA_HOME。
- 变量值: d:\java\jdk(其内容应根据 JDK 安装目录变化)。
- 变量名: PATH。

- 变量值: d:\java\jdk\bin; %path% (其内容应根据 JDK 安装目录变化)。
- 变量名: CLASSPATH。
- 变量值: .;d:\java\jdk\jre\lib\rt.jar; (其内容应根据实际情况变化)。

如图 1-4 所示,添加系统环境变量界面如下。

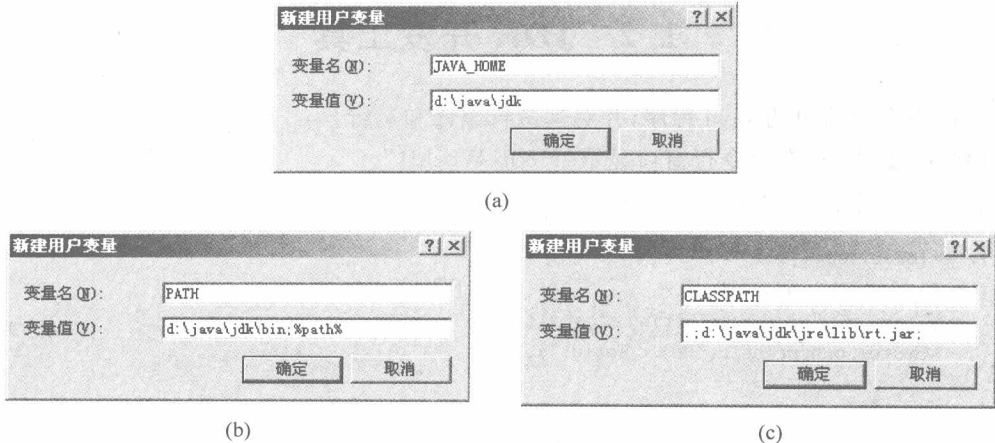


图 1-4 利用系统属性设置环境变量

安装 JDK 一般不需要设置环境变量 CLASSPATH 的值。如果你的计算机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 数据库等,则这些产品在安装后,将会修改 CLASSPATH 的值。那么当你运行 Java 应用程序时,可能加载这些产品所带的旧版本的类库,可能导致程序要加载的类无法找到,使你的程序出现运行错误。例如,某个软件用到的 CLASSPATH 值是:

```
.;d:\pb\jdk1.1.8\jre\lib\classes.zip;
```

如果不想删除这些产品的 CLASSPATH 设置,则可以重新编辑系统环境变量 CLASSPATH 的值,将新版本的类放在 CLASSPATH 路径的前面,修改如下:

```
.;d:\java\jdk\jre\lib\rt.jar;d:\pb\jdk1.1.8\jre\lib\classes.zip;
```

这时 JVM 在加载类时,首先将搜寻到新版本的类库,如果匹配成功,将进行加载,位于 CLASSPATH 后面的类库将被忽略。

(3) 在 MS-DOS 命令行窗口设置

也可以在 MS-DOS 命令行输入下列命令后,按 Enter 键确认,例如:

```
set JAVA_HOME = d:\java\jdk;
set PATH = d:\java\jdk\bin; % PATH % ;
set CLASSPATH = .;d:\java\jdk\jre\lib\rt.jar;
```

注意,这种方式设置的环境变量只对本 DOS 窗口有效。

有关 DOS 常用命令,读者可以参考计算机文化基础等有关计算机基础知识的书籍。

如果读者使用其他操作系统,JDK 的安装和配置也可以参考以上过程,只是表达方式和界面稍有不同。

Java 程序分成 3 类,即 Application(应用程序)、Applet(小程序)和 Servlet(服务器端小程序)。应用程序可以在计算机中单独运行。而小程序只能嵌在 HTML 网页中运行,这需要一些网页知识。Servlet 是运行在服务器端的小程序,它可以处理客户传来的请求(Request),然后传给客户端(Response)。

1.2 JDK 开发工具

下面先介绍简单的 Java 程序,并对其进行编译和运行。

【例 1-1】 要求在命令行窗口显示“Hello World!”。

将该文件命名为 HelloWorldApp.java,其源程序如下:

```
//HelloWorldApp.java
public class HelloWorldApp{           //一个应用程序
    public static void main(String args[]){
        System.out.println("Hello World!");
    }
}
```

该文件保存在 d:\user\chap01 目录下。然后对它进行编译:

```
d:\user\chap01> javac HelloWorldApp.java
```

编译的结果是生成字节码文件 HelloWorldApp.class。最后用 java 解释器来运行该字节码文件:

```
d:\user\chap01> java HelloWorldApp
```

结果在命令行窗口屏幕上显示 Hello World!,如图 1-5 所示。

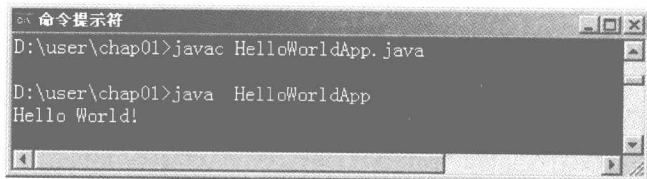


图 1-5 例 1-1 运行的结果

程序中出现的“//一个应用程序”是程序员对语句的注释,其注释部分将不会影响程序的编译和运行。本书中将采用这种方法对部分程序语句进行注释和说明。

上面用到的 javac 和 java 就是 JDK 提供的开发工具。这些程序包括在 d:\java\jdk\bin 目录中,并可以在任何目录中运行,前提是正确设置了环境变量路径 PATH。

