

软件技术系列丛书



普通高等教育“十三五”应用型人才培养规划教材

J2SE

快速入门经典

J2SE KUAISU RUMEN JINGDIAN



主 编 / 梅青平 张 望 刘姗姗

层次递进，任务驱动
案例经典，图文并茂



西南交通大学出版社



J2SE

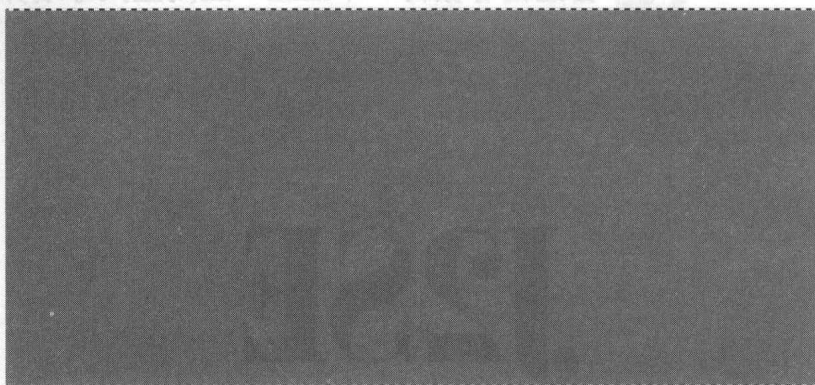


快速入门经典

J2SE KUAISU RUMEN JINGDIAN

主 编 / 梅青平 张 望 刘姗姗
副主编 / 黄吉兰

材料设备类人才应用“十三五”应用型人才规划教材



普通高等教育“十三五”应用型人才规划教材

J2SE 快速入门经典

主编 梅青平 张望 刘珊珊

责任编辑	穆丰
封面设计	墨创文化
出版发行	西南交通大学出版社 (四川省成都市二环路北一段111号 西南交通大学创新大厦21楼)
发行部电话	028-87600564 028-87600533
邮政编码	610031
网址	http://www.xnjdcbs.com
印刷	成都中铁二局永经堂印务有限责任公司
成品尺寸	185 mm × 260 mm
印张	15.75
字数	392 千
版次	2016 年 7 月第 1 版
印次	2016 年 7 月第 1 次
书号	ISBN 978-7-5643-4706-2
定价	35.00 元

课件咨询电话: 028-87600533

图书如有印装质量问题 本社负责退换

版权所有 盗版必究 举报电话: 028-87600562

前言

在最受欢迎的程序设计语言排行榜上，Java 语言已经连续数年位列榜首。一次编写，随处运行，这是一种很有效率的编程方式。跨平台、完全面向对象、既适于单机编程也适合于 Internet 编程等特点，给 Java 语言注入了强大的生命力。而 Java 语言也取得了举世瞩目、全球公认的地位。尽管 Java 很优秀，但是对于第一次接触编程的人来说并不容易学习，尤其是入门，万事开头难，编者讲授多年的 Java 课程，接触到了各种学生，对此深有体会。本书将为您打开一扇通往 Java 世界的大门，使您少走弯路，快速入门，打好坚实的基础。

