



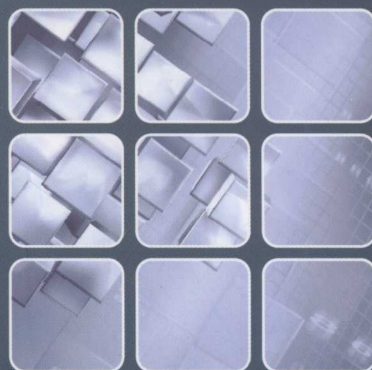
21st CENTURY
实用规划教材

21世纪全国高职高专
计算机系列实用规划教材

全新推出第 **2** 版

Java 程序设计教程与实训 (第2版)

主 编 许文宪 李兴福



COMPUTER

天涯太远，实用耐看的Java就在眼前
记忆更迭，不变的是免费赠送习题答案、源代码
岁月静好，18个实训清风自来



北京大学出版社
PEKING UNIVERSITY PRESS

21 世纪全国高职高专计算机系列实用规划教材

Java 程序设计教程与实训 (第 2 版)

主 编 许文宪 李兴福

参 编 许敬芝



北京大学出版社
PEKING UNIVERSITY PRESS

内 容 简 介

Java 语言是目前最流行的面向对象程序设计语言。本书按照“基本够用、适当扩展”的原则,分 9 章介绍 Java 的运行环境、语言基础、类、对象、继承、多态、数组与常用类、异常处理、数据流、图形用户界面、JDBC 技术等内容,第 10 章提供若干典型实训项目,便于实践教学。全部习题、实训的参考答案均可从 <http://www.pup6.cn> 下载或通过 E-mail 发送。

本书主要面向高职高专学生,可以作为高职高专计算机类和信息管理类专业的专科教材,也可作为其他专业的选学教材。

图书在版编目(CIP)数据

Java 程序设计教程与实训/许文宪, 李兴福主编. —2 版. —北京: 北京大学出版社, 2013.1

(21 世纪全国高职高专计算机系列实用规划教材)

ISBN 978-7-301-20879-3

I. ①J… II. ①许…②李… III. ①JAVA 语言—程序设计—高等职业教育—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2012)第 139671 号

书 名: Java 程序设计教程与实训(第 2 版)

著作责任者: 许文宪 李兴福 主编

策 划 编 辑: 李彦红

责 任 编 辑: 刘国明

标 准 书 号: ISBN 978-7-301-20879-3/TP · 1230

出 版 者: 北京大学出版社

地 址: 北京市海淀区成府路 205 号 100871

网 址: <http://www.pup.cn> <http://www.pup6.cn>

电 话: 邮购部 62752015 发行部 62750672 编辑部 62750667 出版部 62754962

电 子 邮 箱: pup_6@163.com

印 刷 者: 北京鑫海金澳胶印有限公司

发 行 者: 北京大学出版社

经 销 者: 新华书店

787mm×1092mm 16 开本 14.25 印张 323 千字

2005 年 9 第 1 版 2013 年 1 月第 2 版 2013 年 1 月第 1 次印刷

定 价: 28.00 元

未经许可, 不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有 侵权必究

举报电话: 010-62752024

电子邮箱: fd@pup.pku.edu.cn

前 言

Java 语言是一种面向对象的程序设计语言。目前,各高职高专院校的计算机相关专业大都将其作为“面向对象程序设计”首选的基础课程。

高职高专 Java 语言课程的教学目标是具备 Java 程序设计的基本能力,为 Java 的进一步深入学习或专题学习奠定基础;强化培养面向对象的思维方式,为学习后续课程提供必要的知识准备。通常面面俱到地讲述 Java 语言应该涉及的内容对高职高专学生来说是不适合的,也是教学计划所不允许的;同时,在高职高专的教学计划中,开始 Java 语言课程的教学时,学生面向对象程序设计的知识很弱且不系统。

基于上述理念,编者于 2005 年编写了《Java 程序设计教程与实训》教材,其针对高职高专教育的特点和规律,强调教材的实用性和易学性。在内容上,从实际教学出发,以“基本够用、适当扩展”为原则;在知识讲述上,尽量采用高职高专学生能够理解的叙述方式,力求“通俗易懂、逻辑严谨”。参加编写的老师有:许文宪(济南职业学院,主编)、董子建(聊城职业技术学院,主编)、王轶凤(山东商业职业技术学院,副主编)、李伟(昆明冶金高等专科学校,副主编)、李兴福(济南职业学院,参编)、赵晓峰(无锡商业职业技术学院,参编)、周绍景(昆明冶金高等专科学校,参编)、白杨(辽东学院信息技术学院,参编)、王艳红(济南职业学院,参编)、罗毅(湖北教育学院信息学院,参编)、牧笛(河南商业高等专科学校,参编)。

