

© 高等学校计算机基础教材



Java面向对象 程序设计

主 编 赵生慧



高等教育出版社
Higher Education Press

高等学校计算机基础教材

Java 面向对象 程序设计

主编 赵生慧

高等教育出版社

内容提要

本书是在总结编者多年从事 Java 教学实践、Java 应用软件开发、试题库建设经验的基础上,从实际应用的角度出发,采用“案例驱动”的方式编写。每章以引例开头,通过 140 多道例题,介绍 Java 面向对象程序设计方法,以培养读者的基本程序设计能力。全书共 12 章,主要内容包括:Java 概述、基本数据类型和表达式、使用类和对象编程、程序控制语句、数组、Java 的继承与多态、包和接口、异常处理、图形用户界面、Applet 程序设计、Java 的输入与输出流、Java 高级编程。

为了便于教学和加强训练,每章前有内容概述、教学目标,章后有小结和丰富的习题;同时编写了配套用书《Java 面向对象程序设计实训与考试指导》(附光盘),其内容包括上机实训、主教材习题及参考答案、考试指导与试题库练习系统说明,并附两份笔试模拟试题及参考答案。试题库练习系统内嵌了经过多年研制的大型题库,具有自动评分功能,对于 Java 语言的学习和考试有很大的帮助。

本书既可作为高等学校 Java 面向对象程序设计课程教材,也可以作为计算机等级考试的参考书,对从事 Java 应用开发技术人员也具有重要的参考价值。

本书所配电子教案及相关教学资源可以从高等教育出版社高等理工教学资源网下载,网址为 <http://www.hep-st.com.cn>。使用本书的教师也可以与作者联系 (shzhao@ah.edu.cn),索取更多相关教学辅助资源。

图书在版编目 (CIP) 数据

Java 面向对象程序设计/赵生慧主编. —北京:高等教育出版社, 2007.7

ISBN 978-7-04-021476-5

I. J… II. 赵… III. Java 语言—程序设计
IV. TP312

中国版本图书馆 CIP 数据核字 (2007) 第 089709 号

策划编辑	雷顺加	责任编辑	萧 潇	封面设计	于文燕	责任绘图	朱 静
版式设计	陆瑞红	责任校对	杨雪莲	责任印制	朱学忠		

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 100011
总 机 010-58581000

经 销 蓝色畅想图书发行有限公司
印 刷 北京泽明印刷有限责任公司

开 本 787×1092 1/16
印 张 19.5
字 数 480 000

购书热线 010-58581118
免费咨询 800-810-0598
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>
畅想教育 <http://www.widedu.com>

版 次 2007 年 7 月第 1 版
印 次 2007 年 7 月第 1 次印刷
定 价 25.50 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 21476-00

前 言

为了适应信息技术的快速发展与普及,满足高等学校人才培养的需要,计算机基础课程的教学目标、教学内容及教学方法都需要不断改革,且要加强对应用能力和学习方法的培养。在这样的背景下,面向对象程序设计、数据库、Internet 应用等技术也被引入计算机基础课程中,成为教学内容的一部分。

为了加强高等学校学生计算机应用能力的培养,经安徽省高等学校计算机基础课程教学指导委员会研究并报安徽省教育厅批准,将 Java 纳入大学计算机基础课程教学体系之中,并决定自 2006 年上半年起,在全国高等学校(安徽考区)计算机水平考试中新增二级 Java 语种。Java 是一种优秀的面向对象程序设计语言,其平台无关性以及 Internet 应用的支持等多种特点使其成为当今程序设计语言的代表。更重要的是,它已经有了相当广泛的市场基础,几乎成了软件开发人员及程序员的必备技术。同时,由于 Java 与 Internet 的天然联系,使得开发基于 Java 的 Web 应用变得非常轻松。

为了配合大学计算机基础课程教学改革,安徽省高等学校计算机基础课程教学指导委员会组织编写了这本《Java 面向对象程序设计》及其配套辅导书。全书共 12 章。第 1 章介绍 Java 发展历史、基本特点及 Java 程序的组成;第 2 章讨论 Java 中的基本数据类型与表达式;第 3 章是关于类和对象的初步编程讨论;第 4 章说明程序控制结构及相应的语句;第 5 章讨论复合数据类型——数组的创建及应用;第 6 章深入讨论面向对象程序的特性——类的继承与多态;第 7 章介绍包和接口的概念及应用;第 8 章讨论异常处理的方法;第 9 章详细介绍图形用户界面及其设计方法;第 10 章着重讨论 Applet 程序设计的创建及对多媒体的支持;第 11 章讨论 Java 的输入与输出流;第 12 章介绍 Java 的高级编程技术。

本书编写人员多年从事 Java 教学、Java 应用软件开发和“Java 面向对象程序设计”试题库的研发工作,教材编写紧扣教学(考试大纲)要求,结构紧凑,教学内容设计合理,并从应用的角度出发,采用“案例驱动”的编写方式,每章以引例开头,介绍 Java 面向对象程序设计方法,以培养读者的基本程序设计能力。