鲁迅先生曾说过，治学先治史。因此，本书在第 1 章详细介绍了 Java 语言的发展史及语言特点。在介绍第 1 章节之后的第 2、3 章涉及“面向过程的程序设计”内容，是对基本编码能力的培养，先介绍如何使用 Java 语言表示信息以及如何使用 Java 语言处理信息，然后介绍了如何使用选择结构和循环结构。第 4 章“面向对象基础”是对学好 Java 语言乃至其他面向对象编程语言都至关重要的理论篇，首先介绍了如何编写类和如何创建对象，然后进一步介绍了 Java 面向对象的一些高级特性，包括如何实现继承、final 修饰符和 abstract 修饰符的使用、强制类型转换、多态性、Object 类和 Class 类的使用以及内部类的使用等。建议读者学习该篇时要缓进而踏实，精学多悟，可辅以上机实训加强对概念的理解。第 5 章介绍了数组、String 相关类和一些常用的工具类。第 6 章介绍了 Java 异常处理机制。第 7 章介绍了 Java 容器，它是对数组功能的补充。第 8 章为 I/O，介绍了 Java 字符流和字节流以及文件操作。第 9 章讨论了线程和并发处理。第 10 章介绍了基于 TCP 和 UDP 的网络编程。第 11 章介绍了图形用户界面编程，包括 Java 更加高级的功能，如事件响应模型等。Java GUI 编程是重要应用篇，通过该篇的学习，读者可设计出视窗风格的应用程序。本篇的学习策略是实践、再实践，从小的示例到较大的项目设计。由于几乎所有的应用都离不开

对数据库的操作，因此，本书在最后一章讨论了如何使用 JDBC 操作数据库。在本教程最后涉及两个实训项目，建议读者在学习完所有章节后，独立完成实训项目的所有功能，将有助于融会贯通所学知识。

本书在编写过程中得到了张文科老师以及重庆城市管理职业学院 2014 级软件专业许多同学的支持和鼓励,在此表示衷心感谢,由于你们的支持才让此书能尽早面世。同时感谢所有在出版过程中给予帮助的人,谢谢你们,让本书顺利付梓。

由于水平有限，缺点和欠妥之处难免，恳请读者帮助指正。

E-mail: leslie mei@163.com

编者
2016年4月

II

目 录

第 1 章 Java 概述	1
1.1 Java 语言历史	1
1.2 什么是 Java	3
1.3 Java 语言特性	3
1.4 Java 环境	5
1.5 Eclipse 简介	10
本章小结	10
习 题	10
第 2 章 程序、数据、变量和计算	12
2.1 标识符、关键字	12
2.2 Java 数据类型	13
2.3 变 量	18
2.4 运算符	21
2.5 表达式	26
本章小结	27
习 题	27
第 3 章 语 句	31
3.1 决 策	31
3.2 循 环	37
本章小结	43

习 题	43
第 4 章 面向对象基础	45
4.1 类和对象	45
4.2 static 关键字	58
4.3 this 关键字	61
4.4 包	63
4.5 访问权限	65
4.6 类的继承	70
4.7 super 关键字	73
4.8 Object 类常用方法	76
4.9 final 类、final 方法	77
4.10 对象的上转型对象	77
4.11 方法重写	80
4.12 类的多态	81
4.13 abstract 关键字	82
4.14 接口 (interface)	83
本章小结	86
习 题	86
第 5 章 数组与字符串	91
5.1 一维数组	91
5.2 多维数组	94
5.3 数组的常用方法	97
5.4 字符串处理	99
本章小结	107
习 题	107

第 6 章 异 常	111
6.1 异常的概念	111
6.2 异常分类	112
6.3 异常捕获和处理	113
6.4 自定义异常	116
本章小结	117
习 题	117
第 7 章 容 器	120
7.1 Collection 接口	120
7.2 Iterator 接口	123
7.3 增强的 for 循环	124
7.4 Set 接口	125
7.5 List 接口	126
7.6 Comparable 接口	127
7.7 Map 接口	128
7.8 泛 型	129
本章小结	130
习 题	130
第 8 章 I/O	132
8.1 输入/输出流概述	132
8.2 InputStream 类	133
8.3 OutputStream 类	135
8.4 Reader 类	136
8.5 Writer 类	137
8.6 缓冲流	138

