



高等院校规划教材

毕广吉 编著

Java程序设计实例教程

强调程序设计方法和思路，引入典型程序设计案例
注重程序设计实践环节，培养程序设计项目开发技能



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 程序设计实例教程

毕广吉 编著



中国水利水电出版社

内 容 提 要

本书由浅入深地介绍 Java 编程, 注重基础知识、基本概念和基本编程方法的介绍, 学习时不要求读者曾经学习过 C、C++ 等语言, 而是直接学习 Java 的基本概念和编程方法; 注重实践过程, 用给出的 90 个具有图形用户界面的、有针对性的、可供实用的实例来阐述 Java 的编程方法。如果读者把这些实例编制完成并运行成功, 对每一条语句都能正确理解, 就基本掌握了 Java 程序设计方法。

本书可作为高等院校 Java 语言程序设计课程的教材, 也可供其他初学 Java 语言的读者学习。

本书提供免费电子教案和书中所有实例的源代码文件, 读者可以从中国水利水电出版社网站 (<http://www.waterpub.com.cn/softdown/>) 下载。

图书在版编目 (CIP) 数据

Java 程序设计实例教程 / 毕广吉编著. —北京: 中国水利水电出版社, 2007

21 世纪高等院校规划教材

ISBN 978-7-5084-4665-3

I. J… II. 毕… III. JAVA 语言—程序设计—高等学校—教材 IV. TP312

中国版本图书馆 CIP 数据核字 (2007) 第 073015 号

书 名	Java 程序设计实例教程
作 者	毕广吉 编著
出 版 发 行	中国水利水电出版社 (北京市三里河路 6 号 100044) 网址: www.waterpub.com.cn E-mail: mchannel@263.net (万水) sales@waterpub.com.cn
经 售	电话: (010) 63202266 (总机)、68331835 (营销中心)、82562819 (万水) 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	北京蓝空印刷厂
规 格	787mm×1092mm 16 开本 17.25 印张 420 千字
版 次	2007 年 6 月第 1 版 2007 年 6 月第 1 次印刷
印 数	0001—4000 册
定 价	26.00 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越式发展,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培养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培养的需要。本套教材消化和吸收了多年来已有的应用型人才培养的探索与实践成果,紧密结合经济全球化时代高等院校应用型人才培养工作的实际需要,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 语言是典型的面向对象的程序设计语言，与面向过程的编程语言相比，学习起来难度较大，使一些初学者望而却步或半途而废。然而正是 Java 语言的面向对象的特点，使它具有面向过程的编程语言所不具有的诸多优越性，代表了程序设计的最新思想和方法，例如，代码的可重用性、跨平台的特点、适合网上运行、支持多线程等，而且学习 Java 语言又是进一步学习数据库、JavaScript、JSP 等知识的基础，所以大学本科学生学习 Java 是必要的。

Java 语言的语法结构与 C、C++ 相似，类的继承机制等与 C++ 相似，于是图书市场上相当一些教材总是以 C、C++ 为基础来讲解 Java。有的书即使不以 C、C++ 为基础，也经常提到“在 C++ 中……”，似乎学习 Java 就必须先学习 C、C++ 似的，这种书给初学者带来很多困难和困惑。因此，很多读者都在寻找一种真正适合初学者学习的 Java 教材，很多教师也需要真正适合教学的教材。

本书正是针对这样的需求编写而成的 Java 语言基础教程。本书由浅入深地介绍 Java 编程，在有限的篇幅内，以介绍 Java 基础编程为主，包括继承、多态、重载、覆盖、接口、封装等面向对象程序设计的核心概念和机制，Applet 程序设计、Java 常用类、图形用户界面、组件、绘图与图像、异常处理、多线程与动画、文件操作、网络编程等，而那些与 Java 有关的进一步的内容（如 JSP 等），为防止内容分散没有收入在本书中，而且这些内容并不是靠一章的篇幅就能讲清的。

本书特点如下：

（1）注重基础知识、基本概念和基本编程方法的介绍，真正适合初学者。学习本书时不要求读者曾经学习过 C、C++ 等语言，不像很多书那样借用 C、C++ 的概念对比进行解释，而是直接讲述 Java 的基本概念和编程方法，使没有学习过 C、C++ 的读者能直接学好 Java 编程。

