



21世纪高等院校规划教材



Java语言 程序设计教程

刘震 唐铁斌 郝秋燕 娄松涛 编著
曾强聪 主审



中国水利水电出版社
www.waterpub.com.cn

21 世纪高等院校规划教材

Java 语言程序设计教程

刘 震 唐铁斌 郝秋燕 娄松涛 编著

曾强聪 主审



中国水利水电出版社
www.waterpub.com.cn

内 容 提 要

Java 语言具有面向对象、与平台无关、安全、稳定和多线程等优良特性,是目前软件设计中极为强大的编程语言。本书全面、系统地介绍了 Java 语言程序设计的技术和编程方法,将面向对象的程序设计思想贯穿其中。编写过程中注重循序渐进、由浅入深的原则,通过实例对基本概念和基本理论加以说明。

本书分为 13 章,主要内容包括 Java 语言数据类型、运算符、表达式和语句,数组和字符串,类的封装、继承和多态,接口、内部类,异常处理,输入/输出和文件操作,多线程,图形用户界面设计,网络编程,数据库编程等。

本书可作为培养应用性、技能型人才的普通高等学校计算机及相近专业本科、高职高专学生的 Java 语言程序设计课程教材,也可作为使用 Java 语言的软件开发人员的参考书。

本书所配电子教案及相关教学资源可以从中国水利水电出版社网站和万水书苑上下载,网址为: <http://www.waterpub.com.cn/softdown/>和 <http://www.wsbookshow.com>。

图书在版编目(CIP)数据

Java语言程序设计教程 / 刘震等编著. -- 北京 :
中国水利水电出版社, 2011. 12
21世纪高等院校规划教材
ISBN 978-7-5084-9210-0

I. ①J… II. ①刘… III. ①JAVA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(2011)第253000号

策划编辑: 周益丹 责任编辑: 李 炎 加工编辑: 李 冰 封面设计: 李 佳

书 名	21 世纪高等院校规划教材 Java 语言程序设计教程
作 者	刘 震 唐铁斌 郝秋燕 姜松涛 编著 曾强聪 主审
出版发行	中国水利水电出版社 (北京市海淀区玉渊潭南路 1 号 D 座 100038) 网址: www.waterpub.com.cn E-mail: mchannel@263.net (万水) sales@waterpub.com.cn
经 售	电话: (010) 68367658 (发行部)、82562819 (万水) 北京科水图书销售中心(零售) 电话: (010) 88383994、63202643、68545874 全国各地新华书店和相关出版物销售网点
排 版	北京万水电子信息有限公司
印 刷	三河市鑫金马印装有限公司
规 格	184mm×260mm 16 开本 15 印张 370 千字
版 次	2011 年 12 月第 1 版 2011 年 12 月第 1 次印刷
印 数	0001—3000 册
定 价	27.00 元

凡购买我社图书,如有缺页、倒页、脱页的,本社发行部负责调换

版权所有·侵权必究

序

随着计算机科学与技术的飞速发展,计算机的应用已经渗透到国民经济与人们生活的各个角落,正在日益改变着传统的人类工作方式和生活方式。在我国高等教育逐步实现大众化后,越来越多的高等院校会面向国民经济发展的第一线,为行业、企业培养各级各类高级应用型专门人才。为了大力推广计算机应用技术,更好地适应当前我国高等教育的跨越式发展,满足我国高等院校从精英教育向大众化教育的转变,符合社会对高等院校应用型人才培养的各类要求,我们成立了“21世纪高等院校规划教材编委会”,在明确了高等院校应用型人才培养模式、培养目标、教学内容和课程体系的框架下,组织编写了本套“21世纪高等院校规划教材”。

众所周知,教材建设作为保证和提高教学质量的重要支柱及基础,作为体现教学内容和教学方法的知识载体,在当前培养应用型人才中的作用是显而易见的。探索和建设适应新世纪我国高等院校应用型人才培养体系需要的配套教材已经成为当前我国高等院校教学改革和教材建设工作面临的紧迫任务。因此,编委会经过大量的前期调研和策划,在广泛了解各高等院校的教学现状、市场需求,探讨课程设置、研究课程体系的基础上,组织一批具备较高的学术水平、丰富的教学经验、较强的工程实践能力的学术带头人、科研人员和主要从事该课程教学的骨干教师编写出一批有特色、适用性强的计算机类公共基础课、技术基础课、专业及应用技术课的教材以及相应的教学辅导书,以满足目前高等院校应用型人才培养的需要。本套教材消化和吸收了多年来已有的应用型人才培养的探索与实践成果,紧密结合经济全球化时代高等院校应用型人才培养工作的实际需要,努力实践,大胆创新。教材编写采用整体规划、分步实施、滚动立项的方式,分期分批地启动编写计划,编写大纲的确定以及教材风格的定位均经过编委会多次认真讨论,以确保该套教材的高质量和实用性。