8.7 转换流	140
8.8 数据流	142
8.9 打印流	143
8.10 标准输入/输出	144
8.11 对象序列化	145
8.12 文件描述	149
本章小结	152
习 题	152
第 9 章 多线程	157
9.1 线程基本概念	157
9.2 线程的创建和启动	157
9.3 线程的调度和优先级	159
9.4 线程的状态和生命周期	160
9.5 多线程的互斥与同步	164
本章小结	169
习 题	169
第 10 章 网络编程	171
10.1 计算机网络概念	171
10.2 OSI 模型	171
10.3 TCP/IP 模型	172
10.4 Java 网络编程	175
本章小结	187
习 题	187
第 11 章 图形用户界面 (GUI)	190
11.1 图形用户界面概述	190

11.2	Java Applet 基础	191
11.3	Frame 类	194
11.4	布局管理器	195
11.5	事件处理	203
11.6	常用 Swing 组件介绍	208
	本章小结	211
	习 题	211
第 12 章	与数据库通信	215
12.1	JDBC 概述	215
12.2	JDBC 的分类	215
12.3	JDBC 编程步骤	217
12.4	存储过程/函数的调用	221
12.5	事务的执行	222
	本章小结	223
	习 题	223
参考文献		241

就像很多开发新技术的秘密工程一样,工作小组夜以继日地工作到了1992年的夏天。他们能够演示新平台的一部分了,包括 Green 操作系统、Oak 程序设计语言、类库及其硬件。最初的尝试是面向一种类 PDA 设备,被命名为 Sparc7。这种设备有着鲜艳的图形界面并被称为“Desk”的智能代理来帮助用户。

同年11月,Green 计划被转化成了“FirstPerson 有限公司”,一个 SUN 公司的全资子公司,团队也被重新安排到了帕洛阿尔托。FirstPerson 团队对建造一种高度互动的设备感兴趣,当时华纳发布了一个关于电视机顶盒的征求提议书时(Request for proposal),FirstPerson

第1章 Java 概述

1.1 Java 语言历史

1.1.1 Java 语言的起源

Java 是印度尼西亚爪哇岛的英文名称，因盛产咖啡而闻名。Java 语言中的许多库类名称，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。

Java 平台和语言最开始只是 SUN 公司在 1990 年 12 月开始研究的一个内部项目。SUN 公司的一个叫做帕特里克·诺顿的工程师被自己开发的 C 语言和 C 语言编译器搞得焦头烂额，因为其中的 API 极其难用。帕特里克·诺顿决定改用 NeXT，同时他也获得了研究公司的一个叫做“Stealth 计划”项目的机会。

“Stealth 计划”后来改名为“Green 计划”，詹姆斯·高斯林和麦克·舍林丹也加入了帕特里克·诺顿的工作小组。他们和其他几个工程师一起在加利福尼亚州门罗帕克市沙丘路的一个小工作室里面研究开发新技术，瞄准下一代智能家电（如微波炉）的程序设计，因为 SUN 公司预料未来科技将在家用电器领域大显身手。团队最初考虑使用 C 语言，但是很多成员包括 SUN 的首席科学家比尔·乔伊，发现 C 和可用的 API 在某些方面存在很大问题。

由于工作小组使用的是内嵌类型平台，因此可用的资源极其有限，很多成员发现 C 太复杂以致很多开发者经常错误使用。他们发现 C 缺少垃圾回收系统，还缺乏可移植的安全性、分布程序设计和多线程功能。最后，他们决定使用一种易于移植到各种设备上的平台。

根据可用的资金，比尔·乔伊决定开发一种集 C 语言和 Mesa 语言搭成的新语言，在一份报告上，乔伊把它叫做“未来”，他提议 SUN 公司的工程师应该在 C 的基础上，开发一种面向对象的环境。最初，高斯林试图修改和扩展 C 的功能，他自己称这种新语言为 C++，但是后来放弃了。他将要创造出一种全新的语言，被他命名为“Oak”（橡树），以他的办公室外的树而命名。

