

 21世纪高职高专系列规划教材·计算机类专业
 高职高专“十二五”规划教材

Java程序 设计

JAVA CHENGXU
SHEJI

主 编◎张瑞英 郎薇薇
何凤
副主编◎冯艳茹 赵冬玲



北京师范大学出版集团
BEIJING NORMAL UNIVERSITY PUBLISHING GROUP
北京师范大学出版社

21世纪高职高专系列规划教材·计算机类专业
高职高专“十二五”规划教材

Java程序

设计

常州大学图书馆
藏书章

JAVA CHENGXU
SHEJI

主 编◎张瑞英 郎薇薇
何凤
副主编◎冯艳茹 赵冬玲
参 编◎刘妍东 郭亚东



北京师范大学出版集团
BEIJING NORMAL UNIVERSITY PUBLISHING GROUP
北京师范大学出版社

图书在版编目 (CIP) 数据

Java 程序设计/张瑞英, 郎薇薇, 何凤主编. —北京: 北京师范大学出版社, 2014. 2

(21 世纪高职高专系列规划教材)

ISBN 978-7-303-17365-5

I. ①J… II. ①张…②郎…③何… 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 (2013) 第 298668 号

营销中心电话 010-58802755 58800035
北师大出版社职业教育分社网 <http://zjfs.bnup.com>
电子邮箱 zhijiao@bnupg.com

出版发行: 北京师范大学出版社 www.bnup.com
北京新街口外大街 19 号
邮政编码: 100875

印 刷: 保定市中画美凯印刷有限公司
经 销: 全国新华书店
开 本: 184 mm×260 mm
印 张: 11.75
字 数: 245 千字
版 次: 2014 年 2 月第 1 版
印 次: 2014 年 2 月第 1 次印刷
定 价: 20.00 元

策划编辑: 周光明 责任编辑: 周光明
美术编辑: 高 霞 装帧设计: 李 尘
责任校对: 李 菡 责任印制: 孙文凯

版权所有 侵权必究

反盗版、侵权举报电话: 010-58800697

北京读者服务部电话: 010-58808104

外埠邮购电话: 010-58808083

本书如有印装质量问题, 请与印制管理部联系调换。

印制管理部电话: 010-58800825

前言

Java 伴随着 Internet 问世,又随着 Internet 的发展而不断成熟,Java 目前已经成为广泛应用的程序设计语言,它具有简单、面向对象、安全和健壮性等诸多特点,是网络编程的标准开发工具语言。如今计算机及其网络应用已深入人们生活的方方面面,作为一名计算机专业大学生,无疑应该掌握一定的计算机知识和编程技能。同时,为了迎接信息时代的挑战,学习和掌握 Java 语言无疑会给广大学生将来步入社会带来更多的机遇。

本书的目标是帮助学生深入、细致、系统地学习 Java 语言知识,深入理解 Java 语言的精髓,掌握 Java 语言的编程知识和编程技术。主要包括:Java 运行环境,Java 语言基础,Java 类和方法,图形用户界面,异常处理机制,图形用户界面设计,线程,类和常用工具,数据流等。这些内容是编写 Java 程序的基本要素和必备知识。

本书采用案例教学的方法,围绕任务进行,力求深入浅出、叙述详细、概念清晰、通俗易懂。书中每个实例都是作者根据所讲述的知识点精心设计的,展示了如何将 Java 编程知识和编程技巧应用于实践中,具有很强的实用性。

本书在内容布局上,适合高职学生的认知规律。每章的开始都给出本章的内容提要和本章要点,使读者在学习前明白本章要掌握的内容,在学习时可以抓住重点,有针对性地学习。每章的最后都给出具有针对性的习题,每个习题都是精心设计挑选的,用来巩固、消化本章学习的知识。把书中的理论知识通过习题与实践结合起来,有利于读者更快掌握每章的 Java 编程知识,巩固所学内容。

本书的第 2、4 章由郎薇薇编写,第 1、10 章由张瑞英编写,第 6、7、8 章由何凤、冯艳茹编写,第 9 章刘妍东编写、第 11 章由郭亚东和张瑞英共同编写,第 3、5 章由赵冬玲编写,全书由张瑞英负责统稿。

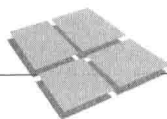
本书适用于高职高专院校的 Java 程序设计类相关课程教材,也可以作为程序爱好者的入门辅导书,还可以作为培训机构的培训教材。

