



“十二五” 高职高专计算机类专业规划教材

JAVA  
CHENGXU SHEJI  
Java程序设计

陈桂生 张 哲 主编

国家级示范性高职高专重点专业教改教材  
行动导向 任务驱动



 河南科学技术出版社



JAVA

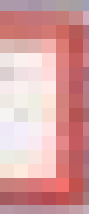
应用案例解密

# Java程序设计

第2版

清华大学出版社

清华大学出版社



“十二五”高职高专计算机类专业规划教材

# Java 程序设计

陈桂生 张 哲 主编

河南科学技术出版社

· 郑州 ·

## 内 容 提 要

Java 作为一种优秀的编程语言,具有面向对象、与平台无关、安全、稳定和多线程等特点,不仅适于开发大型的应用程序,而且特别适合于 Internet 应用的开发,Java 已成为网络时代最重要的编程语言之一。

本书以项目为载体,注重可读性和实用性,项目 1 至项目 4 是对 Java 语言基本语法的讲解,针对每一部分分配了大量的例题;项目 5 至项目 8 是通过四个具体的项目,更好地学习 Java 中一些具体的类和对象的使用方法和功能,使读者能更好地掌握 Java 的编程技巧。

全书共八个项目,分别介绍了 Java 的基本数据类型、语句、类、对象、内部类、异常处理、字符串、实用类、AWT 组件、多线程、输入输出流、网络编程基础等内容。

本书不仅可以作为高等职业院校计算机类专业的教材,也可供自学者及软件开发人员参考使用。

### 图书在版编目(CIP)数据

Java 程序设计/陈桂生,张哲主编. —郑州:河南科学技术出版社,2011.9

“十二五”高职高专计算机类专业规划教材

ISBN 978 - 7 - 5349 - 5312 - 5

I. ①J… II. ①陈…②张… 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(2011)第 186439 号

---

出版发行:河南科学技术出版社

地址:郑州市经五路 66 号 邮编:450002

电话:(0371)65788613 65788629

网址:www.hnstp.cn

策划编辑:范广红 崔军英

责任编辑:徐素军

责任校对:李 军

封面设计:张 伟

版式设计:栾亚平

责任印制:张艳芳

印 刷:郑州文华印务有限公司

经 销:全国新华书店

幅面尺寸:185 mm × 260 mm 印张:16 字数:390 千字

版 次:2011 年 9 月第 1 版 2011 年 9 月第 1 次印刷

定 价:28.00 元

---

如发现印、装质量问题,影响阅读,请与出版社联系调换。

# 前 言

在多年的实际教学中,编者曾使用了多本 Java 程序设计教材,这些教材各有各的特色,但是实际的情况是学生听课的时候可以听懂,学完之后却不知道如何去应用,自己用 Java 语言编写一个小程序还很困难.这几年一直在提倡项目驱动式的教学方法,编者在深入研究了项目驱动的教学理念后,在实际教学中也采用了项目驱动式的教学方法,逐渐发现这种教学方法的效果确实比较好,学生在学习了 Java 语言的基本语法之后,通过覆盖不同知识点的几个小项目,逐渐能够使用 Java 语言来编写一些小程序,即能够做到学有所用.我们建设了立体化的教学资源包,本门课程也被评为院级精品课程,现在把对这门课程的具体设计以教材的形式展示出来,希望对学生学习 Java 语言编程有更大的帮助.

本教材的编写是按照 Java 知识点来划分项目的,基本知识点包括:基本数据类型、语句、类、对象、接口、内部类、AWT 组件、Swing 组件、多线程、输入输出流、多线程、Java 与网络、JDBC 等内容,将基本内容分类整理,穿插到每个项目中,以知识点学习驱动项目,学习项目的同时仍然强调知识点.在实际应用中我们需要解决各种问题,不同的问题需要不同的知识点,即实际中是以问题为驱动的.所以本书以应用为中心,围绕如何解决应用中的问题进行讲解,即完成相关功能需要什么知识点,就介绍什么知识点.

