



21世纪高等院校规划教材

计算机程序设计类

Java程序 设计及应用

李宗军 滕延燕 编 著



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 程序设计及应用

李宗军 滕延燕 编著



中国水利水电出版社
www.waterpub.com.cn

内 容 提 要

Java 从诞生至今发生了许多重大的发展变化,本书是为了满足学习 Java 编程的人员深入研究 Java 而编写的。

本书共分 11 章,分别讲解了 Java 入门、编程基础、面向对象、泛型、异常、线程、I/O、集合框架、常用 API、枚举、GUI 编程等内容。

本书吸收了最新的 Java 技术和语法;融合了编程思想、理论、实践、应用于一体;并且知识点前呼后应,始终站在 Java 编程整体的高度上来分章节讲解,使读者掌握的知识点是有机的整体;为读者留下了一定的思考空间,培养读者思考问题的能力。

本书配有《Java 程序设计实训》,精心设计的典型案例都来源于企业项目,与教材紧密结合,使理论知识与实践技能紧密结合、交互循环贯通。随书附赠光盘内容为本书所有范例源程序、相关软件、教学动画演示等。

本书可作为理工科高等院校的“Java 程序设计”课程教材,也可作为软件培训或者自学的教材。本书是作者长期从事 Java 研究、开发、教学、企业软件项目指导的心得体会,在此奉献给大家,愿本书能够为广大 Java 爱好者提供有益的帮助。

本书配有电子教案,读者可以到中国水利水电出版社网站和万水书苑上免费下载,网址为<http://www.waterpub.com.cn/softdown/>和<http://www.wsbookshow.com>。

图书在版编目(CIP)数据

Java 程序设计及应用 / 李宗军, 滕延燕编著. — 北京: 中国水利水电出版社, 2010.3
21 世纪高等院校规划教材
ISBN 978-7-5084-7271-3

I. ①J… II. ①李… ②滕… 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 034575 号

策划编辑: 雷顺加 责任编辑: 杨元泓 加工编辑: 俞 飞 封面设计: 李 佳

书 名	21 世纪高等院校规划教材 Java 程序设计及应用
作 者	李宗军 滕延燕 编著
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 电话: (010) 68367658 (营销中心)、82562819 (万水)
经 售	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	北京市天竺颖华印刷厂
规 格	184mm×260mm 16 开本 14 印张 343 千字
版 次	2010 年 3 月第 1 版 2010 年 3 月第 1 次印刷
印 数	0001—4000 册
定 价	24.00 元

凡购买我社图书,如有缺页、倒页、脱页的,本社营销中心负责调换

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跃式发展,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的需要。本套教材消化和吸收了多年来已有的应用型人才探索与实践成果,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,注重程序设计算法的实现;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 是世界上最流行的编程语言，在软件产业中占有重要地位，全国各高等院校已经纷纷开设 Java 课程，培养 Java 软件开发人才。

本书主要是为了配合高等院校 Java 教学的深入研究而编写的。按照《教育部关于进一步加强高等学校本科教学工作的若干意见》，全书以最新的《Java 语言规范 3.0》为依据，较深刻地讲解了 Java 语言以及面向对象的编程思想、方法，并且引入了自 Java SE 5.0 以来的新语法，如优化的 for 循环、泛型、自动装包/拆包、枚举、可变参数、静态导入等。本书采用“思想+理论+实践+应用”的编写方式，以企业软件编程为导向，引入案例教学和启发式教学方法，便于激发学习兴趣。本书的编写思想是：把理论知识点融入到编程实践中去，立足于“启发思想，理论够用，重在实践，应用导向”的要求，力求做到博精简易，符合“思想是源泉、理论是基础、实践积经验、应用是目的”的理念。每一章都尽量挖掘理论背后的思想，思想通了，理论就好掌握了，通过编程实例从实践的角度对理论加以验证，理论是抽象的，把抽象的理论用具体的程序代码表示出来，从而进一步提高学生对理论的理解，同时培养学生的实践动手能力，最后达到学习知识、培养能力的目的。另外，在案例的选择上，遵循“在能够验证理论的前提下尽量简单”的原则，这样有利于激发学生的求知欲望。

程序设计的核心问题是思想，这是程序设计的灵魂，程序设计处处都可以凝聚思想，这是本书的出发点，在本书的许多地方都有体现，如面向对象的思想，从物理大世界与计算机小世界这个角度来阐述，初学者就比较容易理解。

