

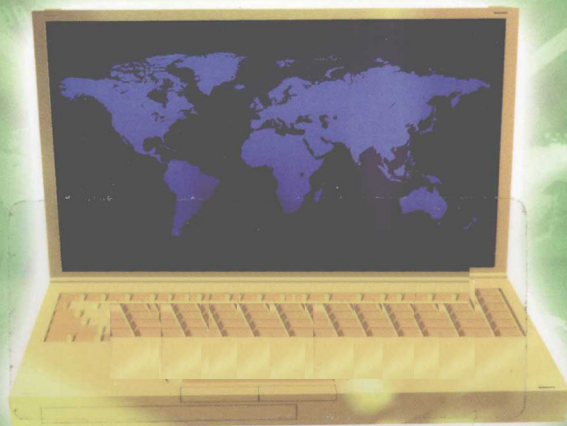


新世纪应用型高等教育
软件专业系列规划教材

Java 面向对象 程序设计与应用开发教程

新世纪应用型高等教育教材编审委员会 组编

主编 匡泰 王岩



大连理工大学出版社



新世纪应用型高等教育
软件专业系列规划教材

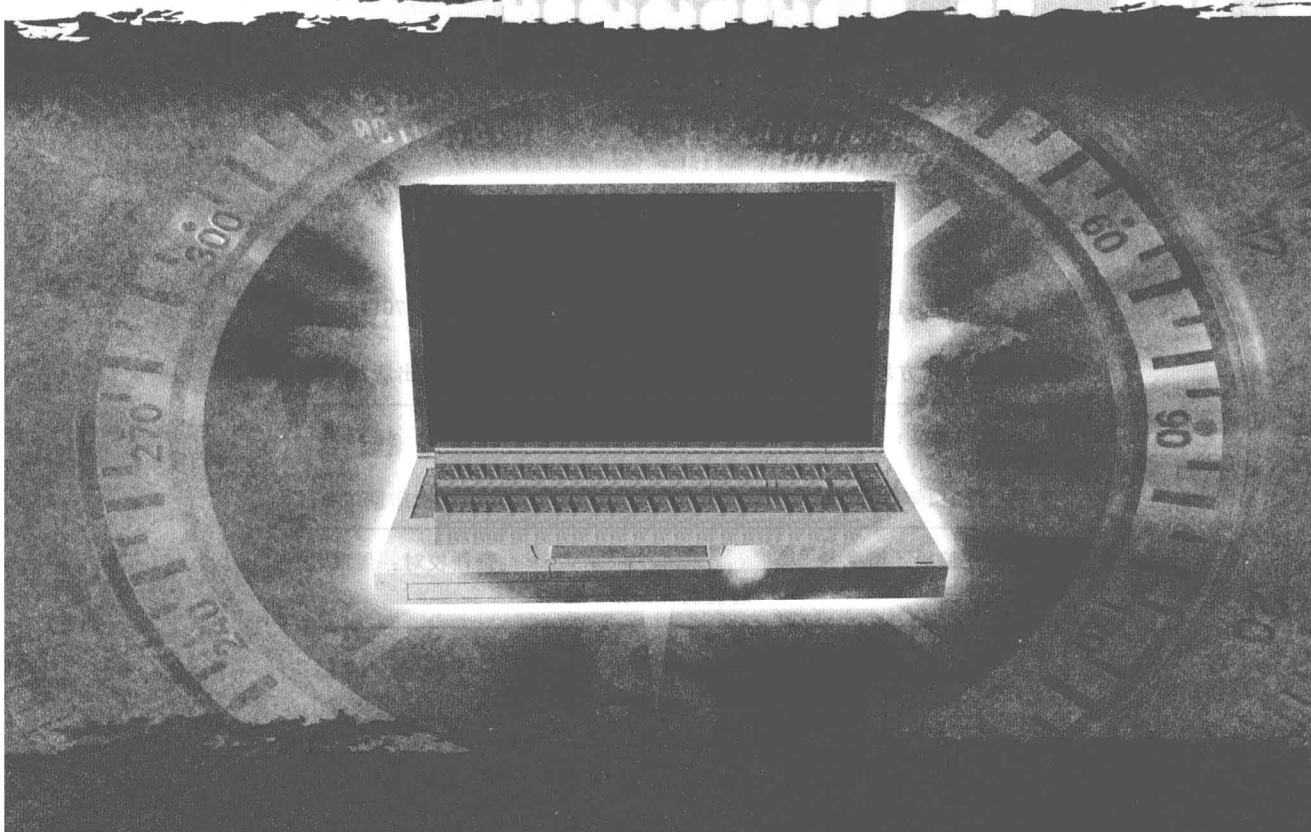
Java 面向对象 程序设计与应用开发教程

新世纪应用型高等教育教材编审委员会 组编

主编 匡泰 王岩

副主编 陈景强 刘世华 谢进军 杨柯

参编 靳新 翁正秋



大连理工大学出版社

图书在版编目(CIP)数据

Java 面向对象程序设计与应用开发教程 / 匡泰, 王岩主编. —大连:大连理工大学出版社, 2011. 1
新世纪应用型高等教育软件专业系列规划教材
ISBN 978-7-5611-5956-9

I. ①J… II. ①匡… ②王… III. ①
JAVA 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(2010)第 246558 号

大连理工大学出版社出版

地址:大连市软件园路80号 邮政编码:116023

发行:0411-84708842 邮购:0411-84703636 传真:0411-84701466

E-mail:dutp@dutp.cn URL:http://www.dutp.cn

丹东新东方彩色包装印刷有限公司印刷 大连理工大学出版社发行

幅面尺寸: 185mm×260mm 印张: 18.75 字数: 432千字

印数: 1~2500

2011年1月第1版

2011年1月第1次印刷

责任编辑: 潘弘喆

责任校对: 潘素君

封面设计: 张莹

ISBN 978-7-5611-5956-9 定 价: 35.00 元

前 言

Java 语言是一门面向对象的语言。它摒弃了其他面向对象语言中存在的一些难于控制的技术和非面向对象特性,如多继承机制和独立于类之外的方法。因此,与其他面向对象语言如 C++ 相比,它显得更纯粹。从这点上讲,Java 语言是相对容易学习和掌握的。