6 年来,各院校同仁对第 1 版教材内容提出了很多宝贵的意见和中肯的建议,Java 语言也有了更成熟的发展。为此,编者对第 1 版进行了修订。此次修订的原则是:保持原有的风格和特色,吸收同仁们的教学实践经验,更加反映技术进展,更加符合目前高职高专教学的需要。修订工作主要包括以下几个方面:将 JDK 版本由 1.4.2 改为 JDK6.0,所有例题的源程序重新进行了调试;删除了有关 Applet 的内容;重写了图形用户界面的内容,由 AWT 改为 Swing 实现;调整、合并了部分内容;增加了“JDBC 技术”一章,满足数据库方面的教学需要;修改了部分例题、习题和实训内容。

修订后的第 2 版教材共分为 10 章。第 1 章至第 9 章主要包括 Java 的运行环境、语言基础、类、对象、继承、多态、数组与常用类、异常处理、数据流、图形用户界面以及 JDBC 技术。第 10 章是实训部分,共给出了 18 个典型的实训。可以从 <http://www.pup6.cn> 下载参考答案和源代码,也可通过 E-mail(xwxian@126.com 或 tingwei0104@126.com)直接索取。

本书第 2 版的修订工作主要由济南职业学院的许文宪、李兴福完成。济南职业学院的许敬芝参加了本书的编写,并负责调试全书的例题代码,给出实训部分的参考答案。

本书作为三年制的高职高专课程教材,建议讲授学时为 48~64 学时,上机实习学时为 32 学时。

由于作者水平有限,书中难免存在不足之处,敬请广大读者特别是讲授此课程的老师批评指正。读者在使用过程中发现的问题和提出的建议可随时发送到邮箱: xwxian@126.com 或 tingwei0104@126.com,以便编者今后对本书进行进一步改进,在此表示衷心感谢!

作 者

2012 年 8 月

目 录

第 1 章 Java 语言概述	1	第 3 章 类和对象	32
1.1 Java 的发展历史和语言特点	1	3.1 类的定义	32
1.1.1 Java 的发展历史	1	3.1.1 类和对象的关系	32
1.1.2 Java 的语言特点	2	3.1.2 类的定义格式	34
1.1.3 Java 运行机制	3	3.2 方法	36
1.2 Java 的运行环境	4	3.2.1 方法的返回值	36
1.3 最简单的 Java 程序	6	3.2.2 方法的参数	37
1.3.1 Hello World 程序的开发		3.3 类的实例化	38
流程	6	3.3.1 创建对象	38
1.3.2 程序分析	6	3.3.2 使用对象成员	39
小结	8	3.3.3 类成员的访问控制	41
习题	8	3.4 构造方法	45
第 2 章 Java 语言基础	9	3.4.1 构造方法的作用和定义	45
2.1 Java 的基本数据类型	9	3.4.2 默认构造方法	46
2.1.1 标识符	9	3.4.3 构造方法的使用	46
2.1.2 基本数据类型	10	3.5 参数传递和 this 引用	47
2.1.3 常量	11	3.5.1 对象作为方法的参数	47
2.1.4 变量	12	3.5.2 this 引用	49
2.1.5 数据类型转换	14	3.5.3 类的封装性	50
2.2 运算符	14	3.6 类的组织	51
2.2.1 算术运算符	14	3.6.1 包的概念	51
2.2.2 关系运算符	16	3.6.2 创建包	52
2.2.3 逻辑运算符	17	3.6.3 访问包	52
2.2.4 位运算符	19	3.7 实例分析	56
2.2.5 其他运算符	20	小结	59
2.2.6 运算符的优先级	21	习题	59
2.3 Java 的控制结构	21	第 4 章 继承与多态	61
2.3.1 if 结构	21	4.1 继承和多态的概念	61
2.3.2 switch 语句	24	4.1.1 继承的概念	61
2.3.3 for 循环	25	4.1.2 多态的概念	63
2.3.4 while 循环和 do-while 循环	26	4.2 类的继承	64
2.3.5 跳转语句	27		
小结	30		
习题	30		