本书一方面可以引导读者学习编程、掌握编程思想和技术，另一方面还可以培养读者好的素质。例如，计算机编程能够锻炼一个人做事的细心和耐心（因为计算机程序不允许出现一丝差错，否则程序就不能正确运行），细节决定成败，这对学生的素质培养很重要。

本书在重点讲解 Java 的同时，尽量和其他相关学科联系起来，适当地穿插贯通，这对学生学习编程很有好处，因为软件编程不是独立的、孤立的，而是和其他学科有着密切的关联，如数据结构、软件工程、哲学认知思想等。

为培养读者的思考能力，本书在讲解知识点或结论时，首先给出知识点或结论，读者读到这里时，根据前面介绍的知识点往往很难完全理解，必然要暂时停下来思考一下，然后再继续往后看就明白了，原来答案就在后面，看到答案后再与自己的思考做比较。这里提醒读者的是，当遇到不理解的知识点或结论时，千万不要一直停下来思考，要继续读下去，这时就会从山穷水尽走入柳暗花明。教材中有些知识点或结论只是给出了，并没有解释，有些是放到了《Java 程序设计实训》的案例中，要求读者编程验证，其用意也在于此。

本书在讲解过程中有些内容用到了后面的知识点，这并非是内容知识点的先后顺序安排有问题，而是这些知识点虽然属于不同的大类，但它们之间的关系非常紧密之故。Java 编程是一项整体综合运用各种编程知识点的事情，有许多知识点紧密交叉，而作为教程只能按照线性关系分章节先后讲解，所以在有关知识点的介绍上采取了前呼后应的方式，以求站在 Java 整体的高度上来看。

本书的主要特点体现在以下几个方面：一是本书吸收了最新的 Java 技术和语法；二是融

合了编程思想、理论、实践、应用于一体，培养编程思想，训练编程方法；三是把物理宇宙大世界与计算机小世界相统一，认知世界，学习编程，使读者学习面向对象编程更直观、简易；四是本书知识点做到了前呼后应，在分章节讲解的过程中始终站在 Java 编程整体的高度上，使读者掌握的知识点不再是零散的，而是有机的整体；五是给读者留下了一定的思考空间，培养读者思考问题的能力；六是提供了精心设计的习题，汇聚编程的常见错误、原因和解决方法，能够帮助学生开阔视野；七是本书配有《Java 程序设计实训》，精心设计了許多案例，与教材一起，使学生的理论知识与实践技能交互循环贯通。

为了便于组织教学和学生自学，本书共分 11 章，第 1 章介绍了 Java 编程的入门知识和开发环境，第 2 章介绍了 Java 语言的基本语法，第 3 章是面向对象的思想和方法，第 4 章是泛型，第 5 章是异常处理，第 6 章是多线程，第 7 章是输入输出，第 8 章是集合框架，第 9 章介绍了一些常用的 API，第 10 章是枚举，第 11 章是 GUI 编程。带*的章节属于选修内容。另外，每章都配有一定数量的习题，这些习题有一定的深度和难度，对启发学生的编程思想、提高学生的软件设计和开发能力很有好处。

本书的总体结构与编写思想由李宗军和滕延燕设计编写。其中第 1、2、4 章以及附录部分由滕延燕编写，其余部分由李宗军编写。同时参加编写的还有李志敏、高峰、王金良、周丽美、刘慧、尹海丽、王广彬（青岛科技大学）、李洪伟（山东科技大学）、钱守国（青岛大学）、杨玉霞、段建丽、姚惠萍、王丽丽等老师。我的学生高素真、罗秀基、李海静、戴明霞、毛静、冯彦君、高金凤、胡克平、刘涛、宋慧、曾雨、邹剑邦等，他们都参与了我们许多项目的研发工作，为本教材编写了大量的程序代码，祝愿他们在以后的工作和生活中一切顺利，祝愿他们取得更大的成绩。全书得到了中国水利水电出版社相关领导的大力支持和北京万水电子信息有限公司策划团队的用心指导，特别是雷顺加编审、俞飞和周益丹编辑在本书的策划和写作中，提出了很好的建议，使得本书能够更好地用于教学，在此深表感谢。在本书编写过程中参考了大量国内外计算机网络文献资料，在此，谨向这些著作者以及为本书出版付出辛勤劳动的同志深表感谢！