全书共由八个项目组成.项目 1 介绍 Java 语言的来历、地位和重要性,对 Java 平台进行了详细讲解.项目 2 介绍 Java 逻辑编程,逻辑编程是任何一种计算机语言的基础,Java 也不例外,主要介绍基本数据类型和基本语句.项目 3 介绍面向对象的编程基础,主要讲解类、对象的定义和使用方法.项目 4 介绍 Java 面向对象高级编程,主要介绍 Java 作为一种面向对象的编程语言,所特有的规范和基础.项目 5 介绍计算器小程序的开发,主要讲解 AWT 组件、Swing 组件和监视器的使用方法.项目 6 介绍文本编辑器的开发,主要讲解对话框和输入输出流的使用.项目 7 介绍网络聊天室的开发,主要讲解套接字、多线程等网络编程基础.项目 8 介绍学生管理信息系统的开发,主要讲解简单的需求分析、Java 与数据库的连接,JDBC 等.

希望本书能对读者学习 Java 有所帮助,在编写过程中,我们力求写出 Java 的精髓,但因作者水平有限,因而书中不妥之处,敬请读者批评指正.

编 者

2011 年 7 月

## 《Java 程序设计》编写人员名单

主 编 陈桂生 张 哲  
副主编 雷 燕 丰 雁 唐文娟  
编 委 (按姓氏笔画排序)  
马 奇 丰 雁 代玉梅  
张 哲 陈桂生 唐文娟  
雷 燕 魏翠萍

# 目 录

项目 1	Java 入门	(1)
任务 1	了解 Java	(1)
任务 2	配置 Java 运行环境	(4)
任务 3	编写第一个 Java 程序	(9)
项目 2	Java 逻辑编程	(16)
任务 1	学习 Java 基本数据类型	(16)
任务 2	使用运算符和表达式	(20)
任务 3	Java 控制语句	(24)
任务 4	学习数组	(34)
项目 3	Java 面向对象编程基础	(40)
任务 1	学习类和对象	(40)
任务 2	学习方法重载	(54)
任务 3	学习包的使用	(59)
项目 4	Java 面向对象高级编程	(67)
任务 1	学习继承的使用	(67)
任务 2	学习区分抽象类、最终类和内部类	(75)
任务 3	学习接口的使用	(78)
任务 4	学习异常类的使用	(82)
任务 5	学习基础类的使用	(98)
项目 5	简单计算器的开发	(107)
任务 1	计算器功能描述	(107)
任务 2	理论指导	(109)
任务 3	计算器总体设计	(124)
任务 4	详细设计与编码实现	(124)
任务 5	计算器程序清单	(129)
任务 6	计算器运行与发布	(139)
项目 6	文本编辑器的开发	(141)
任务 1	文本编辑器功能描述	(141)
任务 2	理论指导	(141)
任务 3	文本编辑器总体设计	(147)

任务 4	详细设计与编码实现	(150)
任务 5	文本编辑器程序清单	(155)
任务 6	文本编辑器的运行与发布	(174)
项目 7	网络聊天室的开发	(176)
任务 1	功能描述	(176)
任务 2	理论指导	(177)
任务 3	总体设计	(193)
任务 4	聊天室详细设计	(203)
任务 5	代码实现	(205)
任务 6	程序的运行和发布	(223)
项目 8	学生管理信息系统的开发	(224)
任务 1	系统分析与描述	(224)
任务 2	理论指导	(225)
任务 3	系统的需求分析	(229)
任务 4	系统总体设计	(231)
任务 5	详细设计与编码	(236)
任务 6	系统部分程序清单	(241)
任务 7	系统运行与发布	(248)
参考文献		(249)



# 项目 1 Java 入门

## 学习目标

Java 是一种简单的,面向对象的,分布式的,健壮安全的,结构中立的,可移植的,性能优异、多线程的动态语言。Java 的诞生是对传统计算机模式的挑战,对计算机软件开发和软件产业都产生了深远的影响。

本项目主要学习 Java 程序设计语言和 Java 平台,JDK 的下载和安装,Java 应用程序的编写、编译和运行等相关知识。

## 任务 1 了解 Java

Java 是由 Sun 公司于 1995 年 5 月推出的 Java 程序设计语言(下称“Java 语言”)和 Java 平台的总称。用 Java 实现的 HotJava 浏览器(支持 Java applet)显示了 Java 的魅力:跨平台、动态的 Web、Internet 计算。从此,Java 被广泛接受并推动了 Web 的迅速发展,常用的浏览器现在均支持 Java applet。

