

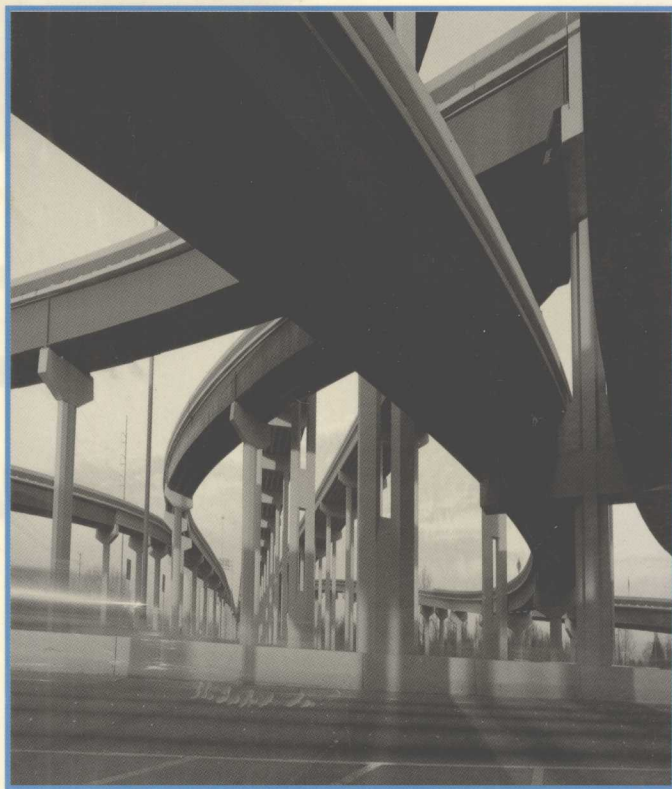
Java 程序设计 与应用技术教程

黄 斐 编著
崔志明 审

突出应用主题

具备理论先进性与后续性

贯彻可行性与熟练技巧培养



科学出版社
www.sciencep.com



TP312
H823

Java 程序设计 与应用技术教程

黄 斐 编著
崔志明 审

突出应用主题

具备理论先进性与后续性

贯彻可行性与熟练技巧培养



内 容 简 介

本书是教育部世界银行贷款 21 世纪初高等理工科教育教学改革项目中《以培养创新人才为目标的地方院校计算机科学与技术专业教学改革的研究与实践》项目成果之一,是“21 世纪高等院校计算机技术教学”丛书中的一本。

随着 Java 的应用越来越广泛,Java 程序设计已越来越受到广大计算机教育工作者的重视。Java 作为一种面向对象的程序设计语言,同时具备了与平台无关性、高可靠性、多线程等特性,是 Internet 上最流行的编程语言之一,正被越来越多的用户了解和使用。

本书共分十一章,主要内容包括:Java 基本语法、科学计算与输入输出、程序流程控制、数组、类和对象、Java 接口和包及异常、标准类库和字符串处理、输入/输出流、图形用户界面(GUI)及网络编程。本书在介绍 Java 的基本语法和常用算法的基础上重点讨论了 Java 的类和对象,并根据程序设计课程的特点,在讲述语法规则的同时,结合面向对象技术的基本原理和思维方法,给出相关的程序实例。本书还特别注重提高读者运用 Java 语言和面向对象技术解决实际问题的能力。书中给出了大量经过调试运行通过的实例,便于初学者入门。每章都附有习题,供课后练习。

需要本书或技术支持的读者,请与北京中关村 083 信箱(邮编:100080)发行部联系,电话:010-62528991,62524940,62521921,62521724,82610344,82675588(总机)传真:010-62520573, E-mail: yanmc@bhp.com.cn

图书在版编目(CIP)数据

Java 程序设计与应用技术教程 / 黄斐编著. —北京: 科学出版社, 2003

(21 世纪高等院校计算机技术教学丛书)