由于本书的编写时间很仓促,书中难免有不妥与疏漏之处,还敬请各位读者不吝指正。

编者
2013 年 12 月

目 录

第 1 章 Java 语言入门	(1)	2.3.2 相关知识点	(16)
1.1 Java 的起源	(1)	2.4 文档	(17)
1.1.1 Java 概述	(1)	2.5 实例 4: 运算符及表达式应用	(18)
1.1.2 Java 的起源	(1)	2.5.1 运算符及表达式的实例	(18)
1.1.3 JDK 版本	(2)	2.5.2 相关知识点	(19)
1.2 Java 的特点	(2)	2.6 综合案例	(24)
1.3 Java 开发环境的建立及环境配置	(4)	2.7 小结	(25)
1.3.1 JDK 的安装	(4)	习题二	(26)
1.3.2 配置环境变量	(6)	第 3 章 结构化编程	(28)
1.4 Java 集成开发工具——Eclipse 介绍	(8)	3.1 实例 5: 分支结构语句	(28)
1.5 实例 1: 建立 Java 应用程序	(9)	3.1.1 分支结构实例	(28)
1.6 综合案例	(11)	3.1.2 分支结构	(30)
1.7 小结	(11)	3.2 实例 6: 循环结构语句	(32)
习题一	(11)	3.2.1 循环结构实例	(32)
第 2 章 数据类型、运算和语法	(12)	3.2.2 循环结构	(34)
2.1 标识符和关键字	(12)	3.3 实例 7: 循环控制及嵌套语句	(34)
2.1.1 标识符	(12)	3.4 综合案例	(36)
2.1.2 关键字	(12)	3.5 小结	(38)
2.2 实例 2: 基本数据类型转换	(13)	习题三	(38)
2.2.1 基本数据类型	(14)	第 4 章 类和对象的使用	(39)
2.2.2 数据类型转换	(15)	4.1 面向对象的编程思想	(39)
2.3 实例 3: 变量及常量应用	(16)	4.2 实例 8: 类的定义及使用	(40)
2.3.1 变量及常量的应用实例	(16)	4.2.1 类的定义实例	(40)
		4.2.2 相关知识点	(41)



4.3 实例 9: 对象的创建和引用	(42)
4.3.1 对象创建的引用实例 ..	(42)
4.3.2 相关的知识点	(44)
4.4 实例 10: 类的继承	(45)
4.5 实例 11: 重载	(47)
4.6 实例 12: 覆盖	(48)
4.7 实例 13: 访问控制符和非 访问控制符	(49)
4.8 实例 14: 接口	(52)
4.8.1 接口实例	(52)
4.8.2 相关知识点	(53)
4.9 综合案例	(54)
4.10 小结	(56)
习题四	(56)
第 5 章 Java 中常见类的属性及使用	(57)
5.1 System 类	(57)
5.2 String 类	(59)
5.3 StringBuffer 类	(67)
5.4 Math 类	(69)
5.5 Date 类	(71)
5.6 GregorianCalendar 类 ..	(73)
5.7 小结	(77)
习题五	(78)
第 6 章 数 组	(79)
6.1 数组的声明	(79)
6.2 数组的初始化	(80)
6.3 匿名数组	(81)
6.4 数组的拷贝	(82)
6.5 命令行参数	(85)
6.6 数组排序	(86)
6.7 小结	(88)
习题六	(88)
第 7 章 Java 中的异常及处理	(89)
7.1 处理错误	(89)
7.1.1 异常的分类	(89)
7.1.2 异常的声明	(90)
7.1.3 异常的抛出	(91)
7.2 异常的捕获	(92)
7.2.1 异常的抛出	(93)
7.2.2 重新抛出异常	(93)
7.3 finally 子句	(94)
7.4 小结	(95)
习题七	(96)
第 8 章 数据流的应用	(97)
8.1 输入输出流	(97)
8.1.1 字节输入流 InputStream	(98)
8.1.2 字节输出流 OutputStream	(100)
8.1.3 字符输入流 Reader ..	(101)
8.1.4 字符输出流 Writer ..	(101)
8.2 文件类	(102)
8.2.1 文件的概念	(102)
8.2.2 File 类	(102)
8.3 I/O 类的使用	(106)
8.3.1 读取文件示例	(106)
8.3.2 写文件示例	(110)
8.3.3 读取控制台示例	(112)
8.4 小结	(119)
习题八	(119)
第 9 章 常用组件及事件处理 ..	(120)
9.1 Java 中常用组件概述	(120)
9.2 实例 15: Java 中容器组件的 使用	(122)
9.2.1 窗口应用实例	(122)
9.2.2 窗口 JFrame	(122)
9.2.3 面板应用实例	(123)
9.2.4 面板 JPanel	(124)
9.3 实例 16: Java 中常用组件 的使用	(124)
9.3.1 按钮组件应用实例	(124)
9.3.2 按钮 JButton	(126)
9.3.3 标签组件应用实例	(126)
9.3.4 标签 JLabel	(127)

