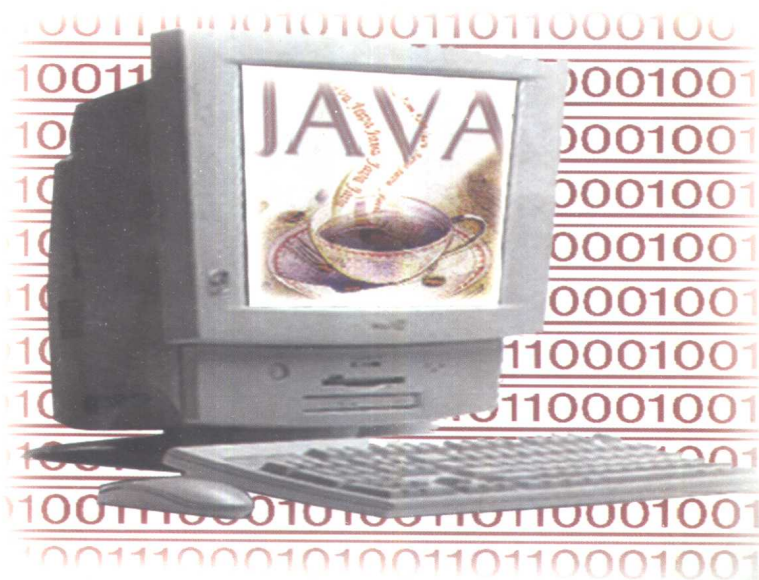


清华大学计算机基础教育课程系列教材

# Java 语言 与面向对象程序设计

印 旻 编著  
王行言 审校



清华大学出版社  
<http://www.tup.tsinghua.edu.cn>





清华大学计算机基础教育课程系列教材

TP312  
534

# Java语言 与面向对象程序设计

印 旻 编著  
王行言 审校



清华大学出版社  
<http://www.tup.tsinghua.edu.cn>

JS66-100-1

**(京)新登字 158 号**

## 内 容 简 介

本书将 Java 语言编程技术和面向对象程序设计两部分内容相结合,在讲授 Internet 上最流行的编程工具 Java 的同时,介绍了它所采用的面向对象技术的基础理论、主要原则和思维方法。书中首先介绍了面向对象软件开发和 Java 语言的基础知识;然后阐述了面向对象程序设计的基本原则与特点,并借助于 Java 把这些原则和特点融入具体的程序中,帮助读者建立由感性到理性的深入认识;最后介绍了 Java 编程的必备工具,包括类库、常用算法、GUI、Swing 和网络编程等。全书不仅介绍理论,更强调实际运用,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的实例,便于初学者入门。

本书可以作为大中专院校低年级学生的教材,也可作为其他同等程度读者的自学用书。

**版权所有,翻印必究。**

**本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。**

### 图书在版编目(CIP)数据

Java 语言与面向对象程序设计/印旻编著. —北京:清华大学出版社,2000  
清华大学计算机基础教育课程系列教材  
ISBN 7-302-03927-5

I. J… II. 印… 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(2000)第 31268 号

出版者:清华大学出版社(北京清华大学学研大厦,邮编 100084)

<http://www.tup.tsinghua.edu.cn>

印刷者:北京市人民文学印刷厂

发行者:新华书店总店北京发行所

开 本:787×1092 1/16 印张:20.25 字数:464 千字

版 次:2000 年 9 月第 1 版 2000 年 9 月第 1 次印刷

书 号:ISBN 7-302-03927-5/TP·2295

印 数:0001~8000

定 价:23.50 元

# 序

计算机科学技术的发展不仅极大地促进了整个科学技术的发展,而且明显地加快了经济信息化和社会信息化的进程。因此,计算机教育在各国倍受重视,计算机知识与能力已成为 21 世纪人才素质的基本要素之一。

清华大学自 1990 年开始将计算机教学纳入基础课的范畴,作为校重点课程进行建设和管理,并按照“计算机文化基础”、“计算机技术基础”和“计算机应用基础”三个层次的课程体系组织教学:

