

21世纪电脑学校



Java程序设计 实用教程

张大治 编著



清华大学出版社

Java 程序设计实用教程

张 大 治 编 著

清华大学出版社

北 京

内 容 简 介

本书结合大量典型的实例,全面而翔实地介绍了 Java 程序设计语言的编程技巧和面向对象编程思想。本书的范例基于最新的 JDK 1.6 版本,内容包括 Java 的基本语法、面向对象的编程思想、Java 异常处理机制、基本输入输出、Java GUI 编程、Java 多线程编程、Java 多媒体应用、Java 数据库编程、Java 网络通信等。书中精选的大量实例都来自实际的工程项目,具有一定的实用性和参考性,广大读者能够从中受益。

本书适合于 Java 初中级读者阅读,可作为高等院校学生的教材,也可供有一定 Java 基础或正在转向进行 Java 开发的各类程序员阅读参考。

本书的电子教案和实例源代码可以到 <http://www.tupwk.com.cn/21cn> 网站下载。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

Java 程序设计实用教程/张大治 编著. —北京:清华大学出版社,2008.4

(21 世纪电脑学校)

ISBN 978-7-302-17184-3

I. J… II. 张… III. JAVA 语言—程序设计—教材 IV. TP312

中国版本图书馆 CIP 数据核字(2008)第 031398 号

责任编辑:胡辰浩(huchenhao@263.net) 袁建华

装帧设计:孔祥丰

责任校对:成凤进

责任印制:李红英

出版发行:清华大学出版社

<http://www.tup.com.cn>

社 总 机:010-62770175

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

地 址:北京清华大学学研大厦 A 座

邮 编:100084

邮 购:010-62786544

印 刷 者:清华大学印刷厂

装 订 者:北京市密云县京文制本装订厂

经 销:全国新华书店

开 本:185×260 印 张:25.5 字 数:653 千字

版 次:2008 年 4 月第 1 版 印 次:2008 年 4 月第 1 次印刷

印 数:1~5000

定 价:38.00 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:(010)62770177 转 3103 产品编号:027620-01

编审委员会

主任：郭 军 北京邮电大学信息工程学院教授

委员：(以下编委顺序没有先后，按照姓氏笔画排列)

王相林 杭州电子科技大学教授

王常吉 中山大学计算机科学系教授

王锁萍 南京邮电大学吴江职业学院教授

闪四清 北京航空航天大学教授

张孝强 南京邮电大学教授

张宗橙 南京邮电大学传媒技术学院教授

杜云贵 长城电脑学校教师

杜耀刚 北京电子科技学院基础部教授

赵树升 郑州大学升达经贸管理学院教授

郭清宇 中原工学院计算机系教授

崔洪斌 河北科技大学教授

焦金生 《计算机教育》杂志社总编

执行委员：许书明 胡辰浩 李万红 荣春献

执行编辑：胡辰浩 袁建华

丛 书 序

清华大学出版社

出版目的

电脑作为一种工具，已经广泛地应用到现代社会的各个领域，正在改变着各行各业的生产方式以及人们的生活方式。在进入新世纪之后，需要掌握更多的电脑应用技能。因此，如何快速地掌握电脑知识和使用技术，并应用于现实生活和实际工作中，成为新世纪人才迫切需要解决的新问题。

为适应这种需求，各类高等院校、高职高专、中职中专、培训学校都开设了计算机专业的课程，另外，许多学校也将非计算机专业学生的电脑知识和技能教育纳入教学计划，并陆续出台了相应的教学大纲。基于以上因素，清华大学出版社组织了一批教学精英编写了一套“21 世纪电脑学校”教材，以满足各类培训学校教学和电脑知识自学人员的需要。本套教材的作者均为各大院校或培训机构的专家和老师，他们熟悉教学内容的编排，深谙学生的需求和接受能力，积累了丰富的授课和写作经验，并将其充分融入本套教材的编写中。

读者定位

本丛书是为从事电脑教学的教师 and 自学人员编写的，可用作各类培训机构和院校的教材，也可作为电脑初、中级用户的自学参考书。

涵盖领域

本套教材涵盖了计算机多个应用领域，包括计算机硬件知识、操作系统、数据库、编程语言、文字录入和排版、办公软件、计算机网络、图形图像、三维动画、网页制作、多媒体制作等。众多的图书品种，可以满足不同读者、不同电脑课程设置的需要。