（2）注重实践过程，使用实例来阐述和示范 Java 的编程方法。本书给出了 90 个具有图形用户界面的、有针对性的、可供实用的实例和一大批小例子，目录中给出了这些实例的索引，其源代码文件可以从中国水利水电出版社网站（<http://www.waterpub.com.cn/softdown/>）下载。学习编程最好的方法就是学习实例，如果读者把这些实例编制完成并对每一条语句都能正确理解，就基本掌握了 Java 程序设计方法。

参加本书编写工作的还有任敏、毕之莹、韩大勇等。本书配套的电子教案（PPT）由苑志旺、钱蓓蓓、钟燕兰、胡世昌制作，谨在此表示感谢。

本书可作为高等院校 Java 语言程序设计课程的教材，也可供其他初学 Java 语言的读者学习。本书的所有实例都在 Windows 98/2000/XP 操作系统以及 Java 1.3.0 环境下运行通过。

毕广吉

2007 年 4 月

目 录

序

前言

第1章 Java 编程语言概述.....1

本章学习目标.....1

1.1 Java 概述.....1

1.2 Java 的软件环境.....2

1.2.1 安装 Java 软件开发包.....2

1.2.2 配置环境变量.....2

1.2.3 安装帮助文档.....3

1.2.4 安装 UltraEdit.exe 编辑器.....3

1.3 Application 程序与 Applet 程序.....3

1.3.1 Java 的 Application 程序与

Applet 程序.....3

1.3.2 Java 程序的编辑、编译和运行.....3

1.3.3 第一个 Application 程序.....4

1.3.4 第一个 Applet 程序.....6

1.3.5 改进第一个 Applet 程序.....8

1.4 HTML 超文本标记语言简介.....10

1.5 UltraEdit 编辑软件 的设置.....12

1.6 初识类与对象.....12

1.7 本章小结.....13

1.8 习题.....13

第2章 Java 语言基础.....15

本章学习目标.....15

2.1 标识符.....15

2.2 关键字.....15

2.3 基本数据类型.....16

2.3.1 基本数据类型简介.....16

2.3.2 关于基本数据类型的说明.....16

2.3.3 基本数据类型的自动转换.....17

2.4 运算符.....18

2.4.1 算术运算符.....18

2.4.2 自增、自减运算符.....18

2.4.3 关系运算符.....19

2.4.4 逻辑运算符.....20

2.4.5 位运算符.....20

2.4.6 条件运算符.....21

2.4.7 赋值运算符.....22

2.4.8 运算符的优先级和结合性.....23

2.5 常量、变量和表达式.....24

2.5.1 变量.....24

2.5.2 常量.....25

2.5.3 表达式.....26

2.6 程序流程控制结构.....27

2.6.1 选择结构.....27

2.6.2 循环结构.....28

2.6.3 控制转移语句.....30

2.6.4 其他语句.....30

2.7 注释.....30

2.8 本章小结.....31

2.9 习题.....31

第3章 类与对象.....34

本章学习目标.....34

3.1 类.....34

3.1.1 类的声明.....34

3.1.2 类体.....34

3.1.3 类的成员变量.....35

3.1.4 类的成员方法.....35

3.2 Java 的类库.....36

3.2.1 自定义类与类库.....36

3.2.2 Java 的常用类库.....36

3.2.3 使用类库的方法.....37

3.3 对象.....	37	5.3 Applet 类.....	73
3.3.1 对象的声明和实例化	37	5.3.1 Applet 类介绍.....	73
3.3.2 引用成员变量和调用成员方法	38	5.3.2 Applet 类的方法.....	74
3.3.3 类的特殊方法——构造方法	38	5.3.3 从 HTML 向 Applet 传递参数 的方法	76
3.4 本章小结	38	5.4 本章小结	77
3.5 习题	39	5.5 习题	77
第 4 章 面向对象的程序设计.....	41	第 6 章 图形用户界面设计.....	79
4 本章学习目标	41	6 本章学习目标	79
4.1 类的封装	41	6.1 图形用户界面设计与 java.awt 包.....	79
4.2 类的继承	41	6.2 组件 Component 的概念	80
4.2.1 创建子类	41	6.2.1 组件类及其子类.....	80
4.2.2 this 对象和 super 对象.....	45	6.2.2 组件类的方法.....	80
4.2.3 final 关键字和 abstract 关键字.....	47	6.3 事件 Event	81
4.3 类的多态性.....	48	6.3.1 事件类 AWTEvent	81
