新编高等院校计算机科学与技术规划教材

（第3版）

Java语言实用教程

JAVA YU YAN SHI YONG JIAO CHENG

丁振凡　编著

北京邮电大学出版社
www.buptpress.com

新编高等院校计算机科学与技术规划教材

Java 语言实用教程

（第 3 版）

丁振凡　编著

北京邮电大学出版社
www.buptpress.com

内 容 简 介

本书以 Java 语言的内容体系为线索，将面向对象程序设计的原则与特点融入具体的 Java 程序实例中。在内容编排取舍方面，覆盖了全国计算机等级考试二级 Java 考试大纲的要求，又结合 Java 语言的新发展，内容中包含 Java 8 的许多新特性。书中含 13 章，主要内容为：Java 概述；Java 语言基础；类与对象；继承、多态与接口；常用系统类；Java 绘图与 Applet；图形用户界面编程；异常处理；流式输入/输出与文件处理；多线程；JDBC 技术和数据库应用；Java 的网络编程；Swing 编程。在讲述上由浅入深，注重理论与实际的结合，例题精练，许多例子是实际应用的写照，有利于培养学生解决实际问题的能力。本书结构合理、内容丰富、通俗易懂。例题一般按照“分析、代码设计、说明、思考”的步骤组织讲解，注意启发学生思考。本书可作为高等学校学生学习 Java 和面向对象程序设计的教材，同时也可作为广大自学者和软件开发人员的参考用书。

图书在版编目(CIP)数据

Java 语言实用教程 / 丁振凡编著. --3 版. --北京 : 北京邮电大学出版社，2016.8

ISBN 978-7-5635-4814-9

Ⅰ. ①J… Ⅱ. ①丁… Ⅲ. ①JAVA 语言—程序设计—教材 Ⅳ. ①TP312

中国版本图书馆 CIP 数据核字(2016)第 166156 号

书　　名：Java 语言实用教程(第 3 版)

著作责任者：丁振凡　编著

责 任 编 辑：徐振华　孙宏颖

出 版 发 行：北京邮电大学出版社

社　　址：北京市海淀区西土城路 10 号(邮编：100876)

发 行 部：电话：010-62282185　传真：010-62283578

E-mail：publish@bupt.edu.cn

经　　销：各地新华书店

印　　刷：北京通州皇家印刷厂

开　　本：787 mm×1 092 mm　1/16

印　　张：19.5

字　　数：510 千字

版　　次：2016 年 8 月第 1 版　2016 年 8 月第 1 次印刷

ISBN 978-7-5635-4814-9　　定　价：39.00 元

前　言

Java 语言是一个由 Sun 公司开发而成的新一代编程语言。从 1995 年 Java 的诞生到现在，已有 20 多年了。这些年 Java 得到了飞速的发展，Java 已经涉及计算机应用的众多领域，如浏览器应用、桌面应用、Internet 服务器、中间件、个人数字代理、嵌入式设备等。Java 语言的面向对象、跨平台、多线程、安全等特性，奠定了其作为网络应用开发的首选工具的基础。长期以来，在 TIOBE 编程语言社区对程序设计语言的排行榜中，Java 一直处于前 2 位。