9.3.5 文本框组件应用实例	(127)	10.2 Java GUI 编程入门 ...	(147)
9.3.6 文本框 JTextField	(128)	10.3 Java GUI 布局管理 ...	(149)
9.3.7 密码框组件应用实例	(129)	10.3.1 FlowLayout 布局管理器	(149)
9.3.8 密码框 JPasswordField	(130)	10.3.2 BorderLayout 布局管理器	(150)
9.3.9 文本区组件应用实例	(131)	10.3.3 GridLayout 布局管理器	(151)
9.3.10 文本区 JTextArea ...	(131)	10.3.4 自定义布局管理器 ...	(153)
9.3.11 单选按钮和复选框组件应用实例	(132)	10.4 Java GUI 事件管理 ...	(154)
9.3.12 相关知识点	(133)	10.4.1 事件处理应用实例 1	(155)
9.3.13 组合框组件应用实例	(135)	10.4.2 事件处理应用实例 2	(156)
9.3.14 组合框 JComboBox ...	(136)	10.5 综合应用	(157)
9.3.15 列表框组件应用实例	(136)	10.6 小结	(159)
9.3.16 列表框 JList	(137)	习题十	(159)
9.4 事件处理概述	(137)	第 11 章 线程	(160)
9.5 实例 16: 事件处理应用	(138)	11.1 线程的基本知识	(160)
9.5.1 动作事件应用实例	(138)	11.2 线程的创建	(162)
9.5.2 事件处理	(140)	11.2.1 用 Thread 类建立多线程	(163)
9.6 综合案例	(142)	11.2.2 用 Runnable 接口建立多线程	(165)
9.7 小结	(144)	11.3 线程的同步	(166)
习题九	(144)	11.3.1 定义同步方法	(166)
第 10 章 图形用户界面设计	(146)	11.3.2 使用同步	(169)
10.1 GUI 功能	(146)	习题十一	(171)
		部分习题答案	(173)

第 1 章 Java 语言入门

内容提要

本章将介绍 Java 语言的起源、特点和应用领域,以及 Java 语言的环境及参数配置。初步编写最简单的 Java 小程序,通过这些内容的学习,程序设计者会选择 Java 开发工具,可以对简单的 Java 程序有一定的了解。

本章要点

- Java 语言的起源。
- Java 语言的特点。
- 能够搭建 Java 的开发环境。
- 简单的 Java 程序。

1.1 Java 的起源

【任务 1-1】了解 Java 语言的起源、版本和特点

通过本次任务的学习,可以初步对 Java 语言的起源、特点以及各种版本有一定的了解。

1.1.1 Java 概述

Java 包括 Java 编程语言、开发工具和环境、Java 类库等。

Java 语言是由美国 SUN 公司开发的一种完全面向对象的程序设计语言。Java 语言由 C++ 语言发展而来,但比 C++ 简单。它是当前网络编程中的首选语言。

从 1995 年 5 月诞生至今,Java 语言随着 Internet 的迅猛发展而成长壮大,现在已经成为 Internet 上的主流编程语言。J2ME/J2SE/J2EE 三大平台强大的应用系统设计能力,使得 Java 无处不在。

1.1.2 Java 的起源

Java 开始并不叫 Java,而是叫 Oak,1991 年 SUN 公司成立 GREEN 项目组,开发一种用于消费电器设备控制的嵌入式系统,该系统最初采用 C++ 语言开发,但是由于 C++ 语言太烦琐而且安全性差,不能满足要求,于是 GREEN 项目组研究开发了一种新的语言,取名 Oak。

但 Oak 是另外一个注册公司的名字。由于商标冲突,Oak 这个名字不能再用了,所以在 1995 年,SUN 公司把 Oak 更名为 Java。Java 是印度尼西亚的爪哇岛的英文名称,因盛产咖啡而出名。Java 语言中的许多类库名称,多与咖啡有关,如 JavaBeans (咖啡豆)、NetBeans (网络豆)以及 ObjectBeans (对象豆)等。SUN 和 Java 的标识也正是一杯正冒着热气的咖啡。虽然可编程控制设备的消费市场并没有像预期的那样大规模



模发展，但是 Java 并没有因此而随之淹没。

1.1.3 JDK 版本

JDK 是 Java Development Kit 的简称，是整个 Java 的核心。JDK 包括了 Java 运行环境、Java 工具和 Java 基础类库。没有 JDK 的话，无法安装或者运行 Java 程序。自从 Java 推出以来，JDK 版本已经从最初的 1.0 发展到最新的 1.6，经历了几次更新。其中 1998 年年底发布的 1.2 版本是一个重要的版本，SUN 称之为 Java2 SDK（注：SDK 是 Software Development Kit 的简称），此后的 Java 就称为 Java 2 平台。由于 SUN 公司的开放策略，用户可以在其网站上免费获取 JDK，所以这也是 Java 语言能够迅速发展的重要原因。

1996 年 6 月 SUN 公司把 JDK1.3 划分为 J2SE、J2EE、J2ME 三个版本，这三个平台分别定位于桌面应用、企业级应用和嵌入式应用，使 Java 技术获得最广泛的应用。用户可以根据实际应用领域的需求选择不同的 Java 平台。

J2SE(Java 2 Standard Edition)，Java 标准版，是我们通常用的一个版本，它包含 Java 编译器、Java 类库、Java 运行时的环境和 Java 命令行工具。