本书的主要特点体现在四个方面。一是理论与实践相结合,既有基本的理论介绍,又注重技术的应用和实践。二是突出基本操作方法,并根据读者的认知规律,由浅入深,循序渐进。例如,类和对象的设计与创建是 Java 程序设计的核心,本书在不同的章节中,都将其作为重点讨论。教材前几章的引例中给出了逐步求精的方法,目的是让读者理解如何编写程序,使读者掌握这种编程思想,为后继课程的学习打下基础。三是例题与习题丰富,教材结合知识点,精选了 140 多道例题,多数例题给出设计思路和结果分析,以帮助读者理解程序,每章后提供了丰富的习题,包括选择、填空、简答、阅读程序、编程等,供读者练习与自测,以便巩固所学知识点。四是方便教与学,本书配有《Java 面向对象程序设

II 前言

计实训与考试指导》(附光盘)辅导书,其内容包括上机实训、主教材习题及参考答案、考试指导与试题库练习系统说明,并附两份笔试模拟试题及参考答案。试题库练习系统内嵌了经过多年研制而成的大型题库,具有自动评分功能,对于Java语言的学习和考试有很大的帮助。

另外,为了适应新的教学需求,提高教学质量,我们研发了整套无纸化考试系统,包括组卷、上机考试、评分、服务器设置、考试监控等模块,方便学校统一测试与阅卷。使用本书的学校可以与作者联系,获取该考试系统及相关教学资源。

本书由赵生慧主编。各章编写分工如下:第1章由孙家启编写,第2、3章由赵生慧编写,第6、9章由陈平编写,第7、11章由刘奎编写,第4、5章由何爱华编写,第8、10章由王精明编写,第12章由刘奎、王精明合编。全书由赵生慧统稿,孙家启审阅。计成超设计了配套光盘中的试题库练习系统软件。李跃民、马隽、周强、戴支祥、刘进军等多位老师参加了本书的校阅和习题编写等工作。

在本书的出版过程中,一直得到安徽省教育厅相关部门领导、安徽省高等学校计算机教学指导委员会专家的指导与支持,袁振发同志主持并参与了试题库建设工作,许多从事Java教学工作的同仁也给予了关心和帮助,高等教育出版社相关部门的领导及策划/编辑给予了大力支持、统筹策划和细心指导,在此表示衷心的感谢!

由于作者水平所限,相信书中还存在着许多不足之处,敬请各方面的专家及广大读者批评指正,以便再版时修正。如果读者有好的建议或要求,欢迎与我们联系, E-mail: shzhao@ah.edu.cn.

编 者

2007年5月

目 录

第 1 章 Java 概述.....1	
1.1 Java 的发展及其特点.....1	
1.1.1 Java 的起源.....1	
1.1.2 Java 的主要版本.....2	
1.1.3 Java 与 Internet.....2	
1.1.4 Java 的特点.....3	
1.2 Java 运行环境与应用程序举例.....4	
1.2.1 Java 程序开发环境.....4	
1.2.2 一个简单的 Java 应用程序.....6	
1.2.3 Java 应用程序结构.....6	
1.2.4 Java 程序开发过程.....8	
1.3 Java Applet 程序举例.....9	
1.3.1 Java Applet 的开发过程.....10	
1.3.2 Applet 程序组成.....11	
1.3.3 Applet 与应用程序.....12	
1.4 面向对象程序设计.....12	
1.4.1 对象和类.....13	
1.4.2 面向对象特性.....14	
1.4.3 面向对象程序设计.....15	
本章小结.....16	
习题 1.....16	
第 2 章 基本数据类型和表达式.....18	
2.1 引例.....18	
2.2 标识符.....20	
2.2.1 标识符的分类.....20	
2.2.2 关键字.....21	
2.3 Java 基本数据类型.....22	
2.3.1 数值型.....22	
2.3.2 字符型.....23	
2.3.3 布尔型.....23	
2.4 常量、变量与赋值.....23	
2.4.1 常量.....23	
2.4.2 变量.....25	
2.4.3 赋值语句.....27	
2.4.4 类型转换.....28	
2.5 运算符与表达式.....31	
2.5.1 算术运算符和算术表达式.....31	
2.5.2 关系运算符和逻辑运算符.....32	
2.5.3 其他运算符.....34	
2.5.4 运算符优先级.....35	
2.6 实例.....37	
本章小结.....40	
习题 2.....41	
第 3 章 使用类和对象编程.....43	
3.1 创建类.....43	
3.1.1 引例.....43	
3.1.2 类的定义.....45	
3.2 成员变量与方法.....47	
3.2.1 成员变量的定义.....47	
3.2.2 成员变量的初始化.....48	
3.2.3 方法的创建.....49	
3.2.4 方法的调用.....50	
3.2.5 成员变量和方法的两种形式.....51	
3.3 对象.....51	
3.3.1 创建对象.....52	
3.3.2 对象的引用.....52	
3.3.3 垃圾回收.....53	
3.4 静态变量和静态方法.....53	
3.5 再谈方法.....56	