Java 的内容体系非常丰富，本书的立足点是 Java 语言基础部分。全书共分 13 章，第 1 章介绍了程序设计语言概述、面向对象的相关知识，以及 Java 程序的调试过程等；第 2 章主要对 Java 语言的基本成分进行了介绍，包括数据类型、变量、表达式、流程控制语句，以及方法和数组的定义与使用等，该章的目标是使读者理解程序逻辑和解题算法；第 3 章和第 4 章以 Java 面向对象编程中的核心概念为线索，介绍了类与对象的关系，同时介绍了包、属性修饰符、接口、访问控制符、内嵌类等重要概念，将封装、抽象、继承、多态等特性融入具体代码设计中；第 5 章介绍 Java 提供的一些重要工具类，如 Math 类、数据类型封装类、字符串、Collection API、Stream 等；第 6 章介绍了 Java Applet 编程和图形绘制等内容；第 7 章以实例为引导，介绍了 AWT 编程的相关概念，如事件处理、布局设计等，并介绍了常用 AWT 组件的使用；第 8 章讨论了 Java 的异常处理，这也是 Java 代码的特色之一，为防错程序设计提供了全新思路；第 9 章流式输入/输出与文件处理的内容非常丰富，本章在让读者初步了解整体的同时，选择性地就典型使用结合实例进行介绍；第 10 章介绍了 Java 多线程编程的方法与机制；第 11 章讨论了 Java 数据库访问编程技术；第 12 章就 Java 的网络编程进行了讨论，同时给出了一个简单聊天程序设计的综合样例；第 13 章结合实例介绍了部分 Swing 组件的使用方法。

Java 语言是一种纯面向对象的编程语言，因此，本书也适合讲述面向对象程序设计课程的教学。面向对象技术总体上包括面向对象分析、设计、编程三方面的内容。本书仅是面向对象编程，要熟悉面向对象分析和设计，读者还需要学习更多的知识和内容，如 UML 建模等，Java 实际是建模实现的最好语言，很多建模工具可以直接将模型转化为 Java 代码。

要学好 Java，一方面，首先必须熟悉 Java 语言的基本语法规则，其次要尽可能熟悉 Java 的类库，掌握类库的体系和常用类的使用方法。对类库的熟悉在某种程度上也决定了程序员的能力和水平。另一方面，计算机编程仅仅知道语言的语法和内容体系并不代表就是一个出色的程序员，软件设计是一个极赋创造性的工作，但同时也是一项工程，只有经过严格系统的训练，才能提高自己的编程能力。

现代软件设计通常是一项集体的劳动，每个人编写的程序要使别人能容易理解，所以，代码的规范化以及适当加注也是提高软件的效率和可维护性的重要保证。程序设计教学的最根本目标是培养学生的计算机逻辑思维和代码组织能力，而代码设计的首要目标是要做到设计算法清晰、代码规范，与此同时也要考虑代码在运行和存储效率上的最佳化。

本书是作者多年来教学和软件开发经验的总结。作者对书中内容进行了精心的设计和安排。书中所有代码均经过调试,读者可以直接选用。希望学完本书后,对 Java 语言的基础内容体系有较全面的了解,同时通过具体的编程实践在程序设计能力方面有较大的提高。

本书可作为高等院校开设 Java 语言的教材,也可以作为读者自学 Java 语言的用书。可以考虑安排 64 学时的教学,每部分安排至少 1/3 以上的时间上机。最好安排一周的课程设计。

全书由华东交通大学丁振凡教授编写。第 3 版在第 2 版的基础上对一些章节的内容进行了删改,并在原书的基础上增加了一些内容,尤其是补充了 JDK 新版本中的一些新增或变化的内容,如 Java 8 的 Lambda 表达式和 Stream 介绍等。

由于编者水平所限,加之时间仓促,疏漏和错误之处在所难免,恳请读者批评指正。

编　者

目 录

第 1 章 Java 概述 …… 1

1.1 程序设计语言与 Java …… 1
1.1.1 程序设计语言概述 …… 1
1.1.2 Java 语言的产生与发展 …… 1
1.2 面向对象概述 …… 2
1.2.1 面向对象与面向过程的区别 …… 2
1.2.2 面向对象程序设计的特性 …… 2
1.3 Java 开发和运行环境 …… 3
1.4 简单 Java 程序及调试步骤 …… 4
1.4.1 Java Application …… 4
1.4.2 Java Applet …… 8
1.5 Java 语言的特点 …… 9
1.5.1 简单的面向对象语言 …… 9
1.5.2 跨平台与解释执行 …… 9
1.5.3 健壮和安全的语言 …… 10
1.5.4 支持多线程 …… 10
1.5.5 面向网络的语言 …… 10
1.5.6 动态性 …… 10
1.6 本章小结 …… 11
习题 …… 11

