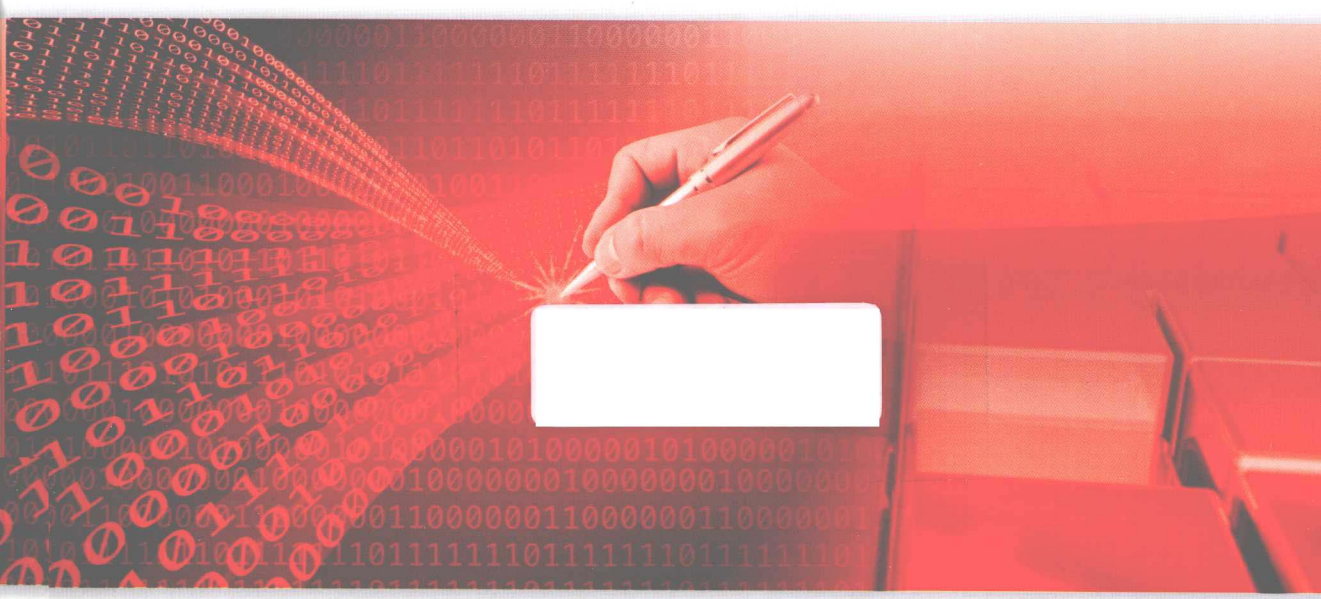
十二五

高等职业教育“十二五”规划教材（计算机类）

Java程序设计项目化教程

郑哲 ◎ 主编



机械工业出版社
CHINA MACHINE PRESS

www.cmpedu.com

配电子课件

高等职业教育“十二五”规划教材（计算机类）

Java 程序设计项目化教程

主　编　郑　哲

副主编　郭双宙　韩越祥　葛茜倩

参　编　蒋　宁　徐志烽　葛科奇

机械工业出版社

本书从 Java 程序设计初学者的角度出发，按照项目化课程的教学方法，通过虚拟的“师生结对编程”的形式详细介绍使用 Java 语言进行程序开发所需掌握的相关知识、技能，并着重强调开发人员需要养成的良好职业习惯。

本书共 9 章，主要内容包括走近 Java 程序、Java 语法基础、Java 面向对象基础、继承和多态、异常、图形、Java I/O、多线程及综合案例等。每章都精心设计了丰富有趣的项目实例，难易适中、趣味性强，便于教学和学生自学。

本书可作为高职院校计算机专业的 Java 程序设计课程的教材，也可作为相关培训机构的 Java 辅导参考书，还可作为从事程序开发、测试及维护的技术人员与编程爱好者的自学图书。