Java 语言又是一门不断发展的语言。从 JDK1.0 开始,Java 的核心 API 已经历了数次变革,基于 Java 语言的各种技术已经扩展到信息领域的每个角落。应用 J2SE (Java 2 Platform Standard Edition),Java 可以用来开发桌面应用程序;应用 J2ME (Java 2 Platform Micro Edition),Java 可以开发嵌入式移动应用程序;应用 J2EE (Java 2 Platform Enterprise Edition),Java 可以用来开发企业级服务器应用程序。另外还有 Java Card 和 Java Web Services 等其他相关技术。这些技术发展迅速,也为 Java 语言的推广应用注入新的生命力。对于当今的程序员来说,不懂 Java 几乎寸步难行。

本教材是为高校培养应用型人才而编写的,由具有丰富教学经验和实际工程经历的双师型一线教师编写。全书共 11 章,分为基础篇和应用篇,其中 1~4 章为基础篇,主要讲解 Java 基本语法和面向对象的编程思想;5~11 章为应用篇,主要以案例为线索将知识融为一体,充分体现编程能力和实践能力的培养。

本教材具有以下特点:

1. 教材知识编排新颖、实用。针对应用型人才的需要和学生的认知特点设计和编写。特别是应用篇,每章首先明确教学目标和教学要求,然后以实例带动知识点,以节为单位结合实例介绍 Java 各知识点使用方法,以提高读者的学习兴趣。同时选用的案例均来自实际应用,

这样就可以把学习的内容快速转化为学生的实际编程经验。

2. 教学案例精彩而丰富。专门精心设计了十几个典型的综合性案例。每个案例都做到了相关知识点针对性强,程序设计规范,功能实现完整,具有很强的应用与借鉴价值。

3. 注重面向对象思想的培养。由于面向对象程序设计思想的建立是一个逐步完善的过程,从人的认识规律出发,本教材在编排上遵循“从简单到复杂”、“从抽象到具体”的原则,较好地将面向对象程序设计思想通过层层拓展的方式展现给读者。

4. 精简理论、强化实践。对于语言类课程的学习,最重要的是培养一种编程的感觉和编程的能力,所以在编写本书时,选用了实用的典型案例,尽可能做到“精简理论、强化实践”,从而达到“培养能力,面向应用”的目的。