本丛书选用应用面最广的流行软件，对每个软件的讲解都从必备的基础知识和基本操作开始，使新用户轻松入门，并以大量明晰的操作步骤和典型的应用实例向读者介绍实用的软件技术和应用技巧，使读者真正对所学软件融会贯通、熟练在手。

丛书特色

一、更为合理的学习过程

- 1、章节结构按照教学大纲的要求编排，符合教学需要和电脑用户的学习习惯。
- 2、细化了每一章内容的分布。在每章的开始，有教学目标和理论指导，便于教师和学生提纲挈领地掌握本章知识的重点，每章的最后附带有上机实验、思考练习，读者不但可以锻炼实际的操作能力，还可以复习本章的内容，加深对所学知识的了解。

二、简练流畅的语言表述

语言精炼实用，避开深奥的原理，从最基本最易操作的内容入手，循序渐进地介绍学习电脑应用最需要的内容。

三、丰富实用的示例

以详细、直观的步骤讲解相关操作，每本图书都包含众多精彩示例。现在的计算机教学更加注重实际的动手操作，学校在教学过程中，有很多的课时用于进行实际的上机操作。因此，本丛书非常注意实例的选材，所选实例都具有较强的代表性。

四、简洁大方的版式设计

精心设计的版式简洁、大方，对于标题、正文、注释、技巧等都设计了醒目的字体，读者阅读起来会感到轻松愉快。

周到体贴的售后服务

本丛书紧密结合自学与课堂教学的特点，针对广大初、中级读者电脑基础知识薄弱的现状，突出基础知识和实践指导方面的内容。每本教材配套的实例源文件、素材和教学课件均可在该丛书的信息支持网站 (<http://www.tupwk.com.cn/21cn>) 上下载或通过 Email(wkservice@tup.tsinghua.edu.cn) 索取。读者在使用过程中如遇到困难可以在 <http://www.tupwk.com.cn/21cn> 的互动论坛上留言，本丛书的作者或技术编辑会提供相应的技术支持。

前 言

随着网络技术应用的飞速发展,Java 作为最优秀的网络编程语言之一也在不断扩展和改进,以适应最新的软、硬件编程环境。为了让更多的 Java 爱好者掌握最新的 Java 技术,我们精选了大量实例,编写了本书。

在编写本书时,我们充分考虑到初学者学习程序语言的特点,在对基本知识的讲解上,力求做到深入细致,一步一步地向读者揭开 Java 的神秘面纱。

全书共分 14 章,从 Java 的基础知识讲起,由浅入深、循序渐进地介绍了 Java 程序设计语言的编程技巧。第 1 章介绍了 Java 语言的发展史、特点,Java 开发环境的建立等内容。第 2 章详细介绍了 Java 的基本语法。第 3 章讲述了面向对象的编程思想和方法。通过前 3 章的学习,使读者对面向对象编程设计思想及 Java 语言有初步的认识。第 4 章介绍了 Java 的异常处理机制。第 5 章介绍了 Java 的数组和字符串处理。第 6 章介绍了流的概念以及 Java 的输入输出和文件操作。第 7 章讲述了图形用户界面(GUI)的知识和 Java 的事件处理技术。第 8 章主要介绍了 Java 多媒体编程技术,包括绘制图形、图像处理和 Java 动画技巧。第 9 章介绍了 Java 小应用程序 Applet 和 JApplet。第 10 章讲述了 Java 多线程程序的开发。第 11 章通过一个简单的实例讲述了 Java 中的对象序列化。第 12 章介绍 Java 的网络编程技术,包括 Java 网络编程的基础知识、套接字编程、数据报通信等内容。第 13 章介绍了 Java 的数据库编程,主要讲述 JDBC 连接和访问数据库的基本步骤。第 14 章是综合应用实例,综合应用本书所述内容开发了一个网络聊天室程序,希望读者通过这个综合实例的学习,达到知识点融会贯通的效果。

本书内容丰富、结构清晰、通俗易懂、图文并茂,所选实例具有广泛的实用性和代表性,读者稍加修饰和完善就是实用软件或程序。全书力求语言生动活泼,示例典型实用,讲解全面细致,希望能给广大 Java 爱好者提供尽可能多的帮助。书中所有程序都基于 Sun 公司的最新 JDK 版本 1.6 编译执行。