4.3.1 多态的概念	48	6.3.2 事件类的接口 interface	82
4.3.2 方法的重载	48	6.3.3 事件适配器 Adapter	83
4.3.3 方法的覆盖	49	6.4 按钮组件 Button.....	83
4.4 静态类成员	51	6.4.1 按钮组件的方法和事件.....	83
4.5 接口	52	6.4.2 按钮的使用.....	84
4.6 包的概念	54	6.5 与文本有关的组件.....	86
4.7 本章小结	55	6.5.1 标签组件 Label.....	86
4.8 习题	55	6.5.2 使用标签组件.....	87
第 5 章 Java 常用类.....	58	6.5.3 TextComponent 类.....	88
5 本章学习目标	58	6.5.4 文本行组件 TextField	89
5.1 常用的 Java 类.....	58	6.5.5 文本域组件 TextArea	90
5.1.1 对象类 Object	58	6.5.6 使用文本行和文本域.....	91
5.1.2 数据类型类	59	6.6 与选项有关的组件.....	92
5.1.3 数学类 Math.....	59	6.6.1 复选框	93
5.1.4 系统类 System	60	6.6.2 单选按钮组 CheckboxGroup	95
5.1.5 Class 类	61	6.7 与列表有关的组件.....	97
5.1.6 日期类 Date 和日历类 Calendar.....	62	6.7.1 选择框组件 Choice	98
5.2 数组和字符串	63	6.7.2 列表框组件 List	100
5.2.1 数组	63	6.8 滚动条组件 Scrollbar	103
5.2.2 字符串 String 类	66	6.9 颜色类和字体类.....	105
5.2.3 Application 应用程序的命令行参数	72	6.9.1 颜色类 Color	106
5.2.4 字符串 StringBuffer 类	72	6.9.2 字体类 Font	108
5.2.5 字符串数组	73		

6.10 容器类及其子类	109	7.6 习题	170
6.10.1 容器类 Container	110	第8章 异常处理	172
6.10.2 框架类 Frame	110	本章学习目标	172
6.10.3 面板类 Panel	115	8.1 错误与异常的概念	172
6.10.4 对话框 Dialog	118	8.2 异常类	172
6.11 布局管理器	122	8.2.1 Throwable 类	172
6.11.1 FlowLayout 布局	122	8.2.2 常用的异常类	173
6.11.2 BorderLayout 布局	122	8.3 异常的处理	174
6.11.3 GridLayout 布局	124	8.3.1 程序异常的实例	174
6.11.4 GridBagLayout 布局	125	8.3.2 try-catch-finally 结构	177
6.11.5 null 布局	128	8.3.3 主动抛出异常	180
6.12 菜单	130	8.4 自定义异常类	184
6.12.1 下拉菜单	130	8.5 将异常交系统处理	187
6.12.2 弹出式菜单	135	8.6 本章小结	187
6.13 Java 中的汉字编码	139	8.7 习题	188
6.14 本章小结	141	第9章 多线程机制与动画	190
6.15 习题	141	本章学习目标	190
第7章 绘图与图像	144	9.1 线程的建立	190
本章学习目标	144	9.1.1 进程与线程的概念	190
7.1 Java 绘图基础	144	9.1.2 继承 Thread 类建立线程	190
7.1.1 绘图容器与坐标系	144	9.1.3 用 Runnable 接口建立线程	193
7.1.2 画布类 Canvas	144	9.1.4 用 Runnable 接口建立多线程	196
7.1.3 paint 方法和 repaint 方法	145	9.2 关于线程的进一步讨论	198
7.1.4 绘图类 Graphics	150	9.2.1 线程的状态	198
7.2 在组件上绘图	154	9.2.2 线程的优先级	199
7.2.1 绘制任意曲线	154	9.2.3 线程组	199
7.2.2 用 drawString 和 drawChars 方法 绘制字符	156	9.3 图像动画	201
7.3 绘图实例	158	9.3.1 用线程控制动画	202
7.3.1 徒手绘图	158	9.3.2 定时器	203
7.3.2 橡皮筋画法	160	9.3.3 定时器的进一步改进	206
7.4 显示图像	163	9.3.4 用定时器控制动画的方法	209
7.4.1 利用 Graphics 类显示图片	163	9.4 本章小结	212
7.4.2 制作相册	164	9.5 习题	212
7.4.3 连续位图动画	167	第10章 文件操作	214
7.4.4 更换图标	168	本章学习目标	214
7.5 本章小结	169	10.1 文件对话框类 FileDialog	214
		10.2 文件类 File	220