教材编委会分析研究了应用型人才与研究型人才在培养目标、课程体系和内容编排上的区别,分别提出了3个层面上的要求:在专业基础类课程层面上,既要保持学科体系的完整性,使学生打下较为扎实的专业基础,为后续课程的学习做好铺垫,更要突出应用特色,理论联系实际,并与工程实践相结合,适当压缩过多过深的公式推导与原理性分析,兼顾考研学生的需要,以原理和公式结论的应用为突破口,注重它们的应用环境和方法;在程序设计类课程层面上,把握程序设计方法和思路,注重程序设计实践训练,引入典型的程序设计案例,将程序设计类课程的学习融入案例的研究和解决过程中,以学生实际编程解决问题的能力为突破口,注重程序设计的实现;在专业技术应用层面上,积极引入工程案例,以培养学生解决工程实际问题的能力为突破口,加大实践教学内容的比重,增加新技术、新知识、新工艺的内容。

本套规划教材的编写原则是:

在编写中重视基础,循序渐进,内容精炼,重点突出,融入学科方法论内容和科学理念,反映计算机技术发展要求,倡导理论联系实际和科学的思想方法,体现一级学科知识组织的层次结构。主要表现在:以计算机学科的科学体系为依托,明确目标定位,分类组织实施,兼容互补;理论与实践并重,强调理论与实践相结合,突出学科发展特点,体现学科发展的内在规律;教材内容循序渐进,保证学术深度,减少知识重复,前后相互呼应,内容编排合理,整体

结构完整；采取自顶向下设计方法，内涵发展优先，突出学科方法论，强调知识体系可扩展的原则。

本套规划教材的主要特点是：

(1) 面向应用型高等院校，在保证学科体系完整的基础上不过度强调理论的深度和难度，注重应用型人才的专业技能和工程实用技术的培养。在课程体系方面打破传统的研究型人才培养体系，根据社会经济发展对行业、企业的工程技术需要，建立新的课程体系，并在教材中反映出来。

(2) 教材的理论知识包括了高等院校学生必须具备的科学、工程、技术等方面的要求，知识点不要求大而全，但一定要讲透，使学生真正掌握。同时注重理论知识与实践相结合，使学生通过实践深化对理论的理解，学会并掌握理论方法的实际运用。

(3) 在教材中加大能力训练部分的比重，使学生比较熟练地应用计算机知识和技术解决实际问题，既注重培养学生分析问题的能力，也注重培养学生思考问题、解决问题的能力。

(4) 教材采用“任务驱动”的编写方式，以实际问题引出相关原理和概念，在讲述实例的过程中将本章的知识点融入，通过分析归纳，介绍解决工程实际问题的思想和方法，然后进行概括总结，使教材内容层次清晰，脉络分明，可读性、可操作性强。同时，引入案例教学和启发式教学方法，便于激发学习兴趣。

(5) 教材在内容编排上，力求由浅入深，循序渐进，举一反三，突出重点，通俗易懂。采用模块化结构，兼顾不同层次的需求，在具体授课时可根据各校的教学计划在内容上适当加以取舍。此外还注重了配套教材的编写，如课程学习辅导、实验指导、综合实训、课程设计指导等，注重多媒体的教学方式以及配套课件的制作。

(6) 大部分教材配有电子教案，以使教材向多元化、多媒体化发展，满足广大教师进行多媒体教学的需要。电子教案用 PowerPoint 制作，教师可根据授课情况任意修改。相关教案的具体情况请到中国水利水电出版社网站 www.waterpub.com.cn 下载。此外还提供相关教材中所有程序的源代码，方便教师直接切换到系统环境中教学，提高教学效果。

总之，本套规划教材凝聚了众多长期在教学、科研一线工作的教师及科研人员的教学科研经验和智慧，内容新颖，结构完整，概念清晰，深入浅出，通俗易懂，可读性、可操作性和实用性强。本套规划教材适用于应用型高等院校各专业，也可作为本科院校举办的应用技术专业的课程教材，此外还可作为职业技术学院和民办高校、成人教育的教材以及从事工程应用的技术人员的自学参考资料。

我们感谢该套规划教材的各位作者为教材的出版所做出的贡献，也感谢中国水利水电出版社为选题、立项、编审所做出的努力。我们相信，随着我国高等教育的不断发展和高校教学改革的不断深入，具有示范性并适应应用型人才培养的精品课程教材必将进一步促进我国高等院校教学质量的提高。