第一层次“计算机文化基础”的教学目的是培养学生掌握在未来信息化社会里更好地学习、工作和生活所必须具备的计算机基础知识和基本操作技能,并进行计算机文化道德规范教育。

第二层次“计算机技术基础”是讲授计算机软硬件的基础知识、基本技术与方法,从而为学生进一步学习计算机的后续课程,并利用计算机解决本专业及相关领域中的问题打下必要的基础。

第三层次“计算机应用基础”则是讲解计算机应用中带有基础性、普遍性的知识,讲解计算机应用与开发中的基本技术、工具与环境。

以上述课程体系为依据,设计了计算机基础教育系列课程。随着计算机技术的飞速发展,计算机教学的内容与方法也在不断更新。近几年来,清华大学不断丰富和完善教学内容,在有关课程中先后引入了面向对象技术、多媒体技术、Internet 与互联网技术等。与此同时,在教材与 CAI 课件建设、网络化的教学环境建设等方面也正在大力开展工作,并积极探索适应 21 世纪人才培养的教学模式。

为进一步加强计算机基础教学工作,适应高校正在开展的课程体系与教学内容的改革,及时反映清华大学计算机基础教学的成果,加强与兄弟院校的交流,清华大学在原有工作的基础上,重新规划了“清华大学计算机基础教育课程系列教材”。

该系列教材有如下几个特色:

1. 自成体系:该系列教材覆盖了计算机基础教学三个层次的教学内容。其中既包括所有大学生都必须掌握的计算机文化基础,也包括适用于各专业的软、硬件基础知识;既包括基本概念、方法与规范,也包括计算机应用开发的工具与环境。

2. 内容先进:该系列教材注重将计算机技术的最新发展适当地引入教学中来,保持了教学内容的先进性。例如,系列教材中包括了面向对象与可视化编程、多媒体技术与应用、Internet 与互联网技术、大型数据库技术等。

3. 适应面广：该系列教材照顾了理、工、文等各种类型专业的教学要求。

4. 立体配套：为适应教学模式、教学方法和手段的改革，该系列教材中多数都配有习题集和实验指导、多媒体电子教案，有的还配有 CAI 课件以及相应的网络教学资源。

本系列教材源于清华大学计算机基础教育的教学实践，凝聚了工作在第一线的任课教师的教学经验与科研成果。我希望本系列教材不断完善，不断更新，为我国高校计算机基础教育做出新的贡献。



1999 年 12 月

---

注：周远清，现任教育部副部长，原清华大学副校长、计算机专业教授。

# 前 言

本书是作者在清华大学开设的一门深受同学欢迎的同名公共任选课的教材。此课程自 1997 年秋季开设以来期期爆满,每学期报名选课的人数都逾千人,一度成为清华大学选课人数最多的公共任选课。为满足同学们强烈的求知欲,这门课程的内容也随着 Java 语言与面向对象编程技术的发展而不断更新。本书既凝聚了两年以来作者的教学经验和体会,也包含了最新的 Java 编程技术,书中的所有例题和程序都在 Java2 环境中编译通过并运行成功。

面向对象技术被称为是程序设计方法学的一场革命,它已经逐步替代了面向过程的程序设计技术,成为计算机应用开发领域的主流趋势,这是因为这种技术所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现代大规模软件开发的要求和潮流。Java 语言是面向对象技术成功应用的著名范例之一。诞生于 1995 年的 Java 语言在短短的四五年间席卷全球,以 20 世纪末网络科技和网络经济所特有的令人瞠目结舌的速度迅速发展。有人预言,不久的将来全世界 90% 的程序代码将用 Java 语言书写或改写。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。本书将以 Java 语言为载体,在介绍 Java 编程的同时讲解面向对象程序设计的主要原则和方法。

本书是面向大中专院校的低年级学生及其他同等程度的对 Java 语言和面向对象编程技术感兴趣的读者编写的。书中对内容的编排、剪裁和例题选择都作了严格的控制,确保了一定的深度和广度,在难易程度上亦遵循由浅入深、循序渐进的原则。本书可以作为大中专院校的公共课教材,也可以作为读者的自学用书。学习本书之前应该对计算机操作有一定的认识,但是不必有开发经验。接触过一门高级语言的读者会感到学习本书得心应手,从未编过程序的读者只要对于前三章多花些心思,一样可以获得良好的学习效果。

