



高等学校计算机基础课程教材

Programming with Java

Java 程序设计

主 编 普运伟

副主编 王建华



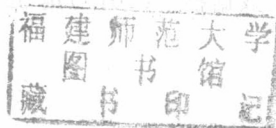
高等教育出版社
HIGHER EDUCATION PRESS

高等学校计算机基础课程教材

Java 程序设计

Java Chengxu Sheji

主 编 普运伟
副主编 王建华
参 编 田春瑾 王樱子
刘领兵 付湘琼



T1092457

1092457



高等教育出版社·北京
HIGHER EDUCATION PRESS BEIJING

内容提要

本书详细介绍了 Java 程序设计的基础知识、编程方法和主要应用方向。全书共分 12 章,第 1~4 章介绍 Java 语言的基本语法知识,内容包括 Java 程序结构、数据类型、流程控制和数组;第 5~8 章介绍 Java 面向对象编程的基本思想和方法,内容包括类和对象、实用类库、异常处理、数据流和文件操作;第 9~12 章介绍 Java 的应用编程,内容包括图形用户界面程序设计、Applet 程序设计、网络程序设计和数据库程序设计。

本书采用“问题导引式”教学模式组织每一章内容,通过问题导引、知识点介绍、专题应用、效果检测等环节使读者迅速掌握 Java 的基本编程思想和方法,提高应用 Java 技术解决实际问题的能力。本书理论与实践相结合,所有知识点都配合具体的范例进行介绍。全书所有程序均在 Java SE JDK 7.0 下编译通过。

本书可作为普通高等院校“Java 程序设计”课程的教材,也可供 Java 初学者和程序开发人员参考使用。配套出版的王建华主编的《Java 程序设计习题与上机实践》可供课后练习和上机实践课程使用。

图书在版编目(CIP)数据

Java 程序设计/普运伟主编.--北京:高等教育出版社,2013.2

ISBN 978-7-04-036953-3

I. ①J… II. ①普… 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(2013)第 025240 号

策划编辑 何新权
插图绘制 尹 莉

责任编辑 何新权
责任校对 杨凤玲

封面设计 赵 阳
责任印制 朱学忠

版式设计 于 婕

出版发行 高等教育出版社
社 址 北京市西城区德外大街 4 号
邮政编码 100120
印 刷 保定市中美凯印刷有限公司
开 本 787mm×1092mm 1/16
印 张 22.5
字 数 550 千字
购书热线 010-58581118

咨询电话 400-810-0598
网 址 <http://www.hep.edu.cn>
<http://www.hep.com.cn>
网上订购 <http://www.landaco.com>
<http://www.landaco.com.cn>
版 次 2013 年 2 月第 1 版
印 次 2013 年 2 月第 1 次印刷
定 价 41.80 元

本书如有缺页、倒页、脱页等质量问题,请到所购图书销售部门联系调换

版权所有 侵权必究

物 料 号 36953-00

前 言

Java 作为一种跨平台的语言,已经成为当前最为流行的程序设计语言之一,越来越多的高等学校将其作为程序设计教学时的首选编程语言。

本书以普通高等院校学生和 Java 初学者为对象,旨在编写一本真正适合学生和初学者学习 Java 程序设计的基础教程。全书采用“问题导引式”教学模式组织教学内容,力求通过问题导引、知识点介绍、专题应用、效果检测等环节使读者迅速掌握 Java 的基本编程思想和方法,提高应用 Java 技术解决实际问题的能力。同时,本书既注重理论方面的介绍,又强调理论与实践的结合,对于每一个重要知识点,都配合贴近实际的范例程序展开介绍,且每个程序均给出了完整代码、运行结果和程序说明。对每章的专题应用程序还详细地对其编程思路进行了分析。此外,本书的内容组织注重由浅入深、循序渐进,编写力求简洁明了、通俗易懂,以便读者理解和掌握 Java 面向对象编程的基本思路以及 Java 技术的主要应用方向和编程方法。

全书共分 12 章,内容包括 Java 开发环境及程序结构、数据类型与表达式、程序流程控制、数组、Java 面向对象程序设计、实用类库、异常处理、数据流和文件操作、图形用户界面程序设计、Applet 程序设计、网络程序设计和数据库程序设计等方面。每章后配有本章小结以及自测与思考,以便于读者对各章的主要内容进行总结和学习效果检测。配套出版的由王建华主编的《Java 程序设计习题与上机实践》可进一步用于课后练习和上机实践课程使用。

本书由普运伟任主编并编写第 1 章和第 5 章,王建华任副主编并编写第 2 章和第 6 章,田春瑾编写第 3 章和第 9 章,王樱子编写第 4 章和第 8 章,刘领兵编写第 7 章和第 12 章,付湘琼编写第 10 章和第 11 章。全书由普运伟负责最后的统稿和修改、定稿工作。

本书在编写过程中得到昆明理工大学教务处、编者所在部门广大教师的大力支持和帮助,在此表示衷心的感谢!同时,对本书所参考的文献、资料的作者表示诚挚的谢意!此外,衷心感谢高等教育出版社的编辑以及校审团队,感谢他们为本书顺利出版所付出的辛勤劳动。