我们期待广大读者对本套规划教材提出宝贵意见，以便进一步修订，使该套规划教材不断完善。

21 世纪高等院校规划教材编委会

2004 年 8 月

前 言

Java 语言是当今计算机软件行业中最热门的网络编程语言，以 Java 为核心的芯片技术、编译技术、数据库连接技术，以及基于企业级应用的 J2EE 技术得到了迅猛的发展。Java 的应用已经深入到我们生活的每一个角落。

Java 作为一种程序设计语言，简单、面向对象、不依赖于机器的结构，具有可移植性、鲁棒性、安全性，提供了并发机制，并且具有很高的效率。Java 语言最主要的特点是“Write once, run anywhere”，这句话一直是 Java 程序员的精神指南，也是 Java 语言深得程序员喜爱的原因之一。本书全面地讲解了 Java 的重要知识，尤其强调面向对象的设计思想和编程方法。通过本书的学习，读者可以掌握 Java 面向对象编程的思想和 Java 编程中的一些重要技术。

全书循序渐进，由浅入深，在编写的过程中通过大量的实例图文并茂地对重点内容加以阐述，力求做到言简意赅。书中的每个实例均上机调试过，便于读者通过实际上机操作进一步理解 Java 的原理、功能和作用。本书共分为 13 章，各章具体内容介绍如下：

第 1 章全面介绍 Java 诞生的背景、特点，介绍 Java 平台，JDK 的安装与配置，并对 Java 开发工具进行了简单描述。

第 2、3、5 章对 Java 语言的基础知识进行了详细地介绍，这是学习 Java 必须掌握的基本内容。

第 4、6 章是 Java 面向对象编程的知识，是面向对象编程的核心内容，包括类、对象、继承、多态和接口等。

第 7 章首先介绍 Java 的异常处理机制以及如何实现这种机制，然后介绍怎样利用 Java 提供的异常类处理异常。

第 8 章主要介绍 Java 语言的输入/输出流处理机制和文件的处理机制。Java 提供了丰富的输入/输出流操作类，功能强大。

第 9 章介绍多线程技术，它是 Java 的特色之一，线程是 Java 中的重要内容，通过大量实例帮助读者理解多线程编程。

第 10 章是基于 Java Swing 的 GUI 图形用户界面设计，讲解了常用的组件和容器、事件处理机制、布局管理器的使用方法。

第 11 章介绍基于 java.net 包编写网络通信程序设计，主要讲解网络编程中常用的 URL、Socket、InetAddress 等重要的类。

第 12 章介绍数据库的基本知识，重点介绍各种 JDBC 驱动程序以及按照 JDBC 编程步骤实现 JDBC 在数据库访问中的作用。

第 13 章介绍一个综合案例的开发。

本书第 1~6 章、第 13 章由刘震副教授编著，第 7~9 章，第 12 章由唐铁斌讲师编著，第 10 章由郝秋燕讲师编著，第 11 章由娄松涛讲师编著，并由曾强聪教授完成全书审稿。

本书可作为培养应用性、技能型人才的普通高等学校计算机及相近专业本科生（或高职高专学生）的 Java 语言程序设计课程教材，也可作为使用 Java 语言的软件开发人员的参考书。