本书第 1 章概述了面向对象软件开发,包括面向对象问题求解的提出、对象的概念以及面向对象软件开发的一般过程,是学习面向对象软件开发的基础知识。第 2 和第 3 章介绍了 Java 编程的基础知识,包括数据类型、变量、表达式和流程控制语句。第 4,5 章以 Java 的面向对象编程为例,详细地讨论了面向对象技术的四大特点:封装、抽象、继承和多态及其在面向对象程序设计中的具体应用。通过这两章的学习,读者可以对面向对象技术和 Java 的面向对象编程有较为扎实的了解和掌握。要掌握好 Java 语言并具有利用它解决实际问题的能力,仅仅学习语法规则是不够的,还需要熟练掌握 Java 的应用程序编程接口,即 Java 的类库。本书从第 6 章开始介绍 Java 的常用标准类库及编程技巧。第 6 章介绍 Java 的类库结构、数组、字符串和查找、排序这些常用算法以及堆栈、链表、树等常见数据结构。第 7 章介绍 Java 图形界面的设计和编程实现。第 8 章介绍 Java 的异常处理、多线程编程、Swing GUI 组件、输入输出处理和网络编程。

计算机学科是注重实践的学科,优秀的软件开发人员无不经过大量上机实践的磨炼,只有在学习书本内容的同时辅以相应的实际练习和实验环节,才能真正掌握书中介绍的知识 and 技能。为此本书中引入了大量的例题,只要读者能够按照书中的要求边学边练,就一定能很快登堂入室,享受在 Java 语言和面向对象技术所构造的无限畅想空间中遨游的乐趣。

最后感谢所有帮助作者完成此书的领导、同事和家人。王行言老师审阅了全书,王蕾老师也对书稿提出了许多修改意见,在此表示衷心的感谢。特别要感谢热情支持和参与本书同名课程的广大清华大学的同学们,正是他们活跃的思维和永无止境的求知欲帮助作者发现错误,鞭策作者不断改进。当然由于水平所限,书中仍然难免疏漏,欢迎各位同行和广大读者批评指正。

作 者

2000 年 2 月

# 目 录

<b>第 1 章 面向对象软件开发概述</b> .....	1
1.1 面向对象问题求解的提出 .....	1
1.2 面向对象问题求解概述 .....	2
1.3 对象、类与实体.....	3
1.4 对象的属性与相互关系 .....	4
1.4.1 对象的属性.....	4
1.4.2 对象的关系.....	5
1.5 面向对象的软件开发过程 .....	6
1.5.1 面向对象的分析.....	6
1.5.2 面向对象的设计.....	8
1.5.3 面向对象的实现.....	8
1.6 面向对象程序设计方法的优点 .....	9
1.6.1 可重用性.....	9
1.6.2 可扩展性 .....	10
1.6.3 可管理性 .....	11
1.7 小结.....	12
习题 .....	12
<b>第 2 章 Java 概述</b> .....	13
2.1 第一个 Java Application 程序 .....	13
2.1.1 源程序编辑 .....	13
2.1.2 字节码的编译生成 .....	14
2.1.3 字节码的解释与运行 .....	16
2.2 第一个 Java Applet 程序 .....	17
2.2.1 源程序的编辑与编译 .....	17
2.2.2 代码嵌入 .....	19
2.2.3 Applet 的运行 .....	20
2.3 图形界面的输入输出.....	22
2.3.1 Java Applet 图形界面输入输出.....	22
2.3.2 Java Application 图形界面输入输出.....	24
2.4 字符界面的输入输出.....	27
2.5 Java 语言的特点 .....	29



