

计算机基础课程解析法系列教材

解析Java 程序设计

甘玲 张璞 李兴春 编著



清华大学出版社

计算机基础课程解析法系列教材

解析Java 程序设计

甘玲 张璞 李兴春 编著

清华大学出版社
北京

内 容 简 介

本书旨在通过实例解析的方式全面而深入地介绍 Java 语言的特点及其面向对象的特性,不但详细介绍了 Java 语言本身;而且讨论了面向对象的设计思想和编程方法、UML、图形用户界面的编程方法、网络和数据库程序的编程方法、线程的使用等实用开发技术。

全书共分 10 章,第 1、2 章是基础部分,介绍了面向对象的基本概念和相关技术,并对 Java 语言进行了概述。第 3~5 章是核心部分,主要介绍 Java 面向对象技术,围绕抽象性、封装性、继承性、多态性以及 I/O、异常等内容由浅入深地展开讨论。第 6~10 章是应用部分,主要介绍运用 Java 语言及其类库在 J2SE 平台下进行图形用户界面编程、网络编程、数据库编程、编写常用数据结构等内容。

本书层次清晰、内容全面、例题丰富、实用性强,全书基础性和实用性并重。本书不仅能使读者掌握 Java 语言,而且能够让读者对现实世界中较简单的问题及其解决方法用计算机语言进行描述。本书适合作为大专院校计算机相关专业的教材及自学 Java 语言的参考书,也可供计算机技术人员参阅。

版权所有,翻印必究。举报电话:010-62782989 13501256678 13801310933

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。

本书防伪标签采用特殊防伪技术,用户可通过在图案表面涂抹清水,图案消失,水干后图案复现;或将表面膜揭下,放在白纸上用彩笔涂抹,图案在白纸上再现的方法识别真伪。

图书在版编目(CIP)数据

解析 Java 程序设计 / 甘玲, 张璞, 李兴春编著. —北京: 清华大学出版社, 2006.9

(计算机基础课程解析法系列教材)

ISBN 7-302-13621-1

I. 解… II. ①甘… ②张… ③李… 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 (2006) 第 093471 号

出 版 者: 清华大学出版社 地 址: 北京清华大学学研大厦

<http://www.tup.com.cn> 邮 编: 100084

社 总 机: 010-62770175 客户服务: 010-62776969

组稿编辑: 付弘宇

文稿编辑: 顾 冰

印 装 者: 北京鑫海金澳胶印有限公司

发 行 者: 新华书店总店北京发行所

开 本: 185×260 印张: 23.75 字数: 583 千字

版 次: 2006 年 9 月第 1 版 2006 年 9 月第 1 次印刷

书 号: ISBN 7-302-13621-1/TP·8225

印 数: 1~4000

定 价: 29.50 元

丛书序

长久以来，计算机语言教育界内一直存在一个缺憾。在计算机程序设计教学过程中，可供读者选择的优秀著作如此之多，其中大部分出自于业界的泰山北斗。我们可以欣喜地从这些著作中感受到学术的源变，欣赏到学者的风采，然而遗憾的是，这些著作的章节安排和行文叙述更适合研讨和参考，却不容易走进课堂。与之相反的是，国内的程序设计教材注重知识的整合和安排，条理清楚，便于教学，却又缺少对知识的发现过程和问题来龙去脉的分析，而这正是国外同类书籍最突出的闪光点，也正是教学的真正意义所在。

出版这套“解析法”系列教材的目的就是为了弥补这一缺憾。“解析法”是指从实际问题入手，剖析求解问题的关键点（知识的发现过程），然后结合问题讲解相应知识点，最后得出问题的求解办法并举一反三。读者可以从本套教材中看到解析法贯穿于始终，每一章节的开始都提出了一个具体的问题，并对问题做出分析，根据分析引出相应的知识点，层层推进，并在每章的最后给出该问题的实现，这样更符合学习者的思维方式以及程序设计课程的特点。更重要的是，解析的目的不仅在于讲授知识，更注重培养学习者独立发现问题、分析问题、解决问题的能力 and 进行知识迁移的能力。这些能力是每个程序开发人员都应具备的。

此外，本系列教材包含大量具有实际意义的例题、思考题和习题，具有较强的启发性，便于教学和自学。本系列教材对高等院校计算机专业以及相关专业的程序设计教学工作有重要作用。我们相信，本系列教材的出版会进一步推动我国教育教学改革特别是计算机基础教学改革步伐。