就像很多开发新技术的秘密工程一样，工作小组没日没夜地工作到了 1992 年的夏天，他们能够演示新平台的一部分了，包括 Green 操作系统，Oak 程序设计语言，类库及其硬件。最初的尝试是面向一种类 PDA 设备，被命名为 Star7，这种设备有着鲜艳的图形界面和被称为“Duke”的智能代理来帮助用户。

同年 11 月，Green 计划被转化成了“FirstPerson 有限公司”，一个 SUN 公司的全资子公司，团队也被重新安排到了帕洛阿尔托。FirstPerson 团队对建造一种高度互动的设备感兴趣，当时代华纳发布了一个关于电视机顶盒的征求提议书时（Request for proposal），FirstPerson

改变了他们的目标，作为对征求意见书的响应，提出了一个机顶盒平台的提议。但是有线电视业界觉得 FirstPerson 的平台给予用户过多的控制权，因此 FirstPerson 的投标败给了 SGI，同时与 3DO 公司的另外一笔关于机顶盒的交易也没有成功。由于他们的平台不能在电视工业产生任何效益，公司再并回 SUN 公司。

1994 年六七月间，在经历了一场历时三天的头脑风暴的讨论之后，约翰·盖吉、詹姆斯·高斯林、比尔·乔伊、帕特里克·诺顿、韦恩·罗斯因和埃里克·斯库米组成的团队决定再一次改变努力的目标，这次他们决定将该技术应用于万维网。他们认为随着 Mosaic 浏览器的到来，因特网正在向同样的高度互动的远景演变，而这一远景正是他们在有线电视网中看到的。作为原型，帕特里克·诺顿写了一个小型万维网浏览器 WebRunner，后来改名为 HotJava。同年，Oak 改名为 Java，因为商标搜索显示，Oak 已被一家显卡制造商注册，因此团队找到了一个新名字。这个名字是在很多成员常去的本地咖啡馆中杜撰出来的（名字是不是首字母缩写还不清楚，很大程度上来说不是）。虽然有人声称是开发人员名字的组合：JamesGosling（詹姆斯·高斯林），ArthurVanHoff（阿瑟·凡·霍夫），AndyBechtolsheim（安迪·贝克托克姆），或者“JustAnotherVagueAcronym”（只是另外一个含糊的缩写）。还有一种比较可信的说法是这个名字是出于对咖啡的喜爱，所以以 Java 咖啡来命名。

1994 年 10 月，HotJava 和 Java 平台对公司高层进行演示。1994 年，Java1.0a 版本已经可以提供下载，但是 Java 和 HotJava 浏览器的第一次公开发布却是在 1995 年 5 月 23 日 SunWorld 大会上进行的，SUN 公司的科学指导约翰·盖吉宣告 Java 技术发布。这个发布是与网景公司的执行副总裁马克·安德森的惊人发布一起进行的，宣布网景将在其浏览器中包含对 Java 的支持。1996 年 1 月，Sun 公司成立了 Java 业务集团，专门开发 Java 技术。