10.3 流的概念	221	10.10 习题	247
10.3.1 数据流的概念	221	第 11 章 网络程序设计	249
10.3.2 标准的输入输出	222	本章学习目标	249
10.3.3 java.io 包中的数据流类	223	11.1 Java 的网络功能	249
10.4 顺序读写文件	225	11.2 使用 URL 类	249
10.4.1 用文件输入流顺序读文件	225	11.2.1 URL 基础知识	249
10.4.2 用字节流顺序写文件	227	11.2.2 URL 类	250
10.4.3 用字符流顺序写文件	229	11.3 URLConnection 类	253
10.4.4 用文件输入输出流读写文件	230	11.4 Java 显示网页的方法	255
10.5 使用 LineNumberReader 类	235	11.4.1 AppletContext 接口的 showDocument 方法	255
10.6 随机读写文件	237	11.4.2 在 Java 程序中显示网页	255
10.7 获取文件信息	240	11.5 Socket 通信	257
10.7.1 SimpleDateFormat 类	240	11.5.1 Socket 通信基础知识	257
10.7.2 获取文件属性	241	11.5.2 Socket 通信实例服务器端	258
10.8 运行可执行文件	243	11.5.3 Socket 通信实例客户端	260
10.8.1 Runtime 类	243	11.5.4 Socket 通信实例的使用	262
10.8.2 在 Java 中执行 EXE 文件	244	11.6 本章小结	263
10.8.3 选择一个可执行文件并运行	244	11.7 习题	263
10.9 本章小结	247		

实例索引

实例 递归累加	4	实例 在字符串中查找字符	69
实例 加法	6	实例 字符串转换成数值	70
实例 改进加法	8	实例 数值转换成字符串之一	70
实例 类的继承	41	实例 数值转换成字符串之二	71
实例 this 引用的例子	45	实例 解析命令行参数	72
实例 super 引用的例子	46	实例 字符串数组	73
实例 方法的重载	48	实例 测试 Applet 类中方法的用途	74
实例 方法的覆盖	49	实例 从 HTML 向 Applet 传递参数	76
实例 类变量	51	实例 按钮的使用	84
实例 接口	53	实例 标签组件的例子	87
实例 使用 Date 类和 Calendar 类	62	实例 文本行和文本域的使用	91
实例 求一个数组中最大的元素及其下标	64	实例 复选框组件的使用	93
实例 用数组的数组存放杨辉三角形	65	实例 单选按钮组的使用	95
实例 获取字符串长度的例子	67	实例 选择框组件	98
实例 字符串比较	68	实例 用列表框实现单选、多选并响应单击 和双击事件	101
实例 取子字符串	68		

实例	滚动条组件的使用	104	实例	完善整数除法	181
实例	用四种方法创建红色	107	实例	自定义异常类	184
实例	使用字体	108	实例	毕达哥拉斯数和完全平方数	191
实例	使用框架	111	实例	滚动字符串	194
实例	框架的事件	113	实例	时钟和字符串	196
实例	面板类的用法	116	实例	多线程	200
实例	对话框的实例	119	实例	改进连续位图动画	202
实例	测试 BorderLayout 布局的效果	123	实例	定时器	204
实例	测试 GridLayout 布局的效果	124	实例	定时器的进一步改进	206
实例	使用 GridBagLayout 布局	126	实例	用定时器控制动画	209
实例	使用 null 布局	128	实例	使用对话框选择文件	215
实例	下拉菜单的实例	132	实例	标准输入输出	222
实例	弹出式菜单的使用	136	实例	打开并读取文件	225
实例	汉字编码	139	实例	用字节流写文件	227
实例	在 Canvas 上绘图	145	实例	用字符流写文件	229
实例	在 Applet 上绘图	149	实例	复制文件	230
实例	使用绘图方法绘制图形	152	实例	复制文件之二	231
实例	旋转的斜椭圆	154	实例	为源程序加行号	235
实例	绘制字符和字符串	156	实例	随机读写文件	238
实例	徒手绘图	158	实例	文件属性列表	241
实例	用橡皮筋画法绘制直线段	160	实例	运行可执行文件 1	244
实例	在 Applet 中显示位图	163	实例	运行可执行文件 2	244
实例	相册	165	实例	用 openStream 读 URL 文件	251
实例	连续位图动画	167	实例	使用 URLConnection 类读 URL 文件	253
实例	更换图标	168	实例	用 showDocument 显示网页内容	255
实例	不完善的除法计算器 1	174	实例	Socket 通信实例服务器端程序	258
实例	不完善的除法计算器 2	176	实例	Socket 通信实例客户端程序	260
实例	除法计算器	178			