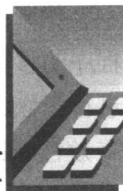
II 目录

3.5.1 重载方法	56	5.5 字符数组	97
3.5.2 构造方法	57	5.5.1 字符数组的声明和创建	97
3.6 内部类	59	5.5.2 字符串与字符数组	98
3.7 实例	61	5.6 多维数组	99
本章小结	64	5.6.1 二维数组的声明和创建以及 初始化	99
习题 3	65	5.6.2 不规则数组	100
第 4 章 程序控制语句	68	5.7 ArrayList 类	101
4.1 引例	68	5.8 对象数组	102
4.2 选择语句	70	5.9 实例	103
4.2.1 简单 if 语句	70	本章小结	105
4.2.2 if...else 语句	72	习题 5	106
4.2.3 嵌套 if 语句	73	第 6 章 Java 的继承与多态	108
4.2.4 switch 语句	74	6.1 类的继承	108
4.3 循环语句	75	6.1.1 创建子类	109
4.3.1 while 循环	75	6.1.2 构造方法的继承	111
4.3.2 do...while 循环	76	6.2 覆盖	112
4.3.3 for 循环	77	6.2.1 方法覆盖	112
4.4 跳转语句	79	6.2.2 使用 super 和 this 关键字	114
4.4.1 break 语句	79	6.2.3 使用 final 关键字	118
4.4.2 continue 语句	79	6.2.4 Object 类	118
4.4.3 return 语句	80	6.3 重载	122
4.5 递归	80	6.3.1 多态的定义与作用	122
4.6 实例	82	6.3.2 构造方法的重载	123
本章小结	86	6.4 实例	124
习题 4	86	本章小结	128
第 5 章 数组	89	习题 6	129
5.1 引例	89	第 7 章 包和接口	131
5.2 数组的声明和创建	90	7.1 包	131
5.2.1 数组的声明	90	7.1.1 创建包	132
5.2.2 数组的创建	90	7.1.2 一个包的示例	133
5.3 数组的赋值和引用	91	7.1.3 import 语句	133
5.4 数组应用	92	7.1.4 再谈作用域修饰符	135
5.4.1 测试数组	92	7.2 接口	137
5.4.2 数组作为方法的参数	93	7.2.1 定义接口	137
5.4.3 数组排序	94		
5.4.4 数组中元素的查找	96		

7.2.2 实现接口	137	9.3 AWT 窗口对象基本组件	178
7.2.3 Collection 接口	139	9.3.1 Frame 类	178
7.3 抽象类	139	9.3.2 Button 类	179
7.4 Java 常用类	140	9.3.3 Label 类	180
7.4.1 File 类	140	9.3.4 TextField 类与 TextArea 类	182
7.4.2 String 类	142	9.3.5 Checkbox 类	185
7.4.3 StringBuffer 类	146	9.3.6 CheckboxGroup 类	186
7.4.4 System 类	148	9.4 布局管理器	187
7.4.5 Math 类	149	9.4.1 FlowLayout 布局管理	188
7.4.6 Random 类	150	9.4.2 BorderLayout 布局管理	189
7.4.7 Vector 类	152	9.4.3 GridLayout 布局管理	190
7.5 实例	154	9.4.4 CardLayout 布局管理	191
本章小结	156	9.4.5 GridBagLayout 布局管理	192
习题 7	157	9.4.6 容器的嵌套	192
第 8 章 异常处理	158	9.5 事件驱动设计	193
8.1 引例	158	9.5.1 委派事件模型	193
8.2 异常和异常类	159	9.5.2 事件处理类	195
8.3 异常处理	160	9.5.3 ActionEvent 类	197
8.3.1 一个异常处理示例	160	9.5.4 TextEvent 类	198
8.3.2 声明异常	161	9.5.5 KeyEvent 类	200
8.3.3 捕获异常	162	9.5.6 MouseEvent 类	202
8.4 抛出异常	164	9.5.7 WindowEvent 类	206
8.5 自定义异常类	167	9.6 AWT 高级组件	208
8.6 实例	168	9.6.1 List 类	208
本章小结	170	9.6.2 Choice 类	210
习题 8	170	9.6.3 Scrollbar 类	212
第 9 章 图形用户界面	172	9.6.4 Dialog 类	214
9.1 GUI 设计概述	172	9.6.5 菜单的设计	216
9.1.1 GUI 组成元素分类	173	9.7 利用 AWT 绘图	220
9.1.2 GUI 的设计原则	174	9.8 Swing 组件库	223
9.1.3 GUI 的设计步骤	174	9.8.1 Swing 概述	224
9.2 AWT 窗口对象类	174	9.8.2 JTree 类	225
9.2.1 简单的窗口对象示例	174	9.8.3 JTable 类	227
9.2.2 AWT 窗口对象类	175	9.9 实例	228
9.2.3 Color 类和 Font 类	177	本章小结	231
		习题 9	231
		第 10 章 Applet 程序设计	233
		10.1 Applet 基础	233

