

21世纪高等教育计算机规划教材

COMPUTER

Java 程序设计教程

(第2版)

Java Programming

■ 刘慧琳 主编

■ 韩雪 王亮 李理 副主编

— 从零讲起，注重基础知识

— 对每个知识点都配有编程实例

— 配有综合案例，提高学生编程能力

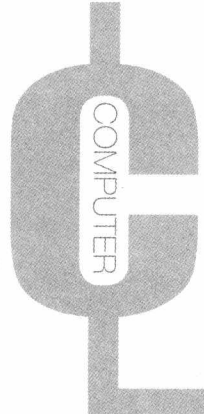


 人民邮电出版社
POSTS & TELECOM PRESS

013032969

TP312JA
1479-2

21世纪高等教育计算机规划教材

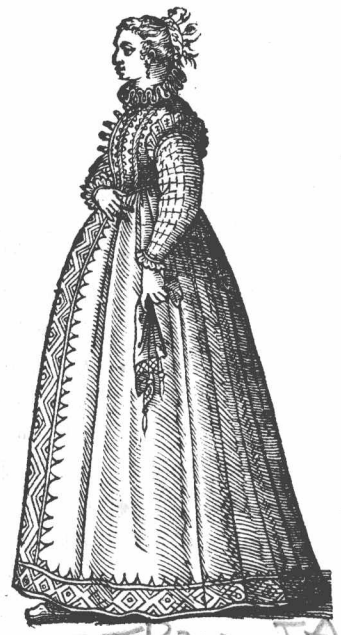


Java 程序设计教程

(第2版)

Java Programming

- 刘慧琳 主编
- 韩雪 王亮 李理 副主编



北航

C1640696

人民邮电出版社
北京

TP312JA

1479-2

0130320810 .

图书在版编目 (CIP) 数据

Java程序设计教程 / 刘慧琳主编. -- 2版. -- 北京:
人民邮电出版社, 2013.5
21世纪高等教育计算机规划教材
ISBN 978-7-115-29733-4

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(2012)第261288号

内 容 提 要

本书循序渐进地介绍了 Java 语言的语法基础、开发工具、程序控制语句、面向对象的程序设计、图形用户界面编程、Java 高级编程等内容。全书共 12 章, 每章附有习题, 供读者复习使用, 同时针对每章内容, 在人民邮电出版社网站上提供了上机指导, 供读者选用。

本书通过附加精选实例的方法, 并穿插少量的设计细节和注意事项, 按照实际开发流程对 Java 语言的程序设计进行了讲解, 简明易懂, 实用性强。本书可作为普通高等院校计算机及理工类专业 Java 相关课程的教材, 同时也可供 Java 语言的初学者参考使用。

21 世纪高等教育计算机规划教材

Java 程序设计教程 (第 2 版)

-
- ◆ 主 编 刘慧琳
副主编 韩雪 王亮 李理
责任编辑 刘 博
 - ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
大厂聚鑫印刷有限责任公司印刷
 - ◆ 开本: 787×1092 1/16
印张: 18.75 2013 年 5 月第 2 版
字数: 489 千字 2013 年 5 月河北第 1 次印刷

ISBN 978-7-115-29733-4

定价: 38.00 元

读者服务热线: (010)67170985 印装质量热线: (010)67129223
反盗版热线: (010)67171154

目 录

第 1 章 Java 语言概述1	2.3.2 注释 (comments)	24
1.1 Java 语言简介.....1	2.3.3 关键字.....27	
1.1.1 Java 发展的历程.....1	2.3.4 标识符 (identifier)	28
1.1.2 Java 的不同平台.....2	2.3.5 分隔符.....28	
1.2 Java 语言的特点.....2	2.4 常量和变量.....28	
1.3 Java 开发工具箱.....3	2.4.1 常量 (literal)	28
1.3.1 系统要求.....4	2.4.2 变量 (variable)	29
1.3.2 下载 JDK	2.5 表达式.....30	
1.3.3 安装 JDK	2.5.1 赋值.....30	
1.3.4 JDK 目录.....6	2.5.2 算术运算符.....31	
1.3.5 设置环境变量.....6	2.5.3 自增自减运算.....36	
1.4 一个简单的 Java 程序.....8	2.5.4 关系运算符.....38	
1.4.1 简单 Java 程序说明.....8	2.5.5 逻辑运算符.....40	
1.4.2 编译和运行 Java 源程序.....9	2.5.6 三元运算符.....42	
1.4.3 程序解析.....9	2.5.7 位运算.....43	
1.5 Java 开发工具概述.....9	2.5.8 移位运算.....44	
1.5.1 文本编辑器 UltraEdit 和 EditPlus.....10	2.5.9 括号及运算符间的优先级关系.....46	
1.5.2 Borland 公司的 JBuilder.....11	2.6 常用数学工具包——java.lang.Math 类.....47	
1.5.3 Eclipse 详细介绍.....11	2.6.1 数学常量.....47	
1.6 简单的 Java 小应用程序.....16	2.6.2 常用数学函数.....48	
小结.....16	小结.....49	
习题.....17	习题.....50	
上机指导.....17	上机指导.....50	
第 2 章 Java 语法基础18	第 3 章 Java 语言程序控制语句51	
2.1 Java 程序的构成.....18	3.1 选择语句.....51	
2.2 Java 基本数据类型.....19	3.1.1 if 语句.....51	
2.2.1 整数类型.....19	3.1.2 switch 语句.....54	
2.2.2 浮点类型.....21	3.2 循环语句.....59	
2.2.3 字符类型 (char)	3.2.1 while 语句.....59	
2.2.4 布尔类型 (boolean)	3.2.2 do-while 语句.....61	
2.2.5 类型之间的转换.....22	3.2.3 for 语句.....62	
2.3 标记.....24	3.3 跳转语句.....65	
2.3.1 空白分隔符.....24	3.3.1 break 语句.....65	
	3.3.2 continue 语句.....67	