本书的编写得到了青岛理工大学和理学院领导的大力支持，在此表示衷心的感谢！另外，感谢理学院计科教研室和数学教研室全体同仁为本书的出版所做的努力！

限于作者水平，书中可能存在错误和不妥之处，敬请各位专家和读者指正，我们的 E-mail 是：li_zjun@126.com。此外，由于受篇幅所限，本书只包含了 Java 编程和软件开发的基础知识，若读者希望了解更全面的 Java，可访问网站 <http://java.sun.com>。

作 者
2009 年 10 月

目 录

序

前言

第1章 Java 入门	1	2.2 Java 运算符	26
1.1 Java 的起源与发展	1	2.2.1 算术运算符	26
1.2 Java 开发环境	2	2.2.2 比较运算符	27
1.2.1 两种开发环境	2	2.2.3 逻辑运算符	27
1.2.2 搭建开发环境	2	2.2.4 位运算符	27
1.2.3 了解环境变量	3	2.2.5 赋值运算符	28
1.2.4 了解 JDK 的目录结构	4	2.2.6 运算符的优先级	28
1.3 初步体验 Java 编程	4	2.3 流程控制	28
1.3.1 HelloWorld 案例	5	2.3.1 顺序结构	28
1.3.2 案例分析	5	2.3.2 条件分支结构	29
1.4 JDK API 文档的使用	7	2.3.3 循环结构	31
1.5 Java 学习的建议	8	2.3.4 跳转语句	34
1.5.1 打好基础	8	本章小结	35
1.5.2 眼手脑并用	8	习题 2	35
1.5.3 JDK API 文档	8	第3章 面向对象	38
1.5.4 有效使用网络资源	8	3.1 物理世界与面向对象	38
1.5.5 计算机英语	8	3.1.1 人与计算机的关系	38
本章小结	8	3.1.2 物理世界的认知	38
习题 1	9	3.1.3 面向对象方法与 UML	42
第2章 Java 编程基础	10	3.2 类与对象	43
2.1 Java 语言基础	10	3.2.1 类的定义	43
2.1.1 注释	10	3.2.2 构造方法	46
2.1.2 关键字和标识符	10	3.2.3 对象的创建与使用	47
2.1.3 基本数据类型	11	3.3 封装	47
2.1.4 字面值与常量	15	3.3.1 访问修饰符	47
2.1.5 变量	16	3.3.2 封装的保护作用	48
2.1.6 基本类型的类型转换	17	3.3.3 增强独立性	48
2.1.7 数组	18	3.4 类的继承	48
2.1.8 方法的参数传递	21	3.4.1 继承	48
2.1.9 方法的可变参数	23	3.4.2 继承的 UML 符号	49
2.1.10 格式化输出	24	3.4.3 this 与 super 关键字	50
2.1.11 包	25	3.5 多态	51

3.5.1 重载	51	4.2 定义简单的泛型	81
3.5.2 覆盖	52	4.3 泛型和子类继承	82
3.6 接口与抽象类	53	4.4 泛型的通配符	83
3.6.1 接口	53	4.5 泛型方法和类型推断	84
3.6.2 抽象类	54	本章小结	85
3.7 引用类型的类型转换	54	习题 4	85
3.7.1 自动类型转换	54	第 5 章 异常处理	86
3.7.2 强制类型转换	55	5.1 什么是异常	86
3.7.3 自动打包/拆包机制	56	5.2 异常处理机制	86
3.8 其他修饰符	56	5.2.1 try/catch/finally	88
3.8.1 final	56	5.2.2 throws 与 throw	89
3.8.2 static	57	5.2.3 异常处理编程原则	89
3.8.3 native	60	本章小结	90
3.8.4 transient	61	习题 5	90
3.8.5 strictfp	61	第 6 章 多线程	91
3.8.6 volatile	61	6.1 基本概念	91
3.8.7 assert	61	6.2 线程对象的创建与启动	91
3.9 类实例对象的创建过程	63	6.2.1 继承 Thread 类	91
3.10 内部类	64	6.2.2 实现 Runnable 接口	93
3.10.1 成员内部类	64	6.2.3 两种创建方式的比较	93
3.10.2 方法内部类	67	6.3 线程的生命周期	93
3.10.3 匿名内部类	67	6.4 前台线程和后台线程	94
3.11 对象之间的关系	69	6.5 线程的分支与合并	95
3.11.1 泛化	69	6.6 多线程的同步	96
3.11.2 聚合、组合	70	6.6.1 线程安全	96
3.11.3 依赖	71	6.6.2 同步代码块	97
3.11.4 关联	71	6.6.3 同步方法	98
3.12 面向对象程序设计的基本原则	72	6.7 线程间的通信	100
3.12.1 单一职责原则	72	本章小结	102
3.12.2 开闭原则	72	习题 6	102
3.12.3 里氏代换原则	73	第 7 章 输入输出 I/O	103
3.12.4 依赖倒转原则	75	7.1 I/O 概述	103
3.12.5 接口隔离原则	76	7.2 字符集编码	104
3.12.6 合成复用原则	77	7.2.1 基本概念	104
3.12.7 迪米特法则	77	7.2.2 字符集编码的发展	105
本章小结	77	7.2.3 汉字编码	107
习题 3	78	7.2.4 内码和 codepage	108
第 4 章 泛型	80	7.2.5 字符集编码的 Java 处理	109
4.1 为什么要引入泛型	80	7.3 File 类与 RandomAccessFile 类	112