第 2 章 Java 语言基础 …… 13

2.1 Java 符号 …… 13
2.1.1 标识符 …… 13
2.1.2 关键字 …… 13
2.1.3 分隔符 …… 14
2.1.4 注释 …… 14
2.2 数据类型与变量 …… 15
2.2.1 数据类型 …… 15
2.2.2 常量 …… 16
2.2.3 变量 …… 18

2.3 表达式与运算符 …… 20
2.3.1 算术运算符 …… 21
2.3.2 关系运算符 …… 23
2.3.3 逻辑运算符 …… 23
2.3.4 位运算符 …… 24
2.3.5 赋值组合运算符 …… 25
2.3.6 其他运算符 …… 25
2.3.7 运算符优先级 …… 26
2.3.8 常用数学函数的使用 …… 27
2.4 数据的输入和输出 …… 28
2.4.1 字符界面数据的输出 …… 28
2.4.2 字符界面数据的输入 …… 29
2.4.3 用 Swing 对话框实现输入/输出 …… 32
2.5 流程控制语句 …… 32
2.5.1 条件选择语句 …… 33
2.5.2 循环语句 …… 37
2.5.3 跳转语句 …… 43
2.6 方法 …… 46
2.6.1 方法声明 …… 46
2.6.2 方法调用 …… 46
2.6.3 参数传递 …… 48
2.6.4 递归 …… 50
2.7 数组 …… 51
2.7.1 一维数组 …… 51
2.7.2 多维数组 …… 53
2.7.3 数组作为方法参数 …… 56
2.7.4 Java 的命令行参数 …… 58
2.7.5 Java 方法的可变长参数 …… 59
2.8 本章小结 …… 59
习题 …… 60

第 3 章 类与对象 …… 64

3.1 Java 的类 …… 64
3.1.1 系统定义的类 …… 64
3.1.2 用户自定义类 …… 64
3.2 对象的创建与引用 …… 66
3.2.1 创建对象及访问对象成员 …… 66
3.2.2 对象的初始化和构造方法 …… 68
3.3 变量作用域 …… 70
3.4 类变量和静态方法 …… 72

3.4.1 类变量 …… 72
3.4.2 静态方法 …… 74
3.5 使用包组织类 …… 75
3.5.1 建立包 …… 75
3.5.2 包的引用 …… 76
3.6 本章小结 …… 78
习题 …… 78

第 4 章 继承、多态与接口 …… 81

4.1 继承 …… 81
4.1.1 Java 继承的实现 …… 81
4.1.2 Object 类 …… 82
4.1.3 构造方法在类继承中的作用 …… 82
4.1.4 变量的继承、隐藏 …… 84
4.2 访问控制符 …… 85
4.2.1 公共访问控制符 …… 85
4.2.2 缺省访问控制符 …… 85
4.2.3 私有访问控制符 …… 85
4.2.4 保护访问控制符 …… 86
4.3 多态性 …… 87
4.3.1 方法的重载 …… 87
4.3.2 方法的覆盖 …… 90
4.3.3 访问继承的成员 …… 91
4.4 this 和 super …… 92
4.4.1 this 的应用 …… 92
4.4.2 通过 super 访问父类成员 …… 93
4.5 final 修饰符的使用 …… 95
4.5.1 final 作为类修饰符 …… 95
4.5.2 用 final 修饰方法 …… 95
4.5.3 用 final 定义常量 …… 95
4.6 抽象类和抽象方法 …… 96
4.6.1 抽象类的定义 …… 96
4.6.2 抽象类的实现 …… 96
4.7 接口 …… 97
4.7.1 接口定义 …… 97
4.7.2 接口的实现 …… 98
4.8 Java 泛型 …… 101
4.8.1 Java 泛型 …… 101
4.8.2 关于 Comparable<T>与 Comparator<T>接口 …… 102
4.9 对象引用转换 …… 104