J2EE(Java 2 Enterprise Edition)，Java 企业版，提供分布式企业软件组件架构规范。

J2ME(Java 2 Micro Edition)，主要用于移动设备、嵌入式设备上的 Java 应用程序，提供 Java Card、Java Telephone、Java TV 等技术，支持智能卡业务、移动通信、电视机顶盒等功能。

1.2 Java 的特点

Java 成为目前网络编程的首选语言，充分说明 Java 语言的设计思想和其所有的特点适应了网络发展的特殊需要。

Java 语言具有以下几个特点。

1. 简单易学

Java 最初是为对家用电器进行集成控制而设计的一种语言，因此它必须简单明了。Java 语言的简单性主要体现在以下三个方面：1) Java 的风格类似于 C++，因而 C++ 程序员是非常熟悉的。从某种意义上讲，Java 语言是 C 及 C++ 语言的一个变种，因此，C++ 程序员可以很快就掌握 Java 编程技术。2) Java 摒弃了 C++ 中容易引发程序错误的地方，如指针和内存管理。3) Java 提供了丰富的类库。这样无论是掌握了 Java 语言再学 C 语言，还是掌握了 C 语言再来学习 Java 语言，都会感到易于入门。

2. 面向对象

面向对象可以说是 Java 最重要的特性。Java 语言的设计完全是面向对象的，它不支持类似 C 语言那样的面向过程的程序设计技术。Java 支持静态和动态风格的代码继承及重用。现实世界中任何实体都可以看作是对象。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。用公式表示,过程式编程语言为:程序=算法+数据;面向对象编程语言为:程序=对象+消息。所有面向对象编程语言都支持三个概念:封装、多态性和继承,Java也不例外。现实世界中的对象均有属性和行为,映射到计算机程序上,属性则表示对象的数据,行为表示对象的方法(其作用是处理数据或同外界交互)。

3. 平台无关性

Java设计之初,并非为在Internet上应用,而是出于对独立于平台的编程语言的需要。如今,Java语言在网络编程上的成功正是归功于它独立于平台的特性。使用Java语言编写的应用程序不需要进行任何修改,就可以在不同的软、硬件平台上运行,因此大大降低了开发、维护 and 管理的开销。这主要通过Java虚拟机(JVM)来实现。Java虚拟机能掩盖不同CPU之间的差别,使J-Code能运行于任何具有Java虚拟机的机器上。

4. 安全性

Java语言通过使用编译器和解释器在很大程度上避免了病毒程序的产生和网络程序对本地系统的破坏。另外,它去除了C++中易造成错误的指针,增加了自动内存管理等措施,保证了Java程序运行的可靠性。此外,当Java用来创建浏览器时,语言功能和浏览器本身提供的功能结合起来,使它更安全。

5. 多线程

多线程也叫并发性,它是当今软件技术的一个重要成果,已成功应用在操作系统、应用开发等多个领域。多线程技术允许同时存在几个执行体,按几条不同的执行线路共同工作,满足了一些辅助软件的需求。Java支持多线程技术,就是多个线程并行机制,多线程是Java的一个重要方法,特别有利于在程序中实现并发任务。Java提供Thread线程类,实现了多线程的并发机制。因为Java实现的多线程技术,所以比C和C++更健壮。

6. 健壮性

Java致力于检查程序在编译和运行时的错误。类型检查帮助检查出许多开发早期出现的错误。Java操纵内存减少了内存出错的可能性。Java还实现了真数组,避免了覆盖数据的可能。这些功能特征大大缩短了开发Java应用程序的周期。Java提供Null指针检测数组边界、检测异常出口和字节代码校验。

7. 动态性

Java的动态特性是其面向对象设计方法的发展。它允许程序动态地装入运行过程中所需要的类,这是C++语言进行面向对象程序设计所无法实现的。在C++程序设计过程中,每当在类中增加一个实例变量或一种成员函数后,引用该类的所有子类都必须重新编译,否则将导致程序崩溃。Java从如下几方面来解决问题。Java编译器不是将对实例变量和成员函数的引用编译为数值引用,而是将符号引用信息在字节码中保存传递给解释器,再由解释器在完成动态链接类后,将符号引用信息转换为数值偏移量。这样,一个在存储器生成的对象不在编译过程中决定,而是延迟到运行时由解释器确定。这样,对类中的变量和方法进行更新时就不至于影响现存的代码。解释执行字节码时,这种符号信息的查找和转换过程仅在一个新的名字出现时才进行一



次，随后代码便可以全速执行。在运行时，确定引用的好处是可以使用已被更新的类，而不必担心会影响原有的代码。如果程序连接了网络中另一系统中的某一类，该类的所有者也可以自由地对该类进行更新，而不会使任何引用该类的程序崩溃。Java 还简化了使用一个升级的或全新的协议的方法。如果你的系统运行 Java 程序时遇到了不知怎样处理的程序，没关系，Java 能自动下载你所需要的功能程序。