7.3.1 File	112	9.4 Scanner 类	151
7.3.2 RandomAccessFile	113	本章小结	151
7.4 流	115	习题 9	152
7.4.1 低级字节流	116	第 10 章 枚举 enum*	153
7.4.2 高级字节流	119	10.1 声明一个简单的枚举	153
7.4.3 字符流	121	10.2 带参数的枚举值与构造方法	155
7.4.4 对象流与序列化	122	10.3 带体的枚举值	155
7.5 NIO 包简介	124	10.4 实现接口	156
7.5.1 为什么要引入 NIO 包	124	本章小结	157
7.5.2 流与块的比较	124	习题 10	157
本章小结	124	第 11 章 GUI 编程	158
习题 7	124	11.1 GUI 概述	158
第 8 章 Java 集合框架	125	11.2 GUI 编程三要素	158
8.1 集合框架	125	11.3 图形组件	159
8.1.1 引入集合框架的原因	125	11.4 容器组件	159
8.1.2 集合框架视图	125	11.4.1 Applet	159
8.2 集合框架中的常用类与接口	126	11.4.2 Frame	161
8.2.1 Iterator	126	11.4.3 Panel	162
8.2.2 Collection	126	11.4.4 Dialog	162
8.2.3 Map	130	11.4.5 ScrollPane	162
8.2.4 集合处理工具	132	11.5 布局管理器	163
本章小结	132	11.5.1 FlowLayout	163
习题 8	133	11.5.2 BorderLayout	164
第 9 章 常用 JDK API	134	11.5.3 CardLayout	165
9.1 Object 类	134	11.5.4 GridLayout	166
9.1.1 Object 介绍	134	11.6 事件处理	167
9.1.2 getClass 方法与 reflection 机制	135	11.6.1 事件源	167
9.1.3 toString 方法	137	11.6.2 事件	168
9.1.4 equals 方法与==	138	11.6.3 监听器	168
9.2 字符串的相关问题	139	11.6.4 监听器接口适配器	170
9.2.1 String、StringBuffer、StringBuilder 类	139	11.7 菜单	173
9.2.2 字符串的解析	140	11.7.1 下拉式菜单	173
9.2.3 正则表达式*	141	11.7.2 弹出式菜单	174
9.3 其他常用的 JDK 类	144	11.8 Graphics 绘图	176
9.3.1 Math、StrictMath 与 Random	144	11.8.1 RGB 颜色模式	176
9.3.2 日期类	144	11.8.2 字体	177
9.3.3 System 类	147	11.8.3 Java 的绘图机制	177
9.3.4 Runtime 类	149	11.9 国际化 I18N	179
		11.9.1 软件的国际化和本地化	179

11.9.2	118N 编程模型	180	11.11	Swing 简介*	193
11.9.3	Locale	181		本章小结	193
11.9.4	本地化的资源	181		习题 11	193
11.10	MVC 简介	185	附录 A	Java 编码规范	194
11.10.1	什么是 MVC	185	附录 B	常用操作命令	206
11.10.2	观察员模式	186	附录 C	Java 发展大事记	209
11.10.3	MVC 综合例程	186		参考文献	212