计算机基础课程解析法
系列教材编委会
2006年6月

前 言

面向对象技术已发展成为一种比较成熟的编程思想，并且逐步成为目前软件开发领域的主流技术。这种技术从根本上改变了人们以往设计软件的思维方式，它集抽象性、封装性、继承性和多态性于一体，实现了代码重用和代码扩充，提高了软件开发的效率。

Java 为面向对象技术提供了全面支持，也是最常用的一种面向对象程序设计语言，目前已经应用于桌面应用程序、嵌入式移动应用、企业级服务器应用程序等各个应用领域中，当前，高等院校计算机专业和相关专业都开设了该课程，其目的是为了让学生掌握面向对象程序设计的概念和方法，深刻理解面向对象程序设计的本质，并用面向对象技术来编写程序、开发软件。根据笔者的教学经验，就该课程而言，很多教材基本都以将课程主要内容按照一定的结构分章节整理、详细描述其知识点这样的思路来编写，这种结构对学生掌握知识点有益处，学生也习惯这种结构，对一些知识点容易以死记硬背的方式来学习，但往往学后不能灵活运用所学知识来解决问题。

基于教学过程的实践和思考，我们产生了编写本书的想法，力图弥补传统教材的不足，在本书中采用实例解析的方式，从典型问题入手，在分析和解决问题的过程中向读者解释新遇到的编程元素和相关知识，使实例和知识点紧密结合，更适合学生的认知规律和计算机程序设计课程的特点，以便更好地实现教学目标。

全书共分 10 章，第 1、2 章是基础部分，介绍了面向对象技术的基本概念和相关技术，并对 Java 语言进行了概述。第 3~5 章是本书的核心，主要介绍 Java 面向对象技术部分，强调面向对象技术的原理，主要以抽象性、封装性、继承性、多态性等面向对象技术特征为线索，结合具体实例进行解析，展开对 Java 的讨论，为 Java 语言的应用奠定基础。第 6~10 章是应用部分，同样在每一章中结合实例解析来分别介绍图形用户界面编程、网络编程、数据库编程、常用数据结构等内容。本书精选了大量的例题，并且都在 JDK1.5 环境下调试通过。书后的附录提供了 Java 语法摘要、Java 标准包介绍、JDK 使用介绍、常用 Java 开发工具介绍、UML 简介等内容，供读者在学习时查阅。

本书由甘玲（编写第 1、2 章及附录 E）、张璞（编写第 4、7、10 章）主编。参加本书编写工作的还有李兴春（编写第 5、8 章）、唐晓军（编写第 3、9 章）、李盘林（编写第 6 章及附录 C）、冯潇（编写附录 A、B、D）。全书由甘玲、张璞统稿和校稿。感谢本书所列参考文献的作者！

由于作者水平有限，加之时间仓促，错误与疏漏之处在所难免，恳请读者批评指正。在使用本书时如遇到什么问题需要与作者联系，或者想索取本书例题的源代码与电子讲稿，请与作者联系，联系方式：ganling@cqupt.edu.cn。

编者

2006 年 6 月

读者意见反馈

亲爱的读者：

感谢您一直以来对清华版计算机教材的支持和爱护。为了今后为您提供更优秀的教材，请您抽出宝贵的时间来填写下面的意见反馈表，以便我们更好地对本教材做进一步改进。同时如果您在使用本教材的过程中遇到了什么问题，或者有什么好的建议，也请您来信告诉我们。

地址：北京市海淀区双清路学研大厦 A 座 602 室 计算机与信息分社营销室 收
邮编：100084 电子邮箱：jsjjc@tup.tsinghua.edu.cn
电话：010-62770175-4608/4409 邮购电话：010-62786544