4.9.1 对象引用赋值转换 …… 104
4.9.2 对象引用强制转换 …… 106
4.10 内嵌类 …… 106
4.10.1 成员类 …… 107
4.10.2 静态 inner 类 …… 109
4.10.3 方法中的内嵌类与匿名内嵌类 …… 110
4.11 Lambda 表达式 …… 111
4.11.1 何为 Lambda(λ)表达式 …… 111
4.11.2 方法引用 …… 114
4.11.3 高阶函数 …… 115
4.12 本章小结 …… 115
习题 …… 116

第 5 章 常用系统类 …… 119

5.1 基本数据类型包装类 …… 119
5.2 字符串 …… 120
5.2.1 String 类 …… 120
5.2.2 StringBuffer 类 …… 125
5.3 Collection API 简介 …… 127
5.3.1 Collection 接口及实现层次 …… 127
5.3.2 Collections 类 …… 130
5.3.3 Map 接口及实现层次 …… 131
5.4 Stream …… 133
5.4.1 Stream 概念 …… 133
5.4.2 中间操作 …… 134
5.4.3 最终操作 …… 135
5.5 日期和时间 …… 137
5.5.1 Date 类 …… 137
5.5.2 Calendar 类 …… 137
5.5.3 Clock 类 …… 138
5.6 本章小结 …… 138
习题 …… 138

第 6 章 Java 绘图与 Applet …… 142

6.1 什么是 Applet …… 142
6.2 Applet 方法介绍 …… 143
6.3 Java AWT 绘图 …… 145
6.3.1 Java 图形坐标 …… 145
6.3.2 各类图形的绘制方法 …… 145
6.3.3 显示文字 …… 146

6.3.4 颜色控制 …… 147
6.3.5 绘制图像 …… 150
6.3.6 Java 2D 图形绘制 …… 153
6.4 Applet 参数传递 …… 158
6.5 本章小结 …… 158
习题 …… 159

第 7 章 图形用户界面编程 …… 160

7.1 图形用户界面核心概念 …… 160
7.1.1 一个简单的 GUI 示例 …… 160
7.1.2 创建窗体 …… 161
7.1.3 创建 GUI 部件 …… 162
7.1.4 事件处理 …… 163
7.1.5 在事件处理代码中区分事件源 …… 165
7.1.6 关于事件适配器类 …… 167
7.2 容器与布局管理 …… 169
7.2.1 FlowLayout …… 169
7.2.2 BorderLayout …… 170
7.2.3 GridLayout …… 171
7.2.4 CardLayout …… 172
7.2.5 GridBagLayout …… 176
7.3 常用 GUI 标准组件 …… 178
7.3.1 GUI 标准组件概述 …… 178
7.3.2 文本框与文本域 …… 180
7.3.3 选项按钮与列表的使用 …… 182
7.3.4 滚动条的使用 …… 190
7.4 鼠标和键盘事件 …… 193
7.4.1 鼠标事件 …… 193
7.4.2 键盘事件 …… 195
7.5 菜单的使用 …… 197
7.5.1 下拉菜单 …… 197
7.5.2 弹出式菜单 …… 201
7.6 对话框的使用 …… 204
7.6.1 对话框的创建与使用 …… 204
7.6.2 文件对话框 …… 204
7.7 本章小结 …… 206
习题 …… 206

第 8 章 异常处理 …… 207

8.1 异常的概念 …… 207

8.1.1 什么是异常 …… 207
8.1.2 异常的类层次 …… 209
8.1.3 系统定义的异常 …… 209
8.2 异常的处理 …… 210
8.2.1 try…catch…finally 结构 …… 210
8.2.2 多异常的处理举例 …… 211
8.3 自定义异常 …… 212
8.3.1 自定义异常类设计 …… 212
8.3.2 抛出异常 …… 213
8.3.3 方法的异常声明 …… 213
8.4 本章小结 …… 215
习题 …… 215