本书适合于专业和准专业的 Java 程序员阅读,可作为各类社会培训机构及高等院校的教材,也是正在转向进行 Java 开发的各类程序员的必备 Java 参考书。

本书是多人智慧的结晶,除封面署名的作者外,参与资料整理和制作的人员还有王强、周惠、杨玉敏、张凤霞、贺宝江、宋军山、王燕、张挂云、陈建兵、张兴武、李志伟、沈世莉、刘福山、王瑞、郭文华、王伟芳、张廷宪、刘开颜等。由于作者水平有限,本书不足之处在所难免,欢迎广大读者批评指正。在本书的编写过程中,参考了一些有关文献,在此向这些文献的作者深表感谢。我们的信箱是: huchenhao@263.net。

作 者

2008 年 2 月

目

第1章 Java 基础知识	1
1.1 Java 语言简介	1
1.1.1 Java 发展史	2
1.1.2 Java 语言的特点	2
1.1.3 Java 的基本元素	3
1.2 面向对象编程	4
1.2.1 面向过程与面向对象	4
1.2.2 客观事物的抽象	5
1.2.3 对象和封装	5
1.2.4 类和继承	6
1.3 建立 Java 开发环境	6
1.3.1 下载并安装 JDK 6	6
1.3.2 设置系统环境变量	8
1.4 JDK 开发环境工具简介	9
1.4.1 Java 编译器(javac.exe)	9
1.4.2 Java 解释器(java.exe)	10
1.4.3 Applet 浏览器 (AppletViewer.exe)	11
1.4.4 API 文件产生器(javadoc.exe)	11
1.4.5 文件归档工具(jar.exe)	12
1.4.6 头文件生成器(javah.exe)	14
1.4.7 反编译器(javap.exe)	14
1.4.8 密钥管理工具(Keytool.exe)	15
1.4.9 转码工具(native2ascii.exe)	16
1.5 编译执行第一个 Java 程序	17
1.5.1 Java 应用程序	18
1.5.2 Java Applet	18
1.6 本章小结	20
1.7 思考和练习	20
第2章 Java 基本语法	21
2.1 标识符和关键字	21
2.1.1 标识符	21

录

2.1.2 关键字	22
2.2 数据类型、常量和变量	23
2.2.1 数据类型	23
2.2.2 常量	24
2.2.3 变量	25
2.2.4 赋值和数据类型转换	28
2.3 运算符和表达式	30
2.3.1 算术运算符和算术表达式	30
2.3.2 关系运算符和关系表达式	33
2.3.3 逻辑运算符和逻辑表达式	35
2.3.4 位运算符和位表达式	36
2.3.5 赋值运算符和赋值表达式	38
2.3.6 条件运算符和条件表达式	38
2.3.7 其他运算符	39
2.3.8 运算符的优先级与结合性	39
2.4 流程控制语句	40
2.4.1 块作用域	40
2.4.2 分支语句	41
2.4.3 循环语句	45
2.4.4 转移语句	50
2.4.5 递归(recursion)	52
2.4.6 注释语句	54
2.5 本章小结	55
2.6 思考和练习	55
第3章 面向对象编程	57
3.1 类	57
3.1.1 类的声明	57
3.1.2 成员变量	59
3.1.3 成员方法	59
3.1.4 构造方法	60
3.1.5 this 关键字	61
3.1.6 main()方法的特点	62
3.2 访问修饰符	63