4.2.1 继承的实现	64	5.5.4 foreach 语句的使用	108
4.2.2 属性和方法的继承	65	小结	110
4.2.3 父类对象与子类对象的转换	66	习题	110
4.2.4 构造方法的继承	67	第 6 章 Java 异常处理	112
4.3 类成员的覆盖	69	6.1 异常处理概述	112
4.3.1 覆盖的概念	69	6.1.1 异常	112
4.3.2 域隐藏的使用	69	6.1.2 异常处理机制	113
4.3.3 方法覆盖的使用	71	6.1.3 异常分类	113
4.3.4 super 引用	72	6.2 Java 异常的处理方法	114
4.4 方法重载	73	6.2.1 try/catch/finally	115
4.4.1 方法的重载	73	6.2.2 声明异常	118
4.4.2 构造方法的重载	74	6.2.3 抛出异常	119
4.5 抽象类和最终类	75	6.2.4 自定义 Java 异常	120
4.5.1 抽象类	75	小结	121
4.5.2 最终类	77	习题	122
4.6 接口	77	第 7 章 Java 数据流	123
4.6.1 接口的定义	77	7.1 Java 数据流概述	123
4.6.2 接口的实现	78	7.2 Java 字节流	125
小结	80	7.2.1 InputStream 类与 OutputStream 类	125
习题	80	7.2.2 System.in 与 System.out	126
第 5 章 数组与常用类	82	7.2.3 FileInputStream 类与 FileOutputStream 类	127
5.1 数组	82	7.2.4 DataInputStream 类与 DataOutputStream 类	129
5.1.1 数组的定义与创建	82	7.3 Java 字符流	132
5.1.2 访问数组元素	83	7.3.1 Reader(字符输入流)类与 Writer(字符输出流)类	132
5.1.3 使用二维数组	86	7.3.2 FileReader 类与 FileWriter 类	132
5.1.4 命令行参数	88	7.3.3 BufferedReader 类与 BufferedWriter 类	133
5.2 Java API 与技术文档	89	7.3.4 InputStreamReader 与 OutputStreamWriter	134
5.3 数据类型类	91	7.4 读写随机文件	135
5.3.1 数据类型类的属性和构造 方法	91	7.5 目录与文件管理	138
5.3.2 数据类型类的常用方法	92	小结	142
5.4 String 类和 StringBuffer 类	94	习题	143
5.4.1 String 类	95		
5.4.2 StringBuffer 类	100		
5.5 Java 中的集合类	103		
5.5.1 Vector 类	103		
5.5.2 Stack 类	105		
5.5.3 Hashtable 类	105		

第 8 章 Java 图形用户界面	144	9.2 连接数据库	185
8.1 Java 图形用户界面概述	144	9.2.1 连接数据库过程	185
8.1.1 AWT 和 Swing	144	9.2.2 配置 JDBC-ODBC 数据源	187
8.1.2 组件和容器	146	9.3 查询数据库	191
8.2 Swing 常用组件	147	9.3.1 查询数据库过程	191
8.2.1 框架与面板	147	9.3.2 查询数据库数据	193
8.2.2 按钮和标签	151	9.4 操作数据库	196
8.2.3 复选框和单选按钮	152	小结	200
8.2.4 单行文本框和多行文本框	152	习题	200
8.2.5 列表框和下拉列表框	155	第 10 章 实训	201
8.2.6 表格与滚动面板	157	实训 1 开发工具和运行环境	201
8.2.7 菜单	158	实训 2 基本数据类型、运算符	202
8.3 布局管理器	163	实训 3 Java 控制结构	202
8.3.1 布局管理器概述	163	实训 4 方法的定义和调用	203
8.3.2 流布局	163	实训 5 对象的创建与使用	203
8.3.3 边界布局	164	实训 6 类的组织——包	204
8.3.4 网格布局	165	实训 7 类的继承	204
8.3.5 空布局	165	实训 8 重载和覆盖	205
8.4 Java 事件处理机制	167	实训 9 接口的实现	206
8.4.1 Java 事件处理概述	167	实训 10 数组及命令行参数	207
8.4.2 Java 常用事件	170	实训 11 String 类和 StringBuffer 类	208
8.4.3 事件适配器	177	实训 12 异常处理	208
小结	179	实训 13 文件属性的访问	209
习题	179	实训 14 文本文件的读写	211
第 9 章 JDBC 技术	181	实训 15 随机文件的读写	211
9.1 JDBC 技术简介	181	实训 16 图形用户界面(一)	212
9.1.1 关系型数据库基础知识	181	实训 17 图形用户界面(二)	213
9.1.2 JDBC 驱动程序	183	实训 18 数据库操作	215
		参考文献	217