---

2.6 小结	30
习题	30
<b>第3章 Java 语言基础</b>	<b>31</b>
3.1 Java 程序的构成	31
3.2 数据类型、变量与常量	32
3.2.1 数据类型	32
3.2.2 标识符	33
3.2.3 常量	34
3.2.4 变量	35
3.3 表达式	39
3.3.1 赋值与强制类型转换	39
3.3.2 算术运算	40
3.3.3 关系运算	43
3.3.4 逻辑运算	44
3.3.5 位运算	45
3.3.6 其他运算符	46
3.3.7 运算符的优先级与结合性	47
3.3.8 注释	47
3.4 流程控制语句	48
3.4.1 结构化程序设计的三种基本流程	48
3.4.2 分支语句	49
3.4.3 循环语句	52
3.4.4 跳转语句	56
3.5 小结	57
习题	57
<b>第4章 抽象、封装与类</b>	<b>59</b>
4.1 抽象与封装	59
4.1.1 抽象	59
4.1.2 封装	60
4.2 Java 的类	60
4.2.1 系统定义的类	60
4.2.2 用户程序自定义类	63
4.2.3 创建对象与定义构造函数	65
4.3 类的修饰符	69
4.3.1 抽象类	69
4.3.2 最终类	70

---

4.4	域	71
4.5	方法	75
4.6	访问控制符	80
4.7	小结	85
	习题	85
<b>第5章</b>	<b>继承与多态</b>	<b>87</b>
5.1	继承	87
5.2	Java 的继承	88
5.2.1	派生子类	88
5.2.2	域的继承与隐藏	91
5.2.3	方法的继承与覆盖	95
5.2.4	this 与 super	98
5.3	多态	102
5.4	Java 的重载	104
5.5	构造函数的继承与重载	107
5.6	包	113
5.6.1	创建包	113
5.6.2	包的引用	115
5.7	接口	116
5.7.1	接口概述	116
5.7.2	声明接口	117
5.7.3	实现接口	119
5.8	小结	120
	习题	121
<b>第6章</b>	<b>工具类与算法</b>	<b>124</b>
6.1	语言基础类库	124
6.1.1	Object 类	124
6.1.2	数据类型类	124
6.1.3	Math 类	126
6.1.4	System 类	126
6.2	Applet 类与 Applet 小程序	127
6.2.1	Applet 的基本工作原理	127
6.2.2	Applet 类	128
6.2.3	HTML 文件参数传递	132
6.3	数组	134
6.4	向量	136

6.5	字符串 .....	139
6.5.1	String 类 .....	139
6.5.2	StringBuffer 类 .....	143
6.5.3	Java Application 命令行参数 .....	145
6.6	递归 .....	147
6.7	排序 .....	150
6.7.1	冒泡排序 .....	150
6.7.2	选择排序 .....	154
6.7.3	插入排序 .....	155
6.7.4	桶排序 .....	156
6.8	查找 .....	158
6.9	链表 .....	162
6.9.1	链表的节点 .....	162
6.9.2	创建链表 .....	163
6.9.3	遍历链表 .....	165
6.9.4	链表的插入操作 .....	166
6.9.5	链表的删除操作 .....	166
6.10	队列 .....	168
6.11	堆栈 .....	170
6.12	二叉树 .....	173
6.13	小结 .....	180
	习题 .....	180
<b>第 7 章</b>	<b>图形用户界面的设计与实现 .....</b>	<b>183</b>
7.1	图形用户界面概述 .....	183
7.2	用户自定义成分 .....	185
7.2.1	绘制图形 .....	185
7.2.2	显示文字 .....	186
7.2.3	控制颜色 .....	188
7.2.4	显示图像 .....	190
7.2.5	实现动画效果 .....	191
7.3	Java 的事件处理 .....	192
7.4	GUI 标准组件概述 .....	195
7.5	标签、按钮与动作事件 .....	197
7.6	文本框、文本区域与文本事件 .....	200
7.7	单、复选按钮,列表与选择事件 .....	203
7.8	滚动条与调整事件 .....	213
7.9	画布与鼠标、键盘事件 .....	217