3.3.3 return 语句	69	第 5 章 Java 面向对象编程进阶	99
3.4 断言	69	5.1 内部类和匿名类	99
3.4.1 断言的定义	69	5.1.1 什么是内部类	99
3.4.2 断言的开启/关闭	70	5.1.2 使用内部类	99
3.4.3 防止断言滥用	72	5.1.3 什么是匿名类	101
3.5 Java 简单程序示例	73	5.1.4 使用内部类需要注意的问题	101
小结	74	5.2 抽象类	102
习题	75	5.2.1 抽象类和抽象方法	102
上机指导	75	5.2.2 如何定义和使用抽象类	102
第 4 章 Java 面向对象编程基础	76	5.3 继承	104
4.1 什么是面向对象编程	76	5.3.1 子类	104
4.1.1 封装性	76	5.3.2 覆盖父类中的方法	106
4.1.2 继承性	77	5.3.3 this 关键字的使用	108
4.1.3 多态性	78	5.3.4 super 关键字的使用	109
4.2 Java 的类	80	5.3.5 继承的层次结构	109
4.2.1 类的基本结构	80	5.4 包	110
4.2.2 类的定义	80	5.5 接口	111
4.2.3 Java 源文件结构	81	5.5.1 接口概述	111
4.2.4 main()入口方法	81	5.5.2 成员变量在接口中的使用	112
4.3 类的成员变量与方法	82	5.5.3 方法在接口中的使用	114
4.3.1 访问修饰符: public, private,		5.5.4 接口与抽象类	121
protect 和 default	82	5.5.5 instanceof 的使用	124
4.3.2 构造方法	83	小结	128
4.3.3 定义变量	84	习题	128
4.3.4 定义成员方法	84	上机指导	129
4.3.5 参数的传递	86	第 6 章 Java 的类库	130
4.3.6 方法的返回值	87	6.1 语言基础类库	130
4.3.7 成员方法重载与过载	87	6.1.1 Object 类	130
4.3.8 static 静态成员	88	6.1.2 封装类——数据类型类	134
4.3.9 final 关键字	89	6.1.3 利用封装类进行数据转换功能	138
4.4 对象概述	90	6.2 数组	141
4.4.1 对象的声明及创建	91	6.2.1 数组的创建	141
4.4.2 调用对象的方法	92	6.2.2 用数组保存对象	143
4.4.3 对象的生存周期	94	6.2.3 数组的排序	146
4.4.4 对象的清理	94	6.2.4 二维数组	147
4.5 简单范例	94	6.3 Java 中集合框架	149
小结	97	6.3.1 集合类的结构层次	149
习题	97		
上机指导	98		

