



21世纪高等院校计算机系列教材

Java 语言实用教程

YUYAN SHIYONG JIAOCHENG

丁振凡 © 主编



北京邮电大学出版社
www.buptpress.com

21 世纪高等院校计算机系

Java 语言实用教程

主 编 丁振凡

副主编 黎 章 曾纯青 汪 斌



北京邮电大学出版社
·北京·

内 容 简 介

本书将 Java 语言与面向对象编程紧密结合,在建立面向对象概念的基础上,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溶入到具体的 Java 程序实例中。本书主要内容有面向对象基础、Java 语言概述、Java 语言基础、Java 面向对象技术、常用类库、异常处理、图形界面编程、多线程、输入输出、JDBC 与数据库、网络编程、Swing 编程等。本书注意理论与实际的结合,例题短小、精练,将实用性与趣味性相结合,在讲清相关概念的同时注意培养读者运用知识的能力。

本书可作为高等院校相关专业的教材和教学参考书,也可作为读者学习 Java 语言的自学用书。

图书在版编目(CIP)数据

Java 语言实用教程/丁振凡主编. —北京:北京邮电大学出版社,2004
ISBN 7-5635-0970-4

I. J... II. 丁... III. Java 语言—程序设计—教材 IV. TP312

中国版本图书馆 CIP 数据核字(2004)第 140080 号

书 名: Java 语言实用教程

主 编: 丁振凡

责任编辑: 时友芬 王玲娣

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)

电话传真: 010-62282185(发行部) 010-62283578(FAX)

电子信箱: publish@bupt.edu.cn

经 销: 各地新华书店

印 刷: 北京源海印刷有限责任公司

开 本: 787 mm × 1 092 mm 1/16

印 张: 18.5

字 数: 449 千字

印 数: 3 001—6 000 册

版 次: 2005 年 2 月第 1 版 2005 年 8 月第 2 次印刷

ISBN 7-5635-0970-4/TP·133

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 语言的诞生到现在,在短短的 9 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。暂且不谈该预言的准确性,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、多态等特性溶入具体代码设计中;第 5 章介绍了 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,还介绍了常用 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的思路;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,还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,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 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 语言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 语言,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 语言的类库,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,仅仅了解语言的语法和内容体系的计算机编程员并不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,才能提高自己的编程能力。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,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。这点我们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,也可以作为读者学习 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,它也许正好能用在你的实际项目开发中。希望学完本书后,读者能对 Java 语言的基础内容体系有较全面的了解,同时使自己的程序设计能力有较大的提高。

本书第 1 章由汪斌编写,第 2 章至第 11 章由丁振凡编写,第 12、13 章由黎章、曾纯青编写,全书由丁振凡统稿。

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,则可以将后面 3 章作为选读内容。

由于编者水平有限,加之时间仓促,疏漏及错误之处在所难免,恳请广大读者批评指正。

编 者

2004 年 12 月于南昌

目 录

第 1 章 Java 语言概述

1.1 程序设计语言与 Java 语言	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	3
1.3 Java 语言的开发和运行环境	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	11
1.5.3 健壮和安全的语言	11
1.5.4 支持多线程	12
1.5.5 面向网络的语言	12
1.6 本章小结	12
习题	12

第 2 章 Java 语言基础

2.1 Java 语言符号	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	16
2.2 数据类型与变量	17
2.2.1 数据类型	17

2.2.2 常量	18
2.2.3 变量	19
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	25
2.3.3 逻辑运算符	25
2.3.4 位运算符	26
2.3.5 赋值组合运算符	28
2.3.6 其他运算符	28
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	46
2.6.1 方法声明	47
2.6.2 方法调用	47
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 语言的命令行参数	58
2.8 本章小结	59
习题	60

第3章 类与对象

3.1 Java 语言的类	63
3.1.1 系统定义的类	63
3.1.2 用户自定义类	63
3.2 对象的创建与引用	65
3.2.1 创建对象及访问对象成员	65
3.2.2 对象的初始化和构造方法	66
3.3 变量作用域	68

3.4 类变量和静态方法·····	70
3.4.1 类变量·····	70
3.4.2 静态方法·····	71
3.5 使用包组织类·····	71
3.5.1 建立包·····	71
3.5.2 包的引用·····	73
3.6 本章小结·····	74
习题·····	74