Java 技术博大精深,发展极其迅速。鉴于编者水平有限,书中难免有不妥之处,恳请各位专家和读者批评指正。

编 者

2013 年 1 月

质 检

目 录

第1章 Java语言概述 1

1.1 探究Java技术 1

1.1.1 Java发展历程 1

1.1.2 Java技术平台 2

1.1.3 Java语言的特点 3

1.2 认识JVM、JRE和JDK 4

1.2.1 Java程序的运行机制 4

1.2.2 Java运行环境 5

1.2.3 Java开发环境 5

1.3 Java开发环境准备 6

1.3.1 JDK的下载、安装和配置 6

1.3.2 常用的Java开发工具 6

1.4 编写第一个Java程序 8

1.4.1 Java程序的编辑 8

1.4.2 Java程序的编译 8

1.4.3 Java程序的运行 9

1.5 Java程序的结构和规范 9

1.5.1 理解第一个Java程序 10

1.5.2 进一步认识Java程序 11

1.5.3 标识符和关键字 13

1.5.4 程序注释 13

1.5.5 对Java程序的再次说明 14

1.6 专题应用:Java程序的输入 15

本章小结 18

自测与思考 18

第2章 数据类型与表达式 20

2.1 Java的基本数据类型 20

2.2 变量与常量 22

2.2.1 变量 22

2.2.2 常量 24

2.3 基本数据类型变量的赋值 24

2.3.1 整型变量的赋值 25

2.3.2 浮点型变量的赋值 26

2.3.3 字符型变量的赋值 26

2.3.4 字符串变量的赋值 29

2.3.5 布尔型变量的赋值 29

2.3.6 基本数据类型变量的默认值 30

2.4 表达式与运算符 31

2.4.1 表达式 31

2.4.2 运算符 32

2.4.3 运算符的优先级 40

2.5 扩展表达式和类型转换 41

2.5.1 扩展表达式 41

2.5.2 表达式的数据类型转换 43

2.6 专题应用:获取数据的方法 45

2.6.1 从键盘输入数据 45

2.6.2 由计算机产生随机数 48

本章小结 49

自测与思考 50

第3章 程序流程控制 51

3.1 典型的程序结构 51

3.2 选择结构 52

3.2.1 if语句 52

3.2.2 switch语句 56

3.3 循环结构 59

3.3.1 for语句 59

3.3.2 while语句 61

3.3.3 do-while语句	63	5.2 类和对象	102
3.3.4 嵌套循环	65	5.2.1 定义类	102
3.4 控制跳转语句	66	5.2.2 成员变量	104
3.4.1 标号语句	67	5.2.3 成员方法	105
3.4.2 continue语句	67	5.2.4 成员变量和局部变量 的区别	106
3.4.3 break语句	68	5.2.5 创建、使用和销毁 对象	107
3.5 专题应用:典型流程控制 算法	69	5.2.6 方法中的参数传递	109
本章小结	72	5.3 构造方法	112
自测与思考	72	5.3.1 构造方法的定义	112
第4章 数组	75	5.3.2 对象的生成过程	114
4.1 数组的引入	75	5.3.3 this关键字	116
4.1.1 引入数组的必要性	75	5.4 类的继承	118
4.1.2 数组的概念	76	5.4.1 继承的概念	118
4.2 一维数组	76	5.4.2 Java继承的实现	119
4.2.1 一维数组的定义	76	5.4.3 访问权限修饰符	121
4.2.2 一维数组的长度	80	5.4.4 构造方法与继承	123
4.2.3 创建一维数组方法	81	5.4.5 super关键字	125
4.2.4 一维数组应用举例	82	5.4.6 Object类	125
4.3 二维数组	84	5.5 类的多态	127
4.3.1 声明二维数组变量	84	5.5.1 多态的概念	127
4.3.2 创建二维数组	85	5.5.2 方法重载	127
4.3.3 二维数组的赋值与 使用	86	5.5.3 方法覆盖	128
4.3.4 二维数组的长度	86	5.5.4 向上转型和动态绑定	130
4.3.5 非矩阵型二维数组	87	5.6 final 关键字	133
4.3.6 二维数组应用举例	89	5.6.1 终极变量	133
4.4 多维数组	92	5.6.2 终极方法	134
4.5 专题应用:数组元素的排序	93	5.6.3 终极类	135
本章小结	97	5.7 静态成员	136
自测与思考	98	5.7.1 静态变量	136
第5章 面向对象程序设计	100	5.7.2 静态方法	137
5.1 面向对象程序设计简介	100	5.8 抽象类	138
5.1.1 程序设计方法概述	100	5.8.1 抽象方法	139
5.1.2 面向对象程序设计的特点	101	5.8.2 抽象类的定义及 应用	139
		5.9 接口	141
		5.9.1 定义接口	141