### 1.1.1 Java 语言简介

Java 语言其实最早是诞生于 1991 年,起初被称为 Oak 语言,是 Sun 公司为一些消费性电子产品而设计的一个通用环境。他们最初的目的只是为了开发一种独立于平台的软件技术,而且在网络出现之前,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,而 Sun 公司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。当 Sun 公司 1995 年正式推出 Java 的时候,几乎所有的 Web 开发人员都惊叹到:噢,这正是我想要的!于是 Java 成了一颗耀眼的明星,丑小鸭一下变成了白天鹅。

Java 的开发环境有不同的版本,如 Sun 公司的 Java Development Kit(Java 开发工具包),简称 JDK。后来微软公司推出了支持 Java 规范的 Microsoft Visual J++,简称 VJ++。

Java 语言的特点主要表现在以下几个方面:

(1)Java 语言是简单的。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、自动的强制类型转换等。特别是 Java 语言不使用

指针,并提供了自动的垃圾回收机制,使得程序员不必为内存管理而担忧。

(2)Java 语言是面向对象的。Java 语言提供类、接口和继承,为了简单起见,只支持类之间的单继承,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 语言是一个纯的面向对象程序设计语言。

(3)Java 语言是分布式的。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、URL Connection、Socket、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。

(4)Java 语言是健壮的。Java 的强类型机制、异常处理、垃圾的自动回收等是 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 更具健壮性。

(5)Java 语言是安全的。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(类 ClassLoader),如分配不同的名字空间以防替代本地的同名类、字节代码检查,并提供安全管理机制(类 Security Manager)作为 Java 应用设置的安全哨兵。

(6)Java 语言是体系结构中立的。Java 程序(后缀为 java 的文件)在 Java 平台上被编译为体系结构中的字节码格式(后缀为 class 的文件),然后可以在实现这个 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)Java 语言是可移植的。这种可移植性来源于体系结构的中立性。另外,Java 还严格规定了各个基本数据类型的长度。Java 系统本身也具有很强的可移植性,Java 编译器是用 Java 来实现的,Java 的运行环境是用 ANSI C 来实现的。

(8)Java 程序既是编译型的,又是解释型的。如前所述,Java 程序在 Java 平台上被编译为字节码格式,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)Java 是高性能的。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)Java 语言是多线程的。在 Java 语言中,线程是一种特殊的对象,它必须由 Thread 类或其子(孙)类来创建。通常有两种方法来创建线程:第一,使用型构为 Thread(Runnable)的构造函数将一个实现了 Runnable 接口的对象包装成一个线程;第二,从 Thread 类派生出子类并重写 run 方法,使用该子类创建的对象即为线程。值得注意的是 Thread 类已经实现了 Runnable 接口,因此,任何一个线程均有它的 run 方法,而 run 方法中包含了线程所要运行的代码。线程的活动由一组方法来控制。Java 语言支持多个线程的同时执行,并提供多线程之间的同步机制(关键字为 synchronized)。

(11)Java 语言是动态的。Java 语言的设计目标之一是适应于动态变化的环境。Java 程序需要的类能够动态地被载入到运行环境中,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。

Java 语言的优良特性使得 Java 应用具有无比的健壮性和可靠性,这也减少了应用系统

的维护费用。Java 对面向对象技术的全面支持使 Java 平台内嵌的应用程序接口能缩短应用系统的开发时间并降低成本。Java 的编译一次就到处可运行的特性使得它能够提供一个随处可用的开放结构和在多平台之间传递信息的低成本方式。特别是,Java 企业应用编程接口(Java Enterprise API)为企业计算及电子商务应用系统提供了有关技术和丰富的类库。

### 1.1.2 Java 平台简介

Java 平台由 Java 虚拟机(Java Virtual Machine, JVM)和 Java 应用编程接口(Application Programming Interface, API)构成。Java 应用编程接口为 Java 应用提供了一个独立于操作系统的标准接口,可分为基本部分和扩展部分。在硬件或操作系统平台上安装一个 Java 平台之后,Java 应用程序就可以运行。现在 Java 平台已经嵌入了几乎所有的操作系统。这样 Java 程序只需编译一次,就可以在各种系统中运行。Java 应用编程接口已经从 1.1x 版本发展到 1.2 版本。目前常用的 Java 平台是基于 Java1.6 的,最新版本为 Java1.7。