第 1 章 Java 编程语言概述

本章学习目标

本章主要介绍 Java 的基本知识,通过初步编程了解 Java 程序的编辑、编译和运行过程,建立对 Java 的感性认识。通过本章的学习,应该掌握以下内容:

- 了解 Java 的基本概念。
- 掌握安装 Java 软件开发包的方法和配置环境变量的方法。
- 掌握 Java 程序的编辑、编译和运行过程。
- 了解 HTML 超文本标记语言。
- 初步理解类与对象的概念。

1.1 Java 概述

Java 是一种编程语言,除了具有所有编程语言的共同特点之外,由于 Java 主要用于网络编程,Java 语言还具有其他编程语言所不具有的诸多特点,可以概括为以下 5 点:

1. Java 是完全面向对象的编程语言

Java 语言是完全面向对象的编程语言,在 Java 语言中一切都是类。Java 语言利用类和对象的机制将数据和对数据的操作封装在一起,并通过统一的接口与外界实现交互,使程序中的各个类彼此独立、自治又能继承,大大提高了程序的可维护性和可重用性,同时大大提高了开发效率。

Java 语言的编程过程就是设计类、继承类、实现类、定义与调用类的属性及方法的过程。

2. Java 是编译解释型的编程语言

以往的编程语言可分为编译型语言和解释型语言两种。

编译型语言的优点是编译成可执行的 EXE 文件后不再需要编程环境的支持,但正是因为这样,编译型语言编译时往往将许多函数放在 EXE 文件之中或者需要带有函数库文件。编译生成的 EXE 文件加上函数库文件往往很大,在网络上传输很困难。

解释型语言的执行离不开解释器,这种解释器往往就是开发程序的编程环境。用户要执行解释程序就必须安装相应的编程环境。从网络传输的角度来讲,解释型语言传输的是源代码文件,虽然源代码文件较之 EXE 文件加上函数库文件小很多,但丧失了程序的保密性。

也有些语言在编程阶段采用解释运行,程序编制完成后再编译成可执行程序。此种方式兼有了二者的优点,但在网络传输方面其缺点依然存在。

Java 是编译解释型的编程语言。Java 语言的源程序不能解释执行,必须使用编译器 javac.exe 进行编译。但是 Java 并不把源程序编译成可执行的 EXE 文件,而是编译成比 EXE 文件小很多的字节码文件,即扩展名为 class 的一个或多个文件,这种很小的字节码文件非常

有利于在网络上传输。用户在网上获取这种字节码文件之后不能直接执行,需要一种“解释器”。然而只要计算机安装了网络浏览器(如 Microsoft 的 IE)就同时安装了这种“解释器”,所以以 Java 的字节码文件形式出现的 Java 程序在网上是畅通无阻的。

顺便提及,Java 除了在网上应用之外,也能编制普通的应用程序。在这种方式下也是先将源文件编译成字节码文件 class,但不是靠浏览器解释执行,此时的解释器是 java.exe。

3. Java 是跨平台的编程语言

由于在网络上传播的是 Java 的字节码,对字节码解释执行的任务交由浏览器负责,所以 Java 程序有良好的跨平台特性,即在一种系统下编制的 Java 程序一经编译成字节码文件,就可以不加修改地在任何系统中运行。这种普遍适用的程序大大降低了程序开发、维护和管理成本。