1.1.2 Java 语言发展史

1995 年 5 月 23 日，Java 语言诞生。

1996 年 1 月，第一个 JDK-JDK1.0 诞生。

1996 年 4 月，10 个最主要的操作系统供应商申明将在其产品中嵌入 Java 技术。

1996 年 9 月，约 8.3 万个网页应用了 Java 技术来制作。

1997 年 2 月 18 日，JDK1.1 发布。

1997 年 4 月 2 日，JavaOne 会议召开，参与者逾一万人，创当时全球同类会议规模之纪录。

1997 年 9 月，JavaDeveloperConnection 社区成员超过十万。

1998 年 2 月，JDK1.1 被下载超过 2000000 次。

1998 年 12 月 8 日，Java2 企业平台 J2EE 发布。

1999 年 6 月，SUN 公司发布 Java 的三个版本：标准版（JavaSE）、企业版（JavaEE）和微型版（JavaME）。

2000 年 5 月 8 日，JDK1.3 发布。

2000 年 5 月 29 日，JDK1.4 发布。

2001 年 6 月 5 日，NOKIA 宣布，到 2003 年将出售 1 亿部支持 Java 的手机。

2001 年 9 月 24 日，J2EE1.3 发布。

2002 年 2 月 26 日，J2SE1.4 发布，自此 Java 的计算能力有了大幅提升。

2004年9月30日, J2SE1.5发布, 成为Java语言发展史上的又一里程碑。为了表示该版本的重要性, J2SE1.5更名为JavaSE5.0。

2005年6月, JavaOne大会召开, SUN公司公开JavaSE6。此时, Java的各种版本已经更名, 以取消其中的数字“2”: J2EE更名为JavaEE, J2SE更名为JavaSE, J2ME更名为JavaME。

2006年12月, SUN公司发布JRE6.0。

2009年4月7日, GoogleAppEngine开始支持Java。

2009年04月20日, 甲骨文74亿美元收购Sun, 取得Java的版权。

2010年11月, 由于甲骨文对于Java社区的不友善, 因此Apache扬言将退出JCP。

2011年7月28日, 甲骨文发布Java7.0的正式版。

2014年3月19日, 甲骨文公司发布Java8.0的正式版。

1.2 什么是Java

Java是一种革命性的程序设计语言, 用它编写的程序可以在不同类型的计算机上运行。能用Java语言编写applet小程序, 并嵌入网页中, 可达到智能交互效果。可以令人满意的方式与用户交互, 包括动画、游戏、交互的事务处理, 几乎无所不能。

把Java程序嵌入网页中对安全性有特别高的要求。当作为Internet用户访问嵌入了Java代码的网页时, 用户需要确信这种访问不会对自己计算机的操作有任何干扰, 也不会破坏自己系统上的数据。Java内部包含了各种措施, 以便将Java applet引起的种种不安全性减到最小。

Java对Internet和基于网络的应用程序支持不局限于applet。例如, JSP(Java Server Pages)提供了强大的建立服务器应用程序的方法。当服务器接收到请求(request)后, 会动态建立并下载HTML网页到客户端, 准确地满足用户请求。用JSP生成的网页也可以包含Java applet。

可用Java编写大型应用程序, 应用程序不加修改, 就能在任何装有支持Java的操作系统计算机上运行。也就是说, 用Java编写的应用程序可以在当今大多数计算机上运行。程序员可以用Java编写普通的应用程序, 也可编写applet应用程序。

Java提供了用来创建带有图形用户界面(GUI)的综合应用程序, 包含大量图形处理和图形编程的应用程序, 以及支持关系数据库(relation database)访问和通过网络与远程计算机通信的网络程序。现在, 用Java几乎可以有效地编写任何应用程序, 而且这些应用程序还具有完全的可移植性。

1.3 Java语言特性

(1) Java是跨平台的。它的设计一开始就是独立于机器的, 无论一个应用程序在多少种不同的计算机平台上运行, 用Java编写的应用程序仅需要一套源代码。用其他语言编写程序时, 常常需要调整源代码, 以便适应不同的计算机环境, 那些需要涉及大量图形界面的应用

更是如此。对于开发、支持和维护在若干不同硬件平台和操作系统上运行的应用程序,用 Java 语言编写可以节省相当可观的时间和资源。

(2) 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。同时,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。

(3) Java 语言是面向对象的。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)。总之,Java 语言是一个“纯的”面向对象程序设计语言。

(4) 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,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(远程方法激活)机制也是开发分布式应用的重要手段。

(5) Java 语言是健壮的。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。

(6) 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。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止替代本地的同名类、字节代码检查,并提供安全管理机制(类 SecurityManager)使 Java 应用设置安全哨兵。

(7) Java 语言是可移植的。这种可移植性来源于体系结构中立性,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 also 具有很强的可移植性,Java 编译器是用 Java 实现的,Java 的运行环境是用 ANSI C 实现的。