第 1 章 Java 语言概述

教学目标：通过本章的学习，了解 Java 语言的特点，掌握 Java 语言的运行机制，学会使用记事本完成最简单的 Java 程序设计

教学要求：

知识要点	能力要求	关联知识
Java 的发展历史	了解 Java 的发展历史及版本	Oak → Java Java SE、Java EE、Java ME
Java 的特点	掌握 Java 语言的特点	面向对象、平台无关、安全等
Java 的运行机制	(1) 掌握 Java 的运行机制 (2) 理解 JVM 的工作过程 (3) 理解 Java 跨平台的实现原理	JVM、字节码的概念 Java 的解释运行过程
Java 环境的建立	(1) 学会 JDK 的安装方法 (2) 学会环境变量的配置	环境变量的配置、Java 编译器、Java 解释器、PATH 环境变量
Java 程序的开发流程	(1) 掌握使用记事本开发 Java 程序的过程 (2) 了解 Java 程序的基本结构	Java 程序的基本框架、main() 方法

重点难点：

- Java 的运行机制
- Java 的环境变量配置
- 编写、运行 Java 程序的流程

1.1 Java 的发展历史和语言特点

1.1.1 Java 的发展历史

1991 年，Sun 公司为了进军家用电子消费市场，成立了一个代号为 Green 的项目组。其目标是开发一个分布式系统，让人们可以利用网络远程控制家用电器。鉴于家用电器制造商众多且制造标准各异，项目组希望新系统具有独立于软件平台的特征，并且安全易用。开始时项目组采用当时使用广泛的 C++ 语言进行系统开发，但是由于 C++ 语言太复杂，安全性也难以满足要求，最后不得不放弃 C++，转而研究设计出了一套新的程序设计语言。这个新的程序设计语言就是 Java 语言的前身，被命名为 Oak(橡树)。可惜的是，由于一些商业上的原因，Sun 公司在以 Oak 为程序设计语言投标“交互式电视项目”时未能中标，这使得 Oak 语言的进一步发展一度遇到很大的问题。

20 世纪 90 年代中期，WWW 的影响在 Internet 上越来越大，WWW 浏览器开始在市场上出现，这预示着计算机网络应用的浪潮即将到来。1994 年，Green 项目组成员认真分析了计算机网络应用的特点，认为 Oak 满足网络应用所要求的平台独立性、系统可靠性和安全性等，并用 Oak 设计了一个称为 WebRunner(后来称为 HotJava)的 WWW 浏览器。1995 年 5 月 23 日，Sun 公司正式发布了 Java 和 HotJava 两项产品。

Java 语言一经推出,就受到了业界的关注。Netscape 公司第一个认可 Java 语言,并于 1995 年 8 月将 Java 解释器集成到它的主打产品 Navigator 浏览器中。接着,Microsoft 公司在 Internet Explorer 浏览器中认可了 Java 语言。Java 语言开始了自己的发展历程。

2009 年 4 月,Oracle(甲骨文)公司以 74 亿美元收购 Sun 公司,从此 Sun 公司旗下的两大产品 Java 和 Solaris 归入 Oracle 门下,甲骨文 CEO 拉里·埃里森(Larry Ellison)表示收购 Sun 之后,将继续加大对 Java 的投资,甲骨文的中间件战略将“100%基于 Java”,Java 的发展进入了新的纪元。

目前使用的 Java 版本包括 Java SE(Java Standard Edition)、Java EE(Java Enterprise Edition)、Java ME(Java Micro Edition)共 3 个版本,分别用于不同的领域。Java SE 用于工作站、PC,为桌面开发和低端商务应用提供了 Java 标准平台。Java EE 用于服务器,构建可扩展的企业级 Java 平台。Java ME,嵌入式 Java 消费电子平台,适用于消费性电子产品和嵌入式设备。

1.1.2 Java 的语言特点

Java 语言是简单的、面向对象的语言,它具有分布式的特点,安全性高,又可以实现多线程,更主要的是它与平台无关,解决了困扰软件界多年的软件移植问题。

1. 面向对象

面向对象(object-oriented)程序设计模式是近代软件工业的一种革新,使软件具有弹性(flexibility)、模块化(modularity)且具有重复使用性(reusability),降低开发与成本。

2. 语法简单

Java 语言的语法结构类似于 C 和 C++,熟悉 C 和 C++的程序设计人员不会对它感到陌生。与 C++相比,Java 对复杂特性的省略和实用功能的增加使得开发变得简单而可靠,例如不再支持诸如运算符重载、多继承及自动强制类型转换等容易混淆且较少使用的特性;去掉了容易导致错误的指针概念;增加了内存空间的自动垃圾收集功能,既避免了内存泄漏现象的发生,又简化了程序设计。简单化的另一个方面是 Java 的系统非常小,其基本解释和类支持部分只占 40 KB,附加上基本标准库和线程支持也只需增加 175 KB。

