



新编高等院校计算机科学与技术应用型规划教材

Java

Java 语言实用教程

(第2版)

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应用型规划教材

Java 语言实用教程

(第 2 版)

丁振凡 编著

北京邮电大学出版社
·北京·

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到具体的 Java 程序实例中。书中主要内容有:面向对象概述、Java 语言基础、Java 面向对象技术、常用类库、异常处理、Applet 编程、AWT 编程、多线程、输入/输出与文件处理、JDBC 与数据库、网络编程、Swing 编程等。本书在讲述上由浅入深,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。

本书可作为大中专院校低年级学生学习 Java 和面向对象程序设计方法的教材,同时也可作为广大自学者和软件开发人员的参考用书。本书第 1 版获得江西省优秀教材二等奖,作者主持的 Java 语言课程被评为省优质课程。

图书在版编目(CIP)数据

Java 语言实用教程/丁振凡编著. —2 版. —北京:北京邮电大学出版社,2008
ISBN 978-7-5635-1559-2

I. J… II. 丁… 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 185072 号

书 名: Java 语言实用教程(第 2 版)

作 者: 丁振凡

责任编辑: 陈岚岚

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: 北京忠信诚胶印厂

开 本: 787 mm×1 092 mm

印 张: 20.25

字 数: 487 千字

版 次: 2005 年 2 月第 1 版 2008 年 1 月第 2 版 2008 年 1 月第 1 次印刷

ISBN 978-7-5635-1559-2

定 价: 28.00 元

· 如有印装质量问题,请与北京邮电大学出版社发行部联系 ·

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,在短短的十多年中,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。有人预言,在不久的将来,全球 90% 的代码将采用 Java 开发。姑且不论该预言的准确性,但足以看出 Java 在未来软件设计中的地位。

Java 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 的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、向量以及 Collection API 等;第 6 章介绍了 Java Applet 编程和图形绘制以及多媒体播放等内容;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,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,并结合一个简单考试系统的设计给出了一个综合设计样例;第 12 章就 Java 的网络编程进行了讨论,并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。

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。按照由浅入深、循序渐进的原则进行组织；程序样例大多简短实用，易于教学使用和读者学习；书中所有代码均经过调试，读者可以直接选用。

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3个方面的内容。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，在计算机编程中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，软件设计是一个极富创造性的工作，但同时也是一项工程，只有经过严格系统的训练，才能提高编程能力。程序设计是一个实践性很强的教学环节，亲自动手编程并上机调试，是提高编程能力的最好途径。希望读者要养成积极思考和不怕困难的作风。读者在阅读书籍时要注意代码的风格，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。这点要学习邻国印度的先进做法，据调查，出一道题给20个印度程序员写代码，结果基本一样，而中国程序员的结果却五花八门。这点对大型软件设计效率的提高是不利的。因此，希望读者对常见问题的解决方法能熟练掌握，这样，一遇到问题，就能快速写出代码。

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语言的自学用书。书中包括有部分难度较大的综合训练例题和练习，希望读者能仔细阅读，它也许正好能运用在你的实际项目开发中。希望学完本书后，读者对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。

本书可以考虑安排64学时的教学，每部分安排一半时间上机。总体上，学

时比较紧,如果课程学时数多,则更宽松一些,最好安排一周的课程设计。如果学时数不足,则可以将第 11~第 13 章这 3 章作为选学内容。

全书由华东交通大学丁振凡教授编写。第 2 版在第 1 版的基础上对一些章节的内容进行了删改,并在原书基础上增加了一些内容,例如:Collection API 中增加了 Map 接口的使用,在 Swing 中增加了表格和树的应用等。从而使内容的阐述更为形象和系统。

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。

作 者

目 录

第 1 章 Java 概述

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 的开发和运行环境	4
1.4 简单 Java 程序及调试步骤	5
1.4.1 Java Application	5
1.4.2 Java Applet	8
1.5 Java 语言的特点	10
1.5.1 简单的面向对象语言	10
1.5.2 跨平台与解释执行	10
1.5.3 健壮和安全的语言	11
1.5.4 支持多线程	11
1.5.5 面向网络的语言	12
1.5.6 动态性	12
1.6 本章小结	12
习 题	12