(8) Java 语言是解释型的。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链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:(1)使用型构为 Thread(Runnable)的构造字将一个实现了 Runnable 接口的对象包装成一个线程,(2)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,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。

1.4 Java 环境

1.4.1 安装和配置 JDK

学习 Java 需要有一个程序开发环境，目前有许多很好的 Java 程序开发环境可用。学习 Java 最好选用 Sun 公司的推出的 Java 开发工具箱 JDK。可以登录到甲骨文公司的网站（<http://www.oracle.com/technetwork/Java/Javase/downloads/index.html>），免费下载最新版本 JDK（见图 1.1）。

Java SE Development Kit 8u65

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	77.69 MB	jdk-8u65-linux-arm32-vfp-hflt.tar.gz
Linux ARM v8 Hard Float ABI	74.66 MB	jdk-8u65-linux-arm64-vfp-hflt.tar.gz
Linux x86	154.67 MB	jdk-8u65-linux-i586.rpm
Linux x86	174.84 MB	jdk-8u65-linux-i586.tar.gz
Linux x64	152.69 MB	jdk-8u65-linux-x64.rpm
Linux x64	172.86 MB	jdk-8u65-linux-x64.tar.gz
Mac OS X x64	227.14 MB	jdk-8u65-macosx-x64.dmg
Solaris SPARC 64-bit (SVR4 package)	139.71 MB	jdk-8u65-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	99.01 MB	jdk-8u65-solaris-sparcv9.tar.gz
Solaris x64 (SVR4 package)	140.22 MB	jdk-8u65-solaris-x64.tar.Z
Solaris x64	96.74 MB	jdk-8u65-solaris-x64.tar.gz
Windows x86	181.24 MB	jdk-8u65-windows-i586.exe
Windows x64	186.57 MB	jdk-8u65-windows-x64.exe

Java SE Development Kit 8u66

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 x86	154.67 MB	jdk-8u66-linux-i586.rpm
Linux x86	174.83 MB	jdk-8u66-linux-i586.tar.gz
Linux x64	152.69 MB	jdk-8u66-linux-x64.rpm
Linux x64	172.89 MB	jdk-8u66-linux-x64.tar.gz
Mac OS X x64	227.12 MB	jdk-8u66-macosx-x64.dmg
Solaris SPARC 64-bit (SVR4 package)	139.65 MB	jdk-8u66-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	99.05 MB	jdk-8u66-solaris-sparcv9.tar.gz
Solaris x64 (SVR4 package)	140 MB	jdk-8u66-solaris-x64.tar.Z
Solaris x64	96.2 MB	jdk-8u66-solaris-x64.tar.gz
Windows x86	181.33 MB	jdk-8u66-windows-i586.exe
Windows x64	186.65 MB	jdk-8u66-windows-x64.exe

图 1.1 甲骨文 JDK 下载页面

从网上下载的 JDK 是一个可执行文件，将其安装在 C 盘下（默认安装即可）。安装完成后目录结构如图 1.2 所示。

bin 目录是一些开发工具，文件夹中含有编译器（Javac.exe、解释器(Java.exe)和一些其他的可执行文件。为了方便用户使用编译器和解释器，应当将 bin 包含在 path 的设置中，如果使用 Windows 操作系统，用户可以在 ms-dos 命令行键入下列命令后回车确定即可：

```
Path C:\Program Files\Java\jdk1.8.0_25\bin
```