4. Java 是适合在网络上运行的编程语言

如前所述,在网络上传输的只是 Java 程序的字节码,而且 Java 程序中的每个类都单独编译成一个字节码文件,所以传输量小、传输速度快,因此 Java 是适合在网络上运行的编程语言。

5. Java 是支持多线程的编程语言

多线程技术是指在一个应用程序中有两个或更多个执行线程,即一个应用程序能够同时做两件以上的东西,这就满足了程序对某些复杂功能(如动画)的要求。Java 语言内置了多线程功能,提供了语言级的多线程支持,预先定义了一些用于建立、管理多线程的类和方法,使得开发多线程应用程序变得简单、方便而且有效。

1.2 Java 的软件环境

1.2.1 安装 Java 软件开发包

建立 Java 软件的开发环境需要从 <http://java.sun.com/products> 网站下载 j2sdk1_3_0-win.exe 文件和 j2sdk1_3_0-doc.zip 文件。

要安装 Java 软件并使其能正常工作需要经过安装 Java 软件、配置环境变量等几个步骤。

运行 j2sdk1_3_0-win.exe 安装 JDK 开发包,安装过程与普通的 Windows 安装程序基本相同,在安装过程中可使用默认选项,不再赘述。

1.2.2 配置环境变量

配置环境变量的方法依操作系统的不同而有所区别,以下分别就 Windows 98 和 Windows NT/2000/XP 两种系统加以说明。

(1) Windows 98 操作系统。在 Windows 资源管理器中找到 C:\AUTOEXEC.BAT 文件并右击,在弹出的快捷菜单中单击“编辑”命令,在 C:\AUTOEXEC.BAT 文件中添加以下三行内容,用于设置文件查找路径:

```
set PATH=%path%;c:\jdk1.3\bin
set CLASSPATH=.;c:\jdk1.3\lib\tools.jar;c:\jdk1.3\lib\dt.jar;c:\jdk1.3\lib
set Java_HOME=C:\jdk1.3
```

其中 c:\jdk1.3 是安装路径, 如果 JDK 开发包不是安装在该路径下, 应做相应的修改。进行上述修改后必须重新启动计算机, 这些设置才起作用。

(2) Windows NT/2000/XP 操作系统。右击桌面上的“我的电脑”图标, 在弹出的快捷菜单中选择“属性”命令, 弹出“系统属性”对话框。在“高级”选项卡中单击“环境变量”按钮, 弹出“环境变量”对话框。在“系统变量”列表框中选 Path 项, 然后单击“编辑”按钮, 弹出“编辑系统变量”对话框。在“变量值”文本框中原有内容的后面输入“;c:\jdk1.3\bin”, 最后逐层单击“确定”按钮, Path 环境变量设置完毕。

如果在“系统变量”列表框中没有找到 Path 项, 则单击“新建”按钮, 弹出“新建系统变量”对话框, 然后在“变量名”文本框中输入 Path, 在“变量值”文本框中输入“c:\jdk1.3\bin”, 最后逐层单击“确定”按钮, Path 环境变量设置完毕。

用同样的方法设置 CLASSPATH 和 Java_HOME 环境变量。

1.2.3 安装帮助文档

双击 j2sdk1_3_0-win.exe, 将该文件解压缩, 解压后文件的存放位置可以任选, 通常可放在 c:\jdk1.3\下。解压后有一个 Index.html 文件, 双击该文件就打开帮助文档的首页。为了以后使用方便, 可以为 Index.html 文件在桌面上建立一个快捷方式。

1.2.4 安装 UltraEdit.exe 编辑器

在网络上下载一个 UltraEdit.exe 文件, 双击执行 UltraEdit.exe 安装程序, 把它安装到默认的目录下, 安装后双击 Uedit32.exe 就打开文本编辑器, 本书中所有 Java 源程序文件(扩展名为 java)和所有 html 文件都是用这个编辑器编辑的。

1.3 Application 程序与 Applet 程序

1.3.1 Java 的 Application 程序与 Applet 程序

Java 程序分为 Application 程序与 Applet 程序两类。Application 程序是普通的应用程序, 但是其编译结果不是通常的 EXE 文件而是 class 文件, 要执行这种 class 文件就要使用解释器 java.exe。Application 程序一般是在本地运行的。