第 1 章 Java 入门

本章内容是 Java 入门，重点是开发环境的搭建和编程入门。通过本章的学习，应了解 Java 的起源和发展，以及学习 Java 的方法和资源，理解 Java 跨平台的原理，熟练掌握搭建 Java 的开发环境，掌握 Java 程序的结构、代码编写格式，初步学会编写、调试简单的 Java 应用程序。

1.1 Java 的起源与发展

学习 Java，首先要弄清楚 Java 是什么。通常可以认为：Java 是一种通用的、并发的、基于类的、面向对象的高级编程语言，吸取了 C/C++ 的优点，摒弃了其缺点，具有简单、面向对象、分布式、多线程、高性能、可移植、健壮、安全、跨平台等特点。Java 在嵌入式软件、桌面软件和网络分布式软件领域具有广泛的应用。

Java 起源于 1991 年 Sun 公司的一个叫“Green”的项目，该项目是为家用消费电子产品开发的一个分布式代码系统，对电视机等家用电器进行交互式控制，Java 之父 James Gosling 为其开发了一个新的程序设计语言，即后来的 Java。

下面简略地列出了 Java 发展的大事记（详见附录 C）：

1995 年 5 月 23 日，Sun 在 SunWorld'95 上正式发布 Java 和 HotJava 浏览器，这标志着 Java 的诞生。触发了 Java 进军 Internet，从此一发而不可收拾。此后，Java 应用之广泛，发展之迅猛是当今任何一种其他的计算机编程语言所无法比拟的。

1996 年 1 月，Sun 推出了 Java 开发工具包 JDK（Java Development Kit）1.0，当年 12 月发布 JDK 1.1。

1997 年 4 月 2 日，Sun 在美国旧金山举行第二届 JavaOne 开发应用者大会，1 万多人参加，与会人数令人瞩目。

1998 年 2 月，JDK 1.1 被下载超过 200 万次，同年发布了 JDK 1.2（此时 Java 改称 Java 2 以表示 Java 发生的重大变化）。

1999 年，Java 技术被分成 J2SE（原来的 JDK）、J2EE、J2ME 三个平台，Java Server Pages（JSP）技术公诸于众，J2SE 平台开始支持 Linux 操作系统。

2000 年 5 月 8 日，J2SE 1.3 发布，Java 开发成员上升到 150 万。

2001 年 9 月 24 日，J2EE 1.3 发布。

2002 年 2 月 26 日，J2SE 1.4 发布，自此 Java 的计算能力和运行速度有了大幅提升。

2004 年 9 月 30 日，J2SE 1.5 发布，是 Java 语言发展史上的又一里程碑事件，Java 语言增加了许多新语法特性（泛型 Generic，For-Each 循环，自动装包/拆包 Autoboxing/unboxing，枚举 Enums，可变参数 Varargs，静态导入 Static Imports），其计算性能再次得到了大幅提升，引入了多线程并发控制包开始支持多核 CPU。为了表示这个版本的重要性，J2SE 1.5 的发布版命名为 J2SE 5.0。

2005 年 6 月, JavaOne 大会召开, 此时, Java 的各种版本已经更名并取消其中的数字“2”: J2EE 更名为 JavaEE, J2SE 更名为 JavaSE, J2ME 更名为 JavaME。

2007 年, Sun 推出 JavaSE 6 (即 1.6), 其计算性能又得到了进一步的提升, 开始支持脚本 (script) 语言并进一步加强对多核 CPU 的支持。

Java 发展之迅猛、应用之广泛、参与者之众, 史无前例。这些都给 IT 界以极大的震撼。促使人们投入极大的热情来学习 Java。希望对读者朋友们也能产生如此之效果, 则作者欣慰矣。

Java 发展到现在已经形成三个平台: JavaSE、JavaEE 和 JavaME。这三个平台分别涉及了桌面平台、网络分布式平台和嵌入式平台。这三个平台合起来覆盖了应用软件领域的方方面面, 甚至在基础软件数据库、操作系统等领域也有很多 Java 的产品。本教材就是讲述如何在 JavaSE 平台上开发 Java 程序的, 这是其他两个平台的基础。

1.2 Java 开发环境

1.2.1 两种开发环境

在企业软件开发中, 一般都采用 IDE (Integrated Development Environment) 作为开发环境, 目前流行的 IDE 有 NetBeans、Eclipse。IDE 的优点是方便易用, 能够大大提高软件的开发效率和质量, 其缺点是不适合 Java 初学者使用。因为学习 Java 编程与企业软件开发的目标是不同的, 前者重点在于学习 Java 编程的原理、经验、技巧等方面, 而后者重点在于软件开发的质量、效率等方面。