ISBN 7-03-012270-4

I. J... II. 黄... 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(2003)第 086051 号

责任编辑: 杨敏

/ 责任校对: 一凡

责任印刷: 东升

/ 封面设计: 梁运丽

科学出版社 出版

北京东黄城根北街 16 号

邮政编码: 100717

<http://www.sciencep.com>

北京市东升印刷厂印刷

科学出版社发行 各地新华书店经销

*

2003 年 9 月第 一 版 开本: 787×1092 1/16

2003 年 9 月第一次印刷 印张: 17 1/4

印数: 1—3 000

字数: 409 000

定价: 23.00 元

21 世纪高等院校计算机技术教学丛书

编委会成员名单

顾问委员：（以姓氏笔划为序）

刘 璟 教授 南开大学

何炎祥 教授 武汉大学

施伯乐 教授 复旦大学

钱培德 教授 苏州大学

主任委员：左孝凌 教授 上海交通大学

副主任委员：徐汀荣 教授 苏州大学

委员：（以姓氏笔划为序）

左孝凌 教授 上海交通大学

朱巧明 教授 苏州大学

宋方敏 教授 南京大学

余雪丽 教授 太原理工大学

徐汀荣 教授 苏州大学

陶树平 教授 同济大学

崔志明 教授 苏州大学

曾 明 教授 西安交通大学

谢康林 教授 上海交通大学

序

本丛书是苏州大学主持的高等院校计算机技术教学丛书，其编委会聚集了国内很多院校的专家，编委会为本丛书的组织、审定起到集思广益、推波助澜的作用。

苏州大学建校百年，蜚声海内外，学校治学严谨、理论底蕴深厚，理科各系，久负盛名。近年来学校多次调整院系，全校各系蓬勃发展，尤其是电子信息各科发展迅速。当前信息时代，信息技术的应用已经深入各行各业，而苏州大学更以信息技术的应用作为各个专业的发展之本。

发展信息技术，苏州大学具有以下三个方面的独特优势：

一是苏州大学计算机工程系成立于1987年10月，至今已15年，这正是信息科学突飞猛进的时期。当前计算机技术的应用已经遍及生产、生活各个方面，苏州大学计算机工程系顺时应势，使全系的教学科研迅速崛起。目前系内除有211工程重点建设的计算机应用学科外，还有江苏省重点实验室（计算机信息处理技术实验室）以及纵横汉字信息技术研究所、多媒体应用技术研究等，在国内外极具影响的研究机构。苏州大学计算机工程系现有博士点一个、硕士点两个，这些都显示了该系具有强劲的得天独厚的科研教育环境。

二是苏州大学地处“天堂”苏州，南邻上海，北依南京，近年新加坡和中国在苏州合建工业园区，而世界著名信息公司Motorola也移师毗邻，使苏州成为其主要研发基地。现在苏州已经成为世界信息技术产业中心之一，信息技术产业产值已达一千亿元以上。苏州大学身临其境，近水楼台，耳濡目染，凭借着培养人才的基础以及软件开发的优势，为新加坡工业园区的建设以及Motorola的芯片研发等项目进行了广泛的优势互补的合作。

三是十几年来信息技术一直是苏州大学发展重点，校长钱培德教授是国内著名信息技术学科的学术带头人，强将手下无弱兵。在苏州大学有一批奋发有为、敬业创新在教学科研上崭露头角的中青年教师队伍，他们把长期从事科学研究的经验与信息教学的实践紧密结合起来编写了一套以理论底蕴为基础，面向应用为目标，充分反映当前计算机进展的实用教材。本丛书即是重要组成部分。

本丛书包括现代软件工程、单片机技术、数据库系统及SYBASE技术、计算机组网技术、计算机图形图像技术、Delphi程序设计、Java程序设计、计算机网页设计等。这些教材有的是计算机专业核心教材，有的是电子类相关专业的实用性应用教材。这套丛书的立足点是以计算机教学应用为主体，也能反映当前的发展动向，在理论上具有先进性和后续性，而在教学实践中努力贯彻可行性和熟练技巧培养。为了改革精简教学内容，突出应用主题，我们把网络原理等大课，分解为组网技术和网页设计等不同主题，以便适应有关专业的不同需要。

我们感谢北京希望电子出版社领导对本套丛书的立题、审稿、编辑、出版等做的大量工作。编委会希望广大读者能够指出丛书中的各种疏漏，期望这套丛书能在信息技术日新月异飞速发展的年代起到一点卵石铺路的作用。

高等院校计算机技术教学丛书编委会 左孝凌

2002年8月

前 言

伴随着 Internet 的飞速发展, Java 正被越来越多的用户了解。Java 的平台无关特性、安全机制、高可靠性和内嵌的网络支持, 使之成为当前编写网络应用程序的首选工具之一。

Java 是一种面向对象程序设计语言。面向对象技术提倡的模拟现实世界的思维方式、数据与操作相捆绑的程序风格符合了现代大规模软件开发的要求和潮流, 正逐步替代面向过程的程序设计技术, 而成为计算机应用开发领域的主流趋势。本书将以 Java 语言为基础, 介绍面向对象程序设计的主要原则和方法。