IV 目录

10.1.1 引例	233	11.4.3 BufferedReader 类和 BufferedWriter 类	263
10.1.2 Applet 与 HTML 和 WWW	234	11.5 随机读写文件	264
10.2 Applet 的创建	235	11.6 实例	265
10.2.1 Applet 的层次结构	235	本章小结	268
10.2.2 Applet 的初始化与终止	236	习题 11	269
10.3 应用程序与 Applet 的比较	238		
10.4 Applet 对多媒体的支持	240	第 12 章 Java 高级编程	270
10.4.1 图像的显示	240	12.1 多线程程序设计	270
10.4.2 声音的加载和播放	242	12.1.1 多线程的概念	270
10.4.3 动画的生成和播放	244	12.1.2 多线程的实现	271
10.5 实例	247	12.1.3 多线程的控制与同步	275
本章小结	249	12.2 网络编程	280
习题 10	249	12.2.1 URL 编程	280
		12.2.2 Socket 编程	282
第 11 章 Java 的输入与输出流	251	12.2.3 基于无连接的数据报通信	286
11.1 引例	251	12.3 数据库编程	289
11.2 概述	252	12.3.1 JDBC 概述	289
11.3 字节流	254	12.3.2 建立与数据库的连接	290
11.3.1 InputStream 类和 OutputStream 类	254	12.3.3 访问数据库	292
11.3.2 标准输入流	255	本章小结	294
11.3.3 FileInputStream 类	256	习题 12	295
11.3.4 FileOutputStream 类	258		
11.3.5 内存的读写	260	附录	296
11.4 字符流	260	附录 1 JCreator 的使用	296
11.4.1 Reader 类和 Writer 类	260	附录 2 运算符优先级	301
11.4.2 FileReader 类和 FileWriter 类	261	参考文献及网站	302



Java 概述

随着计算机应用的普及,人们对应用程序的跨平台特性提出了越来越高的要求。同时,Internet 和 WWW 的快速发展从根本上改变了人们使用计算机的方式,也对程序的可移植性提出了新的要求。在这样的背景下,Java 应运而生。

本章介绍 Java 的基本知识,包括它的起源、发展及主要特点;讨论 Java 应用程序与 Java Applet 的基本形式及结构;对面向对象的概念进行初步的描述。通过本章内容的学习,应该能够:

- 了解 Java 的历史和基本原理
- 理解 Java 平台无关性的含义
- 掌握 Java 应用程序的结构
- 理解 Java 应用程序与 Java Applet 的联系与区别
- 掌握 Java 应用程序与 Java Applet 编辑、编译与运行的方法
- 初步了解类和对象的概念

1.1 Java 的发展及其特点

促使程序设计语言革新的因素通常有两个:程序设计技术的改革及计算环境的改变。Java 也不例外,促使 Java 迅速发展的因素来源于两个方面。一方面,C++等面向对象程序设计语言的发展为 Java 提供了技术基础。事实上,Java 大量继承了 C 及 C++的成果,并增加了体现程序设计技术发展状态的功能。另一方面,针对当时的计算环境,Java 为分布式计算环境提供了流水线程序设计功能。

1.1.1 Java 的起源

Java 是由 Sun Microsystems 的 James Gosling 所领导的开发小组设计的。最初的版本是 1991 年的 Oak,其目标是设计独立于平台且能够嵌入到不同的消费类电子产品的程序。所谓独立于平台,是指通过该语言生成的代码可以在不同的环境下、不同的 CPU 上运行,也可以在异构的操作系统环境下运行。



随着 Internet 及 WWW 的发展,人们发现 Web 也需要在不同的环境下、不同的平台上进行程序的移植,这个变化导致了 Oak 的转型及 Java 的诞生。1995 年, Sun 公司的技术人员对 Oak 进行了修改,用于开发 Internet 应用程序,并将其命名为 Java。今天, Java 技术的多功能性、有效性、跨平台性以及安全性已经使它成为网络计算领域最受欢迎的技术之一。通过 Java 可以:

- ① 在一个平台上编写软件,然后在另一个平台上运行;
- ② 创建可在 Web 浏览器中运行的程序;
- ③ 开发适用于联机论坛、存储、投票、HTML 格式处理以及其他用途的服务器端应用程序;
- ④ 将基于 Java 技术的应用程序或服务组合在一起,以生成高度自定义的应用程序或服务;
- ⑤ 为移动电话、远程处理器、低成本的消费产品以及任何具有数字核心的设备编写强大而高效的应用程序。