本书可以作为高校、培训机构的 Java 语言编程基础教材,也可以作为 Java 爱好者的自学教材。

由于编者水平有限,加之时间仓促,书中难免会存在缺点与不足,希望广大读者与教师提出宝贵意见,并将意见和建议及时反馈给我们,以便下次修订时改进。

所有意见和建议请发往:dutpbk@163.com

欢迎访问我们的网站:<http://www.dutpgz.cn>

联系电话:0411-84707492 84706104

编者

2011年1月

目 录

上 篇 基础篇

第 1 章 认识 Java	3
1.1 Java 概述	3
1.1.1 Java 的由来及其发展	3
1.1.2 Java 的特点	4
1.2 Java 的开发环境	5
1.2.1 JDK 的下载与安装	5
1.2.2 环境变量的设置	5
1.2.3 JDK 介绍	7
1.3 Java 独立应用程序	8
1.3.1 第一个 Java 程序	8
1.3.2 Java 程序的基本结构	8
1.3.3 Java 程序的编译与执行	9
1.4 Java 的相关概念	10
1.4.1 Java 编译器与 Java 虚拟机	10
1.4.2 Java 平台的构成	11
1.4.3 Java 常用开发工具(IDE)	11
1.5 NetBeans 的下载、安装与使用	12
1.5.1 NetBeans 的下载与安装	12
1.5.2 NetBeans 的使用	14
实验 1 熟悉 Java 运行环境(J2SDK)	16
习 题	17
第 2 章 Java 语言基础	18
2.1 Java 语言的基本元素	18
2.1.1 标识符	18
2.1.2 关键字	19
2.1.3 保留字	20
2.1.4 分隔符	20
2.2 数据类型	21
2.2.1 基本数据类型	22
2.2.2 常 量	22

2.2.3 变 量	23
2.3 运算符与表达式	27
2.3.1 算术运算符与表达式	27
2.3.2 关系运算符与表达式	29
2.3.3 逻辑运算符与表达式	29
2.3.4 条件运算符与表达式	30
2.3.5 位运算符与表达式	31
2.3.6 赋值运算符与表达式	32
2.3.7 运算符的优先级与表达式的类型提升	33
2.4 Java 语句及其控制结构	35
2.4.1 Java 语句	35
2.4.2 选择语句	36
2.4.3 循环语句	40
实验 2 基本运算与程序控制结构	48
习 题	50
第 3 章 Java 面向对象的基础	53
3.1 面向对象的思想原理	54
3.2 类的基本概念	55
3.2.1 类的定义	56
3.2.2 成员变量与成员方法	57
3.2.3 构造方法	58
3.3 对象(实例)的产生和使用	59
3.3.1 对象的产生	59
3.3.2 对象的使用	60
3.4 类的封装	61
3.4.1 成员的访问控制	61
3.4.2 静态变量和静态方法	64
3.5 类的继承(Inheritance)	67
3.5.1 继承的产生(extends 关键字)	67
3.5.2 继承与构造方法	69
3.6 类的多态	70
3.6.1 方法覆盖	70
3.6.2 方法重载	71
3.7 类的抽象(Abstract)	72
3.8 接口与包	76
3.8.1 接 口	76
3.8.2 包	79
3.9 关键字 final、this 和 super	82
3.9.1 final 的作用	82

3.9.2 this 和 super	84
3.10 Java 标准类库	86
3.10.1 Math 类	86
3.10.2 Date 类	88
3.11 Java 帮助文档的使用	92
实验 3 类与对象	94
实验 4 方法的重载及方法的覆盖	97
实验 5 实例成员、类成员和包	98
习 题	101
第 4 章 Java 异常处理	105
4.1 异常处理概念	105
4.2 异常分类	106
4.3 异常处理机制	106
4.3.1 系统自动处理异常	106
4.3.2 异常处理结构	109
4.3.3 声明异常(throws)	111
4.3.4 异常的创建和抛出(throw)	112
实验 6 异常处理	113
习 题	116

下 篇 应 用 篇

第 5 章 Java 图形用户界面编程	119
5.1 图形界面编程知识储备篇	119
5.1.1 图形用户界面和命令行用户界面	119
5.1.2 AWT 组件	121
5.1.3 AWT 事件机制	130
5.1.4 使用 NetBeans 进行可视化开发	137
5.2 图形界面应用开发篇	141
5.2.1 初战篇——使用 AWT 制作猜数字游戏	141
5.2.2 晋级篇——使用 AWT 制作简单文本编辑器	144
应用拓展	147
习 题	147
第 6 章 常用类编程	148
6.1 常用类编程知识储备篇	148
6.1.1 数 组	148
6.1.2 字 符 串	153
6.1.3 集 合	158
6.2 名片夹应用开发篇	169