应用 Java 语言进行程序设计的关键是要掌握其特定的思维方式方法, 本书先介绍 Java 的基本语法和常用算法, 在此基础上重点讨论了 Java 类和对象。考虑到程序设计课程的特点, 在讲述语法规则的同时, 给出相关的程序实例。本书第 1~9 章的程序使用 Application 格式, 第 10 章的 GUI 程序采用 Applet 格式, 但也能够以 Application 方式运行。第 11 章中的网络程序设计与数据库接口, 分别采用基于 Window 平台的 PWS 和 Access。

由于 Java 功能强大, 所涉及到的概念、规则等要比其他计算机语言相对复杂一些, 不少读者感到学习困难。本书是一本程序设计基础教材, 不仅适合初学者, 同样也适合具有一定程序设计基础的读者。本书具有以下特点:

1. 把 Java 语言的难点分散, 力求文字流畅、通俗易懂、使读者循序渐进。
2. 书中凡是涉及到的重要概念和技术要点, 均配有适当的实例和文字解说。书中配有一定数量的习题, 使读者能对重要概念和技术要点深入理解, 帮助读者掌握 Java 程序设计的基本方法。

3. 书中所有的程序实例都在 Windows 98 支持下的 JDK1.2.2 基础上调试通过。

本书可作为高等学校计算机及相关专业程序设计课程的入门教材, 或作为各学校程序设计公共选修课的教材, 本书也适用于接受职业教育或从事实际软件开发的读者学习使用。

本书由苏州大学副教授黄斐编著, 崔志明教授审。本书是教育部世界银行贷款 21 世纪初高等理工科教育教学改革项目中《以培养创新人才为目标的地方院校计算机科学与技术专业教学改革的研究与实践》成果之一, 是由苏州大学主持编写的高等院校计算机技术教学丛书中的一本。

本书在编写过程中得到了北京希望电子出版社和苏州大学计算机工程系徐汀荣教授的大力支持, 在此一并致谢!

编 者

2002 年 3 月 18 日

目 录

第 1 章	JAVA 概述	1
1.1	JAVA 的特征	1
1.2	JAVA 程序的类型	3
1.3	简单的 JAVA 程序介绍	3
1.3.1	一个简单的 Java 程序	4
1.3.2	程序的基本要求	5
1.3.3	Java 程序的开发步骤	5
1.4	面向对象的程序设计概述	6
1.4.1	对象及面向对象的程序设计	6
1.4.2	面向对象程序设计的要素	6
1.5	JAVA 程序运行环境	7
习题 1		10
第 2 章	JAVA 的基本语法	11
2.1	词法与句法	11
2.1.1	关键字	12
2.1.2	标识符	12
2.2	JAVA 的基本数据类型	13
2.2.1	整型	14
2.2.2	浮点型	14
2.2.3	字符型	14
2.2.4	布尔型	14
2.3	常量	14
2.3.1	整型常量	14
2.3.2	实型常量	15
2.3.3	字符常量	15
2.3.4	字符串常量	16
2.4	变量	16
2.4.1	变量说明	17
2.4.2	变量赋初值	17
2.4.3	final 变量	18
2.4.4	作用域	18
2.5	基本运算符	19
2.5.1	运算符的优先级	19
2.5.2	算术运算符	20
2.5.3	关系运算符	21
2.5.4	逻辑运算符	22
2.5.5	位运算符	22
2.6	表达式	24
2.6.1	算术表达式	24
2.6.2	关系表达式	25

2.6.3	逻辑表达式	25
2.6.4	赋值表达式	26
2.6.5	复合赋值运算符	26
2.6.6	赋值时的类型转换	27
2.7	表达式语句和空语句	29
2.7.1	表达式语句和空语句	29
2.7.2	自增、自减运算符	29
习题 2		30
第 3 章	科学计算与输入输出	33
3.1	科学计算方法	33
3.2	标准输出方法	36
3.3	标准输入方法	36
3.4	自定义输入法	38
习题 3		41
第 4 章	程序的流程控制	42
4.1	条件选择语句	42
4.1.1	if 语句	42
4.1.2	if.else 语句	43
4.1.3	if 语句的嵌套	44
4.1.4	条件运算符	45
4.2	SWITCH 语句	46
4.3	循环控制语句	49
4.3.1	while 语句	49
4.3.2	do.while 语句	50
4.3.3	for 语句	51
4.3.4	循环的嵌套	54
4.4	跳转语句	54
4.4.1	break 语句	54
4.4.2	continue 语句	56
4.5	控制结构程序实例	57
习题 4		60
第 5 章	数组	62
5.1	一维数组	62
5.1.1	一维数组的声明	62
5.1.2	数组的创建	63
5.1.3	声明和创建的联用格式	63
5.1.4	一维数组的初始化	63
5.1.5	一维数组应用	65
5.2	多维数组	68
5.2.1	二维数组的声明与创建	68
5.2.2	二维数组元素的初始化	69
5.2.3	二维数组应用举例	70
习题 5		72