1.1.2 Java 的主要版本

从诞生到现在, Java 经历了许多变化, 1998 年 Sun Microsystems 发布了 JDK1.2, 即人们通常所说的 Java 2 平台。1999 年开始, Java 2 被分成 J2SE、J2EE 和 J2ME 三种平台。

1. J2SE

J2SE (Java 2 Platform Standard Edition, Java 2 标准版) 是一种开发和部署平台, 提供了编写桌面应用程序所有必需的功能。J2SE 是基于 Java 跨平台技术和强有力的安全模块而开发的, 其最新的特征和功能极大地提高了 Java 语言的伸缩性、灵活性、适用性以及可靠性。

2. J2EE

J2EE (Java 2 Platform Enterprise Edition, Java 2 企业版) 是 Sun 公司针对 Internet 环境下企业级应用推出的一种全新概念的模型, 比传统的互联网应用程序模型更有优势, 适合于开发服务器端应用程序、大型 ERP 系统等。J2EE 也是一组规范集, 其中的每一个规范规定了 Java 技术应该如何提供一种类型的功能, 为应用开发与企业应用集成定义了数目众多的应用程序接口 (Application Program Interface, API) 和多种编程模型。

3. J2ME

J2ME (Java 2 Platform Micro Edition, Java 2 微型版) 被应用于各种各样的消费电子产品上, 例如智能卡、手机、PDA、电视机顶盒等方面。J2ME 也提供了 Java 语言一贯的特性——跨平台和安全网络传输。

以上三种版本中, 作为桌面环境下应用开发工具的主要是 J2SE。关于 Java 的最新发展, 有兴趣的读者可以参考 Java 网站 <http://java.sun.com>。

1.1.3 Java 与 Internet

从某种程度而言, 没有 Internet 的发展就没有 Java 的今天。正是 Internet 的快速发展和计



算环境的变化使 Java 得以普及与流行,并走到了程序设计的前台,成为当今主要的软件开发工具之一。

另一方面,Java 也对 Internet 产生了积极而深远的影响。原因很简单,网络程序是动态的,由此产生了安全及可移植方面的问题,而 Java 解决了这些问题,因为它定义了一种新的程序形式:Java Applet。关于 Applet 的进一步说明,将在本书的第 10 章展开。

1.1.4 Java 的特点

Java 的迅速发展和广泛流行要归功于它所具有的基本特点。Sun 公司在 Java 语言白皮书中将 Java 的特点归纳为:Java 是简单的、面向对象的、分布式的、解释型的、健壮的、安全的、结构中立的、可移植的、高效的、多线程的、动态的。下面对这些概念进行一些简要的解释。

1. 简单性

Java 的简单性是指其容易使用与学习。与同为面向对象程序设计语言的 C++ 相比,Java 要简单一些。在 C++ 中有许多容易混淆的概念(例如指针),在 Java 中或被抛弃,或者以一种更容易理解的方式实现。

2. 面向对象

传统的程序设计语言是面向过程的,程序设计时关注的是程序的控制结构、数据的处理过程等。而基于对象的程序设计更加符合人们的思维模式,程序设计时关注应用程序的数据和处理数据的方法,并将数据及其处理方法封装在对象之中。相比较而言,后者的实现更加容易,效率更高,可维护性更好。

面向对象程序设计的核心是类与对象等概念,但仍然涉及面向过程程序设计中的程序结构及数组等概念。当然,重要的是面向对象程序设计思想。有一种观点认为,学习面向对象的程序设计应该具有面向过程的基础,但从实际学习情况来看,这种基础并不是必需的,至少不是非常重要的。

3. 结构中立

Java 推出之初,最能吸引程序开发人员的便是其体系结构中立(Architecture-neutral)的特点,又称为平台无关,它是指 Java 应用程序不用进行修改就可以在不同的软/硬件平台上运行。

在实际运行时,Java 源代码首先编译成字节码(Bytecode)。字节码是独立于具体的计算机的,通过相应平台上的 Java 解释器,可以将字节码翻译成目标机器上的机器语言,从而使 Java 程序在任何平台上都能运行,如图 1-1 所示。

4. 安全性

Internet 的开放性带来了许多安全问题。作为 Internet 程序设计语言,Java 用于网络与分布式环境。因为 Java 提供了多层安全机制以保护系统不受恶意程序破坏,所以当下载并运行一个 Java Applet 时,它不会损坏本地系统。

