



普通高等教育“十一五”国家级规划教材



21世纪大学本科
计算机专业系列教材

叶乃文 王丹 杨惠荣 编著

面向对象程序设计（第3版）

<http://www.tup.com.cn>

- 根据教育部“高等学校计算机科学与技术专业规范”组织编写
- 与美国 ACM 和 IEEE CS Computing Curricula 最新进展同步



清华大学出版社

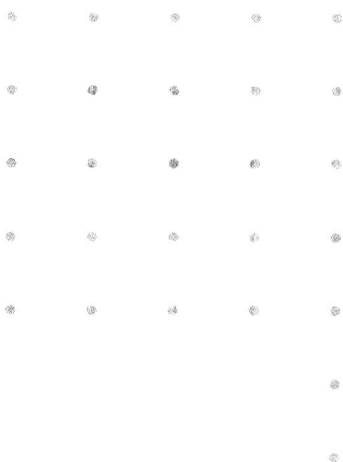


普通高等教育“十一五”国家级规划教材

21世纪大学本科计算机专业系列教材

面向对象程序设计 (第3版)

叶乃文 王 丹 杨惠荣 编著



清华大学出版社
北京

内 容 简 介

面向对象的程序设计方法是当今普遍使用的一种程序设计方法,它是计算机软件开发人员必须掌握的基本技术。本书根据国内外最新的面向对象程序设计课程的教学大纲要求,首先阐述面向对象程序设计方法的相关概念,然后选择具有典型特征的示例,并选择 Java 作为实现工具。本书的此次修订增加了一些 JDK 5.0 以后的新功能,并调整了一些章节的内容。学生通过本书的学习能够掌握面向对象的程序设计方法,学会 Java 程序设计的基本方法,养成良好的程序设计习惯。

本书共分 11 章,内容包括面向对象程序设计概论、Java 程序设计语言概述、抽象与封装、继承与多态、异常处理、流式输入输出及文件处理、泛型程序设计与数据结构、图形用户界面、事件处理、多线程程序设计和数据库访问的编程技术。

本书内容丰富,理论联系实际,可读性强,既可以作为高等院校计算机专业及相关专业本科生学习面向对象程序设计课程的教材,也可供从事软件开发的工程师和自学读者学习参考。

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。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

面向对象程序设计 / 叶乃文,王丹,杨惠荣编著. --3 版. --北京:清华大学出版社,2013

21 世纪大学本科计算机专业系列教材

ISBN 978-7-302-32907-7

I. ①面… II. ①叶… ②王… ③杨… III. ①面向对象语言—程序设计—高等学校—教材

IV. ①TP312

中国版本图书馆 CIP 数据核字(2013)第 136296 号

责任编辑:张瑞庆 徐跃进

封面设计:傅瑞学

责任校对:焦丽丽

责任印制:何 芊

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社 总 机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课 件 下 载: <http://www.tup.com.cn>, 010-62795954

印 装 者:北京国马印刷厂

经 销:全国新华书店

开 本:185mm×260mm 印 张:21.5

字 数:521 千字

版 次:2004 年 8 月第 1 版 2013 年 8 月第 3 版

印 次:2013 年 8 月第 1 次印刷

印 数:1~2000

定 价:35.00 元

产品编号:054662-01

21 世纪大学本科计算机专业系列教材编委会

主 任：李晓明

副 主 任：蒋宗礼 卢先和

委 员：（按姓氏笔画为序）

马华东 马殿富 王志英 王晓东 宁 洪

刘 辰 孙茂松 李仁发 李文新 杨 波

吴朝晖 何炎祥 宋方敏 张 莉 金 海

周兴社 孟祥旭 袁晓洁 钱乐秋 黄国兴

曾 明 廖明宏

秘 书：张瑞庆

第3版前言

FOREWORD

随着计算机技术的迅猛发展,人类对计算机的依赖程度越来越高,期望利用计算机解决各类问题的欲望越来越强烈,从而导致软件开发所面临的问题也越来越复杂,这就需要软件开发人士拥有一种良好的软件开发方法,以便指导软件开发的全过程,使得软件产品的开发效率不断地提高,软件产品的质量确实得到保证。

自从20世纪80年代广泛应用面向对象的程序设计方法以来,软件开发行业慢慢地摆脱了“行业危机”,开始进入良性循环的发展阶段。长期以来,人们在肯定面向对象方法的同时,不断地改进、完善它,使其成为一种科学化、人性化、规范化的软件开发方法。今天,作为一名高等学校计算机及相关专业的本科学生来说,掌握面向对象的程序设计方法已经成为一项基本的专业要求。为此,我们编写了本教材,希望能够对这门课程的教学与学习有一定的帮助。