7.10 布局设计	224
7.10.1 FlowLayout	224
7.10.2 BorderLayout	225
7.10.3 CardLayout	227
7.10.4 GridLayout	228
7.11 Panel 与容器事件	230
7.12 Frame 与窗口事件	233
7.13 菜单的定义与使用	236
7.14 对话框、组件事件与焦点事件	242
7.15 小结	247
习题	247
<b>第 8 章 Java 高级编程</b>	<b>249</b>
8.1 异常处理	249
8.1.1 异常与异常类	249
8.1.2 抛出异常	252
8.1.3 异常的处理	253
8.2 Java 多线程机制	256
8.2.1 Java 中的线程	256
8.2.2 Java 的线程类与 Runnable 接口	258
8.2.3 如何在程序中实现多线程	259
8.3 Swing GUI 组件	264
8.3.1 JApplet	264
8.3.2 JButton	265
8.3.3 JSlider	268
8.3.4 JPasswordField	270
8.3.5 JTabbedPane	271
8.4 流式输入输出与文件处理	273
8.4.1 Java 输入输出类库	274
8.4.2 文件的处理与随机访问	279
8.5 用 Java 实现底层网络通信	288
8.5.1 基于连接的流式套接字	288
8.5.2 无连接的数据报	296
8.6 Java 程序对网上资源的访问	301
8.7 小结	308
习题	308

# 第 1 章

## 面向对象软件开发概述

面向对象的软件开发和相应的面向对象的问题求解是当今计算机技术发展的重要成果和趋势之一。本章将集中介绍面向对象软件开发和面向对象程序设计中的基本概念和基本方法,使读者对面向对象软件开发方法的体系、原则、基本思想和特点有一定的了解。

### 1.1 面向对象问题求解的提出

早期计算机中运行的程序大都是为特定的硬件系统专门设计的,称为面向机器的程序。这类程序的运行速度和效率都很高,但是可读性和可移植性很差,随着软件开发规模的扩大,这类面向机器的程序逐渐被以 FORTRAN、C 等为代表的面向过程的程序所取代。

面向过程的程序遵循面向过程的问题求解方法。其中心思想是用计算机能够理解的逻辑来描述和表达待解决的问题及其具体的解决过程。数据结构、算法是面向过程问题求解的核心组成。其中数据结构利用计算机的离散逻辑来量化表达需要解决的问题,而算法则研究如何快捷、高效地组织解决问题的具体过程。面向过程的问题求解可以精确、完备地描述具体的求解过程(这里的过程通常是指操作),但却不足以把一个包含了多个相互关联的过程的复杂系统表述清楚,而面向对象的问题求解则可以胜任这件工作。面向对象问题求解关心的不仅仅是孤立的单个过程,而是孕育所有这些过程的母体系统,它能够使计算机逻辑来模拟描述系统本身,包括系统的组成,系统的各种可能状态,以及系统中可能产生的各种过程与过程引起的系统状态切换。

面向对象技术代表了一种全新的程序设计思路和观察、表述、处理问题的方法,与传统的面向过程的开发方法不同,面向对象的程序设计和问题求解力求符合人们日常自然的思维习惯,降低、分解问题的难度和复杂性,提高整个求解过程的可控制性、可监测性和可维护性,从而达到以较小的代价和较高的效率获得较满意效果的目的。

最早的面向对象的软件是 1966 年推出的 Simula I,它首次提出模拟人类的思维方法,把数据和相关的操作集成在一起的思想。但是由于当时硬件条件的局限和方法本身不够成熟,这种技术没有得到推广和使用。随着软件危机的出现和过程化开发方法固有局限性的暴露,人们把目光重新转回到面向对象的方法上来。1980 年提出的 Smalltalk-80 语言已经是一种比较成熟、有效的面向对象的工具了,利用 Smalltalk-80 也确实实现了一些面向对象的应用,但是这个语言更重要的作用是提出了一种新的思想观念和解决问题的新思路和新方法,它向人们展示了面向对象这个虽然稚嫩,但却充满希望的发展方向。其

后,先后产生了 Lisp、Clascal、Object Pascal、C++ 等多种面向对象的语言,这中间最有影响,也是对面向对象技术的普及推动最大的当属 C++。