3. 平台无关性

平台无关性是指 Java 能运行在不同的系统平台上。Java 引进了虚拟机概念,Java 虚拟机(Java Virtual Machine, JVM)建立在硬件和操作系统之上,用于实现对 Java 字节码文件的解释和执行,为不同平台提供统一的 Java 接口,这使得 Java 应用程序可以跨平台运行,非常适合网络应用。

4. 安全性

安全性是网络应用系统必须考虑的重要问题。Java 设计的目的是提供一个网络/分布式的计算环境,因此,Java 特别强调安全性。Java 程序运行之前会利用字节确认器进行代码的安全检查,确保程序不会存在非法访问本地资源、文件系统的可能,保证程序在网络间传送运行的安全性。

5. 分布式应用

Java 为程序开发提供了 `java.net` 包, 该包提供了一组类, 使程序开发者可以轻易实现基于 TCP/IP 的分布式应用系统。此外, Java 还提供了专门针对互联网应用的一整套类库, 供开发人员进行网络程序设计。

6. 多线程

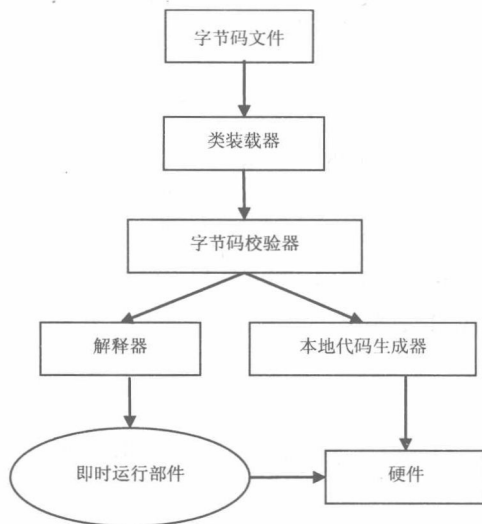
Java 语言内置了多线程控制, 可使用户程序并行执行。利用 Java 的多线程编程接口, 开发人员可以方便地写出多线程的应用程序。Java 语言提供的同步机制可保证各线程对共享数据的正确操作。在硬件条件允许的情况下, 可以将这些线程直接分布到各个 CPU 上, 充分发挥硬件性能, 提高程序执行效率。

1.1.3 Java 运行机制

Java 程序的运行必须经过编写、编译、运行 3 个步骤。编写是指在 Java 开发环境中进行程序代码的输入, 最终形成后缀名为 `.java` 的 Java 源文件。编译是指使用 Java 编译器对源文件进行错误排查的过程, 编译后将生成后缀名为 `.class` 的字节码文件, 这不像 C 语言那样最终生成可执行文件。运行是指使用 Java 解释器将字节码文件翻译成机器代码, 执行并显示结果, 这一过程如图 1.1(a)所示, 其解释器的工作过程如图 1.1(b)所示。



(a) Java 程序运行流程



(b) Java 程序的执行

图 1.1 Java 程序的执行

字节码文件是一种和任何具体机器环境及操作系统环境无关的中间代码,它是一种二进制文件,是 Java 源文件由 Java 编译器编译后生成的目标代码文件。编程人员和计算机都无法直接读懂字节码文件,它必须由专用的 Java 解释器来解释执行,因此 Java 是一种在编译的基础上进行解释运行的语言。

Java 解释器负责将字节码文件翻译成具体硬件环境和操作系统平台下的机器代码,以便执行。因此 Java 程序不能直接运行在现有的操作系统平台上,它必须运行在被称为 Java 虚拟机的软件平台之上。

Java 虚拟机是运行 Java 程序的软件环境,Java 解释器就是 Java 虚拟机的一部分。在运行 Java 程序时,首先会启动 JVM,然后由它来负责解释执行 Java 的字节码,并且 Java 字节码只能运行在 JVM 之上,这样利用 JVM 就可以把 Java 字节码程序和具体的硬件平台以及操作系统环境分隔开来,只要在不同的计算机上安装了针对于特定平台的 JVM,Java 程序就可以运行,而不用考虑当前的硬件平台及操作系统环境,也不用考虑字节码文件是在何种平台上生成的。JVM 把这种不同软硬件平台的具体差别隐藏起来,从而实现了真正的二进制代码级的跨平台移植。JVM 是 Java 平台无关的基础,Java 的跨平台特性正是通过在 JVM 中运行 Java 程序实现的。Java 的这种运行机制可以通过图 1.2 说明。