第 6 章	JAVA 类和对象	76
6.1	面向对象程序设计的概念	76
6.1.1	对象	76
6.1.2	消息	77
6.1.3	类	78
6.1.4	继承	78
6.1.5	抽象	78
6.1.6	多态型	79
6.2	类的描述	79
6.3	成员变量	82
6.3.1	成员变量定义	82
6.3.2	成员变量的访问权限	83
6.3.3	静态变量	85
6.3.4	final 变量	86
6.4	成员方法	86
6.4.1	方法的定义	86
6.4.2	形参与实参	88
6.4.3	成员变量和局部变量的作用域	90
6.4.4	方法的递归调用	91
6.4.5	方法的重载	93
6.5	对象	94
6.5.1	创建对象	95
6.5.2	使用对象	97
6.5.3	访问对象	98
6.5.4	this	110
6.6	类的继承	113
6.6.1	创建子类	113
6.6.2	子类对象的使用	117
6.6.3	抽象类	120
习题 6		122
第 7 章	JAVA 接口、包和异常	124
7.1	接口	124
7.1.1	接口定义	125
7.1.2	接口的实现	125
7.1.3	接口作为数据类型	126
7.1.4	接口继承	127
7.1.5	接口中的变量	128
7.2	包	129
7.2.1	包的定义	130
7.2.2	包的使用	132
7.2.3	标识符的作用域	133
7.3	异常处理	134
7.3.1	异常的分类	135

7.3.2 异常的抛出	137
7.3.3 异常处理机制	138
习题 7	143
第 8 章 标准类库和字符串处理	144
8.1 JAVA 类库简介	144
8.2 创建字符串对象	145
8.2.1 创建 String 对象	146
8.2.2 创建 StringBuffer 对象	147
8.3 使用 STRING 类	148
8.3.1 求字符串长度	148
8.3.2 连接字符串	148
8.3.3 拷贝字符串	149
8.3.4 比较字符串	151
8.3.5 搜索字符	154
8.3.6 搜索子串	155
8.3.7 修改字符串	156
8.4 使用 STRINGBUFFER 类	157
8.4.1 把字符串添加到缓冲区	158
8.4.2 将字符串插入到缓冲区	159
8.4.3 从缓冲区中获取字符串	160
8.4.4 修改缓冲区字符串	160
8.4.5 求缓冲区容量和字符数	161
习题 8	162
第 9 章 JAVA 的输入/输出流	164
9.1 输入/输出包	164
9.1.1 I/O 流	164
9.1.2 文件管理	168
9.2 节点流	171
9.2.1 FileReader 类	172
9.2.2 FileWriter 类	173
9.2.3 FileInputStream 和 FileOutputStream 类	175
9.3 过滤流	178
9.3.1 使用 I/O 缓冲区	178
9.3.2 基本类型数据传输	181
9.3.3 PrintWriter 类	184
9.4 标准输入/输出流	185
9.4.1 标准输入流 in	186
9.4.2 标准输出流 out	186
9.4.3 标准错误输出流 err	187
9.4.4 System 类常用方法	187
习题 9	189
第 10 章 JAVA 图形用户界面	191
10.1 JAVA 图形用户界面概述	191