第4章 继承、多态和接口

4.1 继承·····	77
4.1.1 Java 继承的实现·····	77
4.1.2 构造方法在类继承中的作用·····	78
4.1.3 变量的继承、隐藏·····	79
4.2 访问控制符·····	80
4.2.1 公共访问控制符·····	80
4.2.2 缺省访问控制符·····	80
4.2.3 私有访问控制符·····	81
4.2.4 保护访问控制符·····	81
4.3 多态性·····	83
4.3.1 方法的重载·····	83
4.3.2 方法的覆盖·····	84
4.4 this 和 super·····	84
4.4.1 this 的应用·····	84
4.4.2 通过 super 访问父类成员·····	85
4.5 final 修饰符的使用·····	87
4.5.1 final 作为类修饰符·····	87
4.5.2 用 final 修饰方法·····	87
4.5.3 用 final 定义常量·····	87
4.6 抽象类和抽象方法·····	88
4.6.1 抽象类的定义·····	88
4.6.2 抽象类的实现·····	89
4.7 接口·····	90
4.7.1 接口定义·····	90
4.7.2 接口的实现·····	90
4.8 内嵌类·····	92

4.8.1 成员类	93
4.8.2 静态 inner 类	95
4.8.3 方法中的内嵌类与匿名内嵌类	96
4.9 对象引用转换	97
4.9.1 对象引用赋值转换	97
4.9.2 对象引用强制转换	98
4.10 本章小结	98
习题	99

第 5 章 常用系统类

5.1 语言基础类	101
5.1.1 Object 类	101
5.1.2 Math 类	101
5.1.3 数据类型包装类	103
5.2 字符串	104
5.2.1 String 类	104
5.2.2 StringBuffer 类	109
5.3 Vector 类	110
5.3.1 向量的各种访问	111
5.3.2 向量的应用举例	112
5.4 Collection API 简介	115
5.5 日期和时间	117
5.5.1 Date 类	117
5.5.2 Calendar 类	118
5.6 本章小结	118
习题	119

第 6 章 Java Applet

6.1 什么是 Applet	120
6.2 Applet 方法介绍	121
6.3 Applet 的 AWT 绘制	123
6.3.1 Java 图形坐标	123
6.3.2 各类图形的绘制方法	123
6.3.3 显示文字	124
6.3.4 颜色控制	126
6.3.5 Java 2D 图形绘制	127

6.4 Applet 参数传递	132
6.4.1 在 HTML 文件中给 Applet 提供参数	132
6.4.2 Applet 代码中读取 Applet 参数值	132
6.5 Applet 的多媒体支持	134
6.5.1 绘制图像	134
6.5.2 实现动画	136
6.5.3 播放声音文件	137
6.6 Java 存档文件	138
6.6.1 创建存档(JAR)文件	138
6.6.2 在 HTML 文件中指定 Applet 的存档文件	139
6.7 本章小结	140
习题.....	140

第7章 图形用户界面编程

7.1 图形用户界面核心概念	141
7.1.1 一个简单的 GUI 示例	141
7.1.2 创建窗体	142
7.1.3 创建 GUI 部件	143
7.1.4 事件处理	144
7.1.5 在事件处理代码中区分事件源	146
7.1.6 关于事件适配器类	148
7.2 容器与布局管理	150
7.2.1 流式布局(FlowLayout)	150
7.2.2 边缘或方位布局(BorderLayout)	151
7.2.3 网格布局(GridLayout)	152
7.2.4 卡片式布局(CardLayout)	153
7.2.5 网格块布局(GridBagLayout)	157
7.3 常用 GUI 标准组件	160
7.3.1 GUI 标准组件概述	160
7.3.2 文本框与文本域	161
7.3.3 选项按钮与列表的使用	163
7.3.4 滚动条的使用	171
7.4 鼠标和键盘事件	173
7.4.1 鼠标事件	173
7.4.2 键盘事件	177
7.5 菜单的使用	179

7.5.1 下拉菜单	179
7.5.2 弹出式菜单	183
7.6 对话框的使用	186
7.6.1 对话框的创建与使用	186
7.6.2 文件对话框	188
7.7 本章小结	189
习题	190