3.2.1 访问控制	63	4.3.2 异常捕获	105
3.2.2 static	67	4.3.3 异常的嵌套处理	109
3.2.3 final 修饰符	70	4.4 用户自定义异常	110
3.2.4 抽象类和抽象方法	72	4.5 本章小结	111
3.2.5 内部类	73	4.6 思考和练习	112
3.3 对象创建、使用和清除	78	第5章 数组和字符串类型	113
3.3.1 对象的创建和使用	78	5.1 一维数组	113
3.3.2 匿名对象	78	5.1.1 一维数组的声明	114
3.3.3 对象的克隆	79	5.1.2 一维数组初始化	114
3.3.4 用 finalize 方法清除对象	81	5.1.3 访问数组元素	116
3.3.5 对象的比较	81	5.2 多维数组	118
3.4 类的继承与多态性	83	5.2.1 二维数组的定义与初始化	118
3.4.1 继承	83	5.2.2 访问二维数组元素	119
3.4.2 匿名内部类	84	5.3 字符串类型	121
3.4.3 方法覆盖	86	5.3.1 定义字符串	121
3.4.4 方法重载	87	5.3.2 访问和修改字符串	123
3.4.5 super 关键字	88	5.4 使用 StringTokenizer 类	132
3.4.6 多态性	90	5.5 字符串与其他数据类型的转换	133
3.5 接口	92	5.5.1 其他对象转换为字符串	133
3.5.1 接口的定义	92	5.5.2 字符串转换为其他对象	134
3.5.2 接口的实现	93	5.6 本章小结	135
3.6 包	95	5.7 思考和练习	135
3.6.1 包声明	95	第6章 基本 I/O 及文件操作	137
3.6.2 import 语句	95	6.1 流	137
3.6.3 JDK 中常用的包	97	6.1.1 流的概念	137
3.7 本章小结	98	6.1.2 输入输出流	138
3.8 思考和练习	98	6.1.3 Java 中输入输出处理类的层次结构	140
第4章 异常处理	99	6.2 File 类	141
4.1 Java 异常处理概述	99	6.2.1 创建 File 类的对象	142
4.1.1 什么是异常	99	6.2.2 File 类的方法	143
4.1.2 异常处理范例	100	6.3 使用 Reader 类与 Writer 类	145
4.2 异常处理类	102	6.3.1 FileReader 类和 FileWriter 类	146
4.2.1 异常处理类的层次	102	6.3.2 BufferedReader 类和 BufferedWriter 类	148
4.2.2 异常类 Exception	103		
4.3 异常处理	103		
4.3.1 抛出异常	104		



6.3.3	LineNumberReader 类	150
6.4	RandomAccessFile 类	151
6.4.1	创建 RandomAccessFile 对象	151
6.4.2	RandomAccessFile 类的方法	152
6.5	InputStream 类和 OutputStream 类	155
6.5.1	FileInputStream 类和 FileOutputStream 类	155
6.5.2	PipedInputStream 类和 PipedOutputStream 类	158
6.5.3	SequenceInputStream 类	161
6.6	过滤流	163
6.6.1	BufferedInputStream 和 BufferedOutputStream	164
6.6.2	DataInputStream 和 DataOutputStream	165
6.7	本章小结	167
6.8	思考和练习	168
第 7 章	GUI 编程	169
7.1	GUI	169
7.1.1	GUI 概述	170
7.1.2	创建 GUI 程序	171
7.1.3	AWT 包	173
7.2	Java 事件	174
7.2.1	事件处理模型	174
7.2.2	鼠标和键盘事件	177
7.2.3	窗口事件	183
7.3	容器组件	184
7.3.1	Container 类	184
7.3.2	Panel	185
7.3.3	Frame	185
7.3.4	Dialog	185
7.3.5	ScrollPane 类	187
7.4	AWT 包中的常用组件	188
7.4.1	Label	188
7.4.2	Button	189

7.4.3	Checkbox	190
7.4.4	Scrollbar	192
7.4.5	菜单组件	193
7.5	布局管理器	195
7.5.1	FlowLayout	196
7.5.2	BorderLayout	198
7.5.3	CardLayout	198
7.5.4	GridLayout	200
7.5.5	GridBagLayout	201
7.5.6	Swing 的布局管理器	204
7.6	Swing 基本组件	207
7.6.1	JFrame	207
7.6.2	JLabel	208
7.6.3	JButton、JCheckBox 和 JRadioButton	209
7.6.4	JTextField、JPasswordField 和 JTextArea	213
7.6.5	JComboBox	214
7.6.6	JList	215
7.7	Swing 高级组件	219
7.7.1	面板容器	219
7.7.2	JDialog	223
7.7.3	菜单和工具条	225
7.7.4	JProgressBar 组件	229
7.7.5	JFileChooser	231
7.7.6	JTable	233
7.7.7	JTree	235
7.8	本章小结	236
7.9	思考和练习	236
第 8 章	Java 多媒体编程	237
8.1	Java 绘图概述	237
8.1.1	JDK 中的图形类	237
8.1.2	图形直角坐标系	238
8.1.3	Font 类和 Color 类	238
8.2	简单图形处理	240
8.2.1	绘制规则图形	240