为方便教学，本书配备教学视频、电子课件等教学资源。凡选用本书作为教材的教师均可登录机械工业出版社教育服务网 www.cmpedu.com 免费下载。如有问题请致信 cmpgaozhi@sina.com，或致电 010-88379375 联系营销人员。

图书在版编目（CIP）数据

Java 程序设计项目化教程/郑哲主编. —北京：机械工业出版社，2015.1

高等职业教育“十二五”规划教材，计算机类

ISBN 978-7-111-48867-5

Ⅰ.①J… Ⅱ.①郑… 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（2014）第 293288 号

机械工业出版社（北京市百万庄大街22号 邮政编码 100037）

策划编辑：刘子峰 责任编辑：刘子峰 罗子超

责任校对：黄兴伟 封面设计：陈 沛

责任印制：李 洋

北京华正印刷有限公司印刷

2015 年 2 月第 1 版第 1 次印刷

184mm × 260mm · 18.5 印张 · 449 千字

0001—2500 册

标准书号：ISBN 978-7-111-48867-5

定价：39.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

电话服务

服务咨询热线：010-88379833

读者购书热线：010-88379469

网络服务

机工官网：www.cmpbook.com

机工官博：weibo.com/cmp1952

教育服务网：www.cmpedu.com

金书网：www.golden-book.com

前　言

Java 是 Sun 公司推出的能够跨越多平台的、可移植性较高的一种面向对象的编程语言，自面世以来，凭借其易学易用、功能强大的特点，得到了广泛应用。强大的跨平台特性使 Java 程序可以运行在大部分系统平台上，甚至手机、平板电脑（PDA）等移动电子产品中，真正做到“一次编写，到处运行”。利用 Java 可以编写桌面应用程序、Web 应用程序、分布式系统和嵌入式系统应用程序等，这使得它成为应用范围最广泛的开发语言之一。

本书针对高职高专培养应用技术型人才的要求而编写，提供了 Java 程序设计入门所必备的各类知识，全书共分 9 章。

第 1 章：通过初识 Java、熟悉 Eclipse 开发工具，了解 Java 语言的基本开发环境搭建和配置。

第 2 章：介绍 Java 数据类型、关键字、流程控制、操作符、字符串等内容。

第 3 章：介绍类、对象、方法、构造方法等内容。

第 4 章：介绍数组、集合、接口、继承与多态以及类的高级特性。

第 5 章：介绍异常的概念与分类、创建自定义异常、异常的捕获等内容。

第 6 章：以 SWT/JFace 图形工具为基础，介绍一些常用的 SWT 组件和布局管理器、事件处理。

第 7 章：介绍流的概念和文件的基本操作。

第 8 章：介绍线程的相关知识，包括线程的工作原理、线程的创建、线程的生命周期、线程的同步等。

第 9 章：通过一个完整的微波炉模拟桌面应用程序，运用软件工程的设计思想，让读者学习如何进行软件项目的实践开发。按照“项目需求分析→系统设计→创建项目→实现项目→运行项目→项目打包部署→解决开发常见问题”的流程进行介绍，带领读者一步步体验开发项目的全过程。

本书特点如下：

1）由浅入深，循序渐进。以初级程序员为对象，先从 Java 语言基础学起，再学习 Java 的基础语法，然后学习 Java 的面向对象特性，最后学习开发一个完整项目。本书在讲解过程中步骤详尽、版式新颖，对于重点的语法、API 使用、常见的错误、常用的技巧，都通过图表醒目地进行标记，使读者在阅读时一目了然，从而快速掌握相关知识和语法。

2）结对编程，启发指导。以虚拟化的学生和教师对白呈现结对编程过程。通过教师的启发和指导，让学生掌握相关新知识，并应用这些新的知识以实现相关模块的功能。通过这种模式的训练，在强化学生自己动手能力的同时，培养了学生的独立思考、沟通协调和创新思维的能力。此外，这种身临其境、参与其中的讨论过程，也大大降低了枯燥概念的理解难度，增加了教材的趣味性和可读性，满足高职学生学习的需求。