第8章 异常处理

8.1 异常的概念	191
8.1.1 什么是异常	191
8.1.2 异常的类层次	192
8.1.3 系统定义的异常	193
8.2 异常的处理	194
8.2.1 try...catch...finally 结构	194
8.2.2 多异常的处理举例	195
8.3 自定义异常	196
8.3.1 自定义异常类设计	196
8.3.2 抛出异常	197
8.3.3 方法的异常声明	197
8.4 本章小结	198
习题	199

第9章 流式输入/输出与文件处理

9.1 输入/输出基本概念	201
9.1.1 输入/输出设备与文件	201
9.1.2 流的概念	201
9.2 面向字节的输入/输出流	202
9.2.1 面向字节的输入流	202
9.2.2 面向字节的输出流	206
9.3 面向字符的输入/输出流	209
9.3.1 面向字符的输入流	209
9.3.2 面向字符的输出流	211
9.4 文件处理	212
9.4.1 文件与目录管理	212
9.4.2 文件的顺序访问	214

9.4.3 文件的随机访问	215
9.5 对象串行化	216
9.6 本章小结	219
习题.....	220

第 10 章 多线程

10.1 Java 线程的概念.....	221
10.1.1 线程的状态.....	221
10.1.2 线程的调度与优先级.....	222
10.2 Java 语言多线程编程方法.....	222
10.2.1 Thread 类简介	222
10.2.2 继承 Thread 类实现多线程	223
10.2.3 实现 Runnable 接口编写多线程	225
10.3 线程的控制.....	226
10.3.1 放弃运行(Yielding)	226
10.3.2 挂起(Suspending)	226
10.3.3 睡眠一段时间(Sleeping)	227
10.3.4 阻塞(Blocking)	227
10.3.5 关于用户线程和看守线程(Daemon)	227
10.4 线程资源的同步处理	227
10.4.1 临界资源问题.....	227
10.4.2 wait()和 notify()方法	228
10.4.3 生产者与消费者模型.....	229
10.4.4 死 锁	231
10.5 本章小结.....	232
习题.....	232

第 11 章 JDBC 技术和数据库应用

11.1 关系数据库概述.....	233
11.2 JDBC	233
11.2.1 JDBC 驱动程序	233
11.2.2 ODBC 数据源配置.....	234
11.2.3 JDBC API	236
11.3 JDBC 基本应用	238
11.3.1 数据库查询.....	238
11.3.2 数据库的更新.....	242

11.3.3 用 PreparedStatement 类实现 SQL 操作	243
11.4 数据库应用举例	245
11.5 本章小结	253
习题	253

第 12 章 Java 的网络编程

12.1 网络编程基础	255
12.1.1 网络协议	255
12.1.2 InetAddress 类	255
12.2 Socket 通信	256
12.2.1 Java 的 Socket 编程原理	256
12.2.2 简单多用户聊天程序的实现	258
12.3 无链接的数据报	262
12.3.1 DatagramPacket 类	262
12.3.2 DatagramSocket 类	262
12.3.3 发送和接收过程	262
12.4 URL	265
12.4.1 URL 类	265
12.4.2 URLConnection 类	267
12.4.3 用 Applet 方法访问 URL 资源	268
12.5 本章小结	270
习题	270

第 13 章 Swing 编程

13.1 Swing 包简介	271
13.2 Swing 包典型部件的使用	271
13.2.1 JFrame 类	271
13.2.2 Swing 中的按钮和标签	272
13.2.3 滚动窗格	274
13.2.4 工具栏	275
13.2.5 Swing 中的对话框	276
13.2.6 选项卡(JTabbedPane)	280
13.3 本章小结	282
习题	282

参考文献	283
------------	-----

第 1 章 Java 语言概述

1.1 程序设计语言与 Java 语言

计算机从诞生到现在也只不过短短半个世纪左右。从计算机诞生那一天起人们就为了能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——计算机语言。

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。那时候,除了计算机它自己的设计者之外,恐怕没有几个人能够完全掌握它了。所以后来,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 语言、C 语言、Pascal 语言和 BASIC 语言等,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 语言、Delphi 语言,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 语言来自于 Sun 公司的一个叫 Green 的项目,其原先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,准备采用 C++ 语言,但 C++ 语言太复杂,安全性差,最后基于 C++ 开发一种新的语言 Oak 语言(Java 语言的前身),Oak 语言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,但结果是被 SGI 打败。可怜 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语言编制了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语言进军Internet。Java的取名也有一趣闻,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语言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语言已成为软件开发的主流技术,引起了世界范围学习和使用Java的热潮。