6.3.2 常用集合类的使用——List 的使用	149	8.4.3 动画效果	188
6.3.3 常用集合类的使用——Vector 的使用	151	8.5 对话框与面板	189
6.3.4 常用集合类的使用: Map 概述	153	8.5.1 顶级容器概述	189
6.3.5 常用集合类的使用: Hashtable 的使用	156	8.5.2 JFrame 主对话框	189
小结	157	8.5.3 JDialog 对话框	190
习题	158	8.5.4 JPanel 面板的使用	191
上机指导	158	8.6 常用的组件	191
第 7 章 字符串处理	159	8.6.1 按钮	192
7.1 字符串对象 String 类	159	8.6.2 文本组件	192
7.1.1 String 类常用方法	159	8.6.3 单选按钮 JRadioButton	194
7.1.2 查找指定字符串	163	8.6.4 复选框 JCheckBox	195
7.1.3 转化字符串的大小写	163	8.6.5 下拉列表 JComboBox	196
7.2 改进的 String 对象 StringBuffer 类	164	8.6.6 滑块 JSlider	197
7.2.1 StringBuffer 类——创建和使用	165	8.6.7 进度条 JProgressBar 及 Timer 组件	198
7.2.2 StringBuffer 类——插入和删除操作	167	8.6.8 菜单 JMenu	199
7.3 StringTokenizer 类	168	8.6.9 工具栏 JToolBar	200
7.4 格式化文本	170	8.7 Java 中的事件处理	200
7.4.1 格式化日期	170	8.7.1 事件处理概述	201
7.4.2 格式化数字	172	8.7.2 鼠标事件	202
7.5 正则表达式	173	8.7.3 鼠标移动事件	203
小结	177	8.7.4 对话框事件	203
习题	177	8.7.5 键盘事件	204
上机指导	177	8.8 布局管理器	204
第 8 章 Java Swing	178	8.8.1 概述	204
8.1 Java Swing 概述	178	8.8.2 边框布局 BorderLayout	204
8.2 一个简单的 Swing 程序	179	8.8.3 流动布局 FlowLayout	205
8.3 Swing 和 AWT 的对比	181	8.8.4 卡片布局 CardLayout	205
8.3.1 Swing 与 AWT 之间的关系	181	8.8.5 网格布局 GridLayout	205
8.3.2 Swing 与 AWT 的控件的混用建议	181	8.9 图形用户界面编程综合案例	206
8.4 界面图形的绘制以及文字显示与颜色控制	183	小结	209
8.4.1 图形绘制	183	习题	209
8.4.2 文本显示	186	上机指导	209
8.4.3 动画效果	188	第 9 章 Java Applet 编程	210
8.5 对话框与面板	189	9.1 Applet 基础知识	210
8.5.1 顶级容器概述	189	9.1.1 Applet 的特点	210
8.5.2 JFrame 主对话框	189	9.1.2 Applet 的安全机制	210
8.5.3 JDialog 对话框	190	9.1.3 Applet 的基本工作原理	211
8.5.4 JPanel 面板的使用	191	9.1.4 如何在网页中嵌入 Applet	211
8.6 常用的组件	191		
8.6.1 按钮	192		
8.6.2 文本组件	192		
8.6.3 单选按钮 JRadioButton	194		
8.6.4 复选框 JCheckBox	195		
8.6.5 下拉列表 JComboBox	196		
8.6.6 滑块 JSlider	197		
8.6.7 进度条 JProgressBar 及 Timer 组件	198		
8.6.8 菜单 JMenu	199		
8.6.9 工具栏 JToolBar	200		
8.7 Java 中的事件处理	200		
8.7.1 事件处理概述	201		
8.7.2 鼠标事件	202		
8.7.3 鼠标移动事件	203		
8.7.4 对话框事件	203		
8.7.5 键盘事件	204		
8.8 布局管理器	204		
8.8.1 概述	204		
8.8.2 边框布局 BorderLayout	204		
8.8.3 流动布局 FlowLayout	205		
8.8.4 卡片布局 CardLayout	205		
8.8.5 网格布局 GridLayout	205		
8.9 图形用户界面编程综合案例	206		
小结	209		
习题	209		
上机指导	209		

9.2 Applet 的生命周期和四种基本方法	213	10.5 随机访问文件	244
9.3 向 Applet 传递参数	214	10.5.1 RandomAccessFile 流的构造及内部方法	244
9.4 Applet 综合示例	216	10.5.2 示例——使用随机访问文件流来对文件进行读写	245
9.4.1 示例——使用 Applet 显示当前日期及时间	216	10.6 创建压缩文件	246
9.4.2 示例——使用 Applet 实现计算器功能	217	10.6.1 ZipFile 文件压缩类概述	247
9.4.3 示例——在 Applet 小应用程序中显示图像	221	10.6.2 实现数据文件 zip 方式的压缩方法	247
小结	223	10.6.3 将.class 文件打包成 jar 文件	250
习题	223	小结	251
上机指导	224	习题	251
第 10 章 I/O 输入输出流	225	上机指导	251
10.1 流概述	225	第 11 章 访问数据库	252
10.1.1 什么是流	225	11.1 使用 JDBC 连接数据库	252
10.1.2 输入输出类结构	225	11.1.1 JDBC 的作用及主要接口	252
10.1.3 实例: 使用输入流来读取指定目录下文件的内容	226	11.1.2 连接不同数据库	254
10.1.4 实例: 使用输入输出流来读取并显示文件内容	228	11.1.3 连接数据库的基本步骤	255
10.1.5 实例: 使用 read 方法读取文件内容	229	11.2 执行 SQL 语句	258
10.1.6 实例: 两种“程序接收用户数据输入”方法的实现	230	11.2.1 如何使用 PreparedStatement 接口执行 SQL 语句	258
10.2 文件类 (File) 的使用	231	11.2.2 如何使用 CallableStatement 接口执行 SQL 语句	260
10.2.1 获取指定文件的描述信息	232	11.3 下载及安装 MySQL 数据库	261
10.2.2 创建新文件	234	11.3.1 下载 MySQL5.5 数据库	261
10.2.3 文件改名及文件删除	235	11.3.2 安装 MySQL5.5 数据库	262
10.2.4 获取指定文件目录下的所有文件	236	11.3.3 创建数据库 test 及数据表 userinfo	266
10.3 读取文件	237	11.4 通过 JDBC 访问 MySQL 数据库	267
10.3.1 示例——读取文件并显示	237	11.4.1 查询数据	267
10.3.2 读取文件的分类及所涉及的类	239	11.4.2 添加、删除及修改数据	271
10.3.3 读取文件 Reader 类的基本方法	241	小结	273
10.4 写入文件	241	习题	273
10.4.1 示例——将输入的文本保存到文件	242	上机指导	273
10.4.2 写入文件的分类及所涉及的类	243	第 12 章 学生成绩查询系统	274
10.4.3 写入文件 Writer 类的基本方法	243	12.1 学生成绩查询系统结构	274
		12.2 学生成绩查询系统——数据库层	275
		12.3 学生成绩查询系统——程序预览	278