1.3 Java 开发环境的建立及环境配置

【任务 1-2】Java 环境变量配置

通过本任务的学习能够熟练掌握 JDK 的安装以及进行环境变量的配置，为今后编程做准备。

1.3.1 JDK 的安装

1. 安装环境要求

Java 对计算机的要求不高，以下是基于 Windows 平台的计算机的最低要求。

硬件要求：CPU P II 以上，64MB 内存，100MB 硬盘空间。

软件要求：Windows 98 \ NT \ 2000 \ XP，Internet Explorer 5.0。

2. 下载 JDK

由于 SUN 公司在 2010 年已经被 Oracle 公司收购，所以现在 JDK 的下载都要访问 Oracle 的主页。

(1) 打开 IE 浏览器，在地址栏输入网址 <http://www.oracle.com/technetwork/java/javase/downloads/index.html>，按回车键，打开 JDK 下载主页面，如图 1-1 所示。



图 1-1 JDK 下载主页面

(2)单击最左侧的 Java DOWNLOAD 按钮,出现如图 1-2 所示的页面,在该页面中选择 Windows X86 大小为 123.49MB,名为“jdk-7 u9-windows-i586.exe”。

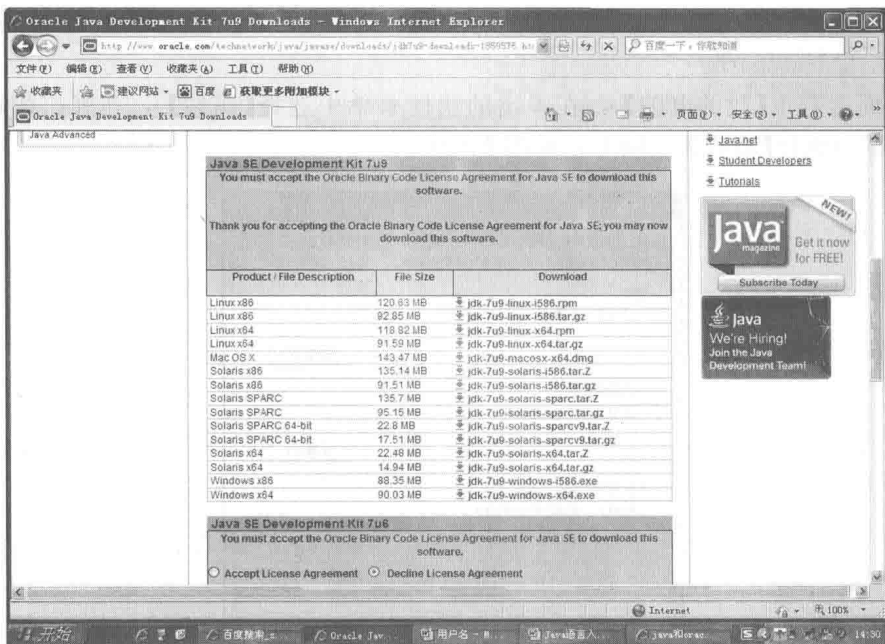


图 1-2 下载界面

3. 安装 JDK

运行 jdk-7 u9-windows-i586.exe 来安装 JDK,安装过程中设置安装路径,并选择组件,默认“组件选择”是全部安装。安装成功。如图 1-3 所示。

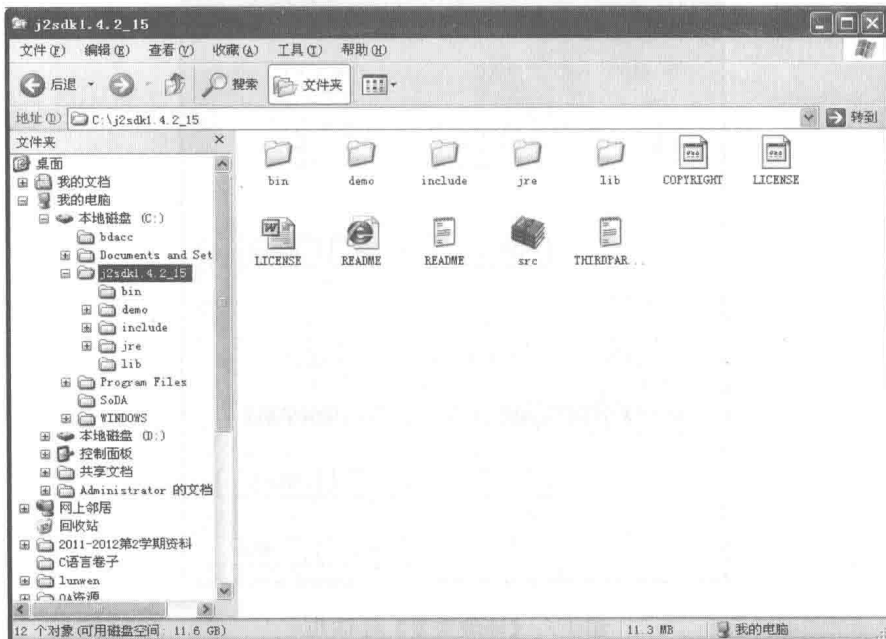
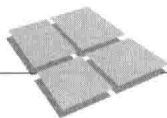