10.1.1	容器与基本组件	191
10.1.2	Java GUI 的组成	194
10.1.3	设计图形用户界面	195
10.2	GUI 程序设计常用技术	197
10.2.1	属性、方法和事件	197
10.2.2	创建 GUI 应用程序	198
10.2.3	事件处理机制	199
10.3	GUI 常用组件	205
10.3.1	标签和文本框	205
10.3.2	命令按钮	210
10.3.3	复选框和单选按钮组	212
10.3.4	组合框和列表框	216
10.4	布局管理和容器	222
10.4.1	布局管理	222
10.4.2	容器组件	228
习题 10		243
第 11 章	JAVA 网络程序设计及相关技术	244
11.1	APPLET 基础知识	244
11.1.1	Applet 和 WWW	244
11.1.2	Applet 的基本工作原理	247
11.1.3	Applet 类的主要方法	248
11.2	APPLET 与 HTML 文件配合使用	251
11.2.1	HTML 中的 Applet 标记	251
11.2.2	向 Applet 传递参数	253
11.3	JAVA 网络编程	255
11.3.1	TCP/IP 编程简介	255
11.3.2	Java 网络编程的基本机制	256
11.3.3	使用 PWS 架设 Web 服务器	259
11.4	线程	261
11.4.1	多线程的概念	261
11.4.2	多线程编程的实例	262
11.5	数据库接口	263
11.5.1	JDBC 简介	263
11.5.2	JDBC 编程实例	264
11.6	JAVA 多媒体编程	266
11.6.1	图像文件的装载及显示	266
11.6.2	声音文件的播放	267
11.6.3	Java 多媒体编程实例	268
习题 11		270
附录 1:	InputData. java	271
附录 2:	Console. java	272

第1章

Java 概述

知识要点:

- Java 的特征
- Java 程序的类型
- 简单的 Java 程序介绍
- 面向对象的程序设计概述
- Java 程序运行环境

本章介绍了 Java 的起源,发展的历史及其特点;Java 程序的基本结构;面向对象程序设计的基本概念;上机操作过程。

1.1 Java 的特征

自从 1995 年 5 月 23 日, Sun Microsystems 公司正式对外发布 Java 语言规范以来,其发展势不可挡,在 Internet 高速发展的今天显示出神奇的魅力。Java 具有简单、灵活和功能强大等优点。用 Java 设计的 Internet 上的应用程序,具有十分快速的实时性、精减性、安全性和完全的开放性。在 Sun 公司流传着这样一句话,Java 语言写一次可以跑一辈子。可以这么说,Internet 的发展孕育了 Java 语言,Java 对 Internet 的发展给予了最好的回报。

1989 年,在日内瓦出现了一种超文本标记语言——HTML,并创建了用于 HTML 文本的协议——HTTP 超文本传输协议。由此产生了 WWW 全球信息网。1993 年出现了轰动世界的 WWW 浏览器,一时间,WWW 便迅速盛行、风靡全球,Internet 也因此被了解、运用。随着多媒体技术的广泛应用,人们感觉到 WWW 上缺少了些什么,平静的页面上没有一点生机;突然 Java 出现了,WWW 完完全全地变了一个样,文字动了,图片动了,Web 网页充满了活力。Java 改变了 WWW,使 Internet 又向前迈进了一步。

Java 已不仅仅只是一种编程语言,而是正在形成一门技术,它包括以下几个方面。

- Java 软件:Java 语言及其编程技术、Java 操作系统 (Java OS)、Java 开发环境与开发工具、数据库接口 JDBC、Java 语言的软件包等等;
- Java 虚拟机及其规范;
- Java 嵌入技术:Java 芯片,基于 Java 技术的新型计算机系统 (网络计算机、基于网络计算的个人计算机、网络计算机体系结构、网络工作站等);
- Java 计算:基于 Java 处理功能的信息综合处理系统。

采用 Java 可以进行所有面向对象程序的开发,包括对象的事件描述、处理、综合等;计算机的可视化、可操作化的软件开发;动态画面的设计,包括图形、图像的调用;交互

操作的设计,包括选择交互、定向交互、控制流程;Internet 的系统管理功能模块(包括 Web 页)的动态设计管理和交互操作等;Intranet(企业内部网)的软件开发;与各类数据库连接查询的 SQL 语句的实现;其他应用的程序设计。

目前 Java 技术的应用领域已不仅仅在信息领域,而是涉及到家用电器(把 Java 芯片 PicoJava 嵌入到移动电话、WebTV 或 InternetTV、洗衣机等),企业的信息综合服务,科学计算,软件产业等等,显示出 Java 技术的广阔的应用前景,因此受到了各界用户的肯定与欢迎。