本书在编写过程中，参考了大量文献、相关著作及其网络上的最新资料，在此表示衷心的感谢。对书中存在的不妥与错漏之处，敬请读者朋友批评指正。

编者于长沙

2011 年 9 月 28 日

目 录

序	
前言	
第 1 章 Java 概述	1
1.1 Java 的诞生	1
1.2 Java 的特点	1
1.2.1 简单性	2
1.2.2 面向对象	2
1.2.3 分布式	2
1.2.4 健壮性	2
1.2.5 安全性	2
1.2.6 平台无关	2
1.2.7 多线程	3
1.3 Java 开发环境安装与配置	3
1.3.1 平台介绍	3
1.3.2 用 JDK 管理 Java	3
1.4 创建并运行一个简单的 Java 程序	6
1.4.1 Java 程序开发步骤	6
1.4.2 一个简单 Java 程序的要素	7
1.4.3 Java Applet 程序	8
1.5 Java 开发工具	9
1.6 小结	10
习题 1	10
第 2 章 数据类型、运算符和表达式	11
2.1 标识符和关键字	11
2.1.1 标识符	11
2.1.2 关键字	11
2.2 基本数据类型	12
2.3 常量与变量	13
2.3.1 常量	13
2.3.2 变量	13
2.3.3 整数类型	14
2.3.4 浮点类型	15
2.3.5 布尔类型	15
2.3.6 字符类型	15
2.4 数据类型的转换	16
2.5 运算符与表达式	18
2.5.1 算术运算符	18
2.5.2 自增、自减运算符	18
2.5.3 关系运算符与关系表达式	19
2.5.4 逻辑运算符与逻辑表达式	20
2.5.5 位运算符	21
2.5.6 三目条件运算符	22
2.5.7 赋值运算符与赋值表达式	22
2.6 运算符的优先级	23
2.7 小结	24
习题 2	24
第 3 章 Java 程序控制语句	26
3.1 语句概述	26
3.1.1 表达式语句	26
3.1.2 复合语句	26
3.1.3 控制结构语句	27
3.2 顺序结构	27
3.3 分支（选择）语句	28
3.3.1 if 条件分支语句	28
3.3.2 switch 语句	30
3.4 循环语句	31
3.4.1 for 循环语句	32
3.4.2 while 语句	32
3.4.3 do-while 语句	33
3.4.4 嵌套循环	34
3.5 跳转语句	35
3.5.1 break 语句	35
3.5.2 break label 语句	36
3.5.3 continue 语句	37
3.5.4 continue label 语句	37
3.5.5 return 语句	38
3.6 小结	39
习题 3	39

第4章 类与对象	42	6.1.2 方法的覆盖	76
4.1 类	42	6.1.3 super 关键字	77
4.1.1 类的定义	43	6.1.4 final 关键字	78
4.1.2 成员变量	43	6.1.5 对象间的类型转换	79
4.1.3 成员方法和构造方法	44	6.2 类之源——Object 类	80
4.2 对象	49	6.3 抽象类和抽象方法	82
4.2.1 创建对象	49	6.4 多态	86
4.2.2 使用对象	50	6.5 接口	87
4.2.3 销毁对象	51	6.5.1 定义接口	87
4.2.4 this 关键字	51	6.5.2 实现接口	87
4.3 static 关键字	52	6.5.3 接口与抽象类的区别	88
4.3.1 实例变量和类变量	53	6.6 小结	88
4.3.2 实例方法和类方法	53	习题6	88
4.4 包	53	第7章 异常处理	90
4.4.1 创建包	53	7.1 异常概述	90
4.4.2 使用包中的类	54	7.2 异常类	91
4.4.3 常用系统包	55	7.2.1 系统定义的异常	91
4.5 访问控制	55	7.2.2 用户自定义的异常	92
4.5.1 public	55	7.3 异常处理	92
4.5.2 private	56	7.3.1 抛出异常	93
4.5.3 protected	57	7.3.2 捕获异常	94
4.5.4 friendly (缺省)	57	7.4 小结	97
4.6 小结	58	习题7	97
习题4	58	第8章 输入/输出流及文件操作	99
第5章 数组与字符串	61	8.1 流概述	99
5.1 数组	61	8.2 字节 I/O 流	101
5.1.1 一维数组	61	8.2.1 标准 I/O 流	102
5.1.2 二维数组	63	8.2.2 文件字节 I/O 流	102
5.1.3 Array 类	64	8.2.3 数据字节 I/O 流	104
5.2 字符串	65	8.2.4 对象字节 I/O 流	105
5.2.1 字符串常量	66	8.3 字符 I/O 流	105
5.2.2 String 类	66	8.3.1 抽象字符 I/O 流	105
5.2.3 StringBuffer 类	68	8.3.2 文件字符 I/O 流	106
5.2.4 其他字符串处理方法	70	8.3.3 缓冲字符 I/O 流	107
5.3 小结	71	8.4 文件操作	108
习题5	72	8.4.1 File 对象的创建	109
第6章 继承和接口	74	8.4.2 File 类成员方法	109
6.1 继承	74	8.5 小结	109
6.1.1 继承的实现	75	习题8	110