应用拓展	191
习 题	191
第 7 章 Java 多线程	192
7.1 线程编程知识储备篇	192
7.1.1 线程的状态和生命周期	192
7.1.2 线程的常用方法	194
7.1.3 线程的实现	194
7.2 线程应用开发篇	197
应用拓展	199
习 题	199
第 8 章 网络编程——聊天室的设计与开发	202
8.1 网络编程知识储备篇	202
8.1.1 网络编程基础	202
8.1.2 基于 Socket 的网络编程	208
8.1.3 基于 UDP 协议的 Socket 通信	214
8.2 聊天室应用开发篇	216
应用拓展	223
习 题	223
第 9 章 Java 画图	224
9.1 画图编程知识储备篇	224
9.2 画图应用开发篇	233
9.2.1 初战篇——绘制一辆可以移动的坦克	233
9.2.2 初战篇——实现坦克的炮弹发射功能	236
应用拓展	240
习 题	241
第 10 章 网络坦克大战游戏制作	242
10.1 初战篇——简单坦克设计的面向对象实现	242
10.2 晋级篇——坦克对战功能的实现	255
10.3 扩展篇——坦克网络对战功能的实现	265
应用拓展	277
第 11 章 Java 数据库连接	278
11.1 JDBC 知识储备篇	278
11.1.1 JDBC 基础	278
11.1.2 使用 JDBC 操作数据库	282
11.2 JDBC 应用开发篇	284
应用拓展	290
习 题	291
参考文献	292

上
篇

基础篇

● 内容提要

Java 语言是由 Sun MicroSystem 公司开发的一种面向对象的程序设计语言,现已成为最具吸引力的编程语言。Java 语言的出现,引起了软件开发的变革,为 IT 业增添了新的活力。

本章主要介绍 Java 的发展过程及其特点,并介绍如何使用 JDK (Java Development Kit,Java 开发工具包)对 Java 程序进行编译和运行。

本章介绍了目前常用的 Java 集成开发工具,并且本文将采用 NetBeans 开发平台进行程序设计。NetBeans IDE 是一个强大的集成化 Java 开发平台,利用它开发人员在编写、编译、调试和部署 Java 应用程序时更为方便。

● 重点难点

1. 掌握 JDK 中环境变量的设置。
2. 使用 NetBeans 开发平台进行程序设计。

● 学习目标

1. 了解 Java 的发展过程及特点。
2. 掌握 JDK 的下载、安装以及环境变量的设置。
3. 了解 Java 语言开发工具 NetBeans。

1.1 Java 概述

Java 是 Sun MicroSystem 公司开发的新一代编程语言,使用它可以在不同的机器以及不同操作平台的网络环境中开发软件。Java 的诞生对整个计算机业产生了深远的影响,对传统的计算模型提出了新的挑战。

1.1.1 Java 的由来及其发展

Java 起源于 20 世纪 90 年代初 Sun 公司的一个叫 Green 的项目,该项目的目标是开发消费性电子产品的分布式软件系统,使电器更加智能化。项目小组在“简单、可靠、紧凑并易于移植”的框架内开发了一个小型的计算机语言,用于编写电子产品中的程序,当时该语言命名为 OAK。在网络出现之前,OAK 可以说是默默无闻,甚至差点夭折,但是,网络的出现改变了 OAK 的命运。

在 Java 出现以前,Internet 上的信息内容都是一些乏味死板的 HTML 文档。这对于那些迷恋于 Web 浏览的人们来说简直不可容忍,他们迫切希望能在 Web 中看到一些交互式的内容,开发人员也极希望能够在 Web 上创建一类无需考虑软硬件平台就可以执行的应用程序,当然这些程序还要有极大的安全保障。对于用户的这种要求,传统的编程语言显得无能为力。Sun 的工程师敏锐地察觉到了这一点,从 1994 年起,他们开始将 OAK 技术应用于 Web 上,并且开发出了 HotJava 的第一个版本。1995 年,Sun 公司正式以 Java 这个名字推出。Internet 的迅猛发展以及万维网的快速增长极大地促进了 Java 语言研制的进展,使得它逐渐成为 Internet 上受欢迎的开发语言,一些著名的计算机公司纷纷购买了 Java 语言的使用权,如 MicroSoft、IBM、Netscape、Novell、Apple、DEC 和 SGI 等。因此,Java 语言被美国的著名杂志 PC Magazine 评为 1995 年十大优秀科技产品之一。Java 成了一颗耀眼的明星。