C++ 语言在兼容原有最流行的 C 语言的基础之上,加入了面向对象的有关内容和规则。由于它的很多语法规则与 C 语言相近,所以很容易为广大的 C 程序员所接受;同时 C++ 所具有的面向对象功能简化了应用软件的开发、设计和维护,为开发大型软件提供了很大的方便。C++ 的广泛推广和成功应用证明了新兴的面向对象技术的实力和前景,C++ 也正在取代 C 而成为主流编程语言。

Java 是 90 年代新出现的面向对象的编程语言。相对于 C++,Java 去除了其中为了兼容 C 语言而保留的非面向对象的内容,使程序更加严谨、可靠、易懂。尤其是 Java 所特有的“一次编写、多次使用”的跨平台优点,使得它特别适合在网络应用开发中使用,成为面向对象开发工具中极具潜力的一员。

面向对象的程序设计方法的出现和广泛应用是计算机软件技术发展中的一个重大变革和飞跃。相对于之前的程序设计方法,面向对象技术能够更好地适应当今软件开发在规模、复杂性、可靠性和质量、效率上的种种需求,因而被越来越多地推广和使用,其方法本身也在这诸多实践的检验和磨练中日趋成熟、标准化和体系化,逐渐成为目前公认的主流程序设计方法。

## 1.2 面向对象问题求解概述

不同于面向过程的程序设计中以具体的解题过程为研究和实现的主体,面向对象的程序设计是以需解决的问题中所涉及到的各种对象为主要矛盾。

在面向对象的方法学中,“对象”是现实世界的实体或概念在计算机逻辑中的抽象表示。具体地,对象是具有唯一对象名和固定对外接口的一组属性和操作的集合,用来模拟组成或影响现实世界问题的一个或一组因素。其中对象名是区别于其他对象的标志;对外接口是对象在约定好的运行框架和消息传递机制中与外界通信的通道;对象的属性表示了它所处的状态;而对象的操作则用来改变对象的状态达到特定的功能。对象的最主要特点是以数据为中心,它是一个集成了数据和其上操作的独立、自恰的逻辑单位。

面向对象的问题求解就是力图从实际问题中抽象出这些封装了数据和操作的对象,通过定义属性和操作来表述它们的特征和功能,通过定义接口来描述它们的地位及与其他对象的关系,最终形成一个广泛联系的可理解、可扩充、可维护、更接近于问题本来面目的动态对象模型系统。

面向对象的程序设计将在面向对象的问题求解所形成的对象模型基础之上,选择一种面向对象的高级语言来具体实现这个模型。相对于传统的面向过程的程序设计方法,面向对象的程序设计具有如下的优点:

(1) 对象的数据封装特性彻底消除了传统结构方法中数据与操作分离所带来的种种问题,提高了程序的可复用性和可维护性,降低了程序员保持数据与操作相容的负担。

(2) 对象的数据封装特性还可以把对象的私有数据和公共数据分离开,保护了私有数据,减少了可能的模块间干扰,达到降低程序复杂性、提高可控性的目的。

(3) 对象作为独立的整体具有良好的自恰性,即它可以通过自身定义的操作来管理自己。一个对象的操作可以完成两类功能,一是修改自身的状态,二是向外界发布消息。当一个对象欲影响其他的对象时,它需要调用其他对象自身的方法,而不是直接去改变那个对象。对象的这种自恰性能使得所有修改对象的操作都以对象自身的一部分的形式存在于对象整体之中,维护了对象的完整性,有利于对象在不同环境下的复用、扩充和维护。

(4) 在具有自恰性的同时,对象通过一定的接口和相应的消息机制与外界相联系。这个特性与对象的封装性结合在一起,较好地实现了信息的隐藏。即对象成为一只使用方便的“黑匣子”,其中隐藏了私有数据和细节内容。使用对象时只需要了解其接口提供的功能操作即可,而不必了解对象内部的数据描述和具体的功能实现。

(5) 继承是面向对象方法中除封装外的另一个重要特性,通过继承可以很方便地实现应用的扩展和已有代码的重复使用,在保证质量的前提下提高了开发效率,使得面向对象的开发方法与软件工程的新兴方法——快速原型法很好地结合在一起。