第 9 章 流式输入/输出与文件处理 …… 217

9.1 输入/输出基本概念 …… 217
9.1.1 I/O 设备与文件 …… 217
9.1.2 流的概念 …… 217
9.2 文件与目录管理 …… 218
9.3 面向字节的输入/输出流 …… 220
9.3.1 面向字节的输入流 …… 220
9.3.2 面向字节的输出流 …… 223
9.3.3 对象串行化 …… 226
9.4 面向字符的输入/输出流 …… 229
9.4.1 面向字符的输入流 …… 229
9.4.2 面向字符的输出流 …… 231
9.5 转换流 …… 232
9.5.1 转换输入流 …… 232
9.5.2 转换输出流 …… 233
9.6 文件的随机访问 …… 234
9.7 本章小结 …… 235
习题 …… 235

第 10 章 多线程 …… 237

10.1 Java 线程的概念 …… 237
10.1.1 多进程与多线程 …… 237
10.1.2 线程的状态 …… 237
10.1.3 线程调度与优先级 …… 238
10.2 Java 多线程编程方法 …… 238
10.2.1 Thread 类简介 …… 238
10.2.2 继承 Thread 类实现多线程 …… 239

10.2.3　实现 Runnable 接口编写多线程 …… 241
10.3　线程的控制 …… 242
10.3.1　放弃运行 …… 242
10.3.2　挂起 …… 243
10.3.3　睡眠一段时间 …… 243
10.3.4　阻塞 …… 243
10.3.5　关于用户线程和看守线程 …… 243
10.4　线程资源的同步处理 …… 243
10.4.1　临界资源问题 …… 244
10.4.2　wait()和 notify() 方法 …… 245
10.4.3　生产者、消费者模型 …… 245
10.4.4　死锁 …… 247
10.5　本章小结 …… 248
习题 …… 248

第 11 章　JDBC 技术和数据库应用 …… 249

11.1　关系数据库概述 …… 249
11.2　JDBC …… 249
11.2.1　JDBC 驱动程序 …… 249
11.2.2　ODBC 数据源配置 …… 250
11.2.3　JDBC API …… 252
11.3　JDBC 基本应用 …… 254
11.3.1　数据库查询 …… 254
11.3.2　数据库的更新 …… 258
11.3.3　用 PreparedStatement 类实现 SQL 操作 …… 259
11.4　本章小结 …… 260
习题 …… 261

第 12 章　Java 的网络编程 …… 262

12.1　网络编程基础 …… 262
12.1.1　网络协议 …… 262
12.1.2　InetAddress 类 …… 262
12.2　Socket 通信 …… 263
12.2.1　Java 的 Socket 编程原理 …… 263
12.2.2　简单多用户聊天程序的实现 …… 265
12.3　无连接的数据报 …… 269
12.3.1　DatagramPacket 类 …… 269
12.3.2　DatagramSocket 类 …… 269
12.3.3　发送和接收过程 …… 269
12.3.4　数据报多播 …… 272

12.4 URL …… 274
12.4.1 URL 类 …… 275
12.4.2 URLConnection 类 …… 276
12.5 本章小结 …… 277
习题 …… 278

第 13 章 Swing 编程 …… 279

13.1 Swing 包简介 …… 279
13.2 Swing 包典型部件的使用 …… 280
13.2.1 JFrame 类 …… 280
13.2.2 JPanel 类 …… 281
13.2.3 Swing 中的按钮和标签 …… 283
13.2.4 滚动窗格 …… 285
13.2.5 工具栏 …… 286
13.2.6 Swing 中的对话框 …… 287
13.2.7 选项卡 …… 292
13.2.8 表格 …… 293
13.2.9 Swing 的其他几个选择部件 …… 296
13.3 本章小结 …… 298
习题 …… 298