1.1.2 Java 的特点

Java 是一种跨平台,适合于分布式计算环境的面向对象编程语言。具体来说,它具有简单、面向对象、分布式、解释型、平台无关、可移植、高性能、多线程、动态性、可靠和安全等特性。

1. 面向对象

Java 语言是一种纯面向对象语言。面向对象其实是现实世界模型的自然延伸,现实世界中任何实体都可以看做是对象,对象之间通过消息相互作用。另外,现实世界中任何实体都可归属于某类事物,任何对象都是某一类事物的实例。如果说传统的过程式编程语言是以过程为中心、以算法为驱动的话,面向对象的编程语言则是以对象为中心、以消息为驱动。所有面向对象编程语言都支持三个概念:封装、多态和继承,Java 也不例外。

2. 分布式

Java 是面向网络的语言。通过它提供的类库可以处理 TCP/IP 协议,用户可以通过 URL 地址在网络上很方便地访问其他对象。

3. 解释型

Java 解释器可以直接在任何已移植该解释器的机器上解释、执行 Java 字节码。字节码本身携带了许多编译时的信息,使得连接过程更加简单。

4. 跨平台移植性

Java 程序执行前先编译成字节码文件,然后加载与校验,再解释成不同的机器码执行,即采用 Java 虚拟机屏蔽了具体的平台环境,所以只要安装 Java 运行系统,Java 可以在任何处理器上运行。

跨平台是确保程序可移植的主要原因,同时 Java 还严格规定数据类型和类结构的标准,使程序可以适用于所有的计算机平台,Java 的链接库还定义了一些可移植的接口以及这些类在不同计算机平台上定义的结构,这些使 Java 具有良好可移植性。

5. 高性能

字节码可以迅速地被转换成机器码。随着 Java 解释器的改进和升级,Java 字节码的执行效率也逐渐增强。

6. 多线程

多线程机制使得应用程序能够并行执行,而且同步机制保证了对共享数据的正确操作。通过多线程机制,程序设计者可以分别用不同的线程完成特定的行为,而不需要采用全局的事件循环机制就能很容易地实现网络上的实时交互行为。

7. 动态性

Java 的设计使它适合于一个不断发展的环境。在类库中可以自由地加入新的方法和实例变量而不会影响用户程序的执行;而且 Java 通过接口来支持多重继承,使之比严格的类继承具有更灵活的方式和扩展性。

8. 可靠性和安全性

Java 设计的目的是要能够使用于网络/分布式运算环境,为此,Java 非常强调安全性和可靠性,以确保建立无病毒且不会被侵入的系统。Java 不支持指针,一切对内存的访问都必须通过对象的实例变量来实现,这样就防止程序员使用木马等欺骗手段访问对象的私有成员,同时也避免了指针操作中容易产生的错误。

1.2 Java 的开发环境

1.2.1 JDK 的下载与安装

运行 Java 程序需要安装 Sun 公司提供的 JDK(Java Development Kit,Java 开发工具包)。JDK 是整个 Java 的核心,包括了 Java 运行环境、Java 工具和 Java 基础类库。

在 Sun 公司的网站可以找到 JDK 的最新版本。当前的 JDK 最新版本是 JDK1.6, JDK 下载地址为:<http://java.sun.com/javase/downloads/index.jsp>,界面如图 1-1 所示,下载文件为:jdk-6u18-windows-i586.exe。

要在 Windows 环境下安装 JDK,运行 jdk-6u18-windows-i586.exe 文件,根据提示进行安装。在安装过程中,选择较为简单的路径,例如,本教材安装地址为 C:\jdk1.6.0_18,其目的是避免在操作时需要拼写复杂的目录名给操作带来不便。

1.2.2 环境变量的设置

在安装完 JDK 之后不能立刻使用,还需要设置环境变量。设置环境变量的目的在于让系统自动查找所需的命令。其具体步骤如下:

(1)右键单击“我的电脑”,在弹出的快捷菜单中选择“属性”选项,进入“系统属性”对话框,如图 1-2 所示。

(2)在“系统属性”对话框中,选择“高级”选项卡,单击【环境变量】按钮,弹出“环境变

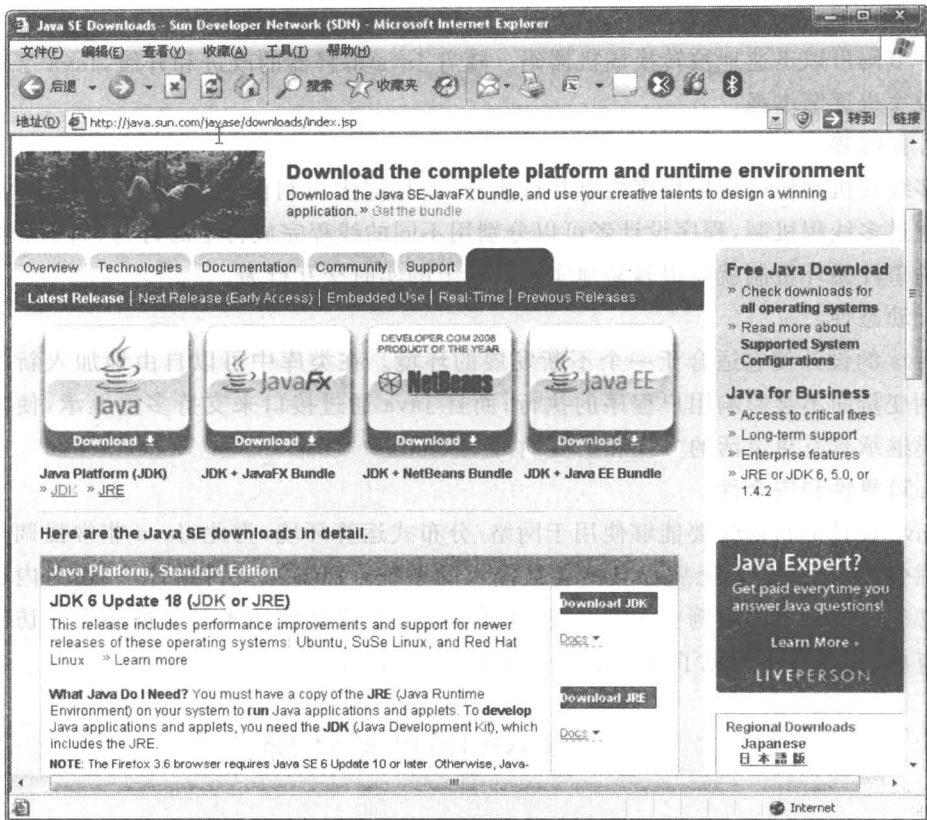


图 1-1 JDK 下载界面

量”对话框,如图 1-3 所示。

(3)单击“系统变量”选项组下的【新建】按钮,打开“新建系统变量”对话框。

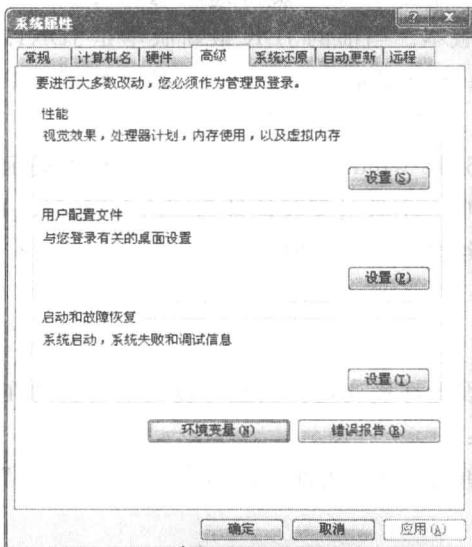


图 1-2 “系统属性”对话框



图 1-3 “环境变量”对话框

(4)为了方便引用,设置 Java_Home 为 JDK 的安装路径。其操作方法为:在“变量名”文本框中输入“Java_Home”,在“变量值”文本框中输入“C:\jdk1.6.0_18”。Java_Home 的设置如图 1-4 所示。

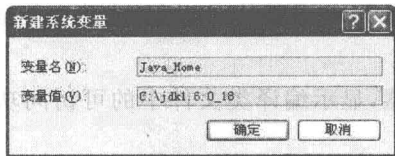


图 1-4 Java_Home 的设置

(5)设置 PATH 变量是为了用户能在系统的任何地方运行 Java 程序。其操作方法为:在“变量名”文本框中输入“PATH”,在“变量值”文本框中输入“.;%path%;%Java_Home%\bin;%Java_Home%\jre\bin;”。