教材名称：解析 Java 程序设计

ISBN：7-302-13621-1/ TP · 8225

个人资料

姓名：_____ 年龄：_____ 所在院校/专业：_____

文化程度：_____ 通信地址：_____

联系电话：_____ 电子信箱：_____

您使用本书是作为：☐指定教材 ☐选用教材 ☐辅导教材 ☐自学教材

您对本书封面设计的满意度：

☐很满意 ☐满意 ☐一般 ☐不满意 改进建议_____

您对本书印刷质量的满意度：

☐很满意 ☐满意 ☐一般 ☐不满意 改进建议_____

您对本书的总体满意度：

从语言质量角度看 ☐很满意 ☐满意 ☐一般 ☐不满意

从科技含量角度看 ☐很满意 ☐满意 ☐一般 ☐不满意

本书最令您满意的是：

☐指导明确 ☐内容充实 ☐讲解详尽 ☐实例丰富

您认为本书在哪些地方应进行修改？（可附页）

您希望本书在哪些方面进行改进？（可附页）

电子教案支持

敬爱的教师：

为了配合本课程的教学需要，本教材配有配套的电子教案（素材），有需求的教师可以与我们联系，我们将向使用本教材进行教学的教师免费赠送电子教案（素材），希望有助于教学活动的开展。相关信息请拨打电话 010-62776969 或发送电子邮件至 fuhj@tup.tsinghua.edu.cn 咨询，也可以到清华大学出版社主页（<http://www.tup.com.cn> 或 <http://www.tup.tsinghua.edu.cn>）上查询。

目 录

第 1 章 面向对象技术概述	1
1.1 面向对象技术的发展	1
1.2 面向对象方法与结构化方法的比较	2
1.2.1 现实问题分析	2
1.2.2 面向对象方法的优越性	3
1.2.3 面向对象与面向过程的不同	4
1.2.4 面向对象技术的基本观点	4
1.3 面向对象技术的基本概念	5
1.3.1 对象和类	5
1.3.2 事件和消息	7
1.4 面向对象技术的基本特征	8
1.4.1 抽象性	8
1.4.2 封装性	9
1.4.3 继承性	10
1.4.4 多态性	11
1.5 面向对象分析与设计简介	11
1.6 面向对象程序设计语言简介	12
1.7 面向对象的 UML 描述	15
本章小结	17
习题	17
第 2 章 Java 语言基础	18
2.1 一个最简单的 Java 应用程序	18
2.1.1 问题的提出与分析	18
2.1.2 main 方法的说明	19
2.1.3 程序的编译与运行	20
2.2 Java 语言编程基础	20
2.2.1 问题的提出与分析	21
2.2.2 常量和变量	21
2.2.3 基本数据类型	23
2.2.4 运算符和表达式	24
2.2.5 类型转换	26
2.2.6 问题的解决	28

2.3	控制语句	29
2.3.1	问题的提出与分析	29
2.3.2	选择语句	29
2.3.3	循环语句	35
2.3.4	特殊的循环流程控制	38
2.4	类和对象的创建和使用	42
2.4.1	问题的提出与分析	42
2.4.2	类的定义	42
2.4.3	对象的生成	44
2.4.4	对象的使用	45
2.5	数组的使用	46
2.5.1	问题的提出与分析	47
2.5.2	一维数组的创建	47
2.5.3	一维数组的使用	48
2.5.4	二维数组的创建	49
2.5.5	二维数组的使用	50
2.6	字符串的使用	51
2.6.1	问题的提出与分析	51
2.6.2	String 类的介绍	52
2.6.3	问题的解决	54
2.6.4	对象转化为字符串	56
2.7	包	57
2.7.1	什么是包	57
2.7.2	包的定义	58
2.7.3	包的引用	59
2.7.4	常用的 Java 标准包	59
2.7.5	程序结构	60
2.8	应用举例	60
	本章小结	63
	习题	63
第 3 章	对象与类	66
3.1	问题的提出与分析	66
3.2	发现对象和类	67
3.3	类的设计	68
3.3.1	成员变量和方法	68
3.3.2	方法的重载	74
3.3.3	类的访问控制	75
3.3.4	成员的访问控制	77