第 2 章 Java 语言基础

2.1 Java 符号	14
2.1.1 标识符	14
2.1.2 关键字	14
2.1.3 分隔符	15
2.1.4 注释	15
2.2 数据类型	16



2.2.1	数据类型	16
2.2.2	常量	17
2.2.3	变量	18
2.3	表达式与运算符	22
2.3.1	算术运算符	23
2.3.2	关系运算符	24
2.3.3	逻辑运算符	25
2.3.4	位运算符	25
2.3.5	赋值组合运算符	27
2.3.6	其他运算符	27
2.3.7	运算符优先级	28
2.4	字符界面常见类型数据的输入	29
2.5	流程控制语句	31
2.5.1	条件选择语句	31
2.5.2	循环语句	36
2.5.3	跳转语句	43
2.6	方法	45
2.6.1	方法声明	45
2.6.2	方法调用	46
2.6.3	参数传递	47
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	59
2.8	本章小结	60
习 题		60

第3章 类与对象

3.1	Java 的类	65
3.1.1	系统定义的类	65
3.1.2	用户自定义的类	65
3.2	对象的创建与引用	67
3.2.1	创建对象及访问对象成员	67
3.2.2	对象的初始化和构造方法	69
3.3	变量作用域	71
3.4	类变量和静态方法	73

3.4.1 类变量	73
3.4.2 静态方法	75
3.5 使用包组织类	76
3.5.1 建立包	76
3.5.2 包的引用	77
3.6 本章小结	78
习 题	79

第 4 章 继承、多态与接口

4.1 继承	81
4.1.1 Java 继承的实现	81
4.1.2 构造方法在类继承中的作用	82
4.1.3 变量的继承、隐藏	84
4.2 访问控制符	84
4.2.1 公共访问控制符 public	84
4.2.2 缺省访问控制符	85
4.2.3 私有访问控制符 private	85
4.2.4 保护访问控制符 protected	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
4.4 this 和 super	90
4.4.1 this 的应用	90
4.4.2 通过 super 访问父类成员	91
4.5 final 修饰符的使用	93
4.5.1 final 作为类修饰符	93
4.5.2 用 final 修饰方法	93
4.5.3 用 final 定义常量	93
4.6 抽象类和抽象方法	94
4.6.1 抽象类的定义	94
4.6.2 抽象类的实现	95
4.7 接口	96
4.7.1 接口定义	96
4.7.2 接口的实现	97
4.8 内嵌类	99
4.8.1 成员类	99
4.8.2 静态 inner 类	101
4.8.3 方法中的内嵌类与匿名内嵌类	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 本章小结 106

习 题 107

第 5 章 常用系统类

5.1 语言基础类 109

 5.1.1 Object 类 109

 5.1.2 Math 类 109

 5.1.3 数据类型包装类 110

5.2 字符串 112

 5.2.1 String 类 112

 5.2.2 StringBuffer 类 117

5.3 Vector 类 118

5.4 Collection API 简介 123

 5.4.1 Collection 接口及实现层次 123

 5.4.2 Map 接口及实现层次 126

5.5 日期和时间 128

 5.5.1 Date 类 128

 5.5.2 Calendar 类 128

5.6 本章小结 129

习 题 129

第 6 章 Java Applet

6.1 什么是 Applet 131

6.2 Applet 方法介绍 132

6.3 Applet 的 AWT 绘制 134

 6.3.1 Java 图形坐标 134

 6.3.2 各类图形的绘制方法 134

 6.3.3 显示文字 136

 6.3.4 颜色控制 137

 6.3.5 Java 2D 图形绘制 138

6.4 Applet 参数传递 143

 6.4.1 在 HTML 文件中给 Applet 提供参数 143

 6.4.2 在 Applet 代码中读取 Applet 参数值 144

6.5 Applet 的多媒体支持 146

 6.5.1 绘制图像 146

6.5.2 实现动画	148
6.5.3 播放声音文件	149
6.6 Java 存档文件	150
6.6.1 创建存档文件	150
6.6.2 在 HTML 文件中指定 Applet 的存档文件	151
6.7 本章小结	152
习 题	152