12.4 视图 (View) 模块——前台 JSP	
页面	279
12.4.1 学号输入页面 main.jsp	279
12.4.2 错误信息页面 errorpage.jsp	280
12.4.3 学生成绩显示页面	
markshow.jsp	281
12.5 模型 (Model) 模块——JavaBean	
和实用类	283
12.5.1 用于保存学生信息的实体	
JavaBean——Student 学生类	283
12.5.2 保存学生成绩信息的实体	
JavaBean——Course 课程类	284
12.5.3 用于连接数据库的 JavaBean——	
DataBase 数据库连接类	285
12.5.4 用于查询学生信息的 JavaBean——	
MarkDB 数据查询类	286
12.6 控制器 (Controller) 模块——Servlet	
控制器	288
12.7 学生成绩查询系统的发布	289
小结	290
习题	290
上机指导	290

第 1 章

Java 语言概述

Java 语言是由 Sun 公司于 1995 年推出的一种新的编程语言，是一种跨平台、适合于分布式计算环境的纯面向对象语言。

1.1 Java 语言简介

Java 的历史要追溯到 1991 年，Sun MicroSystem 公司的 Jame Gosling、Bill Joe 等人，为在家用消费类电子产品上进行交互式操作而开发了一个名为 Oak 的软件，但当时并没有引起人们的关注。后来发现 Oak 是一种已有计算机语言的名字，于是将其更名为 Java。



Java 是爪哇岛出产的美味咖啡，用其命名表示给每个使用 Java 进行开发的程序员一杯美味的咖啡。

1.1.1 Java 发展的历程

Java 从诞生到现在主要经历了如下过程。

(1) 1996 年初，Sun 公司发布了 Java 1.0，但是很快人们就发现它有严重的缺陷。虽然后来 Java 1.1 的发布弥补了其中大部分明显的不足，为 GUI 编程增加了新的事件处理模型，但仍然具有很大的局限性。

(2) 1998 年 Sun 公司推出 Java 1.2。该版本更加接近了“一次开发，随处运行”的承诺。后来的 1.3 及 1.4 版本扩展了标准类库，提高了系统性能，同时为 Java 语言添加了很多新的特性。在这期间随着网络的发展，Java 也在各个领域得到了广泛的应用。

从 Java 1.2 开始也称为 Java 2，并且分 Java SE、Java EE、Java ME 3 个版本。

(3) 2004 年 Java 5.0 发布，为 Java 添加了很多新的特性，如泛型、安全枚举类型、程序注解等，大大简化了 Java 的开发。



Java 5.0 版原来被命名为 1.5 版，在 2004 年的 JavaOne 会议之后，版本数字升为 5.0。

(4) 2006 年底 Java 7.0 发布，不但进一步提升了性能，还加入了对各种脚本语言嵌入式开发的支持等特性，大大提升了 Java 的开发能力和易用性。

(5) 2009 年 04 月 20 日, Oracle (甲骨文) 收购 SUN, Java 因此并归甲骨文公司, 开始了新的发展。

(6) 2011 年, Java 7 发布, 重点在模块化方面进行了改进, 增加了很多新特性。不过 Java 的核心功能没有改变。

本书使用 Java 7 进行讲解。

1.1.2 Java 的不同平台

根据不同的开发用途, 将 Java 分为如下 3 个不同的平台版本。

(1) Java SE (Java Platform, Standard Edition): Java 的标准版, 以前称为 J2SE, 主要用来进行桌面以及普通网络、数据库等程序的开发。

(2) Java EE (Java Enterprise Edition): Java 的企业版, 提供了企业级开发需要的各种技术支持, 如 RMI, JNDI, EJB 等。主要用来进行企业级应用的开发。

(3) Java ME (Java Micro Edition): Java 的微型版, 以前称为 J2ME, 提供了移动、嵌入式开发需要的各种技术支持, 如 CDC, CLDC 等。主要用来进行移动、嵌入式应用的开发。

Java SE 是 Java 平台的标准版, 它包含构成 Java 语言核心的类。如: 数据库连接、接口定义、输入/输出、网络编程。Java SE 适用于桌面系统, 提供了 CORBA 标准的 ORB 技术, 结合 Java 的 RMI 支持分布式互操作环境, 运行在 Java 虚拟机上。Java SE 是可移植性、异构性的实现环境和健壮平台, 也是实现可伸缩性、分布式异构互操作应用软件开发的标准平台。