Java 平台分为三个版本:Java 平台标准版(Java Platform Standard Edition, Java SE), Java 平台企业版(Java Platform Enterprise Edition, Java EE)和 Java 平台微型版(Java Platform Micro Edition, Java ME)。

#### 1. Java SE

Java SE 版本以前称为 J2SE。它允许开发和部署在桌面、服务器、嵌入式环境和实时环境中使用 Java 应用程序。Java SE 包含了支持 Java Web 服务开发的类,并为 Java EE 提供基础。

#### 2. Java EE

Java EE 版本以前称为 J2EE。企业版本帮助开发和部署可移植、健壮、可伸缩且安全的服务器端 Java 应用程序。Java EE 是在 Java SE 的基础上构建的,它提供 Web 服务、组件模型、管理和通信 API,可以用来实现企业级的面向服务体系结构(service-oriented architecture, SOA)和 Web 2.0 应用程序。

#### 3. Java ME

Java ME 版本以前称为 J2ME。Java ME 为在移动设备和嵌入式设备(比如手机、PDA、电视机顶盒和打印机)上运行的应用程序提供一个健壮且灵活的环境。Java ME 包括灵活的用户界面、健壮的安全模型、许多内置的网络协议以及对可以动态下载的连接和离线应用程序的丰富支持。基于 Java ME 规范的应用程序只需编写一次,就可以用于许多设备,而且可以利用每个设备的本机功能。

Java 平台主要是由一个编译器(compiler)、一个运行环境(runtime environment)和一个核心的 API 组成。

JVM(Java Virtual Machine):Java 程序并不是直接在本地机器的操作系统上执行,而是通过 JVM 解释成本地的机器语言再执行,其优点是可以保证 Java 代码在 Windows、Linux、Solaris 等操作系统上的可移植性,其代价是执行速度比 C 代码要慢。考虑到软件越来越庞大,而计算机硬件性能越来越快,而且 Web 应用的发展,其代价是可接受的。

JDK(Java Development Kit):Java 平台开发包目前普遍采用 JDK 1.6。

JRE(Java Runtime Environment):Java 运行环境,一般情况下已经集成到 JDK 中,如果用户只是执行 Java 程序,而不进行 Java 代码的开发,则只需要安装 JRE。

## 任务2 配置 Java 运行环境

### 1.2.1 JDK 简介

JDK 是 Java Development Kit 的缩写,是由 Sun 公司提供. 它为 Java 程序提供了基本的开发和运行环境. 自从 Java 推出以来,JDK 已经成为使用最广泛的 Java SDK. JDK 是整个 Java 的核心,包括了 Java 运行环境、Java 工具和 Java 基础的类库. 了解 JDK 是学好 Java 的第一步,而专门运行在 x86 平台的 Jrocket 在服务器端的运行效率要比 Sun JDK 高得多. 从 Sun 的 JDK5.0 开始,提供了泛型等非常实用的功能,其版本也不断更新,运行效率也得到了非常大的提高.

JDK 主要包括以下几个内容:

(1)Java 编译器程序. Java 编译器程序负责编译 Java 源程序,在 Windows 操作系统中,该程序的文件名为 javac.exe.

(2)Java 虚拟机程序. 该程序负责解释和运行 Java 程序,在各种操作系统平台上都有相应的 Java 虚拟机程序,在 Windows 操作系统中,该程序的文件名为 java.exe.

(3)JDK 类库. JDK 类库提供了最基础的 Java 类及各种实用类. java.lang、java.io、java.util、java.awt 和 javax.swing 包中的类都位于 JDK 类库中.

假定 JDK 安装到本地后的根目录为 D:\jdk,则在 D:\jdk\bin 目录下有一个 javac.exe 和 java.exe 文件,它们分别为 Java 编译器程序和 Java 虚拟机程序.

### 1.2.2 JDK 下载和安装(版本 1.6)

#### 1. JDK 的下载

目前常用的 Java 平台是 Java1.6 版本. JDK1.6 的官方下载地址为:<http://java.sun.com/javase/downloads/index.jsp>,打开该链接显示主页面,如图 1.1 所示.