第 7 章 图形用户界面编程

7.1 图形用户界面核心概念	153
7.1.1 一个简单的 GUI 示例	153
7.1.2 创建窗体	154
7.1.3 创建 GUI 部件	155
7.1.4 事件处理	155
7.1.5 在事件处理代码中区分事件源	159
7.1.6 关于事件适配器类	160
7.2 容器与布局管理	162
7.2.1 BorderLayout	162
7.2.2 BorderLayout	164
7.2.3 GridLayout	164
7.2.4 CardLayout	165
7.2.5 GridBagLayout	170
7.3 常用 GUI 标准组件	172
7.3.1 GUI 标准组件概述	172
7.3.2 文本框与文本域	173
7.3.3 选项按钮与列表的使用	176
7.3.4 滚动条的使用	183
7.4 鼠标和键盘事件	186
7.4.1 鼠标事件	186
7.4.2 键盘事件	189
7.5 菜单的使用	192
7.5.1 下拉菜单	192
7.5.2 弹出式菜单	195
7.6 对话框的使用	197
7.6.1 对话框的创建与使用	197
7.6.2 文件对话框	199
7.7 本章小结	201
习 题	201



第8章 异常处理

8.1 异常的概念	203
8.1.1 什么是异常	203
8.1.2 异常的分类	205
8.1.3 系统定义的异常	205
8.2 异常的处理	206
8.2.1 try...catch...finally 结构	206
8.2.2 多异常的处理举例	207
8.3 自定义异常	208
8.3.1 自定义异常类设计	209
8.3.2 抛出异常	209
8.3.3 方法的异常声明	209
8.4 本章小结	211
习 题	211

第9章 流式输入/输出与文件处理

9.1 输入/输出基本概念	213
9.1.1 输入/输出设备与文件	213
9.1.2 流的概念	213
9.2 面向字节的输入/输出流	214
9.2.1 面向字节的输入流	214
9.2.2 面向字节的输出流	217
9.3 面向字符的输入/输出流	220
9.3.1 面向字符的输入流	220
9.3.2 面向字符的输出流	222
9.4 转换流	223
9.4.1 转换输入流	223
9.4.2 转换输出流	224
9.5 文件处理	224
9.5.1 文件与目录管理	224
9.5.2 文件的顺序访问	226
9.5.3 文件的随机访问	227
9.6 对象串行化	228
9.7 本章小结	231
习 题	232

第 10 章 多线程

10.1 Java 线程的概念	233
10.1.1 多进程与多线程	233
10.1.2 线程的状态	233
10.1.3 线程调度与优先级	234
10.2 Java 多线程编程方法	234
10.2.1 Thread 类简介	234
10.2.2 继承 Thread 类实现多线程	235
10.2.3 实现 Runnable 接口编写多线程	237
10.3 线程的控制	238
10.3.1 放弃运行	238
10.3.2 无限等待	238
10.3.3 睡眠一段时间	239
10.3.4 阻塞	239
10.3.5 关于用户线程和看守线程	239
10.4 线程资源的同步处理	240
10.4.1 临界资源问题	240
10.4.2 wait() 和 notify() 方法	241
10.4.3 生产者与消费者模型	242
10.4.4 死锁	244
10.5 本章小结	244
习 题	245

第 11 章 JDBC 技术和数据库应用

11.1 关系数据库概述	246
11.2 JDBC	246
11.2.1 JDBC 驱动程序	246
11.2.2 ODBC 数据源配置	247
11.2.3 JDBC API	249
11.3 JDBC 基本应用	252
11.3.1 数据库的查询	252
11.3.2 数据库的更新	255
11.3.3 用 PreparedStatement 类实现 SQL 操作	257
11.4 数据库应用举例	258
11.5 本章小结	267
习 题	268



第 12 章 Java 的网络编程

12.1 网络编程基础	269
12.1.1 网络协议	269
12.1.2 InetAddress 类	269
12.2 Socket 通信	270
12.2.1 Java 的 Socket 编程原理	270
12.2.2 简单多用户聊天程序的实现	272
12.3 无连接的数据报	276
12.3.1 DatagramPacket 类	276
12.3.2 DatagramSocket 类	277
12.3.3 发送和接收过程	277
12.3.4 数据报多播	279
12.4 URL	282
12.4.1 URL 类	282
12.4.2 URLConnection 类	284
12.4.3 用 Applet 方法访问 URL 资源	285
12.5 本章小结	287
习 题	287