5.9.2 实现接口	142	7.1.1 异常的概念	188
5.10 内部类	144	7.1.2 异常实例分析	189
5.10.1 内部类的定义及 访问	144	7.1.3 异常发生时的控制权 转移	190
5.10.2 匿名内部类	146	7.1.4 异常机制的优点	191
5.11 专题应用:多类设计	147	7.2 异常的类型	192
本章小结	151	7.2.1 异常类继承规范	192
自测与思考	153	7.2.2 三种异常类型	192
第6章 Java 实用类库	154	7.2.3 方法抛出异常的 声明	193
6.1 Java 包及核心 API	154	7.2.4 常用标准异常类	194
6.1.1 包的概念和作用	154	7.3 异常的抛出与捕获	195
6.1.2 创建包	155	7.3.1 异常的抛出: throw 语句	195
6.1.3 引用包中的类	157	7.3.2 异常的捕获	196
6.1.4 常用的 Java 类库	158	7.4 自定义异常	199
6.2 Math 类和 Random 类	160	7.5 专题应用:账户存款管理	200
6.2.1 Math 类	160	本章小结	204
6.2.2 Random 类	163	自测与思考	205
6.3 Date 类和 GregorianCalendar 类	164	第8章 Java 数据流及文件操作	206
6.3.1 Date 类	164	8.1 流及文件的基本概念	206
6.3.2 GregorianCalendar 类	165	8.1.1 流的概念	206
6.3.3 DateFormat 类	166	8.1.2 流的作用	206
6.4 String 类和 StringBuffer 类	167	8.1.3 文件操作中的流	207
6.4.1 String 类	167	8.1.4 Java 中的输入输出 流类	207
6.4.2 StringBuffer 类	172	8.2 File 类	208
6.5 集合接口与集合类	174	8.2.1 创建文件对象	208
6.5.1 集合接口与相关实 现类	174	8.2.2 文件列表	209
6.5.2 Vector 类	178	8.2.3 文件操作	211
6.5.3 泛型集合	180	8.3 字节流	214
6.6 专题应用:开发一个应用项目 的方法	183	8.3.1 抽象字节流	214
本章小结	186	8.3.2 文件字节流	216
自测与思考	187	8.4 字符流	219
第7章 异常处理	188	8.4.1 抽象字符流	219
7.1 Java 的异常机制	188	8.4.2 文件字符流	221
		8.4.3 缓冲字符流	223

8.5 过滤流	227	9.5.8 菜单	282
8.5.1 过滤流概述	227	9.6 专题应用:GUI 的设计与实现	283
8.5.2 数据流	228	本章小结	290
8.6 对象流	230	自测与思考	291
8.7 专题应用:记录式文件的读写	232	第 10 章 Applet 程序设计	292
本章小结	236	10.1 Applet 程序设计简介	292
自测与思考	237	10.1.1 编写并运行第一个 Applet 程序	292
第 9 章 图形用户界面程序设计	239	10.1.2 Applet 程序的执行流程与生命周期	294
9.1 图形用户界面简介	239	10.1.3 Applet 类	295
9.2 Swing 容器	241	10.1.4 Applet 程序的安全性	295
9.2.1 JFrame 容器	241	10.2 使用 NetBeans 创建 Applet 程序	296
9.2.2 JPanel 容器	244	10.3 HTML 与<APPLET>标记	297
9.3 布局管理器	245	10.3.1 HTML	297
9.3.1 FlowLayout 布局管理器	246	10.3.2 <APPLET>标记	298
9.3.2 BorderLayout 布局管理器	247	10.4 利用 Applet 程序展示多媒体	299
9.3.3 GridLayout 布局管理器	250	10.4.1 图形绘制	299
9.3.4 绝对定位	251	10.4.2 图像处理	301
9.4 Java 事件处理	253	10.4.3 声音输出	302
9.4.1 事件模型	253	10.5 专题应用:图片轮换	302
9.4.2 事件类和事件监听器	255	本章小结	304
9.4.3 事件适配器	259	自测与思考	305
9.4.4 事件监听器的实现方式	260	第 11 章 Java 网络程序设计	306
9.4.5 常见的事件处理	261	11.1 网络编程简介	306
9.5 常用 Swing 组件	265	11.1.1 计算机网络	306
9.5.1 标签	266	11.1.2 协议及 TCP/IP	307
9.5.2 按钮	267	11.1.3 客户/服务器模型	309
9.5.3 文本组件	270	11.1.4 Java 网络地址支持	310
9.5.4 单选按钮和复选框	274	11.1.5 Java 中的网络编程	311
9.5.5 列表框	277	11.2 URL 编程	311
9.5.6 组合框	280	11.2.1 URL 概述	312
9.5.7 对话框	281		

11.2.2	URL类	312	12.2	Java 数据库连接	335
11.2.3	通过 URL访问 WWW	313	12.2.1	JDBC的作用	335
11.3	Socket 编程	316	12.2.2	JDBC驱动程序	336
11.3.1	Socket概述	316	12.2.3	JDBC连接	337
11.3.2	发送和接收数据	318	12.3	数据维护与查询	339
11.3.3	TCP套接字	318	12.3.1	利用 Statement接口 执行 SQL命令	339
11.3.4	UDP套接字	320	12.3.2	利用 ResultSet访问结 果集	341
11.4	专题应用:自动回声机	323	12.3.3	SQLException	342
本章小结		327	12.4	专题应用:电话充值和余额 查询	343
自测与思考		328	本章小结		347
第 12 章	Java 数据库程序设计	329	自测与思考		347
12.1	MySQL 数据库	329	参考文献		349
12.1.1	数据库系统简介	329			
12.1.2	MySQL数据库	331			