参考文献 …… 299

第1章　Java概述

1.1　程序设计语言与 Java

1.1.1　程序设计语言概述

人类的语言是一个渐变发展的过程，直到今天仍在不断改进。计算机程序设计语言也不是一步到位的，而是一个从面向机器语言，到面向过程语言，再到今天的面向对象语言的过程。

面向机器语言，如最早的机器语言，是由 0 和 1 组成的枯燥数字序列，不仅难看、难记，也难理解，后来，计算机科学家们又设计出了一种用英文单词或其缩写形式代替枯燥乏味的二进制数字的语言——助记符语言，即汇编语言，使得操作计算机的方式大大简化了。但其思维编程方式依然是机器式的，人们必须按照计算机固有的方式来设计程序。

面向过程语言，如 Fortran、C、Pascal、BASIC 等，可以让人们用接近数学语言的方式进行程序设计，加快了编程速度，也使得人们能够从繁琐的硬件细节中摆脱出来，而集中注意力在算法本身。

面向对象语言，如 Java、C++等，解决了传统结构化方法中问题空间和解空间在结构上不一致的问题，避免从分析和设计到软件模块结构间的多次转换过程，使软件开发变得简单、高效、合理，是真正最接近人类思维方式的计算机程序设计语言。

1.1.2　Java 语言的产生与发展

Java 来自于 1991 年 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。最开始，Sun 公司准备采用 C++，但 C++太复杂，安全性差，最后基于 C++开发一种新的语言 Oak(Java 的前身)，1995 年 Sun 公司正式推出新一代的面向对象程序设计语言 Java。Java 的取名也有一趣闻，有一天，几位 Java 开发成员讨论给这个新的语言取什么名字，当时在一个叫“爪哇”岛屿的咖啡馆喝着咖啡，有人灵机一动说就叫 Java(爪哇)，得到大家的赞许。

Java 一经问世就给软件行业带来了革命性影响，受到业界的普遍关注和支持，并以极其迅猛的势头发展至今。现在 Java 已成为软件开发的主流技术，引领了世界范围学习和使用 Java 的热潮。Java 语言已作为一门综合性技术在众多领域得到发展和应用。除了本书介绍的 Java 应用程序和 Applet 小应用程序外，Java 内容体系还包括：

- JSP/Servlet，用于基于 Web 的服务端动态网页编程；
- Java Bean，用 Java 语言开发的软件组件，可在分布式环境中移动；

- EJB(企业 JavaBean),用于企业分布式应用系统的构建;
- J2ME,是移动消费产品和嵌入式设备的最佳解决方案。

1.2 面向对象概述

1.2.1 面向对象与面向过程的区别

早期的编程语言如 FORTRAN、C 基本上都是面向过程的语言,主要是采用数学语言的方式组织编程的语言,其编程的主要思路专注于算法的实现。

传统的面向过程的编程在描述问题时,由两部分构成:

- 数据,描述实体状态的数据结构;
- 过程,操作这些状态数据的程序和步骤。

面向过程编程的一个明显特点是数据与程序的分开,数据是静止的东西,不会自行变化,必须通过操作来改变数据,因此,函数调用在面向过程编程中大量使用。

随着计算机软件的发展,程序越做越大,后期维护的工作就越发艰难,甚至出现了因为程序结构不清晰而无法维护改进的局面。面向对象编程提出了一种全新的思路,让计算机语言结构像人类思维方式一样简单、清晰。

面向对象的软件开发中将世界上的事物均看作对象,对象有两个特征:行为与状态。每个对象可以通过自身的行为来改变自己的状态。在面向对象世界中,通过对象间的协作与交互来运作。由于将对象的操作封闭在对象内,外部要与对象进行交互只能通过给对象发送“消息”,这个消息实际上就是调用对象的某个方法以及给方法传递参数。