本书根据“面向对象程序设计”课程的教学大纲要求,按照首先阐述面向对象程序设计方法的相关概念,然后选择具有典型特征的实例,并利用Java程序设计语言举例说明的基本教学策略论述本课程的全部内容,使学生能够掌握面向对象程序设计的基本方法,并且学会利用Java程序设计语言编写具有面向对象特征的程序代码,从中体会面向对象程序设计的精髓。

全书共分11章。

第1章 面向对象程序设计概论,主要介绍结构化程序设计方法与面向对象程序设计方法的基本特征,并对面向对象程序设计方法所涉及的基本概念进行全面的阐述。

第2章 Java程序设计语言概述,主要介绍Java程序设计语言的基本数据类型、Java程序结构、Java程序的基本输入输出方法以及数组类型的应用。

第3章 抽象与封装,主要阐述利用Java程序设计语言实现面向对象的抽象性和封装性的基本方法。

第4章 继承与多态,主要阐述利用Java程序设计语言实现面向对象的继承性和多态性的基本方法。

第5章 异常处理,主要介绍Java程序设计语言提供的异常处理机制。

第6章 流式输入输出及文件处理,主要介绍Java程序设计语言的流式处理及文件的读写方式。

第7章 泛型程序设计与聚合,主要介绍泛型程序设计的相关知识及常用的数据结构接口。

第8章 图形用户界面,主要介绍利用Java程序设计语言设计具有图形用户界面特征的应用程序,使学生能够掌握这类程序设计的基本方法。

第9章 事件处理,主要介绍Java事件处理机制。

第10章 多线程程序设计,主要介绍Java中进行多线程程序设计的相关技术。

第11章 数据库访问的编程技术,主要介绍利用Java语言访问数据库的基本实现方式。

本书列举了大量例子,所有程序均在NetBeans IDE环境下运行通过。NetBeans IDE是Sun公司极力推广的供用户免费使用的一个Java集成开发环境,这个开发环境拥有强大的开发能力,在Java规范化书写、调试、测试、版本管理、移植性等方面给予了极大的支持,近几年深受广大Java开发者的认可。有关NetBeans IDE的使用说明和软件下载可以从网站<<http://www.java.sun.com>>获得。

本书得到了华南理工大学李仲麟教授、北京工业大学蒋宗礼教授的鼎力支持,在此出版之际,一并表示衷心感谢!

由于作者水平有限,加之时间紧张,书稿虽几经修改,仍难免存在缺点和错误,恳请广大读者给予批评指正。

作者

2013年7月



普通高等教育“十一五”国家级规划教材 21世纪大学本科计算机专业系列教材

近期出版书目

- 计算概论(第2版)
- 计算概论——程序设计阅读题解
- 计算机导论(第3版)
- 计算机导论教学指导与习题解答
- 计算机伦理学
- 程序设计导引及在线实践
- 程序设计基础(第2版)
- 程序设计基础习题解析与实验指导
- 程序设计基础(C语言)
- 程序设计基础(C语言)实验指导
- 离散数学(第3版)
- 离散数学习题解答与学习指导(第3版)
- 数据结构(STL 框架)
- 算法设计与分析
- 算法设计与分析(第2版)
- 算法设计与分析习题解答(第2版)
- C++ 程序设计(第2版)
- Java 程序设计
- 面向对象程序设计(第2版)
- 形式语言与自动机理论(第3版)
- 形式语言与自动机理论教学参考书(第3版)
- 数字电子技术基础
- 数字逻辑
- FPGA 数字逻辑设计
- 计算机组成原理(第3版)
- 计算机组成原理教师用书(第3版)
- 计算机组成原理学习指导与习题解析(第3版)
- 微机原理与接口技术
- 微型计算机系统与接口(第2版)
- 计算机组成与系统结构
- 计算机组成与体系结构习题解答与教学指导
- 计算机组成与体系结构(第2版)
- 计算机系统结构教程
- 计算机系统结构学习指导与题解
- 计算机系统结构实践教程
- 计算机操作系统(第2版)
- 计算机操作系统学习指导与习题解答
- 编译原理
- 软件工程
- 计算机图形学
- 计算机网络(第3版)
- 计算机网络教师用书(第3版)
- 计算机网络实验指导书(第3版)
- 计算机网络习题解析与同步练习
- 计算机网络软件编程指导书
- 人工智能
- 多媒体技术原理及应用(第2版)
- 计算机网络工程(第2版)
- 计算机网络工程实验教程
- 信息安全原理及应用