Applet 程序不是独立的程序, 使用时必须把编译时生成的 class 文件嵌入到 html 文件中, 借助浏览器解释执行。

由于这两类程序的执行方式不同, 书写其源程序时的格式也有所不同, 在 1.3.3 节和 1.3.4 节中将结合实例加以介绍。

1.3.2 Java 程序的编辑、编译和运行

开发一个 Java 程序要经过编辑源程序、编译生成字节码文件和运行字节码文件三个步骤。

1. 编辑源程序

Java 的源程序是纯文本文件, 因而可以用任何一种纯文本文件编辑器来编辑, 例如 Windows 的记事本等。本书使用的编辑器是 UltraEdit 编辑软件。该软件的编辑、格式等功能

强大，能够在编辑过程中得心应手，而且该软件能设置关键字、注释、数字、字符串等以不同的颜色和风格显示，并有自动更正关键字的功能，这在编程时是十分方便的。

源程序编辑完毕以 java 为扩展名存盘。

2. 编译生成字节码文件

编译生成字节码文件是在 MS-DOS 方式下进行的，通常的做法是，先打开 MS-DOS 方式窗口，然后在 MS-DOS 窗口中输入以下命令：

```
javac javafilename.java
```

其中 javafilename 是源程序文件的文件名，其扩展名 java 不能省略。如果编译正确，将在该命令的下一行出现 DOS 的提示符；如果有错误发生，将显示错误信息。

3. 运行字节码文件

运行字节码文件的方法视 Java 文件类型的不同而不同，分为两种情况。

(1) Application 程序。运行 Application 程序也是在 MS-DOS 窗口中进行的，使用的 DOS 命令是：

```
java classfilename
```

其中 classfilename 是字节码文件的文件名，其扩展名 class 可以省略。如果程序运行正确，将出现一个窗口显示运行的结果；如果有错误发生，将显示错误信息。

(2) Applet 程序。Applet 程序必须嵌入在 html 文件里由浏览器解释执行，嵌入的方法是在 html 文件中需要调用 Java 程序的地方添加以下内容：

```
<applet code="classfilename.class" width="500" height="200">
</applet>
```

其中 classfilename 是字节码文件的文件名，其扩展名 class 不能省略，不能带有路径，通常与 html 文件放在同一目录中。width 和 height 是两个必要的参数，分别表示 Applet 程序占用网页区域的宽度和高度，单位为像素。

编辑好的网页文件以 html 为文件名存盘，然后在浏览器中打开网页文件，Applet 程序就会在指定的网页区域中运行。如果未出现预定的效果，应当在 java 源文件中查找错误，改正后重新编译 class 文件，然后再一次在网页中观看效果。

以下两节分别通过两个具体的例子说明这些步骤。

事先声明：为了便于解释说明程序中各语句的作用，本书为所有的源程序都加上了行号，读者在输入程序时，不要输入这些行号，而应当直接输入行号后的语句。

1.3.3 第一个 Application 程序

实例 递归累加

作为第一个 Application 程序，它的功能是计算从 1 到 n 的自然数的和，本例中 n=5。

(1) 在 UltraEdit 编辑器中输入以下程序，并以 SumUp.java 为文件名将程序存盘。

```
1 //源文件名: SumUp.java
2 //在“源程序”中的文件夹: 第 01 章\0101 第一个 Application 程序--递归累加
3 public class SumUp
4 {
5     static int f(int n) //递归方法
6     {
```

```

7      if (n==1)
8      {
9          System.out.println("Σ(1)=1");
10         System.out.println("最后结果: ");
11         return 1;
12     }
13     else
14     {
15         System.out.println("Σ("+n+")="+n+"+"Σ("+ (n-1) +")");
16         return n+f(n-1);
17     }
18 }
19 public static void main(String args[])
20 {
21     int i=5;
22     if (args.length>0)
23     {
24         i = Integer.parseInt(args[0]); //将 args[0] 转化为 int 值
25     }
26     System.out.println("递归过程: ");
27     System.out.println("Σ("+i+")="+f(i));
28 }

```

(2) 在 UltraEdit 编辑器中按 F9 键, 打开“DOS 命令”对话框, 在上方的组合框中输入 DOS 命令, 如图 1-1 所示。

```
javac SumUp.java
```