3.4 类的构造方法	80
3.4.1 成员初始化	80
3.4.2 构造方法的重载	81
3.4.3 默认构造方法	82
3.4.4 this 关键字	83
3.4.5 对象的赋值和比较	84
3.5 静态成员	86
3.5.1 静态成员变量	87
3.5.2 静态成员方法	89
3.6 基本的输入输出	90
3.7 问题的解决	92
本章小结	99
习题	99
第4章 继承和多态	103
4.1 问题的提出与分析	103
4.2 类的继承	104
4.2.1 子类的定义	104
4.2.2 子类的构造方法	106
4.2.3 方法的覆盖和成员变量的隐藏	109
4.2.4 子类对象和父类对象间的转换	112
4.3 抽象类和接口	115
4.3.1 抽象类的定义	115
4.3.2 什么是接口	117
4.3.3 接口的定义	118
4.3.4 接口的实现	119
4.4 final 类和 final 方法	122
4.5 多态	122
4.6 问题的实现	127
4.6.1 定义商品类层次	127
4.6.2 定义职工类层次	129
4.6.3 商品列表类的实现	131
4.6.4 顾客类层次的实现	133
4.6.5 销售事件类和销售事件列表类的实现	134
4.6.6 类层次功能的测试	136
4.6.7 功能的进一步扩充	137
本章小结	137
习题	138

第 5 章 输入输出流和异常	141
5.1 问题的提出与分析	141
5.2 输入输出流	143
5.2.1 流的概念	143
5.2.2 输入输出流类层次	144
5.2.3 字节流	151
5.2.4 字符流	157
5.3 异常	160
5.3.1 异常的概念	161
5.3.2 异常类层次	162
5.3.3 捕获及处理异常	164
5.3.4 自定义异常	168
5.3.5 声明异常和抛出异常	169
5.4 问题的实现	170
本章小结	175
习题	175
第 6 章 图形用户界面设计	178
6.1 问题的提出与分析	178
6.2 创建图形用户界面	180
6.2.1 AWT 与 Swing 概述	180
6.2.2 容器与组件	180
6.2.3 布局管理器	188
6.3 事件处理	190
6.3.1 JDK 的事件处理模型	190
6.3.2 事件监听器	191
6.3.3 事件适配器	194
6.3.4 内部类	194
6.4 绘制图形	195
6.5 问题的实现	196
6.5.1 界面框架的实现	197
6.5.2 鼠标事件的响应及图形画笔粗细设置	202
6.5.3 图形的保存	205
本章小结	207
习题	208
第 7 章 网络编程	211
7.1 问题的提出与分析	211

7.2	网络的基本概念	212
7.2.1	TCP/IP 简介	212
7.2.2	Socket 接口	214
7.2.3	客户端/服务器模式	215
7.3	流套接字编程	215
7.3.1	服务器端程序的实现	216
7.3.2	客户端程序的实现	219
7.3.3	编程模型的总结	221
7.4	数据报套接字技术	222
7.4.1	服务器端程序的实现	223
7.4.2	客户端程序的实现	225
7.4.3	编程模型的总结	226
7.5	线程	227
7.5.1	线程的概念	227
7.5.2	线程的创建	228
7.5.3	线程的控制	230
7.5.4	流套接字多线程服务器的实现	231
7.5.5	数据报套接字多线程服务器的实现	233
7.5.6	线程的同步	235
	本章小结	238
	习题	238
第 8 章	Applet	240
8.1	问题的提出与分析	240
8.2	Applet 简介	241
8.2.1	什么是 Applet	241
8.2.2	一个简单的 Applet 程序	242
8.3	Applet 的生命周期	244
8.4	其他常用的方法	247
8.5	Applet 的运行	251
8.5.1	HTML 与 Applet	251
8.5.2	在浏览器中运行 Applet 程序	252
8.5.3	在 appletviewer 中运行 Applet 程序	253
8.5.4	运行 Applet 的安全性	254
8.6	问题的实现	254
8.6.1	定义 Applet 的派生类	254
8.6.2	实现 Applet 周期中的方法	255
8.6.3	线程的实现	257
8.6.4	paint 方法的实现	258

8.6.5 鼠标事件处理	259
本章小结	262
习题	262
第 9 章 JDBC 与数据库	265
9.1 问题的提出与分析	265
9.2 数据库基础	266
9.2.1 数据库简介	267
9.2.2 关系数据库	267
9.2.3 SQL 简介	268
9.3 JDBC 简介	270
9.3.1 引入 JDBC 的背景及其驱动模型	271
9.3.2 通过 JDBC 连接数据库	272
9.4 问题的实现	275
9.4.1 建立数据库	275
9.4.2 连接数据库	275
9.4.3 数据库操作	277
本章小结	285
习题	285
第 10 章 常用数据结构的实现	287
10.1 数据结构简介	287
10.2 单链表	287
10.2.1 单链表的实现	288
10.2.2 单链表的应用（约瑟夫问题的求解）	294
10.3 栈	295
10.3.1 栈的实现	296
10.3.2 栈的应用（数的进制的转换）	297
10.4 队列	298
10.4.1 队列的实现	298
10.4.2 队列的应用（打印杨辉三角形）	299
本章小结	301
习题	301
附录 A Java 语法摘要	302
A.1 语法规则	302
A.2 数据类型	304
A.3 运算符	305
A.4 表达式	306