对于学习 Java 编程而言, 尤其是 Java 初学者, 最好的开发环境就是“JDK+文本编辑器”。我们称这种开发环境为学习型开发环境。JDK (Java Development Kits) 是较早的称呼, 后来改叫 J2SE, 现在则称为 JavaSE, 在 Java 中这三种叫法不再区分, 都是指同一个东西。JDK 是一个平台, 包含了 Java 编译器、运行 Java 程序的虚拟机 JVM (Java Virtual Machine) 等许多 Java 程序开发工具, 还有一整套 API (Application Programming Interface)。可以免费从 Sun 公司的网站 (<http://java.sun.com>) 下载 JDK。文本编辑器用于编写 Java 程序的源代码文件, 常见的有 notepad++、UltraEdit、EditPlus、Vim 等。对文本编辑器没有特别要求, 任何一种文本编辑器都可以, 只要你觉得好用就行。有些初学者认为 MS Word 也是文本编辑器, 这是不对的, 所以不能使用 Word 来编写程序。

初学者使用学习型开发环境会显得笨手笨脚, 而且很容易出错, 而克服这些困难正是学习 Java 编程要求必须掌握的编程经验、技巧。

1.2.2 搭建开发环境

搭建学习型开发环境分为三个步骤: 安装软件、设置环境变量和测试开发环境。下面详解之。

(1) 安装软件。在 Windows 下安装 JDK 和文本编辑器很简单, 只需在资源管理器中双击软件安装包即可 (软件安装包在附带的光盘中), 不过要注意以下几点:

1) 安装路径中不要有空格,也不要有汉字,否则在使用过程中有可能出现问题。

2) 不要安装在操作系统所在的分区中 (Windows 通常为 C 区)。

3) 安装过程中会提示要安装一个公共的 JRE (Java Runtime Environment, 也称为 jvm), 注意其安装路径不要和 JDK 的安装路径相同。

(2) 设置环境变量。安装软件完毕后,接下来需要配置环境变量 path 和 classpath,即向 path 环境变量追加 JDK 安装根目录下的 bin 目录的全路径,新建 classpath 环境变量,并设置为句点“.”,即当前路径。

(3) 测试开发环境。打开命令窗口,执行指令 javac,若出现如下信息则说明开发环境搭建成功:

用法: javac<选项><源文件>

其中,可能的选项包括:

-g	生成所有调试信息
-g:none	不生成任何调试信息
-g:{lines,vars,source}	只生成某些调试信息
-nowarn	不生成任何警告
-verbose	输出有关编译器正在执行的操作的消息
-deprecation	输出使用已过时的 API 的源位置
-classpath <路径>	指定查找用户类文件的位置
-cp <路径>	指定查找用户类文件的位置
-sourcepath <路径>	指定查找输入源文件的位置
-bootclasspath <路径>	覆盖引导类文件的位置
-extdirs <目录>	覆盖安装的扩展目录的位置
-endorseddirs <目录>	覆盖签名的标准路径的位置
-d <目录>	指定存放生成的类文件的位置
-encoding <编码>	指定源文件使用的字符编码
-source <版本>	提供与指定版本的源兼容性
-target <版本>	生成特定 VM 版本的类文件
-version	版本信息
-help	输出标准选项的提要
-X	输出非标准选项的提要
-J<标志>	直接将 <标志> 传递给运行时系统

搭建开发环境的详细过程可参考实训教材附带光盘中的动画演示 JDK Install_windows.swf 或者 JDKInstall_linux.swf。

1.2.3 了解环境变量

环境变量属于操作系统的知识。所谓环境变量,就是一些如 varName=varValue 的一类变量,其目的在于数据共享,即在一定的权限范围内的计算机软件可以共享这些环境变量。按照共享权限的不同,环境变量分为系统环境变量和用户环境变量。系统环境变量允许登录到计算机中的任意用户所能够使用的任意软件所共享,而用户环境变量只允许登录到计算机中的特定用户所能够使用的任意软件所共享。操作系统自身预定义了一部分环境变量,例如 path、pathext、processor_architecture 等;有些特定的应用软件也定义了自己的环境变量,例如 JDK