第9章 多线程	111	11.1 URL 概述	148
9.1 线程概述	111	11.2 URL 类	148
9.1.1 进程	111	11.2.1 创建 URL 对象	148
9.1.2 线程	111	11.2.2 URL 类的成员方法	149
9.2 创建线程	112	11.2.3 显示 URL 资源中的 html 文件	150
9.2.1 Thread 子类创建线程	113	11.3 InetAddress 类	151
9.2.2 Runnable 接口的使用	114	11.4 Socket 套接字	152
9.3 线程的生命周期	115	11.4.1 客户机/服务器模式	152
9.4 线程的控制	117	11.4.2 Socket 通信	153
9.5 线程同步	118	11.4.3 Socket 类	153
9.6 小结	122	11.4.4 ServerSocket 类	153
习题 9	122	11.5 小结	156
第10章 图形用户界面	124	习题 11	156
10.1 图形用户界面概述	124	第12章 数据库编程	157
10.1.1 AWT 和 Swing 包	124	12.1 数据库基础知识	157
10.1.2 Swing 组件和容器	124	12.1.1 数据库发展历史	157
10.2 事件处理	125	12.1.2 关系模型	157
10.3 Swing 组件	126	12.1.3 关系数据库系统	159
10.3.1 javax.swing.JFrame 类	127	12.2 SQL 语言	160
10.3.2 javax.swing.JTextField 类和 javax.swing.JPasswordField 类	128	12.3 JDBC 概述	163
10.3.3 javax.swing.JButton 类	130	12.3.1 JDBC 功能	163
10.3.4 javax.swing.JTextArea 类	130	12.3.2 JDBC 驱动程序的类型	164
10.3.5 javax.swing.JComboBox 类	131	12.3.3 JDBC URL	164
10.3.6 javax.swing.JMenuBar 类、 JMenu 类和 JMenuItem 类	132	12.4 JDBC 访问数据库	165
10.3.7 javax.swing.JTable 类	134	12.4.1 设置数据源	165
10.4 布局管理	136	12.4.2 连接数据库	167
10.4.1 FlowLayout 布局管理器	136	12.4.3 查询数据库	167
10.4.2 GridLayout 布局管理器	138	12.4.4 检索结果	168
10.4.3 BorderLayout 布局管理器	139	12.5 小结	169
10.4.4 CardLayout 布局管理器	140	习题 12	169
10.4.5 GridbagLayout 布局管理器	142	第13章 综合案例——人事资源管理系统	170
10.4.6 null 无布局管理器	144	13.1 需求分析	170
10.5 MVC 结构简介	146	13.2 系统设计	170
10.6 小结	146	13.3 数据库设计	172
习题 10	147	13.4 详细设计	174
第11章 Java 网络编程	148	13.5 程序设计	179
		参考文献	232

第 1 章 Java 概述



- Java 的诞生
- Java 的特点
- Java 开发环境安装与配置
- 创建并运行一个简单的 Java 程序

Java 语言是当今计算机软件行业中最热门的网络编程语言，以 Java 为核心的芯片技术、编译技术、数据库连接技术，以及基于企业级应用的 J2EE 技术得到了迅猛的发展。Java 的应用已经深入到我们生活的每一个角落。

1.1 Java 的诞生

Java 的发展历史，可追溯到 1990 年。当时 Sun 公司为了发展消费性电子产品而进行了一个名为 Green 的项目计划。该计划负责人是 James Gosling。起初他以 C++ 来写一种嵌入式程序，可以放在烤面包机或 PAD 等小型电子消费设备的芯片上，使得机器更智能。但他发现 C++ 并不适合完成这类任务！因为 C++ 有个缺点，就是 C++ 只能针对特定的操作系统和 CPU 芯片进行编译。这样一来，一旦电子设备更换了操作系统或 CPU 芯片就不能保证原程序正确运行，可能程序员需要修改程序并针对新的操作系统或 CPU 芯片重新进行编译。

为了解决遇到的问题，Gosling 决定要发展一种新的语言，来解决 C++ 的潜在性危险问题，这个语言名叫 Oak（Green 小组成员公司楼外有一棵橡树）。Oak 是一种可移植性语言，也就是一种平台独立语言，能够在各种操作系统、各种 CPU 芯片上运行。

1994 年，Oak 技术日趋成熟，这时网络正开始蓬勃发展。Oak 研发小组发现 Oak 很适合作为一种网络程序语言。因此发展了一个能与 Oak 配合的浏览器——WebRunner，后更名为 HotJava，它证明了 Oak 是一种能在网络上发展的程序语言。由于已经有一种计算机语言的名字叫 Oak，且被注册。最后，工程师们便想到以自己常享用的咖啡（Java）来重新命名。“Java”是印度尼西亚一个盛产咖啡的岛屿，中文译名为爪哇。

1.2 Java 的特点

Java 是一种简单的、面向对象的、分布式的、解释的、健壮的、结构中立的、与平台无关的、优异的、多线程的动态语言，也是目前使用最为广泛的网络编程语言之一。

1.2.1 简单性

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。特别地, Java 语言不使用指针, 并提供了自动的垃圾收集, 使得程序员不必为内存管理而担忧。