8.2.2	绘制多边形和折线	242
8.2.3	复制图形	244
8.2.4	显示文本	245
8.3	Java2D	246
8.3.1	Java2D 图形	246
8.3.2	高级绘图效果	247
8.4	XOR 绘图模式和打印图形	250
8.4.1	XOR 绘图模式	250
8.4.2	打印图形	251
8.5	图像处理	253
8.5.1	装载和显示图像	253
8.5.2	生成图像	254
8.5.3	图像过滤	255
8.5.4	图像跟踪器 MediaTracker	257
8.6	Java 实现动画	259
8.6.1	动画技巧	259
8.6.2	动画实例	259
8.7	本章小结	262
8.8	思考和练习	262
第9章	Java Applet	263
9.1	Java Applet 简介	263
9.1.1	Applet 的工作原理	263
9.1.2	Applet 类及其方法	264
9.2	Applet 的使用技巧	269
9.2.1	设置 Applet 的颜色	269
9.2.2	获取 Applet 的显示 空间大小	271
9.2.3	Applet 标记与参数传递	272
9.2.4	Applet 和浏览器间通信	274
9.2.5	显示状态信息	276
9.2.6	带 main 方法的 Applet	276
9.3	使用 JApplet	284
9.4	本章小结	287
9.5	思考和练习	287

第10章	Java 多线程处理	289
10.1	多线程简介	289
10.1.1	进程与线程	289
10.1.2	Java 的多线程	290
10.2	Java 线程类与 Runnable 接口	291
10.2.1	Thread 类	291
10.2.2	Runnable 接口	292
10.2.3	如何在程序中 实现多线程	293
10.3	线程管理	296
10.3.1	线程生命周期	296
10.3.2	线程调度与优先级	297
10.3.3	守护(Daemon)线程	298
10.4	线程同步与线程间通信	300
10.4.1	同步代码块	301
10.4.2	同步方法	302
10.4.3	线程间通信	303
10.5	线程组	306
10.6	本章小结	308
10.7	思考和练习	308
第11章	对象序列化	309
11.1	序列化概述	309
11.1.1	问题的引出	309
11.1.2	序列化机制	310
11.1.3	序列化的实现	310
11.2	控制序列化	313
11.2.1	处理对象流	313
11.2.2	定制序列化过程	314
11.2.3	完全定制序列化过程	316
11.2.4	可序列化类的不同 版本的序列化兼容性	318
11.3	序列化应用实例	318
11.4	本章小结	321
11.5	思考和练习	322



第 12 章 网络编程	323	13.2 JDBC 中主要接口和类	350
12.1 网络编程的基础知识	323	13.2.1 JDBC 4.0API 包	351
12.1.1 Internet 概述	323	13.2.2 DriverManager 类	352
12.1.2 IP 地址和端口号	324	13.2.3 Connection 接口	353
12.1.3 TCP 和 UDP	325	13.2.4 Statement 接口	354
12.1.4 Socket	325	13.3 数据库和 SQL 语言基础	356
12.2 Java 中的网络基础类	326	13.3.1 数据库的基本概念	357
12.2.1 URL 类	326	13.3.2 结构化查询语言—SQL	358
12.2.2 URLConnection 类	329	13.4 使用 JDBC 访问数据库	362
12.2.3 InetAddress 类	330	13.4.1 创建数据源	362
12.3 Socket 通信	331	13.4.2 与数据库建立连接	363
12.3.1 Socket 类和 ServerSocket 类	332	13.4.3 执行查询、处理查询结果	364
12.3.2 Socket 通信实例	333	13.5 本章小结	367
12.3.3 支持多客户连接的 Socket 通信方式	337	13.6 思考和练习	367
12.4 数据报通信	340	第 14 章 综合应用	369
12.4.1 DatagramSocket 类和 DatagramPacket 类	340	14.1 网络聊天室	369
12.4.2 数据报通信实例	342	14.1.1 服务器程序	369
12.4.3 MulticastSocket 类	344	14.1.2 客户端程序	378
12.5 本章小结	346	14.1.3 运行程序	390
12.6 思考和练习	346	14.2 本章小结	391
第 13 章 Java 数据库编程	347	14.3 思考和练习	391
13.1 什么是 JDBC	347	参考文献	393
13.1.1 JDBC 概述	348		
13.1.2 JDBC 驱动程序 的 4 种类型	349		