3）项目驱动，规格表述。本书涵盖项目需求分析、功能模块实现、项目打包发布的完整流程，并通过几个完整项目，反复训练学生实践这一过程以达到提升开发能力的目的，养

成程序员基本的编程习惯和思维习惯。此外，项目的目标明确，任务表述规格化，任务更加精确，让读者在项目开发之前明确自己将要做什么。

4）应用实践，随时练习。每章都提供自测题，读者能够通过对问题的解答重新回顾、熟悉所学知识，举一反三，为进一步的学习做好充分的准备。

本书由宁波城市职业技术学院的郑哲任主编，负责全书的构思和最后的统稿。宁波城市职业技术学院的郭双宙、浙江工业职业技术学院的韩越祥以及浙江工商职业技术学院的葛茜倩任副主编。参与本书编写的还有宁波城市职业技术学院的蒋宁、徐志烽、葛科奇。

在本书编写过程中，我们以科学、严谨的态度，力求精益求精，但由于水平有限，书中错误、疏漏之处在所难免，敬请广大读者批评指正。

编　者

目　录

第1章 走进Java程序

1.1 Java概述

Java是目前非常流行的面向对象程序语言，也是近几年业界最受欢迎的程序语言之一。在世界编程语言排行榜上，Java已经连续好几年排名榜首位置，可见其受欢迎程度。再看一下近几年的就业形势和未来发展的趋势，Java开发人员总是很受用人单位欢迎。此外，它也是开发人员学习其他高级程序语言的基础。

可是对我而言，听说过爪哇咖啡，却从来未接触过Java语言，它难不难学呢？

刚才你说到爪哇咖啡，说起来它还和Java语言真有一段渊源呢。Java是印度尼西亚爪哇岛的英文名称，因盛产咖啡而闻名。“Java之父”James Gosling在为Sun内部开发Green项目（一套为TV机顶盒设计的语言）时，据说在某天喝着爪哇咖啡的时候，突发灵感，创造了Java语言。Sun公司和Java的标志也正是一杯冒着热气的咖啡。而Java的许多类库和工具也都和咖啡有关，比如JavaBeans（咖啡豆）、NetBeans（网络豆）以及ObjectBeans（对象豆）等。从 1995 年诞生至今，经过近 20 年的快速发展，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。

只要你能始终保持一颗好奇的心，并能持之以恒，我想学好Java还是比较容易的。

既然这么重要，看来我得认真学习这门语言啦！那么，Java语言到底是怎样的一门语言呢？

这里我们所指的Java，通常有两种含义。有时候人们谈论Java就是特指Java编程语言（Programming Language），它是一种典型的面向对象编程语言，使用Java语言的语法规则可以编写出相应的程序代码，并让计算机通过解释这些代码来执行一个特定的应用。有时候人们讨论Java，还可能是指Java平台（Platform）。正如我们所理解的底层操作系统平台（如Windows或者UNIX）一样，在平台之上可以运行应用程序。Java平台扮演了这样的平台角色，它提供了开发和执行Java应用程序的完整环境。更确切地讲，Java平台是介于应用和底层操作系统的中间层。作为一个小型的操作系统，有了Java平台，就可以在所有主流操作系统上运行你的代码。

1.2 面向对象编程

面向对象编程从 20 世纪 60 年代就开始出现，但是直到 20 世纪 90 年代，面向对象才开

始真正被广大用户接受。人们逐渐发现了面向对象编程的优点：在编程中使用面向对象技术可以促使软件产品更加成熟和易于扩展，大大提高对软件的维护和管理能力。

1.2.1 对象的定义

那么，什么是对象呢？

对象是某些相关概念在代码级别的抽象。例如，你正在开发学生信息管理系统，对象可以被编程来表现一场考试或一个学生。同样对象也可以是活动的抽象，比如是一次选课。面向对象的做法就是需要对现实需求进行抽象，抽象就意味着需要你“放大本质，去掉无关内容”。例如，一场考试对象会包括一些考试相关信息，比如考场编号、考试时间、考生名单等，但与试卷的纸张类型、纸张颜色之类属性无关，所以你不需要在考试对象中添加这些属性。而在试卷打印系统中，你可能就需要考虑它们了。