第 13 章 Swing 编程

13.1 Swing 包简介	288
13.1.1 Swing 组件的特性	288
13.1.2 Swing 的功能分类	288
13.2 Swing 包典型部件的使用	289
13.2.1 JFrame 类	289
13.2.2 JApplet	290
13.2.3 Swing 中的按钮和标签	292
13.2.4 滚动窗格	293
13.2.5 工具栏	294
13.2.6 Swing 中的对话框	296
13.2.7 选项卡	300
13.2.8 表格	302
13.2.9 树	305
13.3 本章小结	307
习 题	307

第 1 章 Java 概述

1.1 程序设计语言与 Java

计算机从诞生到现在也不过短短半个世纪左右,从其诞生的那一天起人们就为了更好地操作计算机而费尽心机,正像人与人之间的交流是从手势逐渐进化到语言一样,人们操作计算机也是从机械开关到程序设计演进的,因而用到程序设计语言。

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,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。但其编程的思维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,而将注意力集中在算法本身。

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 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),Oak 是一种用于使网络精巧而安全的语言,Sun 公司曾依此投标一个交互式电视项目,但结果是被 SGI 打败。可怜的 Oak 几乎无家可归,恰巧这时 Mark Andreessen 开发的 Mosaic 和 Netscape 启发了 Oak 项目组成员,他们用 Java 编制了 Hot Java 浏览器,得到了 Sun 公司首席执行官 Scott McNealy 的支持,触发了 Java 进军 Internet。Java 的取名也有一则趣闻,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,当时他们正在一个叫“爪哇”的岛屿的咖啡馆喝着咖啡,有人灵机一动说就叫 Java(爪哇)怎样,得到了其他人的赞赏,于是,Java 这个名字就这样传开了。

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,引领了世界范围内学习和



使用 Java 的热潮。

计算机产业的许多大公司购买了 Java 的许可证,如 IBM、Novell、Oracle、SGI 和 Borland等,Microsoft 公司一开始对 Java 无动于衷,比尔·盖茨在仔细研究了 Java 的技术规范之后,不得不承认“Java 确实是有史以来最伟大的程序设计语言”,继而,Microsoft 公司购买了 Java 的使用许可并很快推出了集成化的 Java 开发环境——Visual J++。

众多的软件开发商开始支持 Java 的软件产品。例如:Borland 公司的基于 Java 的快速应用程序开发环境 Latte 在 1996 年发布,推动了 Java 进入 PC 机软件市场。

迄今为止,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 Java Bean),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,由数据和过程两部分构成。

- 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成对象,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。比如笔记本电脑就是一个类。笔记本电脑这个类由许多成员构成,有些是静态的(数据),比如尺寸、重量、显示屏的亮度;有些是动态的(对数据的操作),比如可以通过按下组合键“Ctrl+F4”或“Ctrl+F5”来调整显示屏的亮度,可以卸掉它



的外挂光驱来减轻它的重量。

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。比如笔者所使用的电脑,就是一台具体的笔记本电脑,它和其他具体电脑即便是同一品牌笔记本电脑,都是不同的对象,虽然配置相同,它们的序列号也是不同的,但它们之间毫无疑问存在一些关系,比如说它们都重量较轻,便于携带。其实它们之间的共同属性就是它们所属的类——笔记本电脑类的属性。

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。

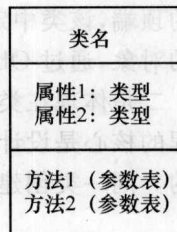


图 1-1 类的表示

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。

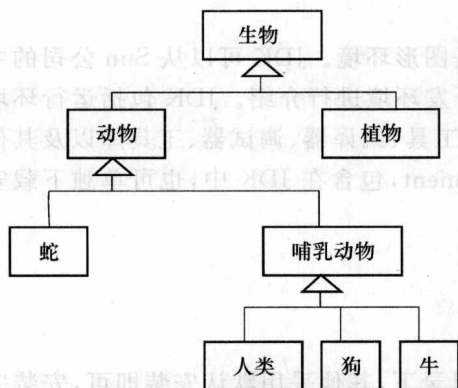


图 1-2 类的继承层次

“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。常见的多态形式有方法的重载与方法覆盖两种。

- 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。
- 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