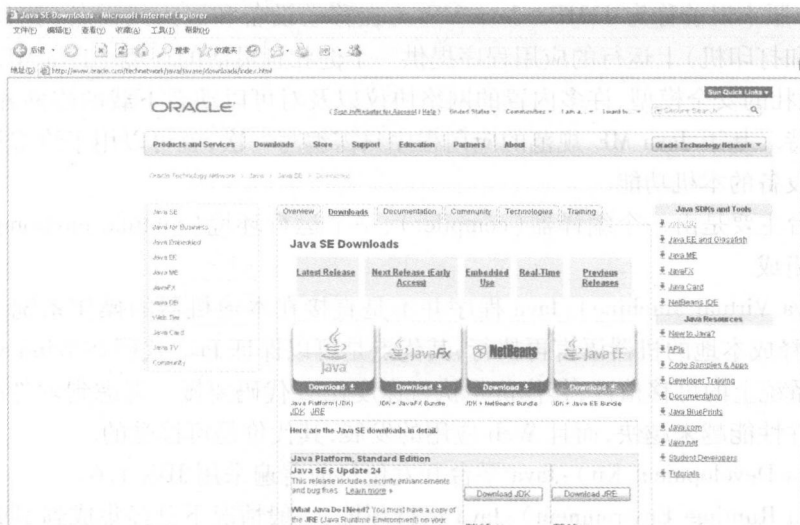


图 1.1 下载主页面

单击 Java 图片或图片下方的 JDK 即可进入下面的页面,如图 1.2 所示。



图 1.2 下载二级页面

首先,在此对话框的“selec...”选项框中选择操作系统版本 Windows;然后选中“[I agree to the Java SE Development Kit 6u24 License Agreement](#)”前面的复选框;最后单击“continue”按钮,进入下面的页面,如图 1.3 所示。

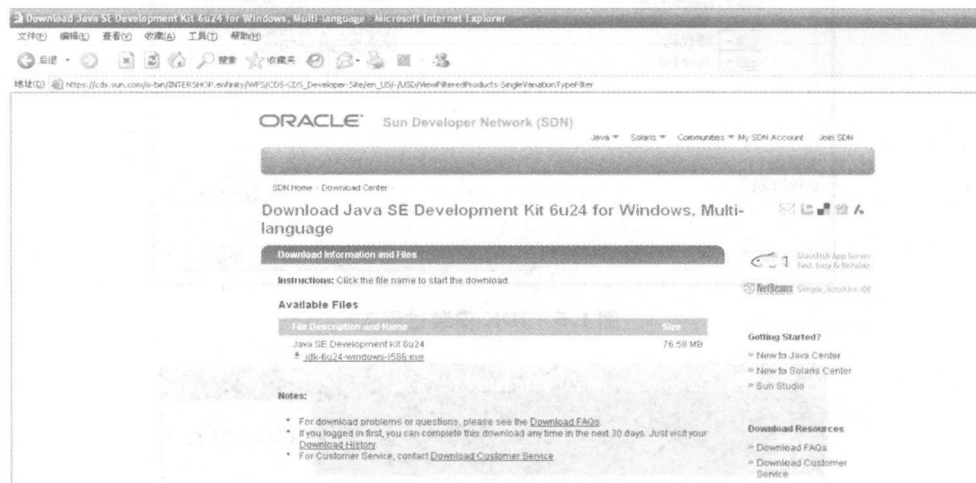


图 1.3 下载三级页面

单击选项中“[jdk - 6u24 - windows - i586.exe](#)”即可免费下载 JDK1.6 版本。

此外,在 JavaThinker.org 网站上也提供了 JDK 的下载,网址为<http://www.javathinker.org/download/software/jdk.rar>。

## 2. JDK 的安装过程

双击 JDK 的安装程序 `jdk - 6u24 - windows - i586.exe`,根据安装向导,选择安装路径。本例安装在 `D:\jdk1.6`,单击“下一步”按钮,完成 JDK 的安装。如图 1.4 ~ 图 1.6 所示。