的 classpath 环境变量, Tomcat 的 Java_HOME 环境变量等。环境变量名在 Windows 中不区分字母大小写, 而 Linux/UNIX 等则为大写。

下面以 path 和 classpath 为例来解释环境变量。

path 是路径的意思, 该变量的值是一些路径, 在 Windows 中不同的路径之间用分号“;”分割, 在 Linux/UNIX 中采用冒号“:”分割, 注意分割符的前后不允许有空格, 类似下面的样子。

```
path=C:\windows;D:\jdk\bin;E:\mypath
PATH=/usr/bin:/usr/local/bin
```

打开 Windows 的命令窗口, 很多人称为 DOS, 在里面输入任意一条外部指令 (非 DOS 指令), 例如 javac, 然后按 Enter 按钮执行, 这时候 Windows 就会取得 path 环境变量指定的路径, 按照从左到右的顺序来寻找 javac 这个可执行文件, 找到后就执行, 若所有路径都搜索完了还找不到, 就给出找不到的提示信息。对于上面的 path 来说是先找 C:\windows 目录, 注意不包括子目录, 找到后就执行, 否则就接着在 D:\jdk\bin 中继续找, 若找到了就执行, 否则就接着在 E:\mypath 中找, 找到就执行, 否则给出找不到的提示信息。

classpath 的作用和 path 类似, 是 JDK 在运行一个 Java 类时, 就会按照 classpath 环境变量指定的路径来寻找这个 Java 类的字节码文件 (后缀为 .class), 找到后就运行, 找不到就给出提示信息。很多初学者在上机试验时, 往往会在这个地方出错。在学习 Java 语言阶段, 设置 classpath 环境变量时, 很多教材中介绍说把 JDK 中 lib 目录下的 jar 文件路径赋给 classpath 环境变量, 这是对 JDK 1.3 及以前的版本而言, 对于 JDK 1.4 以及以后的版本, classpath 的值只取句点“.”就可以了, 而 JDK 6 默认设置了 classpath 为“.”, 故如果使用 JDK 6, 也可以暂时不设置 classpath 环境变量。

1.2.4 了解 JDK 的目录结构

假如 JDK 的安装目录是 D:\Java\jdkroot, 公共 JRE 的安装目录是 D:\Java\jreeroot。注意这里 jdkroot 和 jreeroot 不是真正的目录, 而是代指实际的安装 JDK 和公共 jre 的根目录。打开 jdkroot 目录, 其内容如下:

bin	提供 JDK 的工具程序, 如 javac、java 等等
demo	提供了一些 Java 范例供学习参考之用
lib	JDK 的 java api 类库, 在 jar 压缩包中
jre	自带的 jre

注意 jdkroot\jre 和 jreeroot 都是用于运行 Java 程序的 JRE, 但前者是用于测试运行开发过程中 Java 程序的, 而后者则是用于运行已开发好的 Java 程序的, 后者在安装过程中还会向 Web 浏览器中安装一个 applet plugin 插件, 是用于运行 Java Applet 程序的 (详见 11.4.1)。

1.3 初步体验 Java 编程

Java 程序的开发步骤是: (1) 使用纯文本编辑器输入 Java 程序源代码, 存盘 (文件后缀为 .java); (2) 使用 javac 指令编译源代码文件, 生成后缀为 .class 的字节码文件。字节码是处于二进制码和文本字符之间的混合码, JVM (Java Virtual Machine) 只识别字节码; (3) 使用 Java 指令启动 JVM 来运行 Java 字节码, 如图 1.1 所示。

不同的操作系统有对应的 JVM，所以同样的.class 可以在不同的操作系统上运行，这就是 Java 能够 “Write once, run anywhere” 的原因，如图 1.2 所示。