第1章

Java基础知识

本章导读

Java 是一种基于 C 和 C++语法的纯面向对象的编程语言。本章首先向读者介绍了 Java 语言的发展史、Java 语言的特点、Java 的基本元素以及什么是面向对象编程，使读者对 Java 语言有个基本认识，然后介绍了如何建立 Java 开发环境并对 JDK 开发环境中的常用工具做了简要说明，最后，通过经典的 HelloWorld 程序讲述开发 Java 应用程序的基本流程。通过本章的学习，读者将对 Java 语言有一个概括性的认识，理解什么是面向对象编程，并能在自己的机器上建立 Java 开发环境，编写简单的 HelloWorld 程序。

重点和难点

- Java 语言的特点
- Java 语言的基本元素
- 面向对象编程
- 建立 Java 开发环境
- JDK 常用工具
- 编译并执行 HelloWorld 程序

1.1 Java 语言简介

计算机所使用的是由“0”和“1”组成的二进制数，对计算机的每一种操作，都是使用一个唯一的二进制组合进行定义，该二进制组合称为操作码，也称为机器语言，它们是计算机能够识别和执行的实际语言。

遗憾的是，人类直接处理机器语言比较困难。为了减轻使用机器语言编程的痛苦，人们进行了一种有益的改进：用一些简洁的英文字母、符号串来替代一个特定的指令的二进制串，这种程序设计语言就是汇编语言。

由于汇编语言程序依赖于计算机硬件，其可读性和可移植性都很差；所以后来又出现了很多高级语言。当今常用的高级计算机语言主要有：Ada、Basic、C、C++、COBOL、Fortran、



Java 和 Pascal。其中 Java 是最新的一种语言，它吸收了前面几种语言的精华，并加入了最新的软件工程思想。

1.1.1 Java 发展史

Java 来自于 Sun 公司的一个叫 Green 的项目，其最初的目的是为家用消费电子产品开发一个分布式代码系统，这样我们可以把 E-mail 发给电冰箱、电视机等家用电器，对它们进行控制，和它们进行信息交流。开始的时候研究人员准备采用 C++，但 C++ 太复杂、安全性差，1991 年，Sun 公司的 Jame Gosling、Bill Joe 等人开发了一种新的语言 Oak(Java 的前身)，Oak 是一种用于网络的精巧而安全的语言，Sun 公司曾依此投标一个交互式电视项目，但结果被 SGI 打败了。可怜的 Oak 几乎无家可归，恰巧这时 Mark Ardreesen 开发的 Mosaic 和 Netscape 启发了 Oak 项目组成员，他们用 Java 编制了 HotJava 浏览器，得到了 Sun 公司首席执行官 Scott McNealy 的支持，从而触发了 Java 进军 Internet。

关于 Java 的命名也有一则趣闻，有一天，几位 Java 成员组的会员正在讨论给这个新的语言取什么名字，当时他们正在咖啡馆喝着 Java(爪哇)咖啡，有一个人灵机一动说就叫 Java 并得到了其他人的赞赏，于是，Java 这个名字就这样传开了。

1.1.2 Java 语言的特点

Java 是一种基于 C 和 C++语法的纯面向对象的编程语言，但消除了 C 和 C++语法中大多数易混淆的特征。根据 Sun 公司的“Java 白皮书”中对 Java 的定义，Java 是一种简单、面向对象、分布式、解释性、健壮、安全、结构中立、可移植、高性能、多线程和动态的语言，具有如下特点：

- 简单性：Java 语言是一种面向对象的语言，类似于 C++，因此，C++程序员可以很快就掌握 Java 编程技术。Java 摒弃了 C++中容易引发程序错误的地方，如指针和内存管理。Java 提供了丰富的类库，使得编程比较容易。
- 面向对象：Java 语言的设计集中于对象及其接口，它提供了简单的类机制以及动态的接口模型，是一种纯面向对象的编程语言。
- 分布式：Java 包括一个支持 HTTP 和 FTP 等基于 TCP/IP 协议的子库。因此，Java 应用程序可凭借 URL 打开并访问网络上的对象，其访问方式与访问本地文件系统几乎完全相同。为分布环境尤其是 Internet 提供动态内容无疑是一项非常宏伟的任务，但 Java 的语法特性却使我们很容易地实现这项目标。
- 解释性：Java 解释器(运行系统)能直接运行目标代码指令。链接程序通常比编译程序所需资源少，所以程序员可以在创建源程序上花上更多的时间。
- 健壮性：通过集成的面向对象的例外处理机制，在编译时，Java 会提示出可能出现但未被处理的例外，帮助程序员正确地进行选择以防止系统的崩溃。