第 1 章 Java 语言概述

Java 是目前最流行的程序设计语言之一,它具有简单、健壮、跨平台和面向对象等诸多优点。本章首先介绍 Java 语言的发展历程与语言特点,然后介绍 Java 程序的运行机制、开发环境、开发过程以及程序结构和规范,最后专题介绍 Java 程序的输入/输出方法。

1.1 探究 Java 技术

在当今的信息时代,Java 一词频频出现在网络、智能手持设备、智能家电、各种计算机应用解决方案之中。Java 究竟是一种什么样的技术,为什么具有如此大的魅力?本节通过 Java 历史、现状及其语言特点的回顾与分析,为你揭示其中答案。

1.1.1 Java 发展历程

Java 语言的诞生既充满神奇色彩,又是顺应时代发展的结果。

1990 年年末,Sun 公司启动了一个名为 Green 的内部计划项目。该项目设想将各种类型家用电子产品(如电视机、微波炉等)和消费性数字产品(如手机、PDA 等)联系起来,以形成一个智能化的分布式服务系统。从而可以使上班族在下班的途中通过手机发送信息给微波炉,为他们提前准备好美味的晚餐!但当 Green 项目组成员在实现此类系统时,却发现当时主流的编程语言 C/C++均无法满足要求,因为他们面对的是型号各异的各式芯片。于是,Green 项目负责人 James Gosling(被尊称为 Java 之父)在 C++的基础上开发了一种全新的语言,并根据办公室窗外的橡树将这种语言命名为 Oak。但后来发现 Oak 这个名字已被注册,因此项目组成员在边喝咖啡边讨论新名字的过程中,决定将这种全新的语言取名为 Java。实际上,Java 是印度尼西亚一个盛产咖啡的岛屿,中文名为爪哇。Green 项目组将这种全新的编程语言命名为 Java,其寓意不仅在于 Java 语言给世人带来的是一种咖啡般的美好感觉,或许还在于使用 Java 进行编程就如同喝咖啡般的美妙享受!

可惜的是,市场对消费电子智能化分布系统的需求并不像 Sun 公司设想的那样乐观,Green 项目处于一种市场前景惨淡、甚至面临夭折的尴尬境地。恰在这时,Internet 的兴起给了 Java 充分展现的舞台。James Gosling 意识到 Java 的结构中立特性正好适合开发运行于不同计算机平台上的各种网络软件。因此,在网络浏览器 Mosaic 的启发下,Green 项目组改变研发方向,采用 Java 编制出 HotJava 浏览器,并推出 Java Applet 技术,使 Internet 网页互动技术达到了一个前所未有的高度,并据此确立了 Java 在网络编程方面的王者地位。

1995 年 5 月 23 日,Sun 公司发布了 Java 语言和 HotJava 浏览器,并在次年 1 月正式推出 Java 开发工具包(Java Development Kit,JDK)1.0 版,这标志着一个新的网络计算时代的到来。在之后

的10多年中,Java语言随Internet的高速发展而不断演化,新的版本不断推出,其功能也越来越强大。尤其是1998年12月发布的1.2版本,更是将Java划分为J2EE、J2SE和J2ME三个版本,以分别满足企业级、桌面级和移动设备应用的需要。2004年9月,J2SE 1.5正式发布,在该版本中,Sun公司为Java增添了诸如枚举(Enum)、泛型(Generics)和注释(Annotation)等重要功能。为突出Java功能上的巨大改进,Java 1.5版本直接更名为J2SE 5.0。并从第6版开始,取消了Java各种版本中的数字“2”。现在,Java的三种版本分别被命名为Java EE、Java SE和Java ME。

2010年,Sun公司被另一家著名的IT公司Oracle收购,Java也随之开始了新的历程。2011年7月28日,Oracle公司发布Java SE 7,说明Java的更新与改进并没有停滞。实际上,Java技术已得到业界的广泛认同,成为网络应用和开源软件的前驱和代表。如今,无论是传统的桌面型应用,还是广阔的网络分布式应用,甚至是移动通信嵌入式应用,到处都有Java的身影,Java无疑已成为当今最流行和最通用的软件开发语言之一。

有关Java各主要版本的发布时间如图1-1所示。

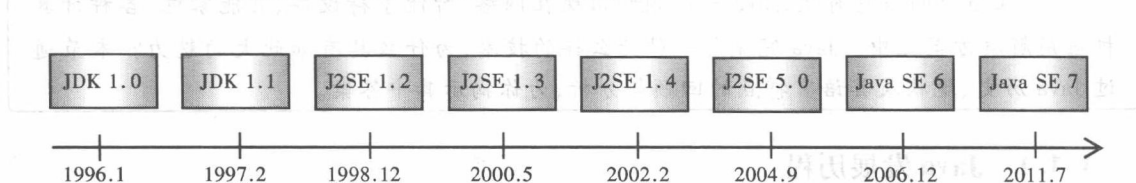


图 1-1 Java 各主要版本发布时间

1.1.2 Java 技术平台

经过10多年的发展,Java已从最初的网络互动展示技术扩展到各类应用。Java也不再是一门单纯的程序设计语言,而是包括Java语言、Java运行环境、各种API(Application Program Interface,应用程序接口)连接库、Java Server Page、Java Servlet、Java RMI、JavaBeans等众多技术于一身的业界解决问题的平台(Platform)。根据其主流应用方向,Java被划分为三个主要的技术分支。