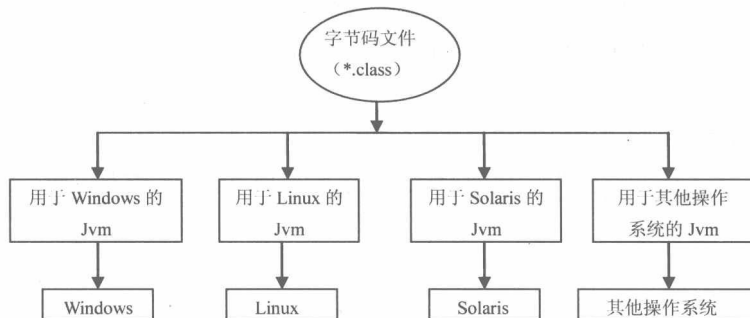


图 1.2 Java 运行机制

Java 语言这种“一次编写,到处运行(write once, run anywhere)”的方式,有效地解决了目前大多数高级程序设计语言需要针对不同系统来编译产生不同机器代码的问题,即硬件环境和操作平台的异构问题,大大降低了程序开发、维护 and 管理的开销。

需要注意的是,Java 程序通过 JVM 可以实现跨平台,但 JVM 不是跨平台的。也就是说,不同操作系统之上的 JVM 是不同的,Windows 平台之上的 JVM 不能用在 Linux 上,反之亦然。

1.2 Java 的运行环境

1. Java 开发工具 Java SDK

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 软件开发工具集(Java Development Kits, JDK)。JDK 包括 Java 的编译器、解释器、调试器等开发工

具以及 Java API 类库。编程人员和最终用户可以利用这些工具来开发 Java 程序。其调试工具主要包括 Java 语言编译器 `javac`，用于编译 Java 源程序；Java 字节码解释器 `java`，用于解释运行 Java 程序，显示程序运行结果；小应用程序浏览工具 `appletviewer`，用于测试并运行 Java 小程序。到目前为止，Sun 公司先后发布了多个主要的 JDK 版本。其最新稳定版本为 JDK 6.0，而 JDK 7.0 Preview 即测试版也已发布，读者可自行去 Java 官方网站下载，本书采用 JDK 6.0 作为开发工具。

2. 安装和设置环境变量

为了搭建起 Java 的运行环境，可以到 Oracle 公司的网站(<http://www.oracle.com/technetwork/java/javase/downloads/index.html>)下载最新的 JDK。建议同时下载其 Java Documentation，这是 Java 帮助文档。

JDK 6.0 版最新下载文件名为 `jdk-6u24-windows-i586.exe`，默认安装在 `C:\Program Files\Java\jdk1.6.0_24` 目录下，安装过程主要遵循提示单击【下一步】按钮即可完成。接下来需要设置运行环境参数，以便能够在 Windows 的任何目录下都能编译和运行 Java 程序，其方法是右击【我的电脑】图标，在弹出的快捷菜单中依次选择【属性】|【高级】|【环境变量】，打开【环境变量】对话框，单击【系统变量】列表框下方的【新建】按钮，新建环境变量 `classpath`，其变量值为 `“.;C:\Program Files\Java\jdk1.6.0_24\lib”`；选择 Path 变量，单击【编辑】按钮，在 Path 变量的变量值后面加上 `“;C:\Program Files\Java\jdk1.6.0_24\bin”`。

3. Java 的编辑、编译和运行

Java 源程序是一种文本文件，可以使用任何的文本编辑器编写，只是要注意存储时的文件后缀名必须是 `.java`。建议把所有源程序文件都保存在一个指定的目录下，便于调试和运行。

这里向初学者推荐两种编辑器：一是 Windows 的记事本，二是文本编辑工具 UltraEdit 和 EditPlus。使用 Windows 记事本编辑 Java 源程序文件，存储时先选择 `*.*(所有文件)` 的文件类型，然后输入带有 `.java` 后缀的文件名；或者直接以带英文双引号的形式 `"XXXX.java"` 输入文件名。UltraEdit 和 EditPlus 是两个非常易用且功能强大的文本编辑工具。编辑时，它们自动地把关键字、常量、变量等不同元素用不同的颜色区分开来，从而有助于减少语法错误。还可以选择其他的 Java 开发工具，如 JCreator、JBuilder、Eclipse、NetBeans 等。

Java 源程序必须先由 Java 编译器进行编译，生成字节码文件(也称为类文件)，然后在 Java 解释器的支持下解释运行。

Java 编译器是 `javac.exe`，其用法如下：

```
javac filename.java
```

其中 `filename.java` 是 Java 源程序文件的文件名。如果编译器没有返回任何错误信息，则表示编译成功，并在同一目录下生成与类名相同的字节码文件 `filename.class`。如果编译出错，则需查找错误原因，进一步修改源程序，并重新编译。