图 1-3 JDK 的目录结构



1.3.2 配置环境变量

由于 Java 是与平台无关的, 安装 JDK 时 Java 不会自动设置路径, 也不会修改注册表, 需要用户自己设置环境变量, 但是不需要修改注册表。设置的环境变量包括 path、java_home 和 classpath。

在桌面上右击【我的电脑】, 在弹出的快捷菜单中选择【属性】, 在弹出的【系统属性】对话框中选择【高级】选项卡, 如图 1-4 所示。

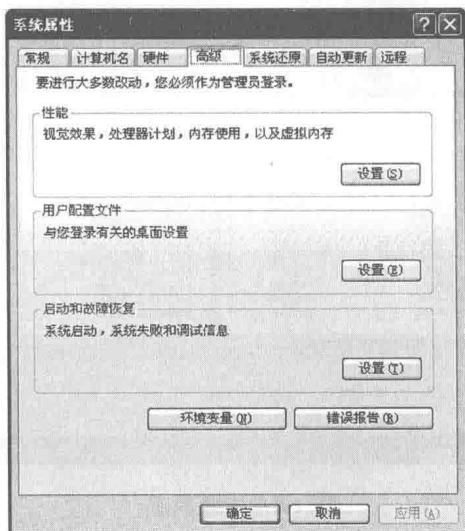


图 1-4 【系统属性】对话框

单击【环境变量】按钮, 打开【环境变量】对话框, 如图 1-5 所示。

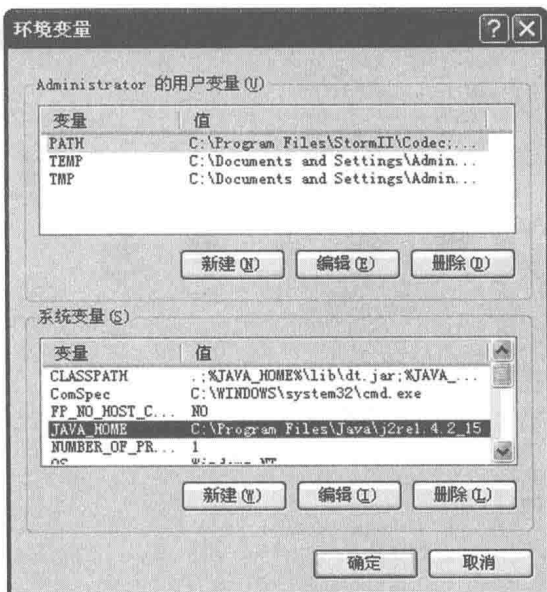


图 1-5 【环境变量】对话框

1. 编辑用户变量 path

path 是 Windows 已经定义的变量, 在【Administrator 的用户变量】列表框中的【变量】一栏, 找到“path”, 单击【编辑】按钮, 在打开的【编辑用户变量】对话框中, 将“C:\Program Files\Storm II\Codec; C:\Program Files\Storm II”加到【变量值】文本框, 如图 1-6 所示。



图 1-6 编辑用户变量 path

2. 新建系统变量 java_home

在【Administrator 用户变量】选项区中单击【新建】按钮, 建立 java_home 系统变量, 【变量值】设置为 C:\Program Files\Java\j2rel.4.2_15 “.;%JAVA_HOME%\lib\dt.jar;%JAVA_HOME%\lib\tools.jar;”, 如图 1-7 所示。单击【确定】按钮, 配置成功。



图 1-7 系统变量 java_home

3. 新建 classpath

classpath 是 JDK 包的路径, 在【Administrator 用户变量】选项区中单击【新建】按钮, 建立系统变量 classpath, 如图 1-8 所示。

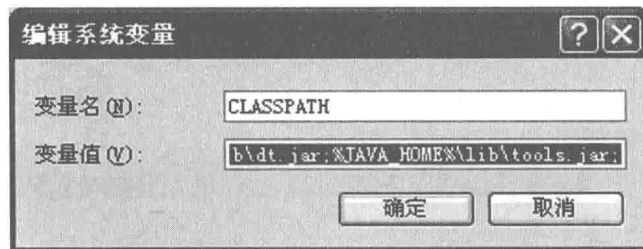


图 1-8 系统变量 classpath

设置好系统变量之后, 单击 Windows 任务栏的【开始】按钮, 选择【运行】命令, 在弹出的对话框中输入“cmd”命令, 进入 DOS 窗口, 在 DOS 窗口下输入“java”命令后,



按回车键，如果出现其用法参数提示信息，则表明 JDK 安装正确。如图 1-9 所示。



图 1-9 Java 安装成功参数提示信息

1.4 Java 集成开发工具——Eclipse 介绍

如图 1-10 所示，Eclipse 是一种可扩展的开放源代码的 Java 集成开发环境，它最大的特点是能够接受由 Java 开发者自己编写的开放源代码插件。本书也主要是以其作为编程环境的。