Java 具有如下特点。

1. 面向对象

Java 语言是在 C++ 这个面向对象程序设计语言的基础上发展形成的,扩充了 C++ 语言中的对象模型,除了 Java 语言的基本数据类型以外,其他一切都是以对象来表示的。Java 同时还删除了 C++ 语言的某些复杂的特性,Java 中没有结构联合,抛弃了多重继承,不提供运算重载,取消了指针和内存分配等。这样,使 Java 语言应用程序更易编写和掌握。

2. 跨平台

用 Java 语言写的程序,经 Java 编译生成字节码(Bytecode)后,可以在任何符合 Java 规范的 Java 虚拟机上运行,与操作系统无关,即用 Java 写的各种程序可在 Win9x/NT、Unix 等任何平台上运行。

Java 采用了一种半编译、半解释的方法,并定义出 Java 虚拟机概念。所谓虚拟机,就是一种并不存在而假设出来的平台。Java 源代码先经过 Java 编译器产生出 Java 虚拟机的机器码——字节码(Bytecode),再经 Java 解释器将字节码转换成实际系统平台上的机器码,去真正执行。

Java 编译出来的字节码与平台无关,这一点正是网络传输所需要的。由于支持 Java 的最主要的客户程序,即 WWW 浏览器内嵌入了 Java 解释器,所以 Java 字节码从服务器端转到客户端后,便会被客户端浏览器轻易解释成当前平台的机器码而运行,根本不用担心 Java 的字节码会由于服务器端的平台不同而编译出不同的目标码。

3. 高度安全

作为一种网络编程语言,其安全性是起决定性作用的。当用户用浏览器在 WWW 上漫游时,Java 小应用程序便会随时传到用户的机器上,所传输的不是一个静态的数据文件,而是活生生的可执行代码。

Java 的安全性是通过多个层次上的措施得到保证的。首先,从内存管理上加强安全,对程序员来说,内存分配是透明的,即程序员没有对内存的分配权,这样大大减少了内存出错的可能。另一方面,在 Java 中编译器也没有对内存布局的确定权,Java 把内存分配权交给运行平台。此外,类装载器也为安全性提供了措施。类装载器在装载字节码时,将本地的类组成一个统一的空间,而将外来的类组成另一个空间,这样为本地类建立了一种安全屏障。

4. 多线程

单线程应用程序只执行一个线程,一次只能做一件事,在它运行时,其他用户接口不会响应。多线程应用程序可以同时独立执行多个线程,这些线程可以交流或合作,如 Windows 95 中的 32 位执行程序,就可称为进程,Windows 95 运行后,就有多达十几个线

程在运行。Java 语言全面地支持多线程机制，并还能实现许多其他语言难以实现的数据同步化过程。

5. 交互性

Java 程序可以使 WWW 上的网页动起来。如果认为 Java 只能实现动画，那就大错特错了，动画只是 Java 功能的一个表面现象，实际上 Java 与众不同的关键之处是 Java 小应用程序 (Applet) 能被传输。Java 的字节码在客户端浏览器解释后，便成为通过网络传输的可执行代码。既然是一种可执行程序，就能在客户端与用户之间实现即时性交互。

1.2 Java 程序的类型

Java 程序分为三种类型：Java 应用程序 Application、Java 小应用程序 Applet 和 JavaScript。

Java 应用程序和其他应用程序几乎没有区别，它不是在浏览器中运行的，但它也必须得到 Java 虚拟机支持才能够运行。

Java 小应用程序则需要客户端 Web 浏览器解释运行的，运行之前必须先将其嵌入 HTML 文件的<applet></applet>标记中。当用户浏览该 HTML 页面时，Java 小应用程序将会从服务器端传送到客户端，从而被执行。这样就使得 WWW 页面增加了交互性和动态性。

Applet 和 Application 是 Java 程序的两种基本类型，从源代码的角度来看，Applet 和 Application 有两个基本的不同点：

- 1) 一个 Applet 必须定义一个从 Applet 类派生出的类，Application 则没有这个必要。
- 2) 一个 Application 必须定义一个包含 main 的方法，以控制它的执行，即程序的入口。而 Applet 不会用到 main 方法，它的执行是由 Applet 类中的几个方法来控制的。两者共同之处是：编程语法是完全一样的。

JavaScript 可以说是 Java 的子类，语法基本相同，它将程序的源代码直接嵌入到 HTML 文件的<SCRIPT></SCRIPT>标记中间。JavaScript 与 Java 小应用程序一样，也必须由客户端的 Web 浏览器解释运行。它是浏览器的脚本语言。Java 语言与 JavaScript 的根本区别是，Java 语言是一种面向对象的编程语言，也就是 OOP，其程序是由对象组成的，而 JavaScript 是浏览器的脚本语言。