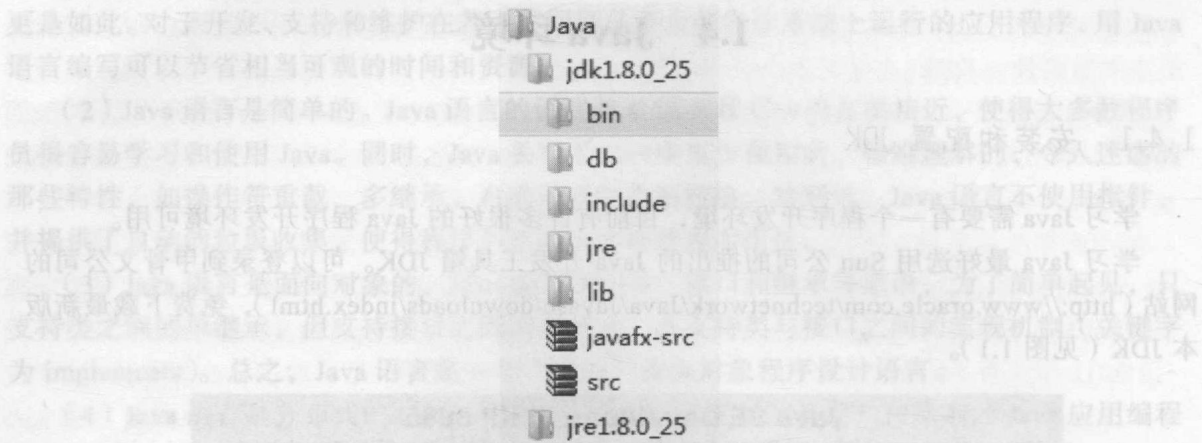


图 1.2 JDK 目录结构

也可以在系统特性中设置 Path。对于 Window 系统，用鼠标右键点击“我的电脑”，弹出菜单，然后选择“属性”，弹出“系统特性”对话框，再单击该对话框中的“高级”选项，然后点击按钮“环境变量”，添加如下的系统环境变量名 Path，变量值 C:\Program Files\Java\jdk1.8.0_25\bin。如果曾经设置过环境变量 Path，可点击该变量进行编辑操作，将需要的值加入即可（环境变量之间以“分号”分隔），如图 1.3 所示。