1. Java SE

Java SE的全称为Java Standard Edition,即Java标准版,包括Java语言和一些Java标准API。可用于开发图形用户界面程序、Java Applet(小程序)、简单网络应用程序、数据库应用程序等传统桌面级应用程序。Java SE是其他各类应用平台的基础和核心,也是本书重点讨论的对象。

2. Java EE

Java EE的全称为Java Enterprise Edition,即Java企业版。该版本是在Java SE的基础上,通过引入Java Server Page、Java Servlet、Java Mail、JavaBeans等技术以构建一个多层次、分布式、以组件和Web为基础的企业级应用解决方案。

3. Java ME

Java ME的全称为Java Micro Edition,即Java微型版。该版本主要用于开发智能手机、PDA等消费性电子产品或嵌入式系统中广为使用的各类应用程序,如Java小游戏、记事程序及其他

各种控制程序等。

1.1.3 Java 语言的特点

Java 语言是整个 Java 技术框架的基础,具有简单、健壮、面向对象、可移植性好、分布式、多线程、动态等众多优良特性。相信读者在逐渐熟悉 Java 语言之后一定会对这些特性有一个清晰的认识。这里仅就其中最重要的几个关键特性进行简要介绍。

1. 简单有效

Java 语言源于 C++ 语言,因此其语法规则和 C++ 类似。但 Java 语言对 C++ 进行了简化和提高,去除了指针和多重继承等复杂概念,转而对通过对象引用和接口实现相似的功能。同时,Java 通过提供自动垃圾回收机制实现对内存的自动管理,大大降低了程序员管理内存的负担。面对相同的问题,Java 比 C++ 显得更为简洁和有效。

2. 面向对象

面向对象是当今最为主流的程序设计方法。Java 是一种完全面向对象的程序设计语言,支持面向对象的三大显著特征——封装、继承和多态。在 Java 程序中,所有信息和操作都被封装在类和对象中,类和类之间通过单重继承实现代码复用,并通过动态接口的编程风格进一步提高代码复用的效率和灵活性。因此,使用 Java 语言可以开发出非常复杂的各类应用系统。

3. 适于网络分布式计算

Java 从诞生开始就和网络密切相关。在 Java 中,内置了 TCP/IP、HTTP、FTP 等协议类库,可以很方便地通过 URL 访问网络上的对象,访问方式就像访问本地文件系统一样。同时,Java 还提供 Applet、Java Server Page 和 Servlet 等技术来丰富网页效果,实现动态页面,构建网络服务器功能。因此,Java 非常适于开发以网络为中心的各种分布式应用。

4. 跨平台特性

跨平台特性是 Java 语言最为重要的特性,这反映了 Java 语言结构中立、与具体运行平台无关的特点。Java 程序(*.java 文件)经 Java 编译器翻译成一种特殊的二进制文件,即字节码文件(*.class 文件),然后交由安装在各运行平台上的 Java 虚拟机(Java Virtual Machine, JVM)解释执行。对于不同的运行平台(如 Windows、Linux 等),所安装的 JVM 并不一样,但这些 JVM 都可以将同一个字节码文件正确翻成本机上可执行的代码。可见,正是 Java 独特的虚拟机运行机制,使得字节码文件可以独立于具体的机器设备。同时,为了保证程序中使用的数据在各种硬件平台上保持一致,Java 定义了独立于平台的基本数据类型和运算。Java 语言的跨平台特性保证了程序良好的可移植性,从而真正实现“一次编写,到处运行(Write Once, Run Anywhere)”的宏伟目标。

5. 丰富、强大的 API 类库

无论是传统的桌面级应用,还是企业级应用及移动应用,Java 语言均提供了丰富且功能强大的各种 API 类库。有了这些 API 作为支撑,Java 开发更像是“搭积木式”的编程,将已经提前准备好的各种组件进行合理组装,便可构建出功能强大的各种应用程序。

1.2 认识 JVM、JRE 和 JDK

正如前述,Java 语言的跨平台特性主要得益于 Java 虚拟机(JVM)运行机制。JVM 究竟是如何成就 Java 程序的平台无关性的?与 JVM 密切相关的 JRE 和 JDK 又是怎样的概念,它们在 Java 程序开发和运行中扮演着什么样的角色?理清 JVM、JRE 和 JDK 的区别和联系,是学习 Java 程序设计开发必须解决的首要问题。

1.2.1 Java 程序的运行机制

Java 程序是与平台无关的程序,可以实现“一次编写,到处运行”,这其中的奥秘正来源于 Java 虚拟机(JVM)的使用。图 1-2 给出了 Java 程序的执行过程,从中可以看出 JVM 的重要作用。