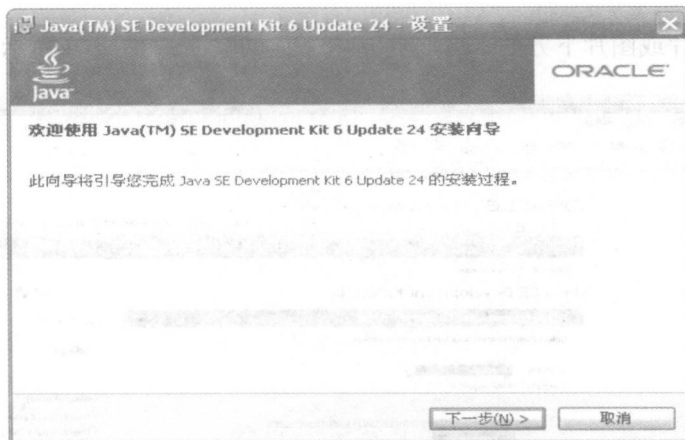


图 1.4 JDK 安装过程 1

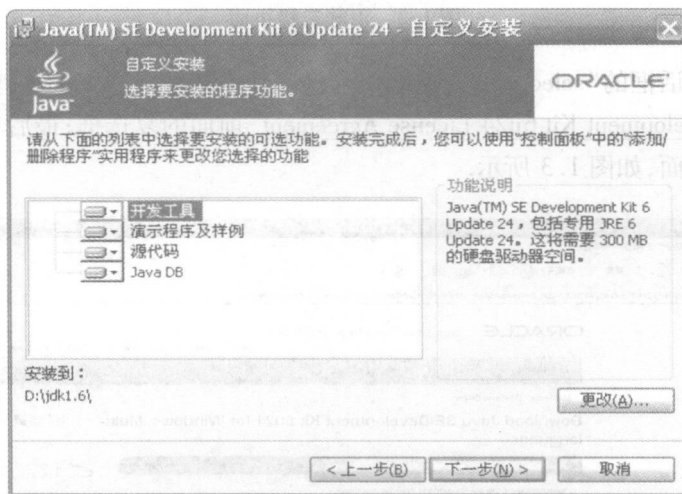


图 1.5 JDK 安装过程 2

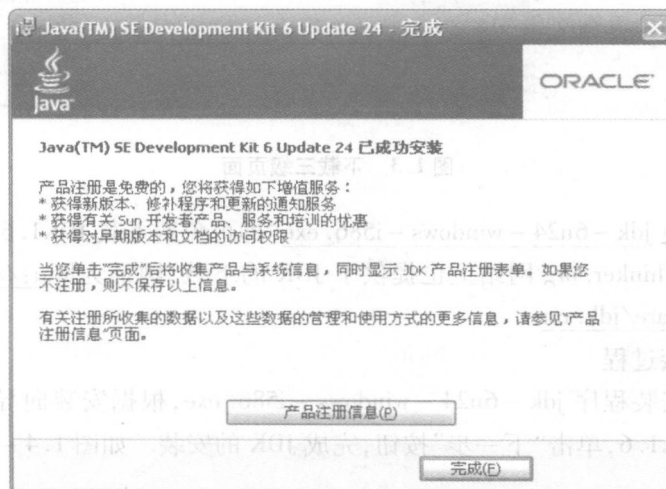


图 1.6 JDK 安装过程 3

JDK 安装完成后,在 D:\jdk\bin 目录下则会有一个 java.exe 和 javac.exe 文件,它们分别为 Java 虚拟机程序和 Java 编译器程序。

在计算机上安装 JDK 的同时也就安装了 Java 运行环境平台。如果只想运行别人的 Java 程序而不进行 Java 程序开发,可以只安装 Java 运行环境 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 的网站免费下载 Java 的 JRE。

### 1.2.3 设置环境变量

为了便于在 DOS 命令行下直接运行 Java 虚拟机程序和 Java 编译器程序,需要设置一下系统环境变量。

如果使用 Win95/98 系统,则在 autoexec.bat 的最后面添加如下 3 行语句:

```
set JAVA_HOME = c:\jdk1.6.0_21\  
set PATH = %JAVA_HOME%\bin;%PATH%  
set CLASSPATH = .;%JAVA_HOME%\lib
```