Java 已成为最重要的 Internet 开发工具，Java 小程序可以很容易地插入到 Web 页里，从而产生 HTML 所无法生成的复杂 Web 页。

无论从哪方面说，Java 都可以说是最成功的新语言。由于 Sun Microsystems 公司是在 C++ 的基础上开发的 Java，所以 Java 语言被设计成面向对象的网络化编程语言。然而，Java 优于 C++，因为它包含了许多特征，例如自动垃圾收集、接口、异常处理以及固有多线程支持。这些新增特征可以使开发者更快地写出高质量的代码。

1.3 简单的 Java 程序介绍

通常，Java 的开发环境约定：当文件的扩展名为 java 时，则为 Java 语言源程序；而当文件的扩展名为 class 时，则为字节码程序，可以直接在 Java 虚拟机 (JVM) 上运行。

1.3.1 一个简单的 Java 程序

我们首先通过一个简单的例子来说明 Java 程序的基本结构及其主要的特点。

例 1-1 一个简单的 Java 程序。

/*源程序文件名为: Jex1_1.java*/	第 1 行
/** Java 程序的基本结构 */	第 2 行
import java.lang.System;	第 3 行
public class Jex1_1{	第 4 行
public static void main(String[] args){	第 5 行
int m=25;	第 6 行
System.out.println(m+"平方的输出是"+m*m);	第 7 行
}	第 8 行
}	第 9 行

该程序经编译后, 运行可执行程序时, 在显示器上显示:

25 平方的输出是 625

对这个程序的基本结构和各语句的作用说明以下几点:

1. 注解或说明信息

在 Java 程序的任何位置可以插入注解信息。可以有三种方法进行注解, 第一种是用“/*”和“*/”把注解信息括起来, 这种注解可以出现在程序中的任何位置, 如上例程序中的第 1 行。第二种是用“/**”和“*/”把注解信息括起来, 如上例程序中的第 2 行。第三种方法是用二个连续的“/”字符, 它表示从此开始到本行结束为注解, 如上例程序中的第 6、7 行。

2. 类文件

第 3 行使用关键字 import 引入了在程序外部定义的类 System, 它是在 java.lang 包中定义的, 因而引入时必须指出完整的路径。第 4 行使用了关键字 class 来声明将要定义的新类, Jex1_1 是类的名称。整个类的定义, 包括代码和数据都在本行的开括号“{”和与之对应的第 7 行上的闭括号“}”之间。

3. 主方法 main()

任何一个 Java 应用程序 (Application) 均有且只能有一个 main() 方法。一个 Java 程序总是从 main() 方法开始执行, 而不管该函数在整个程序中的位置在什么地方。

4. 花括号对 {}

{ } 称为方法或语句括号。任一方法体均以“{”开始, 并以“}”结束。注意, 花括号要配对使用。

5. 方法和语句

任一 Java 程序均由一个或多个方法组成。其中必须有一个主方法 main(), 其余方法可有可无。任一方法可有若干个语句, 每一个语句均以“;”结束。

6. 程序的书写规则

对于 Java 的编译器而言, 一个语句可以写成若干行, 一行内也可以写若干个语句。虽

然 Java 允许的书写格式是非常自由的,但是为了便于阅读和相互交流,程序的书写必须符合以下规则。

- 对齐规则:同一层次的语句必须从同一列开始,同一层次的开花括号必须与对应的闭花括号在同一列上。
- 缩进规则:属于内一层次的语句,必须缩进几个字符,通常缩进 2 个、4 个或 8 个字符的位置。此外,任一函数的定义均从第一列开始书写。

7. 输入和输出

Java 语言没有专门的输入输出语句,输入和输出是通过方法来实现的。如 `System.out.println()` 为输出方法。

8. 严格区分大小写字母

某些高级语言不区分大小写字母,但 Java 语言是严格区分大小写字母的。如: `A` 与 `a` 表示两个不同的标识符。在书写程序或编辑程序时要注意到这一点。

1.3.2 程序的基本要求

用 Java 语言进行程序设计并解决实际问题时,对程序是有质量要求的。通过本课程的学习,对设计的程序要达到以下几方面的基本要求。

1. 程序的正确性

程序的正确性包括两个方面:语法和语义正确;算法描述正确。这是对程序的最基本的要求。

2. 程序的可读性和可理解性好

当设计的程序被阅读时,要做到容易读懂,并且容易理解程序的设计思想和设计方法。这通常也包括三个方面:首先是程序的结构性好,要求采用结构化的程序设计方法或采用软件工程的程序设计方法来设计程序;第二是在程序中增加注解,说明程序的设计思想和方法;第三是程序的书写格式规范。