Java EE 平台主要用于 Java 技术企业级应用, 可以使程序员迅速、快捷地开发和分布式企业级应用程序。由于 Java EE 平台作为一种可扩展的、全功能的平台, 可以将企业应用扩展到任何 Web 浏览器上。目前, 很多厂家都对 Java EE 给予支持, 并将 Java EE 技术作为大型 B2B 市场和海量交易处理的安全稳定的端到端平台。

Java EE 技术巩固了标准版中的许多优点。其最终目的是能够使企业开发者大幅缩短投放市场时间。Java EE 为灵活配置各种多层企业应用软件, 特别是 B2B、B2C 等电子商务应用, 提供了强大的服务功能。

Java ME 被称为 Java 微型版, 被使用在许多消费电子产品上, 如智能卡、手机、PDA、电视机顶盒等方面。同样, Java ME 也秉持了 Java 语言一贯的特性, 那就是跨平台和安全网络传输。Java ME 使用了一系列更小的类包, 而且 `Javax.microedition.io` 为 Java SE 包的子集。

1.2 Java 语言的特点

从 Java 诞生以来, 就凭借其在网络应用开发上独特的魅力以及“一次开发, 随处运行”的可移植性引起了人们极大的兴趣。自 1994 年以来, Internet 与万维网迅速发展, 这大大促进了以网络开发见长的 Java 的发展, 使其逐渐成为 Internet 上最受欢迎的编程平台。下面从几个不同的方面介绍 Java 的一些特点, 并且通过与 C++ 语言的对比进一步指出其在开发方面的优势。

1. 简单性

与 C++ 语法相比, Java 语法是一个“升华”版, 其剔除了 C++ 语法中许多易混淆、难理解的特性, 但其语法的设计还是尽可能地接近 C++ 语法, 以便系统更易于理解。在 Java 中, 略去了头文件、指针运算、操作符重载、多重继承等概念, 并且通过自动垃圾收集大大简化了程序员的内

存管理工作。

另外, Java 也适用于在嵌入式系统上运行, 其基本解释器以及类库的支持仅为 40KB 左右, 加上多线程的支持也只有 175KB 左右, 这是一个杰出的成就。

2. 面向对象

在相当长时间里, 面向对象已经证明了其自身的价值。Java 的面向对象特性与 C++ 旗鼓相当, 实现了模块化和信息隐藏, 类提供了对象的原型, 子类通过继承机制可以使用父类所提供的数据与方法。Java 语言的设计集中于对象及其接口上。Java 与 C++ 不同的是取消了多重继承, 取而代之的是接口概念, 使 Java 更加易用。

3. 分布式

Java 在网络方面的强大易用是其他语言无法比拟的, 可以说 Java 是面向网络的语言。通过其提供的类库可以方便地处理各种网络协议, 方便地进行传统的套接字网络开发, 例如 RMI、CORBA、Web 服务等现在流行的网络开发。

4. 健壮性

Java 在编译和运行时, 都会对程序可能出现的问题进行检查。并将出错信息报告给程序员。同时其提供垃圾收集机制来自动管理内存, 避免了程序员很多无心的错误和恶意的攻击。而 C++ 需要程序员自己编写代码管理内存, 这对大多数程序员来说并不是一项愉快的工作。对曾经花费很多时间来检查悬浮指针引起内存冲突的程序员, 一定非常喜欢 Java 的这一特性。

5. 可移植性

与 C++ 不同, Java 规范中没有“依赖具体实现”的地方。与平台无关的特性可以方便地将 Java 部署到任何不同平台的机器上。同时 Java 的类库封装了不同平台上的实现, 为其提供统一的接口, 这使得同样的类库可以在不同的平台上使用。这也就意味着用 Java 开发的应用可以“一次开发, 随处运行”。

6. 安全性

在安全性方面 Java 也是首屈一指的, 其是目前极少的语言本身具有安全性的开发平台。而 JCP 也大力支持对 Java 安全性的研究, 制定了使用规范, 实现了虚拟机安全机制和安全库, 添加了数字签名类的概念。从一开始 Java 就被设计为有防范各种病毒、袭击的能力, 举例如下:

- (1) 一切对内存的访问都必须通过对象的实例引用来实现;
- (2) 禁止破坏自己处理空间之外的内存;
- (3) 禁止运行时堆栈溢出;
- (4) 未经授权禁止读写文件。



JCP 是 Java Community Process 的缩写, 是制订 Java 标准的一个委员会, 由很多大的厂商和机构参加, 如 Sun、IBM、BEA、Oracle 等都是 JCP 的成员。

1.3 Java 开发工具箱

JDK 是 Java Development Kit 的缩写, 即 Java 开发工具箱, Java 开发工具箱是免费下载和使用的。在这一节中, 将向读者介绍如何获得开发 Java 程序所需要的环境, 以及环境的安装和设置。一切准备好之后, 1.4 小节将用最简单的 HelloWold.java 作为第一个 Java 程序, 向读者介绍 Java 程序最基本的编写、编译和运行的方式, 使读者对 Java 程序有一个初步的认识。