注意:可用 DOS 命令 notepad c:\autoexec.bat 打开记事本,加入上面的 3 行语句。

如果使用 Windows XP 及以上版本操作系统,则选择“控制面板”→“系统”→“高级”→“环境变量”,在新打开的对话框中需要设置三个系统变量:“JAVA\_HOME”、“Path”、“Classpath”。其中,在没安装过 jdk 的环境下,Path 属性是本来存在的,而 JAVA\_HOME 和 Classpath 是不存在的。

#### 1. JAVA\_HOME

在“新建系统变量”对话框中的“变量名”文本框中输入 JAVA\_HOME,该变量的含义就是 Java 的安装路径,然后,在“变量值”输入刚才安装的路径“D:\jdk1.6”。如图 1.7 所示。

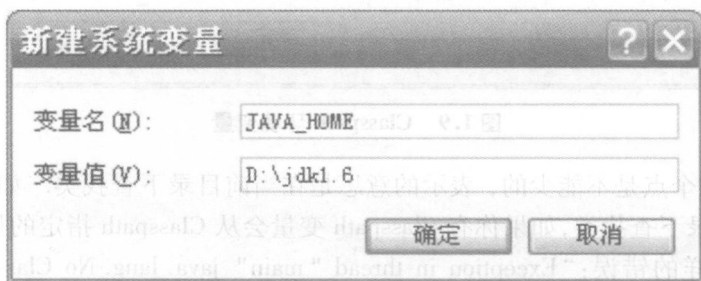


图 1.7 JAVA\_HOME 环境变量

注意:如果安装的路径不是磁盘 D 或者不是在 jdk1.6 这个文件夹下,可对应修改。以下文字都是假定安装在 D:\jdk1.6 这个文件夹里面。

#### 2. Path

在系统变量里面找到 Path,然后单击“编辑”按钮,Path 变量的含义就是系统在任何路径下都可以识别 Java 命令,在“变量值”的最前面加上“.;%JAVA\_HOME%\bin;”,其中“%JAVA\_HOME%”的意思为刚才设置 JAVA\_HOME 的值,也可以直接写上“D:\jdk1.6\bin”。如图 1.8 所示。

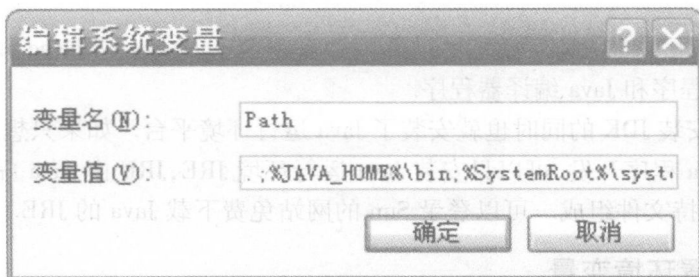


图 1.8 path 环境变量

注意:Java 的环境变量应该设置到最前面,因为环境变量是从前面向后面找的. 系统的环境变量会覆盖用户的环境变量,因为安装程序时有些程序自带的环境变量会覆盖你的 jdk 路径.

### 3. Classpath

单击“新建”按钮,在“变量名”文本框中写上“Classpath”. 该变量的含义是为 Java 加载类(class or lib)路径,只有类在 Classpath 中,Java 命令才能识别. 其“变量值”为“.;%JAVA\_HOME%\lib\dt.jar;%JAVA\_HOME%\lib\toos.jar”,前面要加“.”表示当前路径,与“%JAVA\_HOME%”具有相同的意思. 如图 1.9 所示.

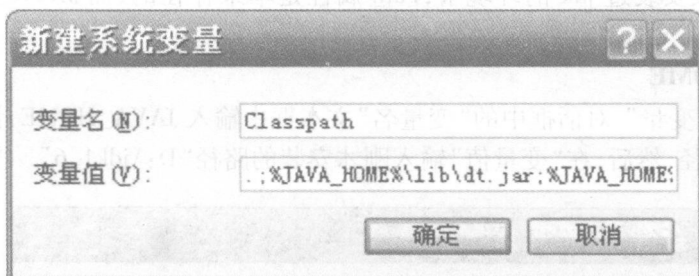


图 1.9 Classpath 环境变量

注意:前面那个点是不能少的,表示的意思是在当前目录下查找类. 如果不设置 Classpath 会从当前目录下查找类,如果你有 Classpath 变量会从 Classpath 指定的目录查找类.