计算机产业的许多大公司购买了Java语言的许可证,如IBM、Novell、Oracle、SGI和Borland等,Microsoft公司一开始对Java语言无动于衷,比尔盖茨在仔细研究了Java语言的技术规范之后,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购买了Java语言的使用许可证并很快推出了集成化的Java语言开发环境——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语言内容体系还包括:

- 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语言和C语言基本上都是面向过程的语言,主要是采用数学语言方式编程的语言,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,必须通过过程操作来改变数据,因此,函数调用在面向过程编程中被大量使用。

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和清晰。

面向对象的软件开发中将世界上的事物均看作对象,任一对象有两个特征: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行为方法,消息中还包括方法的参数。

1.2.2 面向对象程序设计的特性

1. 封装性

面向对象的第一个原则是把数据和对数据的操作都封装在一个类中,比如说笔记本电脑就是一个类。笔记本电脑这个类有许多属性,有些是静态的(数据),例如:显示屏的亮度,本身的重量;有些是动态的(对数据的操作),例如: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。例如:笔者所使用的电脑,就是一台具体的笔记本电脑,它是一个对象,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,即使是配置相同,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,比如说它们都是电脑、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。只不过类是抽象的东西,而对象是具体的,对象具有它隶属类的所有属性,包括静态的和动态的。

每个对象都有自己的存储空间,可容纳其他对象,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高的复杂程度。

每个对象都属于某个类。根据语法,每个对象都是某个“类”的一个“实例”。一个类的最重要的特征就是“能将什么消息发给它?”,也就是类本身有哪些操作。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特征,而方法则描述对象的行为。

类名
属性 1:类型
属性 2:类型
方法 1(参数表)
方法 2(参数表)

图 1-1 类和对象的表示

2. 继承性

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。

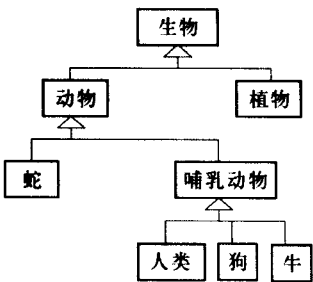


图 1-2 类的继承层次

“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,这样的效果可让软件的开发加速,同时可达到程序共享的好处。通过继承可以直接继承系统提供的或他人写作的类,并加以改写,进而得到功能的不断扩充。

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的子类对象的方法,但不同子类中行为的具体实现可能是不一样的。

子类中对父类行为的重新定义是下面将介绍的多态性的一种体现。

3. 多态性

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:即在同一个类中某个方法有多种形态,其方法名相同,但参数不同,所以也称参数多态;

(2) 方法的覆盖:在子类中重新再定义一个与父类有相同形态的方法,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。参数多态提供了方法的多种使用形式,这样方便使用者的调用;而覆盖多态则使得用户可以用同样的方式对待不同的对象,不同的对象可以用它们自己的方式响应同一消息。用户可以通过父类去引用子类的对象,执行方法时则表现出每个子类对象各自的行为。

4. 抽象性

这里的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 语言中有一个类 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。用户可以用 Object 类的变量去引用任何子类的对象,通过 Object 引用能访问的成员,在子类中总是存在的。

第二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。同一类创建的对象具有共同的属性,但属性值不同。

1.3 Java 语言的开发和运行环境

Java 语言的开发和运行环境有很多,比如:

- Sun 公司的 JDK、NetBeans、Java Studio 5、Java Workshop、Jcreator 等;
- Borland 公司的 Jbuilder;
- IBM 公司的 Visual Age for Java;
- BEA 公司的 WebLogic Workshop;
- Macromedia 公司的 JRUN。

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 SUN 公司的主页 <http://java.sun.com/> 处下载,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(简称 JRE)主要担负三大任务:

- 加载代码——由类加载器执行;
- 检验代码——由字节码校验器执行;
- 执行代码——由运行时解释执行的。

JDK 下载后需要安装,比如安装在 C:\jdk14 目录下,其他采用默认安装即可,安装完后会在此目录下建立 5 个子目录:

bin 目录是二进制文件存放目录,所有 JDK 的可执行程序都是放在此目录下,像 javac、