1.3.1 系统要求

浏览器方面, Java 支持 Solaris 8 (SPARC 和 x86) 上的 Netscape, Windows 平台上的 Internet Explorer。操作系统方面, Java 支持 Intel Architecture (32 位) 的系统, 可以用于 Windows 95、98、NT4.0 (Pack 5)、ME、2000、XP、2003 Server 等操作系统, 某些新操作系统可能仍然用的是 Beta 版本。

1.3.2 下载 JDK

Java 开发工具箱是免费下载和使用的, 读者可以到相关的网站上进行搜索, 也可以根据笔者提供的 URL 地址下载相应的 JDK。JDK 当前的最新版本为 jdk 1.7, 可以根据不同的操作系统平台, 下载相应的 JDK。本书将以基于 Windows 平台的 32 位机为例, 介绍下载的具体过程, 步骤如下。

(1) 在浏览器地址栏输入 “http://www.oracle.com/technetwork/java/javase/downloads/index.html”, 进入如图 1-1 所示的下载网页。

(2) 将滚动条向下拉, 寻找写有 “JDK7” 的一项, 单击进入如图 1-2 所示的 Java (TM) SE Development Kit 7 的下载页面。



图 1-1 Java SE 官方下载页

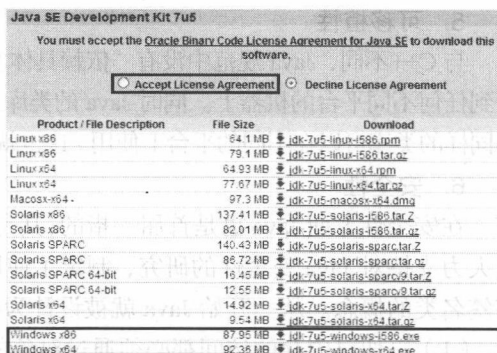


图 1-2 Java (TM) SE Development Kit 7 下载页面

(3) 将滚动条向下拉, 选中 “Accept License Agreement” 单选按钮, 页面会自动刷新, 开始下载 JDK。

(4) 在页面中寻找如图 1-3 所示的写有 “Windows Platform - Java (TM) SE Development Kit 7” 的表格, 在此表中找到第 2 列写有 “jdk-7u5-windows-i586.exe” 的一行, 并单击此行 “Windows Offline Installation, Multi-language” 的超链接进行下载。

Windows x86	87.95 MB	jdk-7u5-windows-i586.exe
Windows x64	92.36 MB	jdk-7u5-windows-x64.exe

图 1-3 Windows Platform - Java (TM) SE Development Kit 7 下载选项



基于其他操作系统平台的 JDK 下载方法与上述介绍的类似, 读者可参照上述的介绍自行操作。同时本书光盘中也附有常用操作系统平台下的 JDK 安装程序。

1.3.3 安装 JDK

下载完成之后,即可进行安装。下面介绍在 Windows 操作系统下安装 JDK 的方法。

(1) 进入存放 Java SE 7.0 软件包安装程序的目录,双击 jdk-7-windows-i586.exe,运行 Java SE 7.0 的安装程序,如图 1-4 所示。