A.5 语句	307
A.6 与 OOP 有关的关键字	309
A.7 与 OOP 有关的语法	310
A.8 JDK 1.5 增强语法	314
A.9 异常	318
附录 B Java 标准包介绍	320
B.1 包的定义	320
B.2 包的引用	320
B.3 标准包列表	321
B.4 常用包的介绍	322
附录 C JDK 使用介绍	332
C.1 安装与设置	332
C.2 JDK 实用程序工具库简介	333
C.3 编译	333
C.4 运行 Application	335
C.5 运行 Applet	336
附录 D 常用开发工具介绍	338
D.1 Eclipse 简介	338
D.2 JBuilder 简介	343
附录 E UML 简介	351
E.1 UML 概述	351
E.2 模型元素的图形表示及其解释	352
E.3 常用的 UML 图	358
E.4 UML 中的四种关系	360
参考文献	362

第 1 章 面向对象技术概述

本章综述

面对各种亟待解决的问题，人们都要有解决的步骤和方法。如何把这些步骤和方法准确无误地告诉给计算机，并且让计算机帮助人们来解决问题，这是程序设计的任务。人与计算机之间通过计算机程序设计语言进行交互。计算机程序设计语言经历了侧重计算机理解逐步过渡到面向人们理解的发展过程，Java 语言是一种成熟的、面向对象的、更符合人们思考习惯的计算机编程语言。

本章从面向对象技术的发展历史入手，引导读者分析客观世界的问题域中的对象，从而理解面向对象程序设计的基本思想和基本概念，理解面向对象的程序设计与结构化程序设计的不同，了解常用面向对象编程语言以及通用建模语言（UML）描述。本章的重点内容包括：

- 面向对象的基本概念；
- 面向对象技术的基本特征；
- 面向对象程序设计方法；
- 面向对象的 UML 描述。

1.1 面向对象技术的发展

面向对象技术源于面向对象的编程语言（object oriented programming language, OOP）。从 20 世纪 60 年代提出面向对象的概念到现在，它已发展成为一种比较成熟的编程思想，并且逐步成为目前软件开发领域的主流技术。

面向对象并非是一个新的概念，寻其根源可以追溯到 20 世纪 60 年代的挪威，当时挪威计算中心的 Kristen Nygaard 和 Ole-Johan Dahl 开发了一种称作 Simula 67 的仿真语言，该语言中引入了类的概念和继承机制，后来一些著名的面向对象编程语言都受到 Simula 的启发，如 Smalltalk、C++、Eiffel。

在 20 世纪 70 年代中期，Alan Kay 在施乐公司设计开发了 Smalltalk 语言，它具备了面向对象语言的继承和封装的主要特征，使面向对象程序设计方法趋于完善，掀起了面向对象研究的高潮。该语言的每个元素都被作为一个对象来实现。Smalltalk 的程序设计环境及其相关的各方面都是面向对象的。即使今天，Smalltalk 仍被认为是最纯的面向对象语言。Simula 67 和 Smalltalk 的开发为今天的面向对象研究开发工作奠定了基础。

从 20 世纪 80 年代中期到 90 年代，大批比较实用的面向对象编程语言涌现出来，如 C++、Objective-C、Object Pascal 等。面向对象编程语言的繁荣是面向对象技术走向实用的重要标志，也是面向对象技术在计算机学术界、产业界和教育界日益受到重视的结果。在 C++ 语言十分热门的时候，人们开始了对面向对象分析（object oriented analysis, OOA）的

研究,进而延伸到面向对象设计(object oriented design, OOD)。特别是 20 世纪 90 年代以后,许多专家都在尝试用不同的方法进行面向对象的分析与设计,在这段时期,面向对象的分析和设计技术逐渐走向实用,最终形成了从分析、设计到编程、测试与维护一整套的软件工程体系。当前,面向对象的概念已经渗透到编程语言、用户接口、人工智能和数据库等方面。