5. 可移植性

可移植性是平台无关性联系在一起的,Java 程序不必重新编译就可以在任何平台上运行,从而具有很强的可移植性。

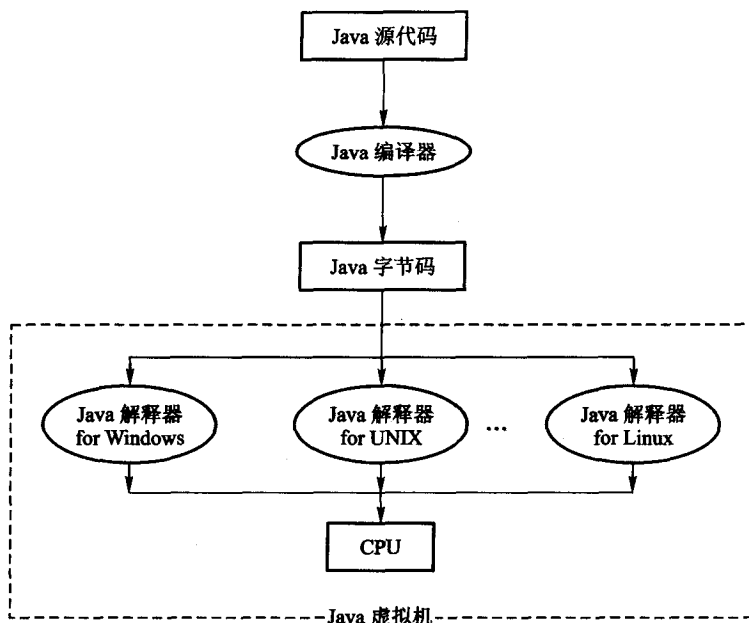


图 1-1 Java 解释器及其平台无关性

1.2 Java 运行环境与应用程序举例

Java 程序分为两类，Java 应用程序（Application）和 Java 小程序（Applet）。应用程序是独立的程序，与其他的高级语言程序相同，能够在任何具有 Java 解释器的计算机上运行。Applet 是一种特殊的 Java 程序，它可以在兼容 Java 的 Web 浏览器中直接运行，适合开发 Web 程序。

1.2.1 Java 程序开发环境

计算机语言都有严格的使用规则（又称语法规则），编写程序时必须遵守这些规则，否则就不能被计算机理解和执行。一种程序设计语言往往会有不同的开发环境，其语法规则也可能会有一些差异。在开始学习或者使用 Java 之前，要先准备好基本的环境。

1. Java 开发工具包的来源

Java 语言的标准由它的发明者 Sun 公司制定并维护，在 James Gosling 等人编著的 Java Language Specification 一书中对 Java 语言的规范进行了详细的描述。

不管是学习 Java 还是通过 Java 开发应用程序，都需要一个程序开发环境。程序开发环境一般要提供编辑、编译与运行工具。Java 程序的编辑可以使用任何一个编辑器，编译与运行则通过 Sun 公司提供的 Java 开发工具包（Java Development Toolkit, JDK）进行。JDK 是一个简单的命令行工具集，包括软件库、编译 Java 源程序的编译器、执行字节码的解释器以及其他的一些实用工具。Sun 还发布了图形用户界面的集成开发环境 NetBeans，但对于初学者来说，还是 JDK 简单、直观一些。

Sun 公司一直在不断升级 JDK，最新的版本可参看 Sun 公司的网站。JDK 软件包中包括 Java



运行时环境 (Java Runtime Environment, JRE) 以及用于编译调试 Java 程序的命令行界面的开发工具。JRE 在运行 Java 程序时是必需的, 可以单独安装。JDK 软件包可以从 Sun 公司的网站 <http://java.sun.com> 下载。

2. 安装 JDK

下载 JDK 安装软件后, 直接双击文件名, 就开始安装。首先显示在屏幕上的是许可证协议, 用户必须同意接受协议条款才能继续后面的安装流程。

在安装 JDK (以 `jdk-6-windows-i586.exe` 为例) 时, 可以选择安装的项目及安装地址, 如图 1-2 所示。安装项目包括开发工具、演示程序及样例、源代码及公共 JRE。开发工具是必须安装的; 演示程序及样例提供了一些示例, 为用户编写程序提供参考; 源代码是 API 的代码, 可以帮助用户了解 API 的编写方法及工作原理; JRE 也是运行 Java 程序必需的; 因此建议这 4 项全部安装。单击“更改”按钮修改安装路径, 将 JDK 安装到 `C:\jdk1.6` 目录下。