1.2.2 面向对象程序设计的特性

1. 封装性(Encapsulation)

面向对象的第一个原则是把数据和对该数据的操作都封装在一个类中,类的概念和现实世界中的“事物种类”是一致的,如笔记本式计算机就是一个类。笔记本式计算机这个类有许多成员构成,有些是静态的(数据),如尺寸、重量、显示屏的亮度;有些是动态的(对数据的操作),如可以通过按下键“F4”或“F6”来调整显示屏的亮度。

对象是类的一个实例化结果,对象具有类所描述的所有的属性以及方法。如笔者所使用的计算机,就是一台具体的笔记本计算机,它和其他具体计算机即便是同一品牌笔记本式计算机都是不同的对象,虽然配置相同,它们的序列号也是不同的。

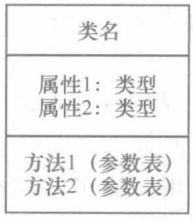

图 1-1 类的表示

每个对象都有自己的存储空间,其中存储对象的所有属性。有些属性本身又可能是其他对象,或者说通过封装现有的对象,可以产生新型对象。因此,尽管对象的概念非常简单,但是经过封装以后却可以在程序中达到任意复杂程度。

每个对象都属于某个类。面向对象程序设计就是设计好相关的类,类中有静态的域(数据)和动态的方法(对数据的操作)。在统一建模语言 UML 中使用如图 1-1 所示的符号来描述对象和类的结构,其中,属性用来描述对象的状态,而方法则描述对象的行为。根据需要

可以给属性和方法设置访问修饰，从而限制外界只能通过特定接口访问对象。

2. 继承性(Inheritance)

继承是在类、子类以及对象之间自动地共享属性和方法的机制。类的上层可以有父类、下层可以有子类，形成一种层次结构。一个类将直接继承其父类的属性和行为，而且继承还具有传递性，因此，它还将间接继承所有祖先类的属性和行为。

图 1-2 给出了以生物的划分对应的类层次关系。

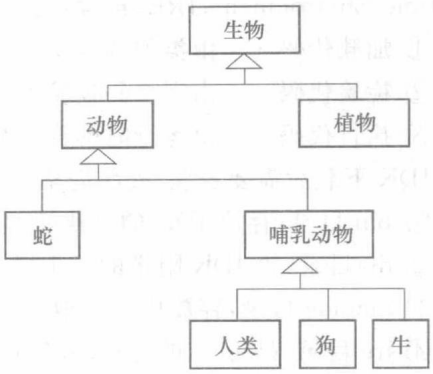

图 1-2　类的继承层次

“人类”继承了“哺乳动物”“动物”以及“生物”的特性。

继承最主要的优点是重复使用性，通过继承可以无限繁衍出更多的类。在继承已有类的基础上加以改写，进而得到功能的不断扩充，可达到程序共享的好处，提高软件开发效率。

继承的另一个优点是在于接口的一致性。当超类繁衍出许多子类时，它的行为接口通过继承可以传给其所有子类。因此，可以通过统一的行为接口去访问不同子类对象的行为，但不同子类中具体行为实现可能不一样。子类中对父类定义的行为重新定义是下面将介绍的多态性的一种体现。

3. 多态性(Polymorphism)

多态是指在表示特定功能时，有多种不同的形态或实现方法。常见的多态形式有两种：

① 方法的重载(overloading)，即在同一个类中某个方法有多种形态。方法名相同，但参数不同，所以也称参数多态；

② 方法的覆盖(overriding)，对于父类的某个方法，在子类中重新定义一个相同形态的方法，这样，在子类中将覆盖从父类继承来的那个方法。

多态为描述客观事物提供了极大的能动性。参数多态提供方法的多种使用形式，这样方便使用者的调用；而覆盖多态则可使得我们以用同样的方式对待不同的对象，不同的对象可以用各自的方式响应同一消息。通过父类定义的变量可引用子类的对象，执行对象方法时则表现出每个子类对象各自的行为，这种特性也称运行时的多态性。