如果你有这样的错误:“Exception in thread "main" java.lang.NoClassDefFoundError: hell”,那么就可能是 Classpath 设置问题.

以上 3 个变量设置完毕,多次单击“确定”按钮,直至属性窗口消失. 下面来验证一下安装是否成功.

首先,打开“开始”→“运行”,输入“cmd”,进入 DOS 系统界面. 在 DOS 命令提示符下输入“java - version”. 如果安装成功,系统会显示 java version jdk"1.6.0". 如图 1.10 所示.

然后,在 DOS 界面中输入“javac”来查看该命令是否合法,同样输入“java”来查看该命令是否合法.



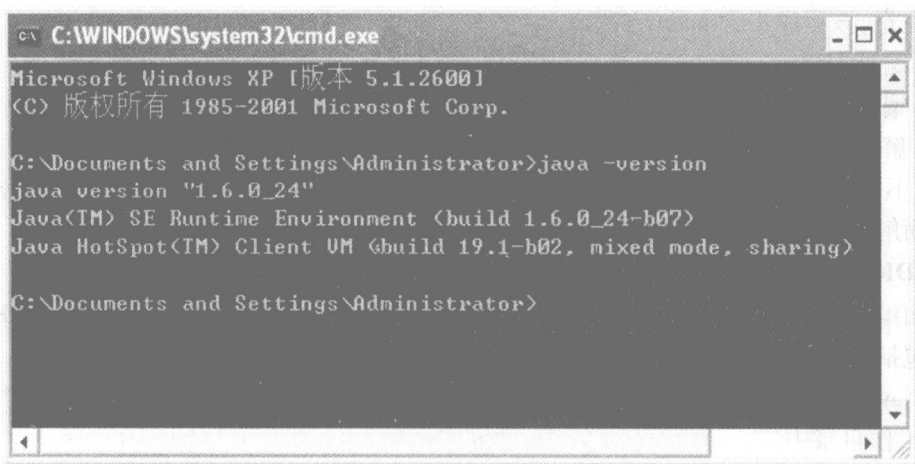


图 1.10 DOS 命令提示符状态

## 任务3 编写第一个 Java 程序

### 1.3.1 Java 程序的编写、编译和运行过程

Java 有两类应用程序:Java Application 和 Java Applet。前者是独立的应用程序,而后者嵌入 HTML 并在浏览器中执行。下面针对这两类应用程序的编程和执行环境进行讨论。

#### 1. Java Application 程序

Java Application 应用程序的编写和执行分 3 步进行:

(1) 编写源代码。首先,要选一个无格式的文本编辑器,如 Windows 的记事本、JCreator、NetBeans、JBuilder、Eclipse、UltraEdit 等。千万不要用 Word 这类带格式的文本编辑器,因为它隐藏有许多 Java 解释器不能识别的格式信息。其次,创建一个文件夹,如 D:\javaProgram 用来存放编写好的 Java 程序。最后,打开编辑器编写程序,写完后以扩展名 .java 存入新建文件夹 D:\javaProgram 中。

(2) 编译源代码。只要安装了 jdk1.6,它已包含有编译器 javac.exe 对 Java 程序进行编译,需要进入 MS-DOS 方式,在 DOS 提示符下输入命令:cd D:\javaProgram,设置运行目录,再输入编译命令:javac 源文件全名(带扩展名.java)。如果没有语法错误,在文件夹 D:\javaProgram 中就会出现一个二进制字节码文件:源文件名.class,它由编译器自动生成;如果源代码有语法错误,则给出错误报告,按行指出错误,编者按报告改正错误后,重复上面编译命令,直至编译成功。

(3) 解释执行。利用 jdk1.6 的解释器 java.exe 解释执行程序。仍在 DOS 方式下,输入命令:java 源文件名(不带 .Java 扩展名),如果执行成功,显示结果;如果执行有错,显示错误报告,设法排错直至获得正确结果。

#### 2. Java Applet 应用程序

Java Applet 应用程序的编写和执行共分 4 步进行:

(1) 编写源代码。这一步与 Java Application 应用程序相同,编辑一个源文件存入指定文件