一个面向对象系统首先需要关注的是对象的行为，对象之间通过相互发送消息影响对象行为。通俗地讲，其工作原理就是：一个对象发送一条消息给另一个对象，告诉它去做某些事情。

这种方式也能在我们日常生活中找到原型。例如，当我打开电灯控制开关时，它（一个对象）向一盏电灯（一个对象）发送了一条打开的消息，告诉电灯置于打开的状态（如图 1-1 所示）。控制开关作为消息的发送者只关心电灯提供的可被调用的抽象行为（如打开、关闭，也可能是调亮、调暗），而不必了解电灯是如何关闭、如何打开、如何调亮的细节。所有实现打开、关闭、调亮的细节由消息的接受者——电灯提供。这就体现了面向对象另一个重要特性——封装性。封装性的目的就是：对系统中的其他对象，隐藏所有不必要的细节。在后面的课程中将有更详细的介绍。

1.2.2 类的概念

看来面向对象的思想还是比较直观，易于理解的。那么，在Java程序中如何来描述对象信息呢？

要解答你的问题，还需要介绍另一个非常重要的概念：类。虽然教室中的灯会分布在不同的位置，类型也不一，但它们都有可以被打开、关闭的共同行为。因此，它们可以归为一类。类提供了定义一组相关对象的方式，就好像一个模板或者蓝图，程序可以通过它来创建新的对象。例如，可以定义一个Light类表示灯，并规定每个灯对象都需要保存其位置信息location，提供开灯open和关灯close的两个行为，如图 1-2 所示。

图 1-1 发送一个消息

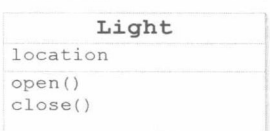


图 1-2 Light 类

图 1-2 使用一个矩形表示类的定义，最上方的格子中的 Light 表示类的名字，第二行的格子中描述每个 Light 对象都应该存储的属性信息，第三行的格子列出每个 Light 对象支持的

行为，这些行为就是灯对象收到消息后能做出的行为。

这让我想到了造房子需要施工图样，按照这个图样施工出来的房子具有相同的机构和套型。这里的施工图就是类，而根据图样建筑的房子就是对象。

你理解得非常正确。其实在现实生活中，人们都在不知不觉地使用面向对象技术。无论是某一型号的手机，还是计算机，厂家都定义了一份该型号的模板，类似于Java中的类，它描述了该型号产品的共同特性。而根据该模板创建出来的每部手机或者每台计算机就类似Java中的对象。每个具体的对象可以存储自己特有的信息，比如颜色、产品序列号，但作为同一类，它们总是具有共同的行为。在后面的章节中我们将详细地介绍Java中的面向对象设计。

1.2.3　UML 简介

正如读者在 1.2.2 节中看到的那样，类的图形化表述就是采用了 UML（统一建模语言）。UML 不但是一种方法，而且是一种图形化语言工具。本书在后续的章节中，会采用 UML 阐述代码设计。UML 的优点在于，作为一种面向对象建模语言，它不依赖于特定的程序语言，可以让其他程序员甚至是客户很快地理解开发者的设计思想。作为一种有效的沟通工具，UML 是有价值并且值得去学习的。

本书在必要的时候会介绍一些 UML 的初步知识，读者也可以从网络中得到 UML 规范的最新文档。

http://www.omg.org/technology/documents/formal/uml.htm

1.3　J2SDK 简介

老师，听您这么一说，我都有些等不及了，快教我如何开发Java程序吧！

“工欲善其事，必先利其器”，我们需要先把开发Java的工具和运行环境搭建好！你需要下载Java软件开发包（Java Develop Toolkit，JDK），它提供了以下 3 个主要组件：

1）编译器（javac）。

2）虚拟机（java）。

3）一套类库或者API（应用程序接口）。

编译器是一个程序，用以读取 Java 源文件，并对它们进行语法检查，保证它们包含正确的 Java 代码，然后输出 class 文件。源文件是一个包含代码的文本文件。编译器生成的 class 文件包含字节码（ByteCode）。字节码是一种能被虚拟机快速读取和解析的数据格式。