Java 解释器是程序 `java.exe`，其用法如下：

```
java filename
```

其中 `filename` 是编译生成的 Java 字节码文件的文件名，注意不要带后缀名 `.class`。

1.3 最简单的 Java 程序

1.3.1 Hello World 程序的开发流程

本节将介绍 Java 应用程序的一个简单示例,以此来说明 Java 应用程序的开发流程和程序的基本结构。

【例 1.1】 编写一个应用程序,在屏幕上显示字符串“Hello, World!”。

第一步,编写源程序。使用记事本编写程序的源代码,代码的内容如图 1.3 所示。将源代码保存为文件 HelloWorld.java,并存放在一个指定的目录例如 C:\test 中。注意:输入文件名时必须区分大小写。

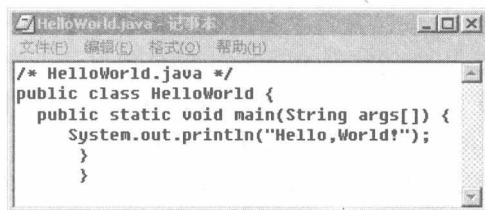


图 1.3 HelloWorld 程序

第二步,编译源程序。选择【开始】|【运行】命令,在打开的对话框中输入“cmd”并按回车键,打开一个 MS-DOS 命令提示符窗口;将当前目录转换到 Java 源程序所在的目录 C:\test;输入“javac filename.java”形式的命令进行程序编译。在本例中应输入“javac HelloWorld.java”。如果编译正确,编译结果应如图 1.4 所示。

第三步,执行程序。在同样的命令提示符窗口中输入“java filename”形式的命令,在本例中应输入“java HelloWorld”,运行结果如图 1.5 所示。

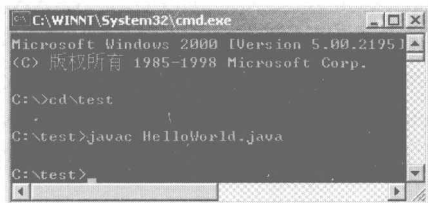


图 1.4 HelloWorld 程序编译结果

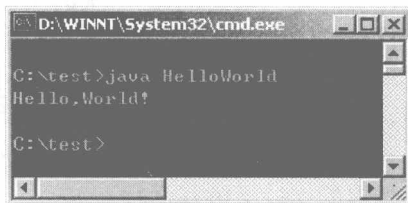


图 1.5 HelloWorld 程序运行结果

至此就完成了这个简单程序的开发。查看 C:\test 目录,其中应该有两个文件,分别是 HelloWorld.java 和 HelloWorld.class。

1.3.2 程序分析

由 HelloWorld 程序可知,Java 应用程序的基本框架如下:

```
public class 类名{
    public static void main ( String args [ ] ){
```

```
.....//程序代码  
}  
//其他属性和方法  
}
```

在下面的分析中,为了说明方便,特地为 HelloWorld 程序的源代码行加上了编号:

```
1  /* HelloWorld.java */  
2  public class HelloWorld {  
3      public static void main(String args[]) {  
4          System.out.println("Hello, World!");  
5      }  
6  }
```

第 1 行是注释行。Java 语言主要有 3 种注释:形如“/* 注释内容 */”的格式,可以注释一行或多行文本;形如“// 注释内容”的格式,可以注释一行文本;形如“/** 注释内容 */”的格式,可以注释一行或多行文本,并可用于生成专门的 Javadoc。注释可以放在一行的开头或某个语句之后,为程序增加必要的解释,提高程序的可读性。

第 2 行是类的定义。使用关键字 `class` 定义了一个 HelloWorld 类, `class` 前面的 `public` 关键字表示这个类的访问特性是公共的。

Java 语言中的基本程序单位是类,在一个程序文件中可以定义多个类,但仅允许有一个公共的类。源程序的文件名要与公共类的名称相同(包括大小写),其扩展名为 `.java`。因此,HelloWorld 程序的源程序文件名必须是 `HelloWorld.java`。

第 2 行最后到第 6 行的一对花括号内部是类体。在类体中可以声明类的变量(属性)和类的方法(函数),它们是类的成员。本例中没有声明类的成员变量。

第 3 行是类的成员方法的声明,这是一个主方法 `main()`。Java 应用程序必须含有一个主方法。`public` 关键字表示这个方法是公共的,可以从其他类中访问;`static` 关键字表示这个方法是静态的,指出这个方法是针对这个类而不是针对类生成的对象;`void` 关键字表示这个方法没有返回值。