(2) 在经过短暂的初始化工作后,进入安装界面,如图 1-5 所示,并单击【下一步】按钮。

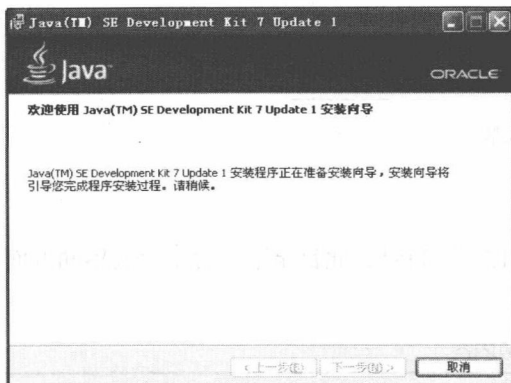


图 1-4 开始安装界面

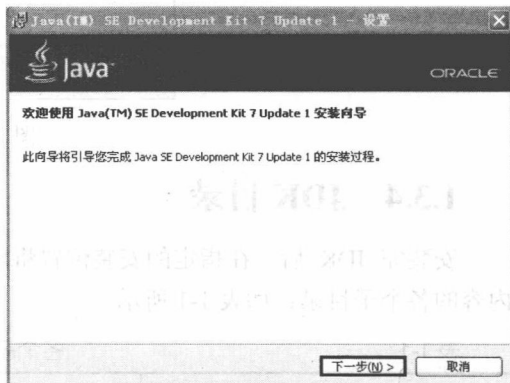


图 1-5 选择安装功能与目录

(3) 弹出如图 1-6 所示的对话框,在此处配置 Java 运行时环境 (JRE) 安装。

(4) 在此处选择安装的功能组件,单击【更改】按钮选择安装目录。完成设置后,单击【下一步】按钮继续安装,安装界面如图 1-7 所示。

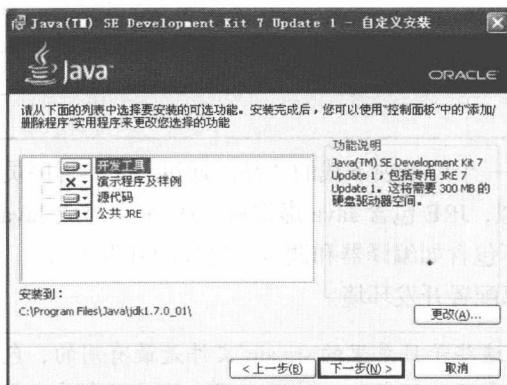


图 1-6 【安装】对话框

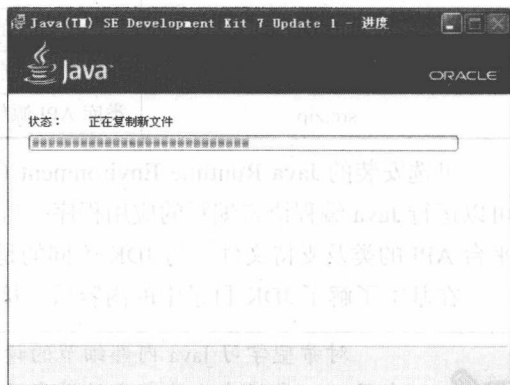


图 1-7 安装 JRE

(5) 如图 1-8 所示,安装完成,单击【完成】按钮即可。

(6) 在图 1-8 所示的对话框中选择是否显示自述文件,完成选择后,单击【完成】按钮退出。这样就完成了 JDK 的安装。



在其他操作系统 (Solaris、Linux) 中安装 Java SE 7.0 的过程与在 Windows 下的安装过程基本相同,读者可以参照 Windows 下的安装过程自行操作。

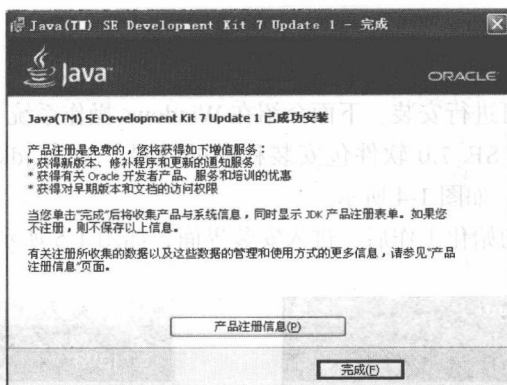


图 1-8 完成安装

1.3.4 JDK 目录

安装完 JDK 后, 在指定的安装位置将出现“jdk1.7.0”目录, 此目录下包含了存放不同方面内容的各个子目录, 如表 1-1 所示。

表 1-1 各个子目录的存放内容

“jdk1.7.0”目录下的各个子目录	描 述
bin	开发中需要用到的工具, 如编译程序 javac, 运行命令 java 等
demo	开发工具包自带的演示程序, 有兴趣的读者可以参照学习
db	使用嵌入式数据库 Derby 开发所需要的资源以及一些案例
include	用于编译本地方法的 C++头文件
jre	Java 运行时环境
lib	开发工具包的类库文件
sample	开发工具包自带的示例程序, 有兴趣的读者可以参照学习
src.zip	类库 API 源代码文件

可选安装的 Java Runtime Environment (JRE) 是一个可单独下载的产品。通过 JRE, 程序员可以运行 Java 编程语言编写的应用程序。与 JDK 相似, JRE 包含 Java 虚拟机 (JVM)、组成 Java 平台 API 的类及支持文件。与 JDK 不同的是, JRE 不包含如编译器和调试器这样的开发工具。

在基本了解了 JDK 目录中的内容后, 接下来可以配置开发环境。



说明

对希望学习 Java 内部细节的读者而言, 这些子目录里的 src.zip 文件是最有用的, 包含了 Java 类库中公共部分的源代码。如果需要深入研究 Java 的源代码, 就可以在其中寻找相应的源代码文件进行研究, 例如, 对 String 类的内部工作机制感兴趣, 就可以查看其中的 src/java/lang/String.java 文件。