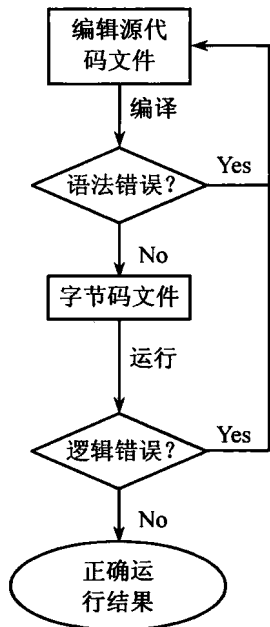


图 1.1 Java 程序开发流程

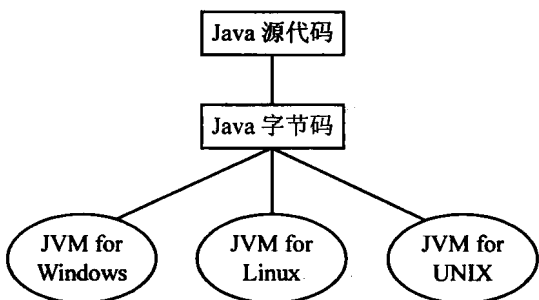


图 1.2 Java 跨平台原理

1.3.1 HelloWorld 案例

首先，建立自己的工作目录，这是一个好习惯，以后自己的程序都放在自己的工作目录中；然后，在自己的工作目录下，使用文本编辑器建立 java 源代码文件（保存文件名为 Test.java），内容如下：

```
1 class HelloWorld {
2     public static void main (String [] args) {
3         System.out.println("Hello World !");
4     }
5 }
```

然后在命令窗口中输入命令：javac Test.java，进行编译处理。若出现错误，请检查源代码文件排错，否则键入命令：java HelloWorld，运行程序。若程序运行正常，则会在显示器屏幕上显示出字符串 “Hello World !”。

1.3.2 案例分析

Java 程序源代码区分英文字母的大小写，这点在编程过程中务必注意。Java 的程序代码在格式上比较自由，但为了程序阅读和交流的方便，格式上应遵循 Java 编码规范（参考附录 A）。

所有的 Java 源代码文件必须以.java 作为扩展名，否则编译不通过。一般地，一个源代码文件至多包含一个 public 类，若在源代码文件中定义了 public 类，则源代码文件的文件名必须

与 `public` 类的类名相同(包括英文字母的大小写也要相同);若在源代码文件中没有定义 `public` 类,则源代码文件的文件名任意。例如,1.3.1 节中的程序中没有定义 `public` 类,故源代码文件名可以任意,程序中为 `Test.java`,若 `HelloWorld` 类的定义如下:

```
public class HelloWorld {
    //...
}
```

则源代码文件名只能是 `HelloWorld.java`。

程序源代码编译完成后,接下来的工作就是编译源代码文件,采用命令:

```
javac javasourcefile.java
```

其中 `javasourcefile.java` 在这里应该是 `HelloWorld.java`。

编译过程中若编译器检查到语法错误,则会中断编译并报告错误,这样就需要不断调试排错。若编译通过,则执行命令: `dir`, 并发现新生成了一个 `HelloWorld.class` 文件, `.class` 文件是字节码文件,是介于纯文本文件和二进制文件之间的一种文件。一般地,一个源代码文件中允许定义任意多个类,在编译阶段,编译器会为每个类都生成各自的 `.class` 文件,在 `.class` 文件中存放着可被执行的类的定义。

接下来执行刚编译完毕的 Java 程序,采用命令:

```
java className
```

其中 `className` 是含有 `public static void main (String [] args)` 方法(也叫函数,在 Java 中习惯叫方法)的那个类,在这里是 `HelloWorld`。注意 `className` 是类名,不是含有该类名的 `.class` 文件,例如: `java className.class` 就是错的。

运行该类时, `JDK` 会根据环境变量 `classpath` 的值在指定的目录中寻找 `className.class` 文件,找到后从 `className.class` 文件中提取出 `className` 类,然后再运行;若 `JDK` 寻找不到 `className.class` 文件,就抛出异常信息 `ClassNotFoundException`。

最后谈谈 `main` 方法,任何 Java 应用程序都必须有 `main` 方法,这个方法是程序开始运行的入口,其方法头如下:

```
public static void main(String[]args)
```

其中 `public` 和 `static` 不分先后顺序。若在执行 Java 程序时,在命令行上传递了一个或多个参数,则 `JVM` 会把这些参数封装到 `main` 方法的形式参数 `args` 中, `args` 数组的索引顺序与命令行参数的先后顺序一致,例如:

```
1 class TestMain {
2     public static void main (String [] args) {
3         for(int i=0;i<args.length;i++){
4             System.out.print(args[i]+" ");
5         }
6     }
7 }
```

在命令行上输入: `java TestMain A B C D`

输出结果为: `A B C D`

在这里只是说明了 Java 编程的基本操作命令和过程,关于其中的道理和更深入的内容在后续章节中我们再一一讲解。