虚拟机是一个程序，用以执行 class 文件中的代码。之所以称为虚拟机，是由于它的行为如同一台完整的平台或者操作系统。与 C 或者 C++语言直接调用底层 Windows API 进行 Windows 应用开发不同，Java 代码不直接调用操作系统提供的编程 API，绝大多数情况下，它只需针对 JDK 提供的类库 API 进行编程。

下面就来认识一下这个 Java 开发工具包。

1.3.1 认识 J2SDK

Java 历史上有几个重要的版本，第一个正式版本是 Java 1.0，接着是 Java 1.1、Java 1.2，大致每年发布一个新的版本。在版本 1.2 时，为了体现这个版本的重要性，Sun 公司将这个平台的名称从 Java 改成 Java 2，并在后续版本中使用新的版本命名方案。本书以 Java 2 标准版 5.0（简称 J2SE）为基础，同时，也介绍最新 Java 7 的一些实用新特性。

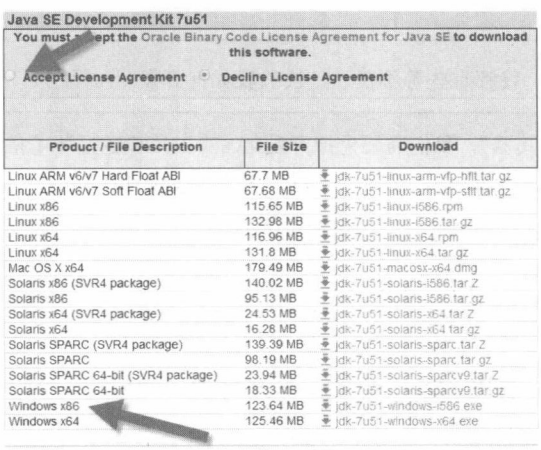
JDK 是 Java 软件开发工具箱（Java Development Kits），里面包括了运行的虚拟机、编译器等所有开发过程中需要的工具。Sun 公司为 Solaris、Linux 和 Windows 提供了 Java 2 标准版（J2SE）最新、最安全的版本，可以从相关网站下载最新的版本。目前最新版本为 JDK 1.7。

1.3.2 J2SDK 下载

你可以通过下列地址下载并安装最新的SDK。然后按照下列步骤进行SDK的安装。需要注意的是，Oracle公司还提供了简装的JRE（Java运行环境）版本，它只提供Java程序运行所需的最小环境，如果你只打算运行而不是编译Java程序，可以选择下载安装JRE。单独的JRE版本在部署Java应用程序时非常有用，只需在发布时绑定最小的组件集合。而我们选择安装的SDK版本中包含了JRE，因此无须另行安装JRE。

输入下载地址，打开如图 1-3 所示的下载页面，选择【Accept License Agreement】，接受许可协议。随后，根据自己的操作系统类型选择 JDK 的版本类型。对于 32 位操作系统，需要选择 x86 后缀的 JDK，对于 64 位操作系统，选择 x64 或 64-bit 类型的版本。

http://www.oracle.com/technetwork/java/javase/downloads/index.html



Java SE Development Kit 7u51

You must accept the Oracle Binary Code License Agreement for Java SE to download this software.