1.3.5 设置环境变量

下面将介绍如何在 Windows 下设置 PATH 环境变量, 步骤如下。



提示

其他操作系统环境变量的设置这里不再赘述, 读者可查阅其他相关资料自行设置。

(1) 单击【开始】按钮，在弹出的菜单中依次选中【设置】|【控制面板】命令。系统将弹出【控制面板】对话框，如图 1-9 所示。

(2) 在【控制面板】对话框中双击【系统】图标，将弹出【系统属性】对话框，如图 1-10 所示。

(3) 在弹出的【系统属性】对话框中，单击【高级】标签，如图 1-11 所示。



图 1-9 【控制面板】对话框

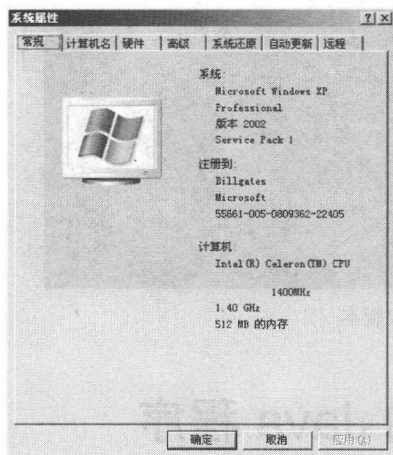


图 1-10 系统属性对话框

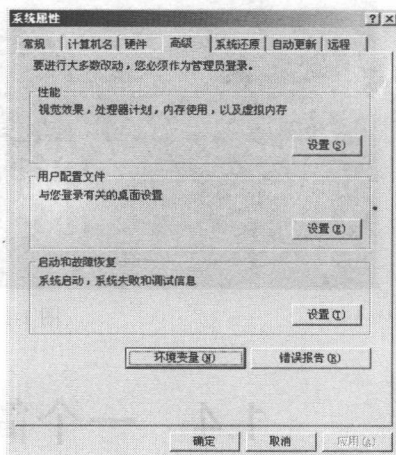


图 1-11 高级标签

(4) 在高级标签对话框中单击【环境变量】按钮，进入【环境变量】设置对话框，如图 1-12 所示。

(5) 在【环境变量】对话框中的【系统变量】列表框中找到【Path】选项并选中，然后单击【编辑】按钮进入【编辑系统变量】对话框，如图 1-13 所示。（图 1-12 选中的是用户的环境变量，不是选择的系统变量中的 Path）

(6) 在【变量值】文本框内容的最前边插入 JDK 目录下 bin 目录所在的路径，如“C:\Program Files\Java\jdk1.7.0_01\bin”，并用分号与后边原来的内容相隔。



读者可以根据计算机 JDK 的安装情况进行设置。

(7) 完成编辑后, 逐步单击【确定】按钮退出。

设置完成后, 测试环境变量是否设置成功; 打开 Windows 的【命令提示符】对话框, 输入 java-version, 然后按下 Enter 键, 如果是出现如图 1-14 所示的提示信息, 显示了读者目前安装的版本是 jdk1.7.0_01 等详细信息; 键入 java、javac 等命令同样都会出现提示信息; 则表明读者安装 JDK 成功。读者还可以使用一个简单的 Java 实例进一步测试 JDK。

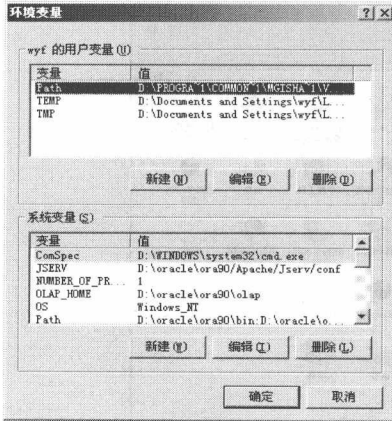


图 1-12 【环境变量】设置对话框

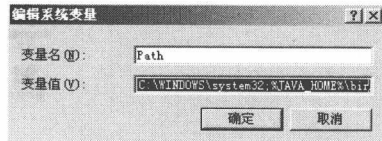


图 1-13 【编辑系统变量】对话框

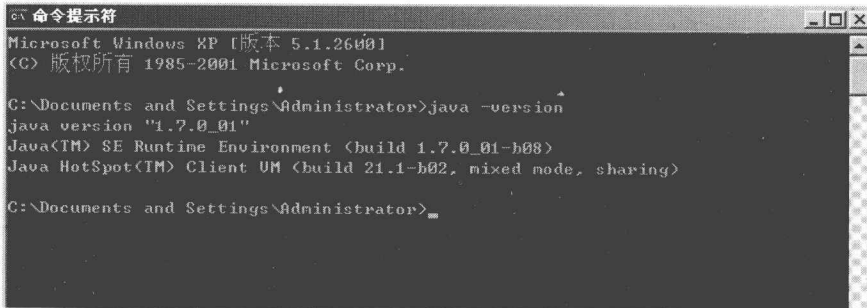


图 1-14 JDK 安装版本

1.4 一个简单的 Java 程序

安装并配置了 JDK 以后, 就可以开始编写和运行代码了。和学习其他语言一样, 先来考虑一个最简单的应用程序: HelloWorld.java, 从而对 Java 代码有一个初步的印象。

1.4.1 简单 Java 程序说明

如代码 1-1 所示, 完成了在屏幕上输出“HelloWorld!”一行文本。

【代码 1-1】 HelloWorld 代码

```

class HelloWorld
{
    public static void main(String argv[])           //类中主方法, 程序的入口点
    {
        System.out.println("Hello World!");         //在屏幕输出字符串“Hello World!”
    }
}

```