4. 抽象性(Abstraction)

这里，抽象有两个层次的含义，一是体现在类的层次设计中，高层类是底层类的抽象表述。类层次设计体现着不断抽象的过程。例如，Java 中有一个类 Object，它处于类层次结构的顶端，该类中定义了所有类的公共属性和方法。可以用 Object 类的变量去引用任何子类的对象，通过 Object 对象引用能访问的成员，在子类中总是存在的。

二是体现在类与对象之间的关系上，类是一个抽象的概念，而对象是具体的。面向对象编程的核心是设计类，但实际运行操作的是对象。类是对象的模板，对象的创建是以类为基础的。同一类创建的对象具有共同的属性，但属性值不同。

1.3　Java 开发和运行环境

Java 开发和运行环境有很多，例如，Oracle 公司的 JDK、NetBeans；IBM 公司的 Visual

Age for Java;BEA 公司的 WebLogic Workshop;Xinox 公司的 JCreator;开源组织提供的 Eclipse;Spring Source 公司的 STS 等。

在以上工具中,只有 JDK 是字符环境,其他均是图形环境。JDK 可以从 Oracle 公司的网站下载,本书以 JDK 为开发环境进行介绍。JDK 包括运行环境(Java 虚拟机,即 Java 类和支持文件的平台)和开发工具(编译器、调试器、工具库以及其他工具)。Java 运行环境(Java Runtime Environment,JRE,包含在 JDK 中,也可单独下载安装)主要担负三大任务:

① 加载代码——由类加载器执行;

② 检验代码——由字节码校验器执行;

③ 执行代码——由运行时解释执行。

JDK 下载后需要安装,安装完毕后会在安装目录下建立 5 个子目录:

① bin 目录,存放 JDK 的可执行程序,如 javac. exe、java. exe、appletviewer. exe 等;

② db 目录,含 JDK 附带的一个轻量级的关系数据库,名为 Derby;

③ include 目录,存放用于本地方法的文件;

④ jre 目录,是 Java 的运行环境,如果只是为了能够运行 Java 字节码,可以只安装 jre,它会在系统中已安装的浏览器下安装相应的解析环境,使浏览器能够运行 Java;

⑤ lib 目录,含有一些库文件。

1.4 简单 Java 程序及调试步骤

根据结构组成和运行环境的不同,Java 程序可以分为两类:Java 应用程序(Java Application)和小应用程序(Java Applet)。

1.4.1 Java Application

Java Application 是一个完整的程序,不过仍然需要独立的解析器来解析运行。也就是说,Java 不像其他大多数语言一样,可以编译链接成可独立运行的. exe 等可执行程序,Java 需要编译,但不需要链接,编译后生成. class 字节码文件,这是 Java 平台无关性的重要体现,因为. class 不是可直接执行的程序文件,而是需要解析器解析的二进制目标码,可以在任何装有 jre 的系统下运行。例如,可以在 Windows 下编译 Java 源程序,然后到 Linux 下去运行它;当然反过来也可以。这正是 Java 的"Write once, run anywhere."(一次编写,到处运行)跨平台的重要特性。调试 Java Application 包括编辑、编译、解释运行 3 个步骤,如图 1-3 所示。

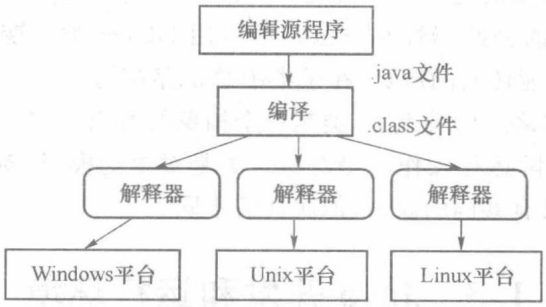

图 1-3 Java 程序的调试过程