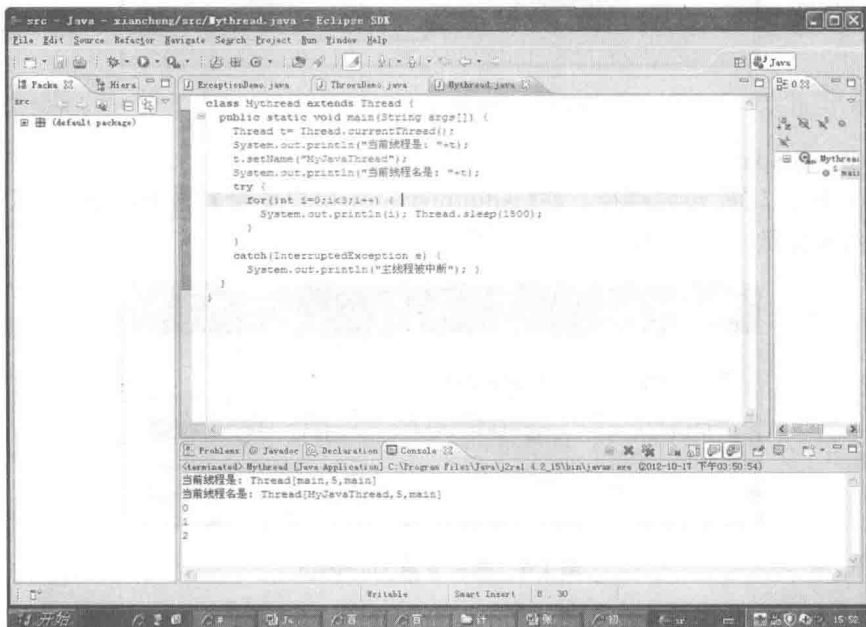


图 1-10 Eclipse 的工作环境

目前 Java 的集成开发工具除了 Eclipse 之外, 还有 SUN 公司的 NetBeans 以及英国 Kent 大学和澳大利亚 Deakin 大学开发的 BlueJ。

1.5 实例 1: 建立 Java 应用程序

【任务 1-3】使用 Eclipse 来编写简单的 Java 程序

通过本次任务的学习, 能够熟练掌握用 Eclipse 的开发环境进行 Java 程序设计, 并能初步树立起面向对象程序设计中类的概念。

安装完 Eclipse 之后, 双击 Eclipse 图标, 打开工作界面, 在工具栏中单击【New】按钮, 选择【Java Project】, 弹出如图 1-1 所示对话框, 在弹出的对话框中, 输入工程名字, 输入“1”, 如图 1-11 所示。

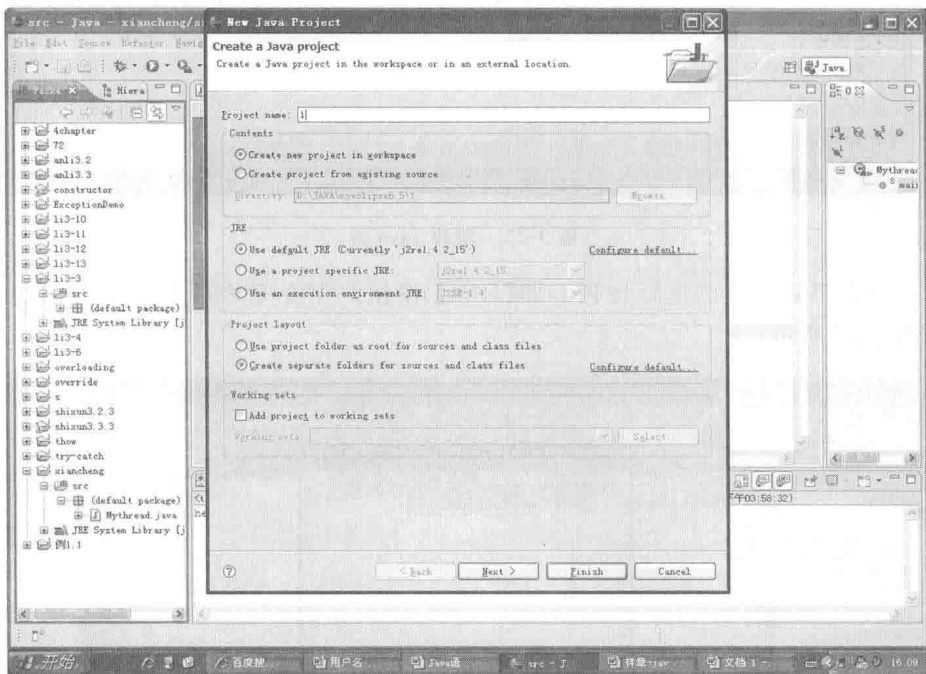


图 1-11 新建 Java 工程

然后单击【Next】按钮, 再单击【Finish】按钮, 会在左侧出现工程 1 的文件夹。双击该文件夹, 出现“src”, 然后右击“src”, 选择 new class, 也就是新建一个类, 注意类名要符合 Java 的命名规矩, 在之后的章节我们会具体讲解, 这里, 类名叫“Hello”, 如图 1-12 所示。然后再单击【Finish】按钮。