Accept License Agreement　Decline License Agreement

Product / File Description	File Size	Download
Linux ARM v6/v7 Hard Float ABI	67.7 MB	jdk-7u51-linux-arm-vfp-hflt.tar.gz
Linux ARM v6/v7 Soft Float ABI	67.68 MB	jdk-7u51-linux-arm-vfp-sflt.tar.gz
Linux x86	115.65 MB	jdk-7u51-linux-i586.rpm
Linux x86	132.98 MB	jdk-7u51-linux-i586.tar.gz
Linux x64	116.96 MB	jdk-7u51-linux-x64.rpm
Linux x64	131.8 MB	jdk-7u51-linux-x64.tar.gz
Mac OS X x64	179.49 MB	jdk-7u51-macosx-x64.dmg
Solaris x86 (SVR4 package)	140.02 MB	jdk-7u51-solaris-i586.tar.Z
Solaris x86	95.13 MB	jdk-7u51-solaris-i586.tar.gz
Solaris x64 (SVR4 package)	24.53 MB	jdk-7u51-solaris-x64.tar.Z
Solaris x64	16.28 MB	jdk-7u51-solaris-x64.tar.gz
Solaris SPARC (SVR4 package)	139.39 MB	jdk-7u51-solaris-sparc.tar.Z
Solaris SPARC	98.19 MB	jdk-7u51-solaris-sparc.tar.gz
Solaris SPARC 64-bit (SVR4 package)	23.94 MB	jdk-7u51-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	18.33 MB	jdk-7u51-solaris-sparcv9.tar.gz
Windows x86	123.64 MB	jdk-7u51-windows-i586.exe
Windows x64	125.46 MB	jdk-7u51-windows-x64.exe

图 1-3　JDK 下载页面

1.3.3 JDK 的安装

具体的安装步骤如下：

1）运行 jdk-7u51-windows-i586.exe 可执行文件，在弹出的安装向导中根据提示进行选择，单击“下一步”按钮。

2）选择安装开发工具，可以通过“更改”按钮改变默认安装路径，例如安装到 D:\Java\jdk1.7.0-15，单击“下一步”按钮，如图 1-4 所示。

3）安装运行环境 JRE，把运行环境的目录也设置在 D:\Java\jre7 文件夹下，单击“更改”按钮，再单击“下一步”按钮，如图 1-5 所示。

图 1-4　JDK 安装

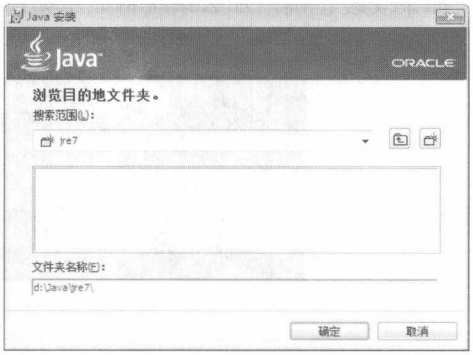


图 1-5　JRE 安装

因为 JDK 默认自带了 JRE，因此，在完成的默认安装目录 Java 中找到 JDK 和 JRE 两个文件夹，其中 JDK 放置了 Java 开发工具包相关的文件，JRE 放置的是运行环境相关的文件。打开 JDK 的 bin 子目录，可以看到很多扩展名为 exe 的可执行文件（见图 1-6），这些可执行文件都是在开发过程中经常需要用到的工具。