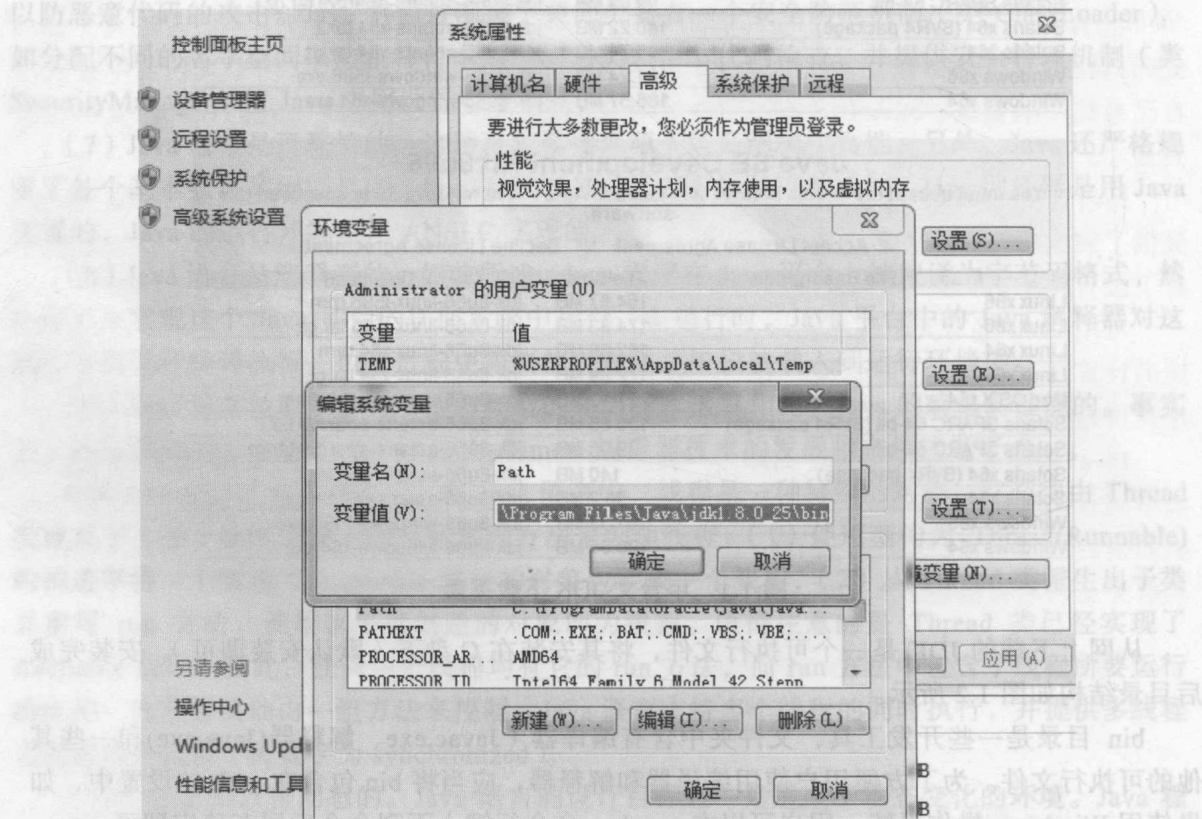


图 1.3 环境变量配置

JDK 的安装目录的 JRE 文件夹中包含着 Java 应用程序运行时所需要的 Java 类库和虚拟

机, 这些类库被包含在“jre\lib”中的压缩文件 rt.jar 中。安装 JDK 有时还需要设置环境变量 classpath 的值, 并修改为:

C:\Program Files\Java\jre1.8.0_25\lib\rt.jar;

1.4.2 一个 Java 程序的开发过程

Java 程序的开发过程如图 1.4 所示。字节码文件是与平台无关的二进制码, 执行时由解释器解释成本地机器码, 解释一句, 执行一句。

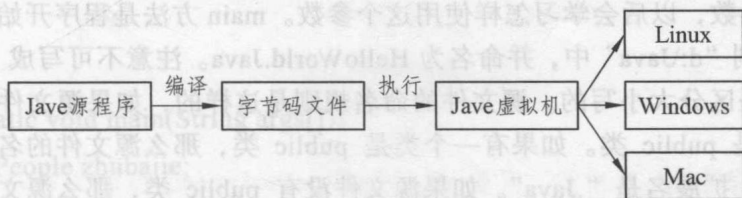


图 1.4 开发 Java 程序流程图

- (1) 编写源文件使用一个文字编辑器(如 Edit 或记事本), 来编写源文件。不可使用 Word 编辑器, 因它含有不可见字符。将编好的源文件保存起来, 源文件的扩展名必须是“.Java”。
- (2) 编译 Java 源程序。使用 Java 编译器 Javac.exe 编译源文件得到字节码文件。
- (3) 运行 Java 程序。

Java 程序分为两种: Java 应用程序和 Java 小应用程序。

- (1) Java 应用程序必须通过 Java 解释器 Java.exe 来解释执行其字节码文件;
- (2) Java 小应用程序必须通过支持 Java 标准的浏览器来解释执行。

你马上就会知道怎样使用解释器和浏览器来运行程序, 普遍使用的 Netscape Navigator 和 Microsoft Explorer 都完全支持 Java。

1.4.3 第一个 Java 程序 (HelloWorld)

1. 编写源文件

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

注意: Java 源程序中语句所涉及的小括号及标点符号都是英文状态下输入的括号和标点符号, 比如"HelloWorld"中的引号必须是英文状态下的引号, 而字符串里面的符号不受汉语输入或英文输入状态的限制。

一个 Java 源程序是由若干个类组成的。如果你学过 C 语言, 你就会知道一个 C 源程序是由若干个函数组成的。上面的这个 Java 应用程序简单到只有一个类, 类的名字是由我们起的, 叫“HelloWorld”。