1.2 面向对象方法与结构化方法的比较

为了更好地理解面向对象方法与结构化方法的不同,下面通过对一个实例的分析来认识这个问题。

1.2.1 现实问题分析

【例 1-1】 某校需要设计图书管理系统,要求该系统具有借阅、归还、预订、查询等基本功能,其功能要求如下:

(1) 借阅:图书管理员输入读者借书证号码,系统首先检查借书证是否有效,若有效,则检查该读者所借图书或杂志是否超过 10 本,若已达到 10 本,则拒借;若未达到 10 本,则办理借书手续,修改图书库存目录并将读者借书情况保存到借书文件。

(2) 归还:图书管理员通过系统从借书文件中读出与读者有关的记录,查阅所借日期,如超期(3 个月)作罚款处理。否则,修改相应图书库存目录与借书文件。

(3) 预定:借书者可以预定目前借不到的书或杂志,一旦预定的书被其他人返还,就通知预定者。

(4) 查询:通过借书文件、图书库存目录文件查询读者情况、图书借阅及库存情况,打印统计表。

分析:按照结构化方法的思路,采用“自顶向下、逐步求精”的方法,可以把问题描述中的系统分解为一系列的模块,如图 1-1 所示。结构化方法强调功能抽象与模块化,因而结构化方法把问题看作一系列处理的过程,也就是以模块(即过程)为中心的开发方法。

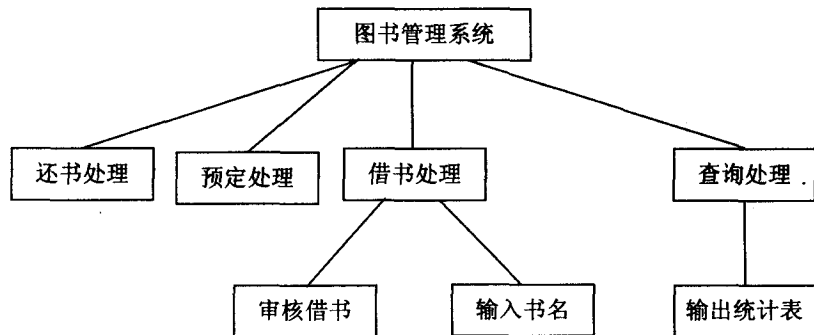


图 1-1 “图书管理系统”结构化方法的功能图

然后对每个模块再按相同的方法进行分解，例如还书处理模块可进一步分解为下面的子模块，如图 1-2 所示。

借书处理模块也可以进一步分解为相应子模块，如图 1-3 所示。

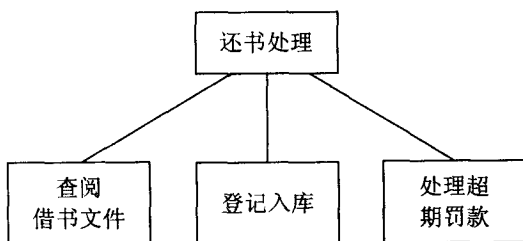


图 1-2 “还书处理”模块

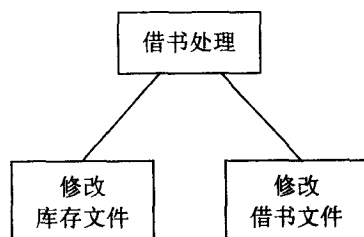


图 1-3 “借书处理”模块

按照面向对象方法来分析本系统则不是从功能入手，而是从对象入手。待解决的问题所涉及的业务范围称为该软件的问题域（problem domain）。面向对象方法从问题域中的事物出发，围绕问题域中的事物来思考问题、认识问题，并根据这些事物的本质特征，把它们抽象地表示为系统中的对象，作为系统的基本构成单位。这可以使系统直接地映射问题域。

对于图书管理系统，首先可以识别出一些直接的对象，包括图书、杂志、借阅者、图书管理员，还需要一个实体对象来记录图书的借还信息。到这一步，基本完成了实体对象的识别。另外，还需要一个借书的控制类来执行借书的动作。这样，初步的对象模型就可以建立了，如图 1-4 所示。