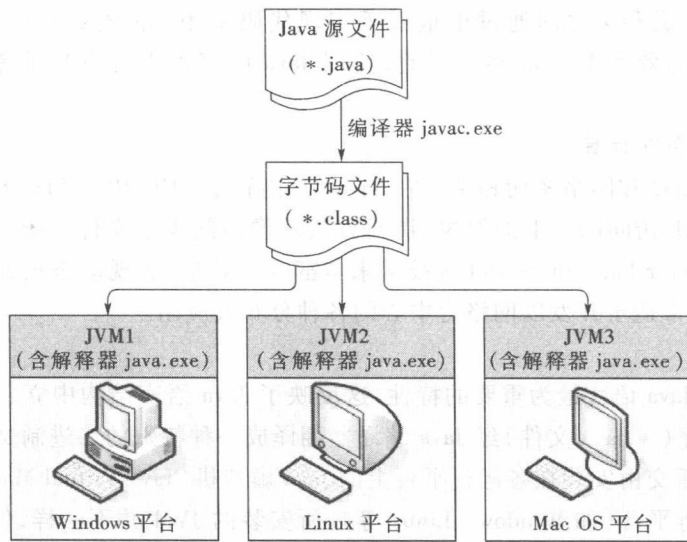


图 1-2 Java 程序的执行过程

从图 1-2 可见,Java 源程序的扩展名为.java,经过 Java 编译器(javac.exe)翻译成扩展名为.class 的字节码文件。字节码文件又称为类文件,它是一种特殊的二进制文件,独立于具体的机器平台。也就是说,字节码文件并不直接面对具体的机器平台,而是面对 Java 虚拟机,由 Java 虚拟机中的解释器(java.exe)负责解释转换成特定系统的机器代码并执行。这样,对于同一个.class 文件,便可实现在不同的平台上运行。要注意的是,每一种平台上的 JVM 和解释程序都不相同,这些安装在本地机器上的 JVM,负责为.class 文件提供统一的虚拟运行平台,并和底层实际平台进行沟通。JVM 正如同一个“当地翻译”,可以将.class 文件的内容翻译为本地计算机能够理解的机器语言。

不难看出,JVM 是可运行 Java 字节码文件的“虚拟计算机”。JVM 相当于给各种实际运行平台(Windows、Linux、Mac OS 等)包装上一层 Java 虚拟操作系统,Java 字节码文件便可直接在此虚拟平台上运行。因此,对于特定的计算机和各种消费电子产品,只要按照 JVM 规范实现了该设备相应的 JVM,便可保证 Java 字节码文件在该系统上运行。

1.2.2 Java 运行环境

Java 运行环境(Java Runtime Environment,JRE)其实就是执行 Java 程序所必备的各种要素。从 1.2.1 节的讨论可知,JVM 为 Java 程序的运行提供了一个虚拟的平台,Java 字节码文件正是在 JVM 的支持下才得以运行的。从这个意义上说,JVM 是 JRE 的基础和核心。除此之外,运行 Java 程序还需要别的什么东西呢?

其实,现代程序设计已经摆脱了“白手起家”的时代。可以设想,如果我们要设计一个类似 Microsoft Word 的应用程序,若每一个按钮和菜单选项都需要从头开始编写代码,那将是多么痛苦的事情!软件厂商通过提供各种已经设计好的程序组件和大量的各种 API 类库解决了这个问题,实现了程序代码的复用,提高了编程的效率。这也正是所有现代面向对象程序设计语言的一个共同和显著的特点。就 Java SE 而言,Sun、Oracle 以及其他 Java 阵营的厂商已经为我们准备好了数量丰富、功能强大的各种 Java SE API 类库,如基本语言和工具库(java.lang 和 java.util)、输入和输出库(java.io)、窗口程序工具库(java.awt 和 javax.swing)以及数据库连接(JDBC)等,直接调用这些类库中的组件,便可快速进行程序开发。因此,在 JVM 将.class 文件转换为本地可执行代码的过程中,为了能够理解 Java 程序中使用的各种 API,JRE 还包括 Java SE API 类库。

此外,为了能够将设计好的各种 Java 程序方便、快速地安装和应用到客户端,JRE 还包括软件部署(Deployment)技术。JRE 的组成示意如图 1-3 所示。

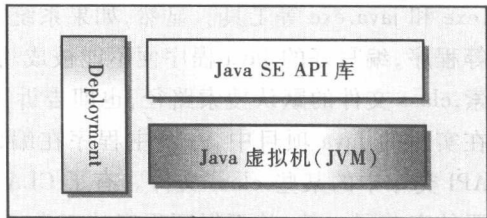


图 1-3 JRE 的组成

1.2.3 Java 开发环境

JRE 是运行 Java 程序的必备环境,而不是开发 Java 程序的环境。也就是说,如果仅仅只运行 Java 程序,亦即只想使自己的机器能够理解执行 Java 的字节码文件,则只需安装 JRE 即可,Oracle 官方网站提供了单独的 JRE 下载(<http://www.oracle.com/technetwork/java/javase/downloads/index.html>)。