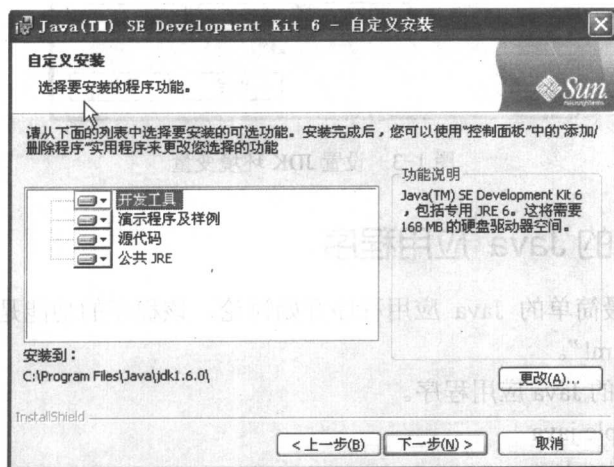


图 1-2 选择 JDK 安装组件

上述安装过程完成后, 接着安装公共 JRE, 根据屏幕提示选择即可。

3. JDK 环境变量设置

JDK 提供的是命令行用户界面, 为了保证在用户工作目录下能够正常调用 Java 编译工具, 需要进行路径设置, 即在 `path` 变量值中加入 JDK 的路径。新建变量 `java_home`, 变量值为 JDK 的安装位置。再新建变量 `classpath` 设置类路径, 使 Java 程序可以方便地调用 JDK 中的类。设置的方法是, 右击“我的电脑”, 选择“属性”, 在打开的对话框中选择“高级”选项卡, 单击“环境变量”按钮, 打开如图 1-3 所示的“环境变量”对话框。在系统变量中新建变量 `java_home`, 变量值为“`c:\jdk1.6`”。选中系统变量中的 `Path` 项, 在其变量值的末尾添加“`;%C:\%java_home%\bin`”。再新建变量 `classpath`, 将其变量值设为“`.;%JAVA_HOME%\lib\rt.jar;%JAVA_HOME%\lib\tools.jar;c:\%JAVA_HOME%\jre`”。

除了 Sun 的 JDK, 市场上还有一些 Java 开发软件包。例如 JCreator(参考附录 1)、JBuilder 和 Eclipse。IBM 公司也推出了一系列的 Java 开发工具, 主要有 WebSphere 及 Developer Kits for Java Technology 等。

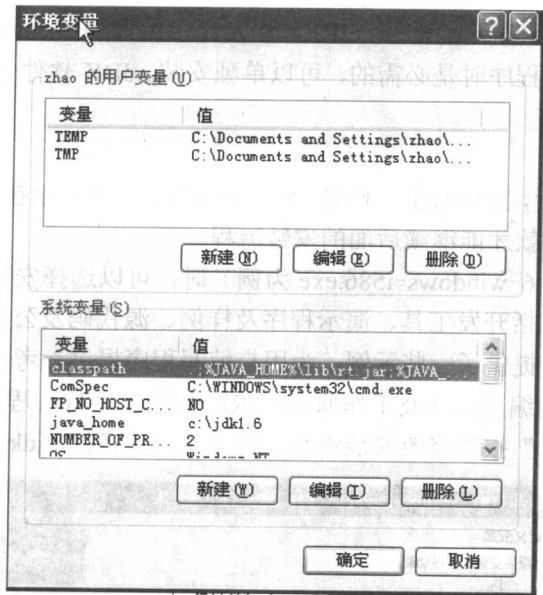


图 1-3 设置 JDK 环境变量

1.2.2 一个简单的 Java 应用程序

我们首先从一个最简单的 Java 应用程序开始讨论，该程序的功能是在显示器上显示信息“**This is a simple program!**”。

例 1.1 一个简单的 Java 应用程序。

```
//文件名 Simple.java
public class Simple
{
    public static void main(String[] args)    //输出一个字符串
    {
        System.out.println("This is a simple program!");
    }
}
```

Java 并没有对程序的书写格式提出明确的要求，但为了增加程序的可读性，设计具有良好风格的程序，建议在书写程序时采用缩进格式书写，即按照程序的层次，下一个层次比上一个层次缩进两格。

Java 应用程序的基本结构都是相似的。上面的程序比较简短，没有涉及类及对象的定义与操作，但也包含了 Java 应用程序的许多特性，下面对其进行详细分析。

1.2.3 Java 应用程序结构

一般情况下，一个 Java 应用程序由类、对象与方法等若干部分组成。下面结合例 1.1 的程序对这些组成部分进行简单的解释。



1. 类

类是 Java 的基本概念，也是面向对象程序设计的基本概念。所有 Java 程序都是由一个或者多个类组成的，一个 Java 程序至少包含一个类，子程序都包含在类定义的块内。要编写 Java 程序，必须理解并会使用类。关于类的进一步讨论将在本书的第 3 章和第 6 章展开。

例 1.1 中，Simple 就是以 public 修饰的用户定义的一个类。类定义开始于第 2 行的大括号，结束于最后一行的大括号。public 类在程序中最多只有一个，是程序的主类（含有主方法 main() 的类），Java 程序的文件名就是这里的 public 修饰的主类名。

注意：Java 是大小写敏感的，如程序中的 println 均为小写，如果写成 Println 会出错。

2. 方法与 main()方法

方法是执行一个操作而组合在一起的语句组。例 1.1 中，System.out.println() 是一个方法，其中的 println() 是已经预先定义的方法，可以直接使用。这个方法的目的是在屏幕上输出字符串，本例中是输出 “This is a simple program!”。

方法可以由用户自己定义，但每个 Java 应用程序必须有一个用户声明的 main() 方法，用来表示 Java 程序的执行入口。Java 程序依次执行 main() 方法内的每一条语句，直到方法结束。main() 方法的定义格式有严格的规定，详细内容参见本书的第 2 章。