1.2.2 面向对象

面向对象是当前软件开发的重要方法。它是在编程过程中采用封装、继承、多态的编程方法。面向对象的概念和应用已超越了程序设计和软件开发, 扩展到很宽的范围。如数据库系统、交互式界面、应用结构、应用平台、分布式系统、网络管理结构、CAD 技术、人工智能等领域。

Java 语言提供类、接口和继承等原语, 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。

本书将在后续章节详细、准确地讨论面向对象的基本概念。

1.2.3 分布式

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`、`URLConnection`、`Socket`、`ServerSocket` 等。Java 的 RMI (远程方法调用) 机制也是开发分布式应用的重要手段。

1.2.4 健壮性

Java 的强类型机制、异常处理、垃圾的自动收集等是 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。

1.2.5 安全性

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 `SecurityManager`) 让 Java 应用设置安全哨兵。

1.2.6 平台无关

无论哪种语言编写的应用程序的运行都需要通过操作系统和处理器来完成。与平台无关是指程序代码不因操作系统、处理器的变化导致发生无法运行或出现运行错误。

Java 能运行于不同的平台, 这是因为 Java 引进虚拟机原理, 并运行于虚拟机, 在不同平台的 Java 接口之间实现。使用 Java 编写的程序能在世界范围内共享。Java 的数据类型与机器

无关, Java 虚拟机 (Java Virtual Machine) 是建立在硬件和操作系统之上, 实现 Java 二进制代码的解释执行功能, 提供于不同平台的接口的。

1.2.7 多线程

多线程机制能够使应用程序在同一时间并行执行多项任务, 而且相应的同步机制可以保证不同线程能够正确地共享数据。使用多线程, 可以带来更好的交互能力和实时行为。

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。

1.3 Java 开发环境安装与配置

对于初学者要进行 Java 的开发, 必须首先搭建 Java 开发环境 (运行平台)。目前, Java 2 平台有三个版本

1.3.1 平台介绍

(1) Java SE (以前版本称为 J2SE, Java 2 Platform Standard Edition)

——Java 标准版或 Java 标准平台。Java SE 提供了标准的 Java Development Kit (JDK)。利用该平台可以开发 Java 桌面应用程序和低端的服务器应用程序, 也可以开发 Java Applet 程序, Applet 是嵌入在 HTML 文件中的 Java 程序。当前最新的 JDK 版本为 JDK 6.0。

(2) Java EE (以前版本称为 J2EE, Java 2 Platform Enterprise Edition)

——Java 企业版或 Java 企业平台。使用 Java EE 可以构建企业级服务应用, Java EE 建立于 Java SE 基础之上, 增加了附加类库, 具有 Web 服务、组建模型, 以及通信 API 等特性, 这些为面向服务的架构 (SOA) 以及 Web 2.0 应用开发提供了支持。

(3) Java ME (以前版本称为 J2ME, Java 2 Platform Micro Edition)

——Java 微型版或 Java 小型平台。是一个技术和规范的集合, 它为移动设备 (消费类产品、嵌入式设备、高级移动设备等) 提供了基于 Java 环境的开发与应用平台。Java ME 目前分为两类配置, 一类是面向小型移动设备的 CLDC (Connected Limited Device Profile), 一类是面向功能更强大的移动设备, 如智能手机和机顶盒, 称为 CDC (Connected Device Profile)。

有关 Java SE、Java EE 和 Java ME 的详细内容, 可登录官方网站 <http://java.sun.com> 或 <http://www.oracle.com/technetwork/java/index.html> 进行了解。

1.3.2 用 JDK 管理 Java

1. JDK 简介

JDK 是 Java Development Kit (Java 开发工具包) 的缩写, 是 Sun Microsystems 针对 Java 开发的产品。它为 Java 应用程序提供了基本的开发环境。自从 Java 推出以来, JDK 已经成为使用最广泛的 Java 软件开发工具包。JDK 也是 Java SE 平台提供学习 Java 语言的最佳平台, 只有掌握了 Java SE 的 JDK 才能进一步学习 Java EE 和 Java ME。

一般初学 Java 时都选用 JDK 作为开发环境, 而目前很多优秀的 Java 集成开发环境 (IDE), 如 Eclipse, NetBeans, JCreator 等, 都是在 JDK 基础上建立的, 换句话说, 如果没有 JDK, 其他集成开发环境是无法工作的。

Sun 公司的 JDK 是免费的工具，可以到 Sun 公司网站 (<http://java.sun.com>) 或提供相关下载的网站下载。不同的版本适合不同的操作系统，用户可以根据自己所用的操作系统下载相应的 JDK 版本。本书以 Windows 操作系统平台为例进行开发环境的搭建，下载的版本为 jdk-6u24-windows-i586.exe (JDK6.0)。