综上所述,面向对象程序设计是将数据及数据的操作封装在一起,成为一个不可分割的整体,同时将具有相同特征的对象抽象成为一种新的数据类型——类。通过对象间的消息传递使整个系统运转。通过对象类的继承提供代码重用的有效途径。

在面向对象程序设计方法中,其程序结构是一个类的集合和各类之间以继承关系联系起来的结构,有一个主程序,在主程序中定义各对象并规定它们之间传递消息的规律。从程序执行这一角度来看,可以归结为各对象和它们之间的消息通信。面向对象程序设计最主要的特征是各对象之间的消息传递和各类之间的继承。

实际上,面向对象的程序设计只是面向对象方法学的一个组成部分。完整地看,面向对象的问题求解还应该包括面向对象的分析和面向对象的设计。但是由于面向对象的方法学尚未完全成熟及得到广泛意义上的公认,所以在本书里将把重点放在它可操作性较强的方面,即结合 Java 语言主要介绍面向对象的程序设计方法。

### 1.3 对象、类与实体

对象的概念是面向对象技术的核心所在。以面向对象的观点看来,所有的面向对象的程序都是由对象来组成的,这些对象首先是自治、自恰的,同时它们还可以互相通信、协调和配合,从而共同完成整个程序的任务和功能。

更确切地,面向对象技术中的对象就是现实世界中某个具体的物理实体在计算机逻辑中的映射和体现。比如,电视机是一个具体存在的,拥有外形、尺寸、颜色等外部特性和开、关、频道设置等实在功能的实体;而这样一个实体,在面向对象的程序中,就可以表达成一个计算机可理解、可操纵、具有一定属性和行为的对象。

类也是面向对象技术中一个非常重要的概念。简单地说,类是同种对象的集合与抽象。例如,日常接触的电视机有很多,如小张的便携电视机、老王的彩色电视机等都属于电视机的范畴,这些实体在面向对象的程序中将被映射成不同的对象。不难看出,这些代表不同的电视机实体的对象之间存在着很多实质性的共同点。例如,都可以接收并播放电视信号,都可以调节画面效果、音量……因此,为了处理问题的方便,在面向对象的程序设计

中定义了类的概念来表述同种对象的公共属性和特点。从这个意义上来说,类是一种抽象的数据类型,它是所有具有一定共性的对象的抽象,而属于类的某一个对象则被称为是类的一个实例,是类的一次实例化的结果。面向对象技术中类和对象的这种关系在现实世界中也很容易理解。如果类是抽象的概念,如“电视机”,那么对象就是某一个具体的电视机,如“老王家那台 1983 年出产的金星牌彩色电视机”。

图 1-1 表示了类、对象、实体的相互关系和面向对象的问题求解的思维方式。在用面向对象的软件方法解决现实世界的问题时,首先将物理存在的实体抽象成概念世界的抽象数据类型,这个抽象数据类型里面包括了实体中与需要解决的问题相关的数据和属性;然后再用面向对象的工具,如 Java 语言,将这个抽象数据类型用计算机逻辑表达出来,即构造计算机能够理解和处理的类;最后将类实例化就得到了现实世界实体的面向对象的映射——对象,在程序中对对象进行操作,就可以模拟现实世界中的实体上的问题并解决之。

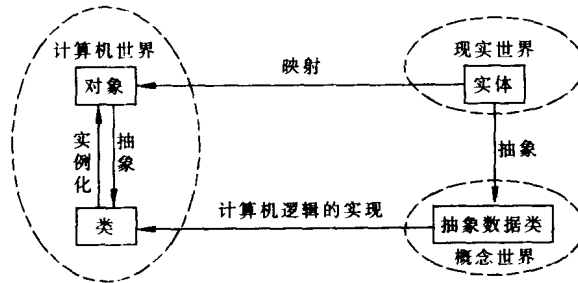


图 1-1 对象、实体与类

实质上,面向对象技术的一个关键的设计思想就是要让计算机逻辑来模拟现实世界的物理存在,即让计算机世界向现实世界靠拢。这一点与传统的程序设计中把现实世界的问题抽象成计算机可以理解和处理的数据结构的思路,即使现实世界向计算机世界靠拢的思路是完全不同的。面向对象技术提出的这种新的解决问题的思路,使得我们可以用更接近于人类自然思维模式和更接近于现实问题本来面目的方法来设计解题模型。这样,无论是当时的设计实现本身,还是日后的维护、修改和扩充,都可以比较顺利、容易地在已有工作的基础之上完成,避免了用面向过程方法实现时需要面对的种种困难。