目 录

CONTENTS

第 1 章 面向对象程序设计概论	1
1.1 结构化程序设计	1
1.1.1 结构化程序设计方法的产生背景	2
1.1.2 结构化程序设计方法	3
1.1.3 利用结构化程序设计方法求解问题域的基本过程	4
1.2 面向对象的程序设计	5
1.2.1 面向对象程序设计方法的产生背景	5
1.2.2 面向对象程序设计方法	7
1.3 基本概念	11
1.3.1 抽象	11
1.3.2 封装	12
1.3.3 对象	12
1.3.4 类	13
1.3.5 消息	14
1.3.6 继承	14
1.3.7 多态	14
1.3.8 UML	15
1.4 面向对象的程序设计语言	16
1.4.1 什么是面向对象程序设计语言	16
1.4.2 几种具有代表性的面向对象的程序设计语言	17
本章小结	20
课后习题	20
第 2 章 Java 程序设计语言概述	21
2.1 Java 程序设计语言的发展	21
2.2 Java 程序设计语言的基本特征	22
2.3 Java 程序设计语言环境	25
2.4 Java 程序的基本结构	26

2.5	Java 程序的基本数据类型	31
2.6	标识符、注释、直接量、变量和常量	35
2.7	Java 程序的输入输出	40
2.8	运算符和表达式	43
2.9	流程控制语句	52
2.10	一维数组	66
2.10.1	一维数组的声明与创建	66
2.10.2	一维数组的初始化	67
2.10.3	一维数组元素的访问	67
2.10.4	一维数组的复制	68
2.10.5	Arrays 类的应用	70
2.10.6	一维数组的应用举例	71
2.11	二维数组	75
2.11.1	二维数组的声明与创建	75
2.11.2	二维数组的初始化	77
2.11.3	二维数组元素的访问	77
2.11.4	二维数组的应用举例	78
2.12	字符串常量 String	80
	本章小结	83
	课后习题	83
	上机实践题	84
第3章	抽象与封装	85
3.1	抽象与封装的实现技术	85
3.2	类	86
3.2.1	类的定义	86
3.2.2	成员变量的声明与初始化	90
3.2.3	成员方法的声明	92
3.2.4	成员方法的重载	93
3.2.5	构造方法	95
3.3	对象	96
3.3.1	对象的创建	96
3.3.2	对象成员的使用	98
3.3.3	对象的清除	101
3.4	访问属性控制	102
3.4.1	默认访问属性	102
3.4.2	public 访问属性	102
3.4.3	private 访问属性	104
3.4.4	protected 访问属性	106

3.5 静态成员	106
3.5.1 类变量的声明及初始化	106
3.5.2 类方法	109
3.6 对象拷贝	110
3.7 几个 Java API 中的标准类	112
3.7.1 随机数类 Random	113
3.7.2 字符串类 StringBuffer	115
3.7.3 高精度数值类 BigInteger/BigDecimal	117
3.8 应用举例	118
本章小结	124
课后习题	125
上机实践题	126
第 4 章 继承与多态	127
4.1 继承与多态的实现技术	127
4.2 类的继承	128
4.2.1 定义子类	128
4.2.2 子类的构造方法	134
4.2.3 通用父类 Object	135
4.3 类成员的隐藏与重载	136
4.3.1 成员变量的继承与隐藏	136
4.3.2 成员方法的继承、重载与覆盖	137
4.4 多态性的实现	139
4.5 抽象类	144
4.6 接口	146
4.7 包	148
4.8 应用举例	150
本章小结	155
课后习题	155
上机实践题	156
第 5 章 异常处理	157
5.1 异常概述	157
5.1.1 异常的概念	157
5.1.2 Java 语言中的异常类	158
5.2 异常处理机制	159
5.2.1 抛出异常	160
5.2.2 捕获异常	160
5.2.3 处理异常	163

5.2.4 用户声明异常类	163
本章小结	165
课后习题	166
上机实践题	166
第6章 流式输入输出及文件处理	168
6.1 流式输入输出处理机制	168
6.2 Java 的输入输出流库	169
6.2.1 Java 的输入输出流库的标准类	169
6.2.2 字节输入流 InputStream	169
6.2.3 字节输出流 OutputStream	170
6.3 文件	171
6.3.1 文件的创建与管理	171
6.3.2 顺序文件的读写	175
6.3.3 随机文件的访问	180
6.4 字符流	182
6.5 对象的串行化	187
6.5.1 对象串行化概述	187
6.5.2 对象串行化的处理	187
6.5.3 应用举例	188
本章小结	190
课后习题	190
上机实践题	191
第7章 泛型程序设计与聚合	192
7.1 泛型程序设计	192
7.1.1 泛型类的定义与使用	193
7.1.2 对象包装器	195
7.2 基本的数据结构接口	196
7.2.1 Collection 接口	197
7.2.2 Set 接口	198
7.2.3 List 接口	199
7.2.4 Map 接口	201
本章小结	207
课后习题	207
上机实践题	208
第8章 图形用户界面	209
8.1 Java 图形用户界面概述	209