2. 安装 JDK

JDK6.0 的下载地址：

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

下载完后双击 jdk-6u24-windows-i586.exe 文件就可以进行开发工具的安装，安装过程非常简单。当出现选择安装路径界面时，如图 1-1 所示，建议读者将默认的安装路径 C:\Program Files\Java\jdk1.6.0_24 改为 D:\jdk1.6.0_24，这样做是为了便于今后环境变量的设置。



图 1-1 JDK 安装路径的选择

表 1-1 显示了 JDK 安装成功后，在 D:\jdk1.6.0_24 目录下形成的目录结构。

表 1-1 JDK 主要目录内容

目录	描述
bin 子目录	存放开发工具和实用程序，如编译器 javac.exe 和解释器 java.exe
demo 子目录	存放 Java 平台的编程示例，含源码
include 子目录	存放支持使用 Java 本机界面、JVM 工具界面以及 Java 平台的其他功能进行本机代码编程的头文件
jre 子目录	存放 Java 运行环境包括 Java 虚拟机、类库以及其他支持执行以 Java 语言编写的程序的文件
lib 子目录	存放开发工具所需的其他类库和支持文件
sample 子目录	存放 Java API 编程样例，含源码

3. 配置 Path 和 ClassPath 环境变量

JDK 安装结束后，为了方便 Java 程序的编译和运行，还需要对其运行环境进行配置，即 Path 和 ClassPath 的设置。Path 的设置主要是为了能够在 DOS 命令行下找到 Java 编译与运行

所用的程序，如 `javac.exe` 和 `java.exe` 等；而 `ClassPath` 的设置主要是为了让 Java 虚拟机能够找到需要的类库。不同的操作系统设置方法略有差异，下面以 Windows 7 为例分别讲解 `Path` 和 `ClassPath` 的设置步骤。

(1) 系统环境 `Path` 的设置。

在 Windows 7 操作系统中选择【控制面板】→【系统】→【高级系统设置】→【高级】→【环境变量】命令就可以设置环境变量 `Path`，如图 1-2 所示的【环境变量】对话框，在系统变量中找到 `Path` 变量，点击【编辑】按钮，在图 1-3 所示的对话框中对 `Path` 变量进行编辑或修改。



图 1-2 【环境变量】对话框

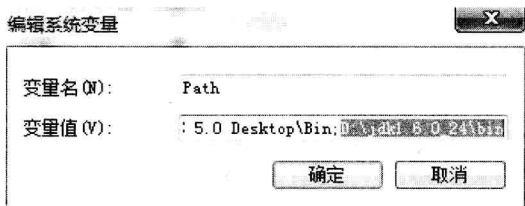


图 1-3 设置系统环境变量 `Path`

(2) 系统环境 `ClassPath` 的设置。

在 Windows 7 操作系统中选择【控制面板】→【系统】→【高级系统设置】→【高级】→【环境变量】→【新建系统变量】命令就可以设置环境变量 `ClassPath`，如图 1-4 所示。

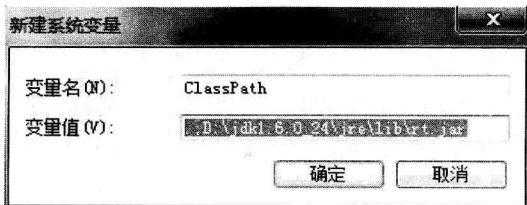


图 1-4 设置环境变量 `ClassPath`

1.4 创建并运行一个简单的 Java 程序

Java 程序分为三类, 即 Application (Java 应用程序)、Java Applet (Java 小程序) 和 Servlet (服务器端应用程序)。Java 应用程序可以在安装了 Java 标准平台的任何计算机上运行。Java Applet 不需要安装 Java 标准平台, 只能嵌在 HTML 网页中由支持 Java 的浏览器直接运行。Servlet 是运行在服务器端的小程序, 它可以处理客户传来的请求 (request), 然后对客户的请求给予响应 (response)。本节以 Java 应用程序开发为例。

1.4.1 Java 程序开发步骤

开发一个 Java 应用程序分为三个步骤:

(1) 创建一个带有文件扩展名 *.java 的源文件。

1) 使用编辑器 (如记事本), 输入如图 1-5 的例 1.1 中所示的 6 行文本。

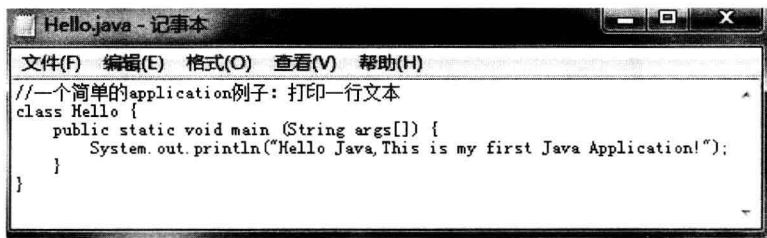


图 1-5 Hello.java 程序

2) 把该文件命名为 Hello.java, 并保存。为了方便起见, 本章实例文件保存的目录为 E:\Java_pro。

(2) 使用 Java 编译器 javac.exe 编译源文件生成一个带有文件扩展名 *.class 的字节码文件。

1) 打开 Windows 命令提示符。单击【开始】→【运行】命令, 在文本框输入 “cmd” 命令后单击【确定】按钮进入 DOS 窗口。

2) 使用 cd 命令将路径转入源文件 “Hello.java” 所在的路径 (E:\Java_pro), 输入 “javac Hello.java” 命令, 并按回车键, 如图 1-6 所示。



图 1-6 编译 Hello.java 源文件

(3) 使用 Java 解释器 java.exe 运行字节码。

上一步中执行“javac Hello.java”命令后，如果没有任何提示信息，则表示编译成功，且生成一个字节码文件“Hello.class”。在当前路径下，输入“java Hello”命令，回车。运行结果如图 1-7 所示。

```

E:\Java_pro>dir
驱动器 E 中的卷是 WORK
卷的序列号是 A2D7-1376

E:\Java_pro 的目录

2011/02/26 00:51 <DIR>      .
2011/02/26 00:51 <DIR>      ..
2011/02/26 00:51              449 Hello.class
2011/02/26 00:40              201 Hello.java
                2 个文件          650 字节
                2 个目录 65,652,113,408 可用字节

E:\Java_pro>java Hello
Hello Java, This is my first Java Application!

E:\Java_pro>

```

图 1-7 运行结果

1.4.2 一个简单 Java 程序的要素

【例 1.1】Hello.java

```

1 // 一个简单的 application 例子：打印一行文本
2 class Hello {
3     public static void main (String[] args) {
4         System.out.println("Hello Java, This is my first Java Application!");
5     }
6 }

```

第 1 行在程序中是注释行，编译器会忽略注释内容，注释可以加在程序中任何位置。注释的目的是为程序增加必要的解释，提高程序的可读性，适当给代码增加注释是一个良好的编程习惯。Java 支持三种注释方式：单行注释、多行注释和 Javadoc 文档注释。例 1.1 中的第 1 行使用符号“//”表示单行注释开始，该行从“//”后为注释内容。多行注释使用“/* ... */”进行注释，以“/*”表示注释开始，“*/”表示注释结束。例如：

```

/*一个简单的 application 例子
功能：打印一行文本
*/
class Hello { //声明类
    public static void main (String args[]) {
        System.out.println("Hello Java, This is my first Java Application!");
        //调用 println() 方法，显示字符串
    }
}

```

Javadoc 文档注释以符号“/**”表示注释开始，“*/”表示注释结束。Javadoc 注释可以被 JDK 内置的 javadoc.exe 工具制作程序的开发文档。将例 1.1 代码进行如下修改，并保存。

```

/**
 *Simple Java Application
 *@author smalltang

```

```

    *@version 1.0 2011.2.1
    */
    class Hello {
        public static void main (String args[]) {
            System.out.println("Hello Java,This is my first Java Application!");
        }
    }

```

然后使用下列命令：

```
javadoc -d e:\html document hello.java
```

在 E:\html document 目录中得到 Hello 类的 HTML 文档，用浏览器打开 HTML 文档，效果如图 1-8 所示。



图 1-8 Hello 类的 Javadoc 网页

第 2 行声明类的名称 Hello。当源文件被编译时，根据该源文件中的类名会创建一个 Hello.class 文件。

第 3 行 main()方法是程序的执行的开始。

第 4 行程序调用 println()方法将字符串写入到标准输出流。

1.4.3 Java Applet 程序

Java Applet 就是用 Java 语言编写的一些小应用程序，它们可以直接嵌入到网页中，并能够产生特殊的效果。包含 Applet 的网页被称为 Java-Powered 页，可以称其为 Java 支持的网页。

【例 1.2】HelloApplet.java

```

import java.applet.*;    //引入 java.applet 包中的系统类
import java.awt.*;        //引入 java.awt 包中的系统类

```