## 1.4 对象的属性与相互关系

### 1.4.1 对象的属性

状态和行为是对象的主要属性。

对象的状态又称为对象的静态属性,主要指对象内部所包含的各种信息,也就是变量。每个对象个体都具有自己专有的内部变量,这些变量的值标明了对象所处的状态。当对象经过某种操作和行为而发生状态改变时,具体地就体现为它的属性变量的内容的改变。通过检查对象属性变量的内容,就可以了解这个对象当前所处的状态。仍然以电视



机为例。每一个电视机都具有以下这些状态信息：种类、品牌、外观、大小、颜色、是否开启、所在频道等。这些状态在计算机中都可以用变量来表示。

行为是对象的第二个属性，又称为对象的操作。它主要表述对象的动态属性，操作的作用是设置或改变对象的状态。比如一个电视机可以有打开、关闭、调整音量、调节亮度、改变频道等行为或操作。对象的操作一般都基于对象内部的变量，并试图改变这些变量（即改变对象的状态）。如“打开”的操作只对处于关闭状态的电视机有效，而执行了“打开”操作之后，电视机原有的关闭状态将改变。对象的状态在计算机内部是用变量来表示，而对象的行为在计算机内部是用方法来表示的。方法实际上类似于面向过程中的函数，对象的行为或操作定义在其方法的内部。

从图 1-2 中可以看出，对象的方法一方面把对象的内部变量包裹、封装、保护起来，使得只有对象自己的方法才能操作这些内部变量；另一方面，对象的方法还是对象与外部环境和其他对象交互、通信的接口，对象的环境和其他对象可以通过这个接口来调用对象的方法，操纵对象的行为和改变对象的状态。

在面向对象的方法学中，“对象”是现实世界的实体或概念在计算机逻辑中的抽象表示。具体地，对象是具有唯一对象名和固定对外接口的一组属性和操作的集合，是用来模拟组成或影响现实世界问题的一个或一组因素。其中对象名是区别于其他对象的标志；对外接口是对象在约定好的运行框架和消息传递机制中与外界通信的通道；对象的静态属性表示了它所处的状态；而对象的操作则用来改变对象的状态以达到特定的功能。对象最主要的特点是以数据为中心，它是一个集成了数据和其上操作的独立、自恰的逻辑单位。

面向对象的问题求解就是力图从实际问题中抽象出这些封装了数据和操作的对象，通过定义属性变量和操作来表述它们的特征和功能，通过定义接口来描述它们的地位及与其他对象的关系，最终形成一个广泛联系的可理解、可扩充、可维护、更接近于问题本来面目的动态对象模型系统。

#### 1.4.2 对象的关系

一个复杂的系统必然包括多个对象，这些对象间可能存在的关系有三种：包含、继承和关联。

##### 1. 包含

当对象 A 是对象 B 的属性时，称对象 B 包含对象 A。例如，每台电视机都包括一个显像管。当把显像管抽象成一个计算机逻辑中的对象时，它与电视机对象之间就是包含的关系。

当一个对象包含另一个对象时，它将在自己的内存空间中为这个被包含对象留出专门的空间，即被包含对象将被保存在包含它的对象内部，就像显像管被包含在电视机之中一样，这与它是电视机组成部分的地位是非常吻合的。

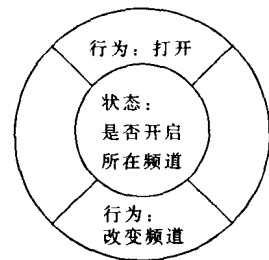


图 1-2 对象的属性