单击“浏览”按钮, 在弹出的“浏览文件夹”对话框中选择工作文件夹为保存 SumUp.java 的文件夹, 在本例中是“E:\Java\实例\第 01 章\第一个 Application 程序--递归累加”, 单击“确定”按钮即把 SumUp.java 编译成 SumUp.class, 并存放在指定的目录中。

如果出错, UltraEdit 编辑器会在新的编辑窗口中显示错误信息。

(3) 再一次在 UltraEdit 编辑器中按 F9 键, 打开“DOS 命令”对话框, 在上方的列表框中输入 DOS 命令, 如图 1-2 所示。

```
java SumUp
```

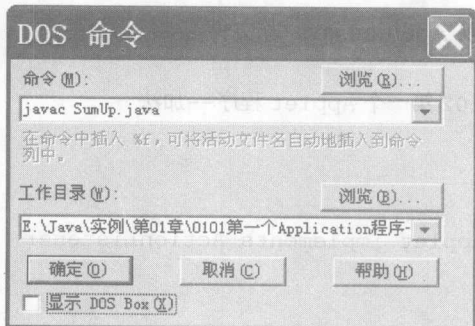


图 1-1 编译成字节码文件

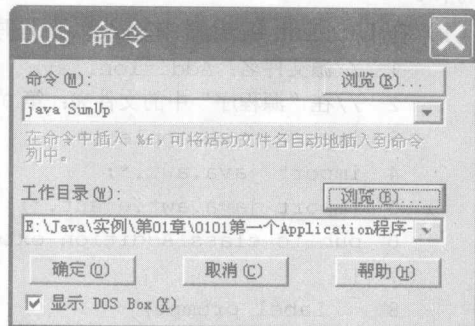


图 1-2 运行字节码文件

单击“浏览”按钮，在弹出的“浏览文件夹”对话框中选择保存 SumUp.java 的文件夹，在本例中是“E:\Java\实例\第 01 章\第一个 Application 程序--递归累加”（这时该文件夹应是默认的文件夹），单击“确定”按钮就运行了 SumUp.class 程序，并在新的窗口中显示运行的结果。本例的运行结果如图 1-3 所示。

```

1 递归过程:
2  $\Sigma(5)=5+\Sigma(4)$ 
3  $\Sigma(4)=4+\Sigma(3)$ 
4  $\Sigma(3)=3+\Sigma(2)$ 
5  $\Sigma(2)=2+\Sigma(1)$ 
6  $\Sigma(1)=1$ 
7 最后结果:
8  $\Sigma(5)=15$ 

```

图 1-3 SumUp.class 的运行结果

(4) 读者现在不必过细地追究该程序编程原理的细节，通过本例只需弄清以下几个问题。

1) 程序的第 3 行：

```
public class SumUp
```

定义了一个类，类名是 SumUp。类的内容即类体放在其后的一对大括号中。类名就是存盘时的文件名，扩展名为 java。Java 语言是大小写敏感的，由于类名是 SumUp，本例存盘的文件名必须是 SumUp.java，写成 sumUp.java 或 sumup.java 等都是错误的。

2) 程序的第 19 行：

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

定义了一个名为 main 的方法，这是 Application 程序所要求的。main 方法是 Application 程序开始运行时的入口处，每一个 Application 程序必须有一个 main 方法。

3) 通过本例读者还应当学会编辑、编译、执行 Application 程序的方法和步骤。

1.3.4 第一个 Applet 程序

实例 加法

运行本程序时，在网页中出现一个加法计算器，该计算器有三个文本输入域，分别用来输入被加数、加数和显示计算的得数——和，该计算器还用标签显示加号“+”，用一个命令按钮显示等号“=”，输入完被加数和加数之后，只要单击命令按钮就在第三个文本域中显示二者的和。

(1) 在 UltraEdit 编辑器中输入以下程序，并以 Addition.java 为文件名将程序存盘。

```

1 //源文件名: Addition.java
2 //在“源程序”中的文件夹: 第 01 章\0102 第一个 Applet 程序--加法
3 import java.applet.*;
4 import java.awt.*;
5 import java.awt.event.*;
6 public class Addition extends Applet implements ActionListener
7 {
8     Label prompt;
9     TextField input1,input2,output1;
10    Button button1;

```