然后就可以输入程序代码:

```
public class Hello {  
    public static void main(String args[]){  
        System.out.println("hello");  
    }  
}
```

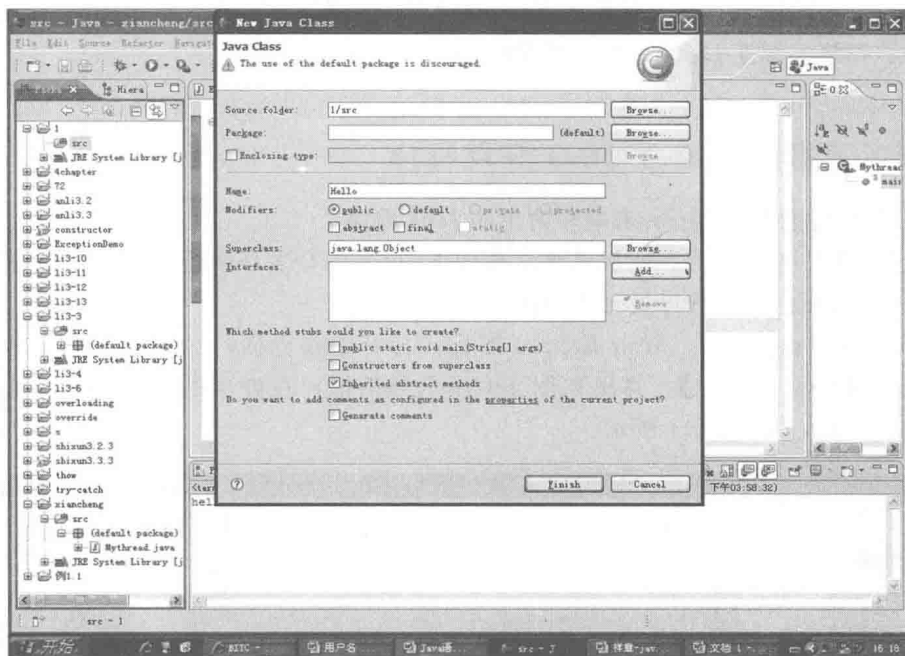
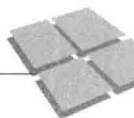


图 1-12 新建 Java 类

然后单击工具栏上的运行按钮，即【Run】按钮，保存完之后会在下方出现运行结果。如图 1-13 所示。

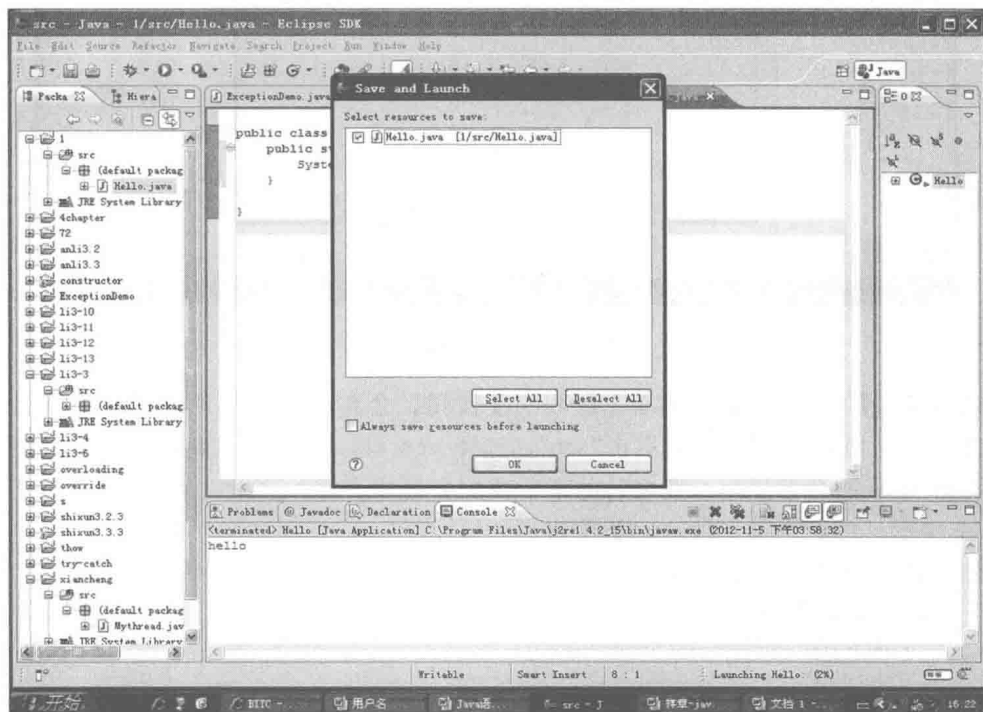


图 1-13 第一个 Java 程序