- **安全性:** Java 的安全性可以从两个方面得到保证。一方面 Java 语言不支持指针和释放内存等 C++ 的功能,这样就避免了非法内存操作;另一方面,类装载通过将本机类与网络资源类的名称分离开来保持安全性。
- **多线程:** Java 提供多线程功能使得在一个程序里可同时执行多个小任务,而且同步机制保证了对共享数据的正确操作。通过使用多线程可以很容易的实现网络上的实时交互行为。
- **动态性:** Java 的动态特性是其面向对象设计方法的扩展。它允许程序动态地装入运行过程中所需要的类,这是 C++ 语言进行面向对象程序设计所无法实现的。而且 Java 通过接口来支持多继承,使之比严格的类继承具有更灵活的方式和扩展性。
- **结构中立:** 为了让 Java 与网络融为一个整体,Java 将它的程序编译成一种结构中立的中间文件格式,只要有 Java 运行系统的机器都能执行这种中间代码。
- **可移植性:** 与平台无关的特性使得 Java 应用程序可以方便的移植到网络上的不同机器。同时 Java 类库中也实现了与不同平台的接口,使得这些类库可以移植。这种“编写一次,任何地方均可使用”的哲学意味着一个组织不会被一种特定类型的计算机硬件所锁定。
- **高性能:** 如果解释器速度不慢,Java 可以在运行时直接将目标代码翻译成机器指令,从而得到较高的性能。

当然 Java 也有一些不足之处。其中之一是它缺乏其他语言所具有的一些重要特征。例如, Fortran、C 和 C++ 等语言具有简单的输出语句,允许程序员以一种特定的格式显示数值数据,并容易制定小数点后数字的长度和个数,从而产生输出数据表,这在许多工程和科学应用中非常重要,而在标准 Java 语言中则不包含此功能。

另外,Java 的运行速度相对较慢。当每次使用编译后的 Java 程序时,它是被一个实时(JIT)编译器解释或编译执行。因此 Java 应用程序的执行速度比本机语言(如 C++)程序要慢得多。

1.1.3 Java 的基本元素

Java 由 3 种独立的基本元素组成: Java 编程语言, Java 运行时环境, Java 应用编程接口(API)。

Java 与其他计算机语言不同的是,所有的 Java 程序都在一种特定类型的计算机上编译执行,称为 Java 虚拟机(Java VM)。Java VM 是一种具有特定机器语言(字节码)的计算机。之所以称为“虚拟”,是因为这种计算机实际上是不存在的,它是由一种称为 Java 运行环境(JRE)的特定程序模拟的。人们已经开发了一个版本的 JRE,该 JRE 适合于每一台计算机和众多的操作系统(如 Windows、Sun Solaris、Linux、Macintosh 等)。所有的 Java 编译器都产生字节码,字节码能够直接在任何 Java VM 上运行而不管该计算机是什么类型的,这使得 Java 程序真正做到“编写一次,任何地方均可使用”。

Java 运行环境包含了在一台给定计算机上运行 Java 的所有必要支持。当 Java 程序运行



时, JRE 装载并验证字节码以确保程序的有效性, 然后运行一个称为 Java 解释器(也可能是一个实时编译器)的特定程序, 将 Java VM 字节码转换为特定计算机能够识别的机器语言。对于一种给定类型的计算机, 只要编写了针对该计算机的 Java 解释器, 该计算机就能够运行任何 Java 程序。

编译和执行 Java 程序的过程如图 1-1 所示。可以使用任何文本编辑器编辑 Java 程序并存储为一个磁盘文件, 文件的扩展名为“.java”(例如, MyProg.java)。Java 编译器将这个程序编译为能够在 Java VM 上执行的字节码, 并将字节码存储为一个文件, 文件的扩展名为“.class”(本例中是 MyProg.class)。当程序在计算机上执行时, Java 解释器实时地将 Java VM 字节码转换为计算机能够执行的指令。编译过程仅执行一次, 解释过程在每次运行 Java 程序时都要执行。Java 字节码独立于任何特定的计算机硬件, 因此任何具有 Java 解释器的计算机都能够执行编译过的 Java 程序而不管程序编译时的计算机是什么类型。