为了可以通过命令的方式来启动这些工具，我们需要设置系统的环境变量，以便操作系统知道这些新安装的可执行工具在逻辑磁盘的哪个位置，进而能够准确地调用它们。

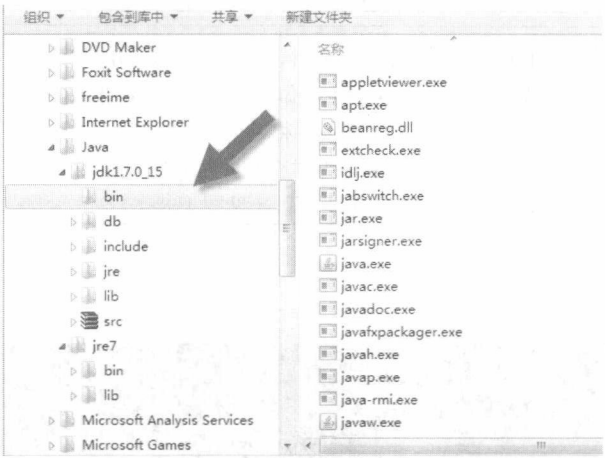


图 1-6　JDK 的 bin 目录

1.3.4　测试安装

安装完成后，如何检查是否安装成功呢？

JDK安装完成后，可以通过如下步骤测试是否安装成功。

1）按<Win+R>快捷键打开【运行】对话框，在文本框中输入“cmd”，打开命令提示符窗口。

2）在命令提示符窗口中输入如下命令：

```
Java -version
```

3）如果安装正确，那么系统将显示如图 1-7 所示的信息。

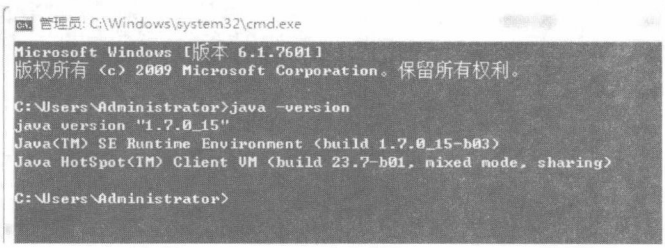


图 1-7　验证 JDK 是否成功安装

视频 01：JDK 的下载安装

要查看 JDK 的下载、安装过程，请播放教学视频 01.mp4。

1.3.5　JDK 的配置

1．理解 PATH

真厉害！通过在【运行】对话框中输入“cmd”就可以打开命令提示符窗口。

事实上，输入的命令恰好对应了位于C:\Windows\System32\下的cmd.exe可执行命令，虽然你没有指定该命令的完整路径信息，但是操作系统会查找一个叫做PATH的系统环境变量，提取其中的变量值，这些值就是用户或者系统预先设定的搜索路径，操作系统将以此查找各路径下执行该命令的可执行文件。你刚才运行的命令提示符工具cmd.exe就是通过这种方式查找并运行的。你还可以通过在命令提示符中输入如下命令来查看目前系统环境变量中包含哪些路径信息（见图 1-8）。

```
echo %PATH%
```

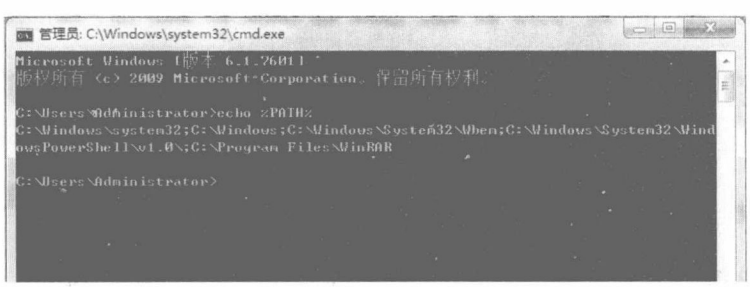


图 1-8　显示系统环境变量

由于在安装JDK的过程中，安装程序在C:\Windows\System32 下自动放了一份java（.exe），因此，我们可以在命令提示符里直接输入Java命令来加以执行（见图 1-9）。

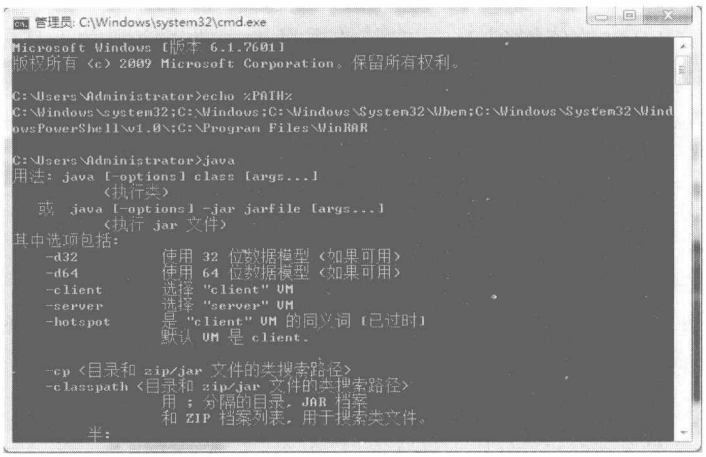


图 1-9　Java 命令

操作系统在识别Java命令时，根据图 1-8 所设的系统环境变量给出的路径，依次查找java（.exe）程序。当搜索C:\Windows\System32 路径时，发现存在java.exe可执行文件，随即加以执行，显示图 1-9 所示的命令。