一个类可以声明多种方法,但最多只能有一个主方法 `main()`。应用程序从 `main()` 方法获得入口点开始运行,并通过主方法调用类中的其他方法。

`Main()` 方法后的小括号中是方法的参数列表,它们是方法内的局部变量,接收从外部向 `main()` 方法中传递的参数。通常将其写成 `String args[]`,表明所接收的参数是一个名为 `args` 的字符串数组。

第 3 行最后到第 5 行的一对花括号内部是方法体。在方法体内部,可以声明方法的局部变量及书写执行语句,实现数据处理功能。

第 4 行是 `main()` 方法中唯一的一条语句,其作用是在标准输出设备(屏幕)上输出一行字符“Hello, World!”。这是一个字符串,必须用引号括起来。最后的分号是必需的,表明这是一条 Java 语句。

为了实现屏幕输出,这里使用了系统包 `java.lang` 中的 `System` 类,该类中有静态成员变量 `out`。`out` 是一个标准输出流,主要用于输出用户信息,通过 `out` 调用标准输出流类中的 `println` 方法。`println` 方法用于将作为参数的字符串输出到屏幕上并换行。与此相似的方法是 `print`,不同之处是 `print` 输出内容后不换行。

Java 语言系统以包的形式提供了许多的标准类库, 这些类库是编制 Java 程序的基础。调用类库中的方法之前, 先要使用 `import` 语句导入相应的类库(包), 而系统包 `java.lang` 是 Java 最基本的类库, 由系统自动导入, 因此, HelloWorld 程序没有使用 `import` 语句。

小 结

Java 语言是当今流行的网络编程语言, 特别适合于开发网络应用程序, 具有面向对象、简单、平台无关、多线程等优秀特性。

Java 应用程序的开发必须经过编写、编译、运行 3 个步骤。使用记事本等文本编辑工具进行程序代码的编写, 使用 Java 开发工具集 JDK 提供的编译器进行编译, 最后使用 Java 解释器解释运行。Java 虚拟机 JVM 使 Java 应用程序实现了跨平台运行。

本章介绍的 HelloWorld 程序可以作为初学者编写简单应用程序的入门模板。

习 题

1. Java 语言有哪些特点?
2. 简述 Java 的运行机制。
3. 简述 Java 应用程序的开发过程。
4. 在计算机上安装、配置 Java 运行环境, 并编辑运行本章中的例题。

第 2 章 Java 语言基础

教学目标：通过本章学习，掌握 Java 语言的基本数据类型、操作符、表达式的使用方法；掌握 Java 程序的流程控制方式，重点是分支结构和循环结构。

教学要求：

知识要点	能力要求	关联知识
基本数据类型	(1) 掌握 Java 基本数据类型的用法 (2) 学会数据类型的转换	常量、变量、标识符、显式类型转换、隐式类型转换
运算符	掌握常用运算符的用法	算术运算符、关系运算符、逻辑运算符
Java 控制结构	(1) 掌握 if、swith 分支语句的用法 (2) 掌握 for、while、do-while 循环语句的用法 (3) 掌握 break、continue 跳转语句的用法	if(){}else{} swith(){case;break;default:;} for(int i=初值;i<=终值;步长){} while(){}

重点难点：

- if 语句的使用方法
- for 循环语句的使用方法
- while 循环语句的使用方法

2.1 Java 的基本数据类型

在不同的计算机程序设计语言中，数据类型的定义和处理方法各不相同，但却是必不可少的部分。数据类型用于表示数据的格式和结构。在 Java 语言中，主要有两种类型的数据：基本类型和引用类型(复合类型)。相应地，也就有了两种类型的变量。对于常量和变量的概念，可以这样来理解：计算机在处理数据时，必须将其装入内存，按照不同的数据类型分配不同的存储空间，借助于对内存单元的命名来访问这些数据。被命名的内存单元就是常量和变量。本章首先介绍 Java 的基本类型。

2.1.1 标识符

符号是构成语言和程序的基本单位。Java 语言不采用计算机语言系统通常所采用的 ASCII 代码集，而是采用更为国际化的 Unicode 字符集。在这种字符集中，每个字符用 2 个字节即 16 位表示。这样，整个字符集中共包含 65535 个字符。其中，前面 256 个字符表示 ASCII 码，使 Java 对 ASCII 码具有兼容性；后面 21000 个字符用来表示汉字等非拉丁字符。Java 符号按词法可分为如下 5 类。