3. 标识符与关键字

编写程序时使用的各种单词称为标识符。关键字是程序设计语言中具有特殊意义的一组标识符，它只能按照预先定义好的方式使用，不能用于其他目的。在 Simple.java 程序中，出现的标识符有 public、class、Simple、static、void、main、String、args、System、out 和 println，其中 public、class、static、void 是关键字。例如在程序的第一行中，当编译器看到 class 时，就知道其后的标识符 Simple 是一个类的名字。关于标识符及关键字的详细描述参考第 2 章。

有些特定的关键字又被称为修饰符，Java 使用它们来指定数据、方法和类的属性与使用方式。例 1.1 中的 public 和 static 就是修饰符，其他修饰符还有 private、final、abstract 和 protected 等。

4. 语句

一条语句表示一个操作，也可以表示一系列操作。例 1.1 中，“system.out.println (“This is a simple program!”)” 就是一条语句，在 Java 中，语句用分号 “;” 结束。例如，下面的代码行都是合法的 Java 语句：

```
x=3;
y=5;
x=x+y;
```

这些语句都是赋值语句。

5. 块

将程序中的一些成分组合起来，就构成了一个块 (Block)。在 Java 中，块使用 “{” 及 “}” 表示其开始与结束。每个类都有一个组合该类的属性和方法的类块，每个方法也有一个组合该方法语句的方法块。块也可以嵌套，即一个块可以放到另一个块内，图 1-4 表示了这个关系。



6. 注释

为了便于理解程序，增加程序的可读性，通常的做法是在程序中增加适当的注释。注释语句都是不执行的，编译器在编译源程序时会将其忽略。

在 Java 中，注释有两种格式。如果是单行注释，则用两个斜杠 (//) 引导；如果是多行注释，则用/*和*/将注释的内容括起来。

除了上述两种注释外，Java 还支持一种称为文档注释的特殊注释，它以/**开头，以*/结尾。主要用于描述类、数据和方法，它可以通过 JDK 的 javadoc 命令转为 HTML 文件。具体请参看 <http://java.sun.com/products/j2se/javadoc/index.html>。

注释是程序的重要组成部分，一个具有良好风格的程序必须有清晰而具体的注释。

1.2.4 Java 程序开发过程

一个 Java 程序的开发过程通常包括创建、编译、运行等几个阶段，其流程如图 1-5 所示。这个过程是反复的，不管是在创建源代码，还是在编译或者运行时，只要有错误，就必须通过修改程序源代码以纠正错误，然后再重新编译或者运行。这里介绍的开发环境为 JDK 支持的 DOS 模式，应用 JCreator 的开发过程参见附录 1。

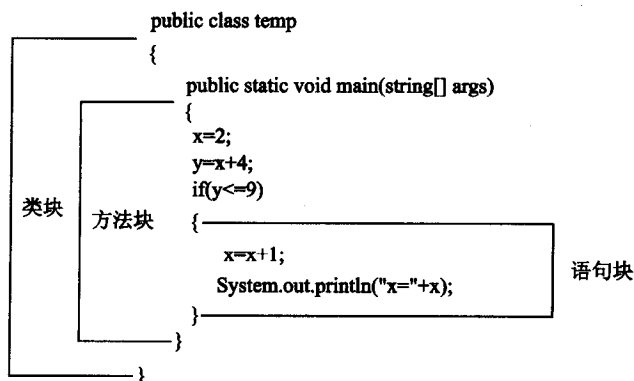


图 1-4 块的示例

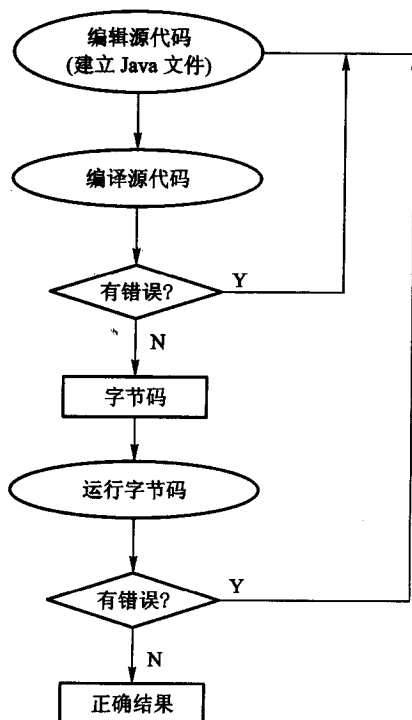


图 1-5 Java 程序开发过程

1. 编辑

编辑源代码可以使用任何一个文本编辑器，源代码文件的文件名必须与程序中定义的公用 (public) 类的类名相同，扩展名必须是“.java”。例 1.1 的源代码文件必须命名为 Simple.java。