下面将详细介绍一些常用的开发工具。

1. javac: Java 语言的编译器

```
D:\user\chap01> javac - help
用法: javac <选项> <源文件>
```

其中,可选项见表 1-1。

表 1-1 javac 的选项

选 项	含 义
-g	生成所有调试信息
-g:none	不生成任何调试信息
-g:{lines,vars,source}	只生成某些调试信息
-nowarn	不生成任何警告
-verbose	输出有关编译器正在执行的操作的消息
-deprecation	输出使用已过时的 API 的源位置
-classpath <路径>	指定查找用户类文件的位置
-cp <路径>	指定查找用户类文件的位置
-sourcepath <路径>	指定查找输入源文件的位置
-bootclasspath <路径>	覆盖引导类文件的位置
-extdirs <目录>	覆盖安装的扩展目录的位置
-endorseddirs <目录>	覆盖签名的标准路径的位置
-d <目录>	指定存放生成的类文件的位置
-encoding <编码>	指定源文件使用的字符编码
-source <版本>	提供与指定版本的源兼容性
-target <版本>	生成特定 VM 版本的类文件
-version	版本信息
-help	输出标准选项的提要
-X	输出非标准选项的提要
-J<标志>	直接将 <标志> 传递给运行时系统

(1) -d

-d 是为了指定输出目录。这个命令中要使用相对路径来指定 classes 目录的位置。假设当前目录为 d:\user\chap01,并在其下创建源代码 Hello.java。

创建一个 Hello.java 文件,代码如下:

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

编译的命令为:

```
d:\user\chap01>javac -d . Hello.java
```

编译后的结果如图 1-6 所示。“-d”选项可以自动创建和包名相应的下级文件夹,并把编译后的类文件存放于此。“.”代表当前目录。(注意“.”的前后均有空格)

(2) -classpath

如果当前要编译的 java 文件中引用了其他类(如继承),但该引用类的.class 文件不在当前目录下,则这种情况下就需要在 javac 命令后面加上 -classpath 参数(这时环境变量 classpath 设置的值失效),可以通过使用以下 3 种方法来指导编译器在编译时去指定的路

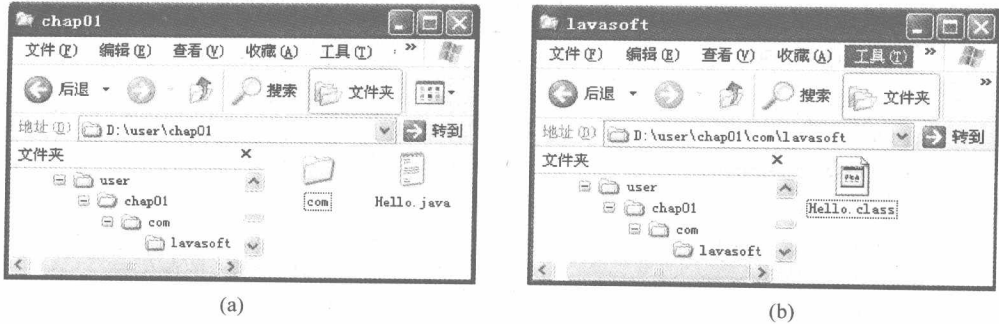


图 1-6 Java 原文件和类文件的文件夹

径下查找引用类。

- ① 绝对路径: `javac -classpath c:\junit3.8.1\junit.jar Xxx.java`。
- ② 相对路径: `javac -classpath ..\junit3.8.1\Junit.jar Xxx.java`。
- ③ 系统变量: `javac -classpath %CLASSPATH% Xxx.java`。

注意: `%CLASSPATH%`表示使用系统变量 `CLASSPATH` 的值进行查找,这里假设 `junit.jar` 的路径就包含在 `CLASSPATH` 系统变量中。

(3) `-verbose`

`-verbose` 输出有关编译器正在执行的操作的消息,通过它可以查看编译时加载的 jar 包等信息。

例如:

```
D:\user\chap01> javac -verbose -d com\lavasoft Hello.java
[源文件的搜索路径: [., d:\java\jdk\jre\lib\rt.jar, ...]]
[类文件的搜索路径: [d:\java\jdk\jre\lib\resources.jar, ...]]
[解析开始时间 Hello.java]
[解析已完成时间 78ms]
[正在装入 d:\java\jdk\lib\ct.sym(META-INF/sym/rt.jar/java/lang/Object.class)]
[正在装入 d:\java\jdk\lib\ct.sym(META-INF/sym/rt.jar/java/lang/String.class)]
[正在检查 com.lavasoft.Hello]
[正在装入 d:\java\jdk\lib\ct.sym(META-INF/sym/rt.jar/java/lang/System.class)]
[正在装入 d:\java\jdk\lib\ct.sym(META-INF/sym/rt.jar/java/io/PrintStream.class)]
[正在装入 d:\java\jdk\lib\ct.sym(META-INF/sym/rt.jar/java/io/FilterOutputStream.class)]
[正在装入 d:\java\jdk\lib\ct.sym(META-INF/sym/rt.jar/java/io/OutputStream.class)]
[已写入 .\com\lavasoft\Hello.class]
[总时间 1594ms]
```

2. java: Java 语言的解释器

```
d:\user\chap01> java -help
```

Java 解释器的使用方法有两种:

- (1) `java [-options] class [args...]` (运行 class 文件)
- (2) `java [-options] -jar jarfile [args...]` (运行 jar 文件)

部分选项见表 1-2。