(6)设置 CLASSPATH 变量是当用户开发的 Java 程序需要引用别人写好的类时,让 Java 解释器知道到哪里去找这个类。通常,Sun 为用户提供了一些额外的丰富的类包,一个是 dt.jar,一个是 tools.jar,这两个 jar 包都位于 C:\jdk1.6.0\lib 目录下,所以通常要把这两个 jar 包加到“CLASSPATH”环境变量中。其操作方法为:在“变量名”文本框中输入“CLASSPATH”,在“变量值”文本框中输入“.;%Java_Home%\lib\dt.jar;%Java_Home%\lib\tools.jar;”。

(7)安装好 JDK 之后,选择“开始”→“运行”命令,在文本框中输入“cmd”命令后打开 DOS 窗口。在窗口中输入“java -version”,如果看到如图 1-5 所示的提示信息,则说明安装正确。

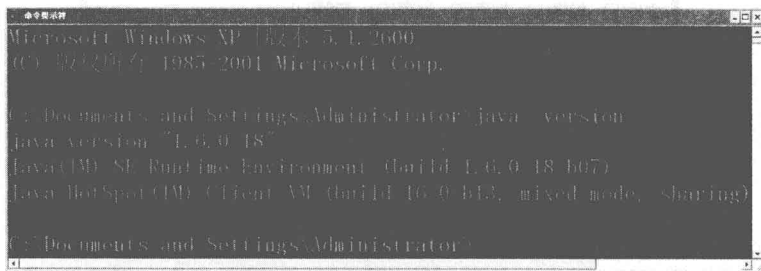


图 1-5 Java 版本提示信息

1.2.3 JDK 介绍

在 JDK 中,主要包括 Java 编译器、Java 虚拟机、Java 文档生成器、Java 调试器等。下面介绍几种常用的 JDK 组件:

- (1)javac:编译器,将源程序转成字节码。
- (2)jar:打包工具,将相关的类文件打包成一个文件。
- (3)javadoc:文档生成器,从源码注释中提取文档。
- (4)jdb:debugger,查错工具。

(5)java:运行编译后的 java 程序(. class 后缀)。

(6)appletviewer:小程序浏览器,一种执行 HTML 文件上的 Java 小程序的 Java 浏览器。

(7)Javah:产生可以调用 Java 过程的 C 过程,或建立能被 Java 程序调用的 C 过程的头文件。

(8)Javap:Java 反编译器,显示编译类文件中的可访问功能和数据,同时显示字节码含义。

(9)Jconsole:Java 进行系统调试和监控的工具。

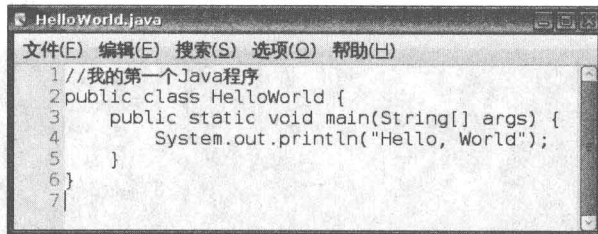
1.3 Java 独立应用程序

1.3.1 第一个 Java 程序

安装并配置好 JDK 之后,就可以开始编译和运行程序了。下面通过一个具体的程序对 Java 代码有一个初步的认识。

编写 Java 程序的工具很多,最简单的就是文本编辑器或者写字板。首先开启一个写字板程序,然后新建一个文本文件,文件名为 HelloWorld.java,表示这是一个 Java 源程序。为了后面测试方便,该文件保存在 D 盘根目录下。

在写字板中输入以下文字,如图 1-6 所示。



```
1//我的第一个Java程序
2public class HelloWorld {
3    public static void main(String[] args) {
4        System.out.println("Hello, World");
5    }
6}
7|
```

图 1-6 第一个 Java 程序 HelloWorld

1.3.2 Java 程序的基本结构

Java 程序由程序注释、类的定义、方法的声明、程序的执行等部分构成。Java 源文件的命名格式是类名加后缀名(. java)。英文习惯将类名第一个字母大写。

首先看源文件第 1 行://我的第一个 Java 程序

第 1 行是注释文字,一个好的程序都应该包含足够的注释。也就是说,这一行文字不是程序的正式内容,只是给看这个程序代码的人一些提示,或者纯粹就是编程的时候自己做的一个记号。

第 2~6 行是 Java 程序的全部内容:

```
2 public class HelloWorld {
```