你说得非常对。如果想要让安装在JDK的bin文件里的开发工具命令都能被操作系统搜索到，一个非常重要的步骤就是需要指定系统的环境变量。

2．设定 PATH

将常用的可执行工具所在路径设置到 PATH 变量中，以便系统通过 PATH 变量找到你要执行的命令。通常有如下两种方式进行设定：使用 SET 指令和设置系统环境变量值。

（1）SET 指令

在命令提示符中，使用 SET 指令，其命令格式如下：

```
SET PATH=路径
```

很多时候，需要一次性设定多个搜索路径，可以在设置时用分号（;）作为分隔。为了保留系统原有的 PATH 变量信息。在最后通常加上%PATH%，这里的%PATH%引用了原有的 PATH 变量值。

```
SET PATH=新添加的完整路径 1;新添加的完整路径 2;%PATH%
```

（2）设置系统环境变量值

在 Windows 7 中右键单击【计算机】，在弹出的快捷菜单中选择【属性】，单击【高级系统设置】，进入【系统属性】对话框，切换到【高级】选项卡，单击【环境变量】按钮。系统提供【用户变量】和【系统变量】的设置。系统变量和用户变量的区别在于：系统变量对所有使用本系统的用户都有效，而用户变量仅对当前登录的用户设置有效。

这里以系统变量设置为例。首先，自定义一个环境变量。选择系统变量中的【新建】按钮，在打开的【新建系统变量】对话框中，输入变量名 JAVA_HOME，其值设置为 JDK 的安装位置，如 D:\Java\jdk1.7.0_15，如图 1-10 所示。

其次，将该变量值追加到 PATH 系统变量中。先找到系统环境变量的 PATH 变量，单击

【编辑】按钮，在【变量值】的文本框最后添加一个英文字符分号（;）以隔开现有的变量值。通过引用%JAVA_HOME%的值得到 Java 安装的根目录。要指定开发工具命令所在目录，只需追加“\bin”，如图 1-11 所示。单击【确定】按钮以完成保存。

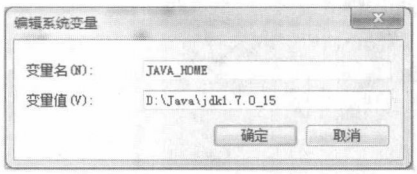


图 1-10　新建 JAVA_HOME 环境变量

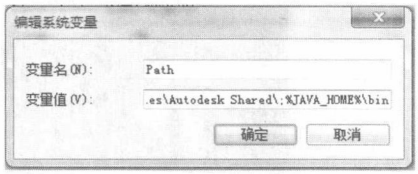


图 1-11　设置 PATH 环境变量

```
...;%JAVA_HOME%\bin
```

3. 测试环境变量

打开【命令提示符】窗口，在其中输入如下命令。

```
javac
```

如果环境变量设置正确，那么系统将显示如图 1-12 所示的信息，否则将显示：

```
'javac' 不是内部或外部命令，也不是可运行的程序或批处理文件。
```

图 1-12　javac 命令

视频 02：JDK 的配置

要查看 JDK 的配置过程，请播放教学视频 02.mp4。

需要注意的是，每次对环境变量的修改，需要重启命令提示符才能生效。

1.3.6　理解 CLASSPATH 和 SOURCEPATH

操作系统执行 JavaC 命令时，需要提供 PATH 搜索路径，以正确找到该命令并加以执行。同样的，虚拟机（JVM）就好像 Java 程序的操作系统，也可以执行不同的搜索路径完成特定的目的。例如，用户可以在执行编译源代码命令的同时，通过指定源代码路径（SOURCEPATH），告诉虚拟机你要编译的源代码文件所在的位置；在运行 Java 程序时，还可以指定类路径（CLASSPATH）来告诉虚拟机那些参与运行的字节码文件所在的位置。

表 1-1 展示了 PATH、SOURCEPATH、CLASSPATH 之间的区别。