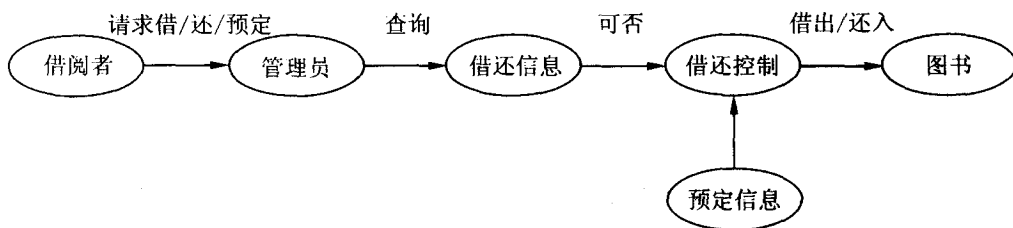


图 1-4 图书管理系统中的“对象”及关系

说明：在这个问题中，如果借书功能有变动，要求借阅前登记借阅者详细信息，如爱好、专长、借阅图书类型倾向等，两种设计方法的修改不同。如果采用面向过程方法设计，则要修改程序（借书模块）。如果采用面向对象方法设计，则只需要在借还控制类中增加一个方法，整个系统功能图不发生改变。

从这个例子可以看出，面向对象方法与结构化方法相比较而言，它以数据为中心而不是以功能为中心来描述系统，而数据相对于功能而言具有更强的稳定性，通常而言，以面向对象方法开发的系统的可维护性更好。

1.2.2 面向对象方法的优越性

以对象为中心的开发方法比以功能为中心的开发方法优越，它将客观世界直接映射到

面向对象程序空间，程序结构清晰、简单，提高了代码的重用性，有效地减少了程序的维护量，提高了软件的开发效率。具体表现在如下几个方面：

(1) 以对象为中心的开发方法能更自然更直接地反映真实世界的问题空间，对象、类、子类都自然对应于实际问题的物理或逻辑实体，这样使问题转换工作量达到最小程度。事实上，采用面向对象方法较之结构化方法在编程量方面可降低 40%~90%。

(2) 以对象为中心的开发方法采用消息传递机制作为对象之间相互通信的唯一方式。对象本身具有很强的自治性和独立性，接收消息的对象有权负责响应与处理所收到的消息，按消息名激活本对象内相应的操作（称为方法）并执行该操作，返回适当的结果。发送消息的对象还可将同一条消息同时发送至多个接收消息的对象中，并允许这些接收同一条消息的对象按照自身的适当方式加以响应。这样就使消息传递的机制能很自然地与分布式并程序、多机系统、网络通信等模型取得一致，从而强有力地支持复杂大系统的分析与运行。

(3) 以对象为中心的开发方法具有独特的继承性和更丰富的多态性，使这种开发方法更易于扩充，能很好地适应复杂大系统不断发展与变化的要求。

在结构上，面向对象程序与面向过程程序有很大不同。面向对象程序由类的定义和类的使用两部分组成：在主程序中定义各对象并规定它们之间传递消息的规律，程序中的一切操作都是通过向对象发送消息来实现；对象接到消息后，启动消息处理函数完成相应的操作。

1.2.3 面向对象与面向过程的不同

面向对象方法与结构化方法相比较，主要有以下不同：

(1) 它以数据为中心而不是以功能为中心来描述系统，数据相对于功能而言具有更高的稳定性。

(2) 它将数据和对数据的操作封装在一起，作为一个整体来处理，采用数据抽象和信息隐藏技术，将这个整体抽象成一种新的数据类型——类，并且考虑不同类之间的联系和类的重用性。类的集成度越高，就越适合大型应用程序的开发。

(3) 面向对象程序的控制流程由运行时各种事件的实际发生来触发，而不再由预定顺序来决定，更符合实际。事件驱动程序的执行围绕消息的产生与处理，靠消息循环机制来实现。

(4) 它可以利用不断扩充的类库，在实际编程时采用搭积木的方式来组织程序，站在“巨人”的肩上实现自己的目标。

1.2.4 面向对象技术的基本观点

如果说传统的面向过程的编程是符合机器运行指令的流程的话，那么面向对象的思维方法就是符合现实生活中人类解决问题的思维过程。把人类解决问题的思维方式逐步翻译成程序能够理解的思维方式的过程，在这个翻译的过程中，软件也就逐步被设计好了。总结起来，面向对象技术的基本观点如下：