如果除了运行 Java 程序之外,还需要编写和测试 Java 程序,则需要 JDK 的支持。JDK 除了包括 JRE 外,还包括许多用于开发和测试 Java 程序的实用工具。例如,用于将 Java 源代码程序翻译成字节码文件的编译器 javac.exe,用于解释执行 class 文件的解释器 java.exe,用于测试 Java Applet 运行效果的 appletviewer.exe,用于依据程序说明格式生成 HTML 帮助文档的 javadoc.exe,以及用于将程序进行打包发布的 jar.exe 等。

因此,学习 Java 程序设计的第一步便是下载和安装 JDK,并对 JDK 进行合适的配置,以使计算机能够编写、编译和运行 Java 程序。

1.3 Java 开发环境准备

② 了解了 JVM、JRE 和 JDK 的各自作用及彼此联系之后,接下来便可以搭建 Java 开发环境,为开始编写 Java 程序做好最后的准备。Java 程序的编写与测试要做好哪些准备工作呢?还有,就如 Excel 是用于专门设计电子表格的工具软件一样,有没有专门用于 Java 程序开发的工具软件?

1.3.1 JDK 的下载、安装和配置

Java 程序的编译和运行离不开 JDK 的支持。因此,开发 Java 程序的准备工作的第一步便是下载、安装和配置 JDK。这个过程本身并不复杂,详细过程可参阅配套《Java 程序设计习题与上机实践》(以下简称配套上机实践教材)的第二部分。但在这里,有必要向大家解释一下为什么要对 JDK 进行配置,这有助于帮助读者理解 Java 程序的编译和运行过程。

JDK 的配置主要包括系统环境变量 PATH 和类变量 CLASSPATH 的配置。其中,PATH 变量用于设置 Windows 系统的默认搜索路径,即告诉操作系统默认情况下到什么地方去寻找 javac.exe 和 java.exe 等工具。显然,如果系统在 PATH 指定的路径信息中找不到 javac.exe 和 java.exe 等程序,编写好的 Java 程序便不能被成功编译和运行。另外,CLASSPATH 变量用于设置 JVM 搜索.class 文件的默认搜索路径,也即告诉 Java 虚拟机默认情况下到什么地方去寻找.class 文件。在实际的 Java 项目中,一个主程序在编译或运行时总会调用到其他的.class 文件,或者使用到 API 类库中的某些.class 文件。有了 CLASSPATH 变量,JVM 便会根据该变量的设置去查找所需要的字节码文件,从而保证程序的正常编译和执行。

通常,JDK 的 PATH 和 CLASSPATH 变量应作如下配置:

(1) 在原 PATH 变量后添加“;C:\Program Files\java\jdk1.7.0_05\bin”(这里假设 JDK 被安装到 C:\Program Files\java\jdk1.7.0_05 位置,最前面的分号用于和原 PATH 路径列表进行分隔)。

(2) 创建 CLASSPATH 变量,并设置 CLASSPATH 的值为“.; C:\Program Files\java\jdk1.7.0_05\lib\dt.jar; C:\Program Files\java\jdk1.7.0_05\lib\tools.jar”,这里的“.”表示当前目录。

1.3.2 常用的 Java 开发工具

Java 源程序本质上是一种无格式的文本文件。因此,在安装和配置好 JDK 后,便可使用类似 Windows 记事本的无格式编辑器编写 Java 程序,然后使用 JDK 提供的命令行工具 javac.exe 对源程序进行编译,得到相应的.class 文件,之后再用 java.exe 对该.class 文件进行解释执行,便可得到程序的运行结果。但这样的程序开发工具组合效率极低,不易于查错,且要求用户对 DOS 操作系统的命令行方式有一定的了解。

为了提高 Java 程序的开发效率,简化程序设计和调试过程,人们开发出一些方便的工具软

件和专门的集成开发环境(Integrated Development Environment, IDE),如 UltraEdit、JCreator、NetBeans 和 Eclipse 等。下面简要介绍其中 3 种。

1. UltraEdit

从严格意义上讲,UltraEdit 并不能算是一个专门的 Java 开发工具软件。实际上,UltraEdit 是一个功能超强且灵活方便的文本编辑器,主要用于取代记事本程序,实现对各种文本文件(*.txt)、批处理文件(*.bat)、系统配置文件(*.conf)和十六进制文件的直接编辑和修改。但值得一提的是,UltraEdit 内置了对 C/C++、C#、Visual Basic、Java、JSP、PHP、HTML 和 XML 等主流设计语言的支持,可以对其源代码文件进行很好的语法着色,以清晰地区分程序代码的各个组成部分。同时,通过对 UltraEdit 进行相应的配置,便可直接在图形用户界面下编译和运行程序,获得程序的最终执行结果。正因为这些优点,许多人将 UltraEdit 作为最常用的源代码编辑器。

本书编者也建议初学者开始学习 Java 编程时使用该工具软件(如学习本书第 1~4 章时),以利于理解和掌握 Java 程序的结构,熟悉 Java 程序关键词和常用 API 类库名称的准确写法。有关如何配置该软件,使之能够适于编辑、编译和运行 Java 程序的方法,详见配套上机实践教材。