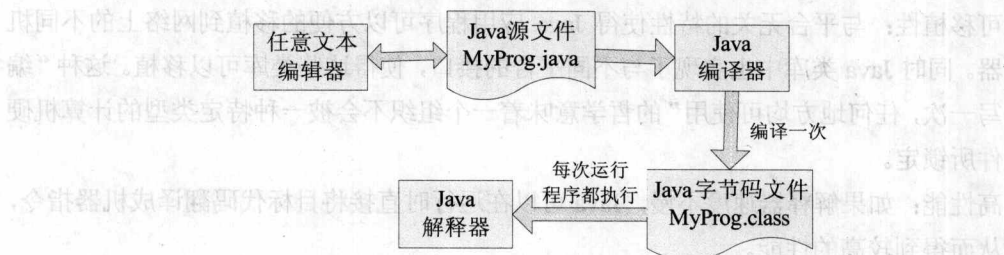


图 1-1 Java 程序编译和执行过程

Java 应用编程接口是一个非常大的软件组件集合, 这些组件提供了读写文件、操作字符串、绘图以及其他许多必要的功能。Java API 中相关的组件被组合成库(或称为包)。通过使用这些标准包中的对象, 程序员能够节省大量的时间。所有的 Java 实现都支持这些包中的组件, 因此使用组件的程序能够在任何支持 Java 的计算机系统中正确运行。另外, 组件是经过调试的, 因此使用组件将减少编写和调试程序的工作量。

1.2 面向对象编程

Java 是面向对象的编程语言, 因此要理解 Java 语言的编程技术就必须了解面向对象的编程技术。

1.2.1 面向过程与面向对象

编程的主要目的是为了实现现实世界的某些功能, 因此, 传统的高级程序语言, 如 Basic、C、Pascal 等, 很容易采用面向过程的编程方法。面向过程的程序设计方法从解决问题的每一个步骤入手, 适合解决比较小的简单问题。当编写比较大的应用程序时, 使用面向过程的编程方法设计就会变得很困难, 这主要是因为面向过程的处理模型是数据驱动的, 与现实世界



模型具有不一致性。在现实世界模型中,对象是基本的组成单元,对象具有状态和行为,这两者紧密相关,面向过程的编程方法就是将两者分开处理,而不考虑这两者的相互关系。

面向对象的程序设计就是按照现实世界的特点来管理复杂的事物,把它们抽象为对象,具有自己的状态和行为,通过消息传递来完成一定的任务。面向对象的程序设计主要有事物抽象、封装、继承和多态等特点,进而对应面向对象的基本概念:对象(Object)、类(Class)、方法(Method)和实例(Instance)等。

1.2.2 客观事物的抽象

现实世界的对象都有两个相同的特征——状态和行为。例如,汽车有状态(颜色、品牌、排量)和行为(加速、刹车、换档),狗也有状态(名字、颜色、种类)和行为(叫、走、吃)等。为了使计算机能够处理和理解客观事物,程序开发人员必须对事物进行抽象,以求得事物的本质。

利用面向对象的编程语言设计对象时,使用属性来包含状态,使用方法来实现对象的行为,因此如果我们对汽车进行抽象将得到如图 1-2 所示的关系图。

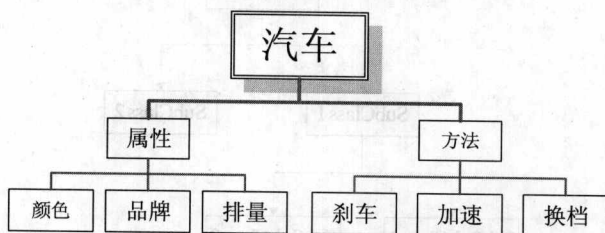


图 1-2 对汽车进行抽象

1.2.3 对象和封装

在对客观事物进行抽象之后,就可以用抽象出来的数据模型来描述对象。因此,对象就是把数据和相关的操作封装在一起而构成的实体。对象中的数据称为数据成员或属性,对象中的一些函数称为方法或成员函数。方法的调用是通过消息的发送和传递实现的。

封装是将相关的数据及其操作结合在一起,使其结构严谨,屏蔽交叉,并使除该对象方法以外的其他方法不能使用这些数据、改变这些数据的状态。封装也称为消息隐藏。

封装允许对象拥有完全独立的数据和代码,这样人们在使用对象时,不需要考虑对象的具体实现,而只需考虑各个对象的接口。因此,每个对象可以看成是一个“黑匣子”,消息从“黑匣子”的一端进入,从另一端出来,如图 1-3 所示。

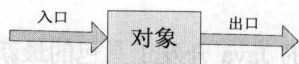


图 1-3 封装的对象