3. 程序的可维护性好

程序易于修改,易于增加新的功能。这要求程序的结构性好,各模块的独立性强。

4. 程序的构思好,程序简短,执行速度快

前面的三点是最基本的要求,而最后一点的要求略高一点,也应尽力做到。必须指出,要设计出高质量的程序,仅学习本课程的知识是不够的,还应具有数据结构、算法设计与分析、软件工程及程序设计方法学等方面的知识。

1.3.3 Java 程序的开发步骤

对于一个高级语言编写的源程序而言,编译是指将其翻译成可运行的目标程序,目标程序作为一个独立的文件可以无数次地运行。编译过程所需要的存储空间较大,同时编译所需要的时间较长,但目标程序执行速度快,这种方式特别适用于重复使用的程序。而解释则是指对高级语言编写的源程序翻译一句执行一句,翻译和运行交叉进行,如果要再运行一次,那就必须重新翻译、重新执行。解释型语言适用于需要经常修改程序的情况。由于边翻译边执行,解释型语言的执行速度远远低于编译型语言。

Java 是一种半编译半解释的语言。一个 Java 程序的执行过程是:先由 Java 编译器将

其编译成字节码,运行时由运行系统上的字节码解释器将其翻译成机器语言。Java 字节码解释器的工作是边翻译边执行,但因为字节码已非常接近于机器码,所以速度仍相当快,而且在一个系统上编写并编译好的 Java 程序字节码可以不加任何变动地放到另一系统上运行。

设计好一个 Java 源程序后,需要经过编译,生成可在 Java 虚拟机上可执行的程序文件,然后执行并调试程序。Java 程序的开发步骤如下:

- 1) 根据要解决的问题,分析需求,并用合适的方法描述之。
- 2) 编写 Java 源程序,并利用一个编辑器将源程序输入到计算机的某一个文件中。文件的扩展名为 java。
- 3) 编译源程序,并产生字节码文件。文件的扩展名为 class。
- 4) 调试程序。运行可执行文件,分析运行结果。若结果不正确,则要修改源程序,并重复以上的过程,直到得到正确的结果为止。

Java 程序的开发、执行流程如图 1-1 所示。

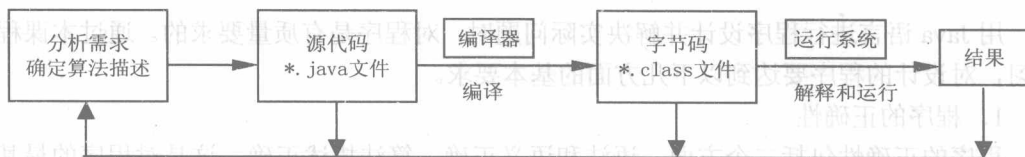


图 1-1 Java 程序的开发、执行流程

1.4 面向对象的程序设计概述

面向对象的程序设计 (Object Oriented Programming, 缩写为 OOP) 是目前占据主流地位的一种程序设计技术,它试图用客观世界中描述事物的方法来描述一个程序要解决的问题。Java 通过引入类(Class)、对象(Object)、消息(Message)、继承(Inherit)等概念,增加了程序模块的独立性和可扩展性,为软件开发人员和软件产品提供了很多便利。

1.4.1 对象及面向对象的程序设计

对象 (Object) 是 OOP 中最重要的概念之一。对象是一个抽象的概念,简单地说,它是对一个客观实体的描述,是既有数据又有对数据进行操作的代码的一个逻辑实体。

由此,面向对象的程序设计,就是用对象来描述客观世界中所需说明的有关事物。

1.4.2 面向对象程序设计的要素

Java 是一种面向对象程序设计语言,它支持面向对象程序设计的几个要素:封装性、继承和派生性、重载性和多态性。下面对这些要素分别作简单介绍。

1. 封装性

将描述对象的数据及对这些数据进行处理程序代码有机地组成一个整体,形成一个模块,对其数据及代码的存取权限加以限制后,模块完全独立。对象的这种特性称为封装性。这种特性使得描述对象的数据只能通过对象中的程序代码来处理,而其他任何程序代