2. NetBeans

NetBeans 最早是由 Sun 公司于 2000 年创建的开放源代码计划,旨在向程序员提供一个一流的 Java 程序集成开发环境。利用 NetBeans,可以非常方便地进行 Java 程序的编辑、编译、调试和运行,还可实现清晰、直观的项目管理,以及灵活、高效的 Java 应用程序部署。NetBeans 最基本的几个特性包括:语法着色、代码完成与提示、代码折叠、代码错误提示等,这些优点极大地提高了用户开发 Java 程序的效率。

NetBeans 支持英语、葡萄牙语、日语、俄语和简体中文等多种语言,且完全由 Java 语言编写而成,确保其能可靠地运行在 Windows、Solaris、Linux 和 Macintosh OS X 等主流平台上。同时,通过插件支持,NetBeans 还支持 C/C++、PHP、Ruby 和 Ajax 等程序的开发,这是开源软件的共同显著特征。目前,NetBeans 由开源组织 netbeans.org 负责管理并提供技术交流。通过专门的社区网站(<http://netbeans.org/index.html>)可下载最新版本的 NetBeans,并得到各种最新资讯。

本书编者建议:从第 5 章开始,尝试使用 NetBeans IDE 进行 Java 程序开发。

3. Eclipse

Eclipse 是另外一个较为出色的开放源代码项目,是在 IBM 公司早期开发工具 Visual Age for Java 的基础上发展起来的非常优秀的 Java 开发工具。Eclipse 是一个基于 Java 的、跨平台的、可扩展的应用开发平台,它为编程人员提供了一流的 Java 集成开发环境。就功能而言,Eclipse 和 NetBeans 差不多,同样支持语法着色、代码完成与提示等基本特性,并提供了方便、高效的项目管理等典型功能。

Eclipse 是一个成熟的可扩展的体系结构。它本身只是一个框架和一组服务,通过安装和集成各种各样的插件,可支持 Java Web 和 Java ME 等各种主流开发和应用,并可作为其他语言的开发工具。随着 Java 技术的广泛应用,许多著名 IT 企业纷纷加入 Eclipse 架构的开发中,并发布了数量众多的支持各种应用的插件,使 Eclipse 的发展非常迅速。目前,Eclipse 由非营利软件供应商联盟 Eclipse 基金会(Eclipse Foundation)负责管理并提供技术服务,其官方网站(<http://www.eclipse.org/>)提供了最新的软件和插件下载。

1.4 编写第一个 Java 程序

❓ 了解了 Java 程序的运行机制,并安装、配置好 Java 开发环境之后,接下来便可亲自体验开发 Java 程序的整个过程。设计一个 Java 程序,一般需要经历哪些步骤呢?

Java 程序的开发一般需经历编辑、编译和运行三个步骤。本节将通过一个简单的范例,说明设计 Java 程序的一般过程和方法。有关 Java 程序的结构、语法规则和每条语句的含义,将在下一节进行详细阐释和说明。

1.4.1 Java 程序的编辑

Java 程序的编辑是指利用 Java 开发工具创建 Java 源文件,并将其保存为扩展名为 .java 的文本文件。以 UltraEdit 为例,在编辑窗口中输入以下的程序。

【程序 1-1】第一个 Java 程序。

```
01 //我的第一个 Java 程序,文件名为 HelloJavaDemo.java
02 public class HelloJavaDemo {
03     public static void main(String[] args) {
04         System.out.println("Hello,Java!");           //屏幕输出
05     }
06 }
```

程序说明:

(1) 01, 02, ..., 06 是为了方便说明程序而添加的行号,并非 Java 程序的组成部分,因此输入程序时不必输入。本书的所有范例程序均采用这种形式,以后不再说明。

(2) 输入完上述程序后,可执行菜单命令“文件→保存”或单击工具栏上的“保存”按钮将程序保存为 HelloJavaDemo.java 文件,注意文件名中字母的大小写。

(3) 输入程序时,也要特别注意每个单词的大小写,且所有标点符号均为英文标点。编辑结束后,本例的结果如图 1-4 所示。

1.4.2 Java 程序的编译

Java 程序的编译是指利用编译器将源文件编译成与具体机器平台无关的字节码文件。例如,对程序 1-1 的 HelloJavaDemo.java 进行编译,可得到字节码文件 HelloJavaDemo.class。此过程若在 DOS 命令行方式下完成(Windows 环境下,可通过“开始→程序→附件→命令行提示符”或直接运行命令 cmd.exe 打开命令行窗口),需使用如下命令:

```
javac HelloJavaDemo.java
```

若在 UltraEdit 中进行编译,可选择菜单命令“高级→编译 Java 程序”或直接单击工具栏上相应的按钮。实际上,该菜单命令和按钮本质上使用的也是上述命令,只是在准备开发环境时,已经对 UltraEdit 进行了相应的配置。详见配套上机实践教材。

若程序书写无误,编译过程中不会产生任何提示信息,并将在源文件 HelloJavaDemo.java 的