8.2	用 Swing 创建图形用户界面	210
8.2.1	Swing 概述	210
8.2.2	Swing 容器	211
8.3	布局管理器	217
8.3.1	FlowLayout 布局管理器	218
8.3.2	BorderLayout 布局管理器	220
8.3.3	GridLayout 布局管理器	221
8.3.4	CardLayout 布局管理器	223
8.3.5	使布局管理器无效	224
8.4	常用 Swing 组件	226
8.4.1	标签	226
8.4.2	按钮	228
8.4.3	文本框	237
8.4.4	列表	241
8.4.5	组合框	242
8.4.6	菜单	244
8.5	在窗口中绘制图形	250
8.5.1	坐标系统与变换	250
8.5.2	图形设备文本	250
8.5.3	设置颜色	251
8.5.4	绘制几何图形	252
8.5.5	填充几何图形	254
	本章小结	256
	课后习题	257
	上机实践题	257
第 9 章	事件处理	259
9.1	Java 事件处理机制	259
9.2	事件的处理过程	259
9.3	事件类	260
9.3.1	低级事件	261
9.3.2	语义事件	262
9.4	事件监听器	262
9.5	事件类及其对应的监听器接口	263
9.6	处理事件	264
9.6.1	窗口事件的处理	264
9.6.2	监听适配器	268
9.6.3	键盘事件的处理	270
9.6.4	鼠标事件的处理	272

9.6.5	ActionEvent 事件处理	276
9.6.6	ItemEvent 事件处理	283
	本章小结	285
	课后习题	285
	上机实践题	286
第 10 章	多线程程序设计	288
10.1	创建线程	288
10.1.1	利用 Thread 类创建线程	288
10.1.2	利用 Runnable 接口创建线程	290
10.2	线程状态的转换	291
10.2.1	线程的状态	291
10.2.2	线程的优先级及其调度	292
10.3	线程控制	293
10.3.1	基本的线程控制方法	293
10.3.2	线程控制举例	295
10.4	多线程的同步与互斥	297
10.4.1	临界区与互斥	298
10.4.2	Java 的互斥锁机制	299
10.4.3	线程的同步	301
	本章小结	307
	课后练习	307
	上机实践题	308
第 11 章	数据库访问的编程技术	310
11.1	Java 语言的数据库访问接口——JDBC	310
11.1.1	JDBC 框架结构	310
11.1.2	JDBC 访问数据库的应用模型	311
11.1.3	JDBC 驱动程序	312
11.1.4	JDBC 中的主要类和接口	312
11.2	JDBC 访问数据库	314
11.2.1	利用 JDBC 访问数据库的基本步骤	314
11.2.2	加载 JDBC 驱动程序	314
11.2.3	创建数据库连接	316
11.2.4	创建 SQL 语句对象	317
11.2.5	执行 Statement	318
11.2.6	处理查询结果集	319
11.2.7	关闭数据库连接	320
11.3	一个简单的 JDBC 应用程序	320

11.3.1 注册 ODBC 数据源	321
11.3.2 JDBC 数据库应用程序	322
本章小结	324
课后习题	324
上机实践题	325
参考文献	327

第 1 章

面向对象程序设计概论

软件是计算机的灵魂,而软件的开发方法是主导这个灵魂的关键。几十年来,众多的专家、学者将毕生的研究方向定位于探索软件开发方法的基础理论上,他们与奋斗在软件开发领域最前沿的软件精英们共同携手,将理论与实践紧密地结合在一起,提出了各种有效的软件开发方法,使软件开发行业历经磨难,终于步入了科学化、工程化和规范化的良性发展时代。

简要地说,一个规范的软件开发过程需要经历系统分析、系统设计、编码、测试和维护几个阶段。软件开发方法是指导软件开发各个阶段工作的理论和方法,它决定了审视问题域的角度、各个开发阶段的工作任务以及最终软件系统的构成方式。其中,编码阶段的主要任务是按照系统设计的要求编制最终的程序代码,即程序设计。它是软件开发过程的一个重要阶段,是软件系统的具体实现。在程序设计过程中,选择一种良好的程序设计方法将有助于提高程序设计的效率,保证程序的可靠性,增强程序的可扩充性,改进程序的可维护性。时至今日,用于指导程序设计的方法已有许多种类,它们都有各自的特点,其中结构化和面向对象是两种发展最为成熟、应用最为广泛的程序设计方法。本章主要介绍有关程序设计方法的概念,并阐述结构化程序设计方法和面向对象程序设计方法的具体内容,以便读者对它们有个初步认识,为日后更好地指导程序设计打下良好的基础。

1.1 结构化程序设计

程序设计是指设计、编制和调试程序的方法和过程。由于程序是应用系统的本体,是软件质量的具体体现,因此,研究程序设计中涉及的基本概念、描述工具和所采用的方法就显得格外重要。

这里所说的基本概念主要包括程序、数据、子程序、模块,以及顺序性、并发性、并行性和分布性等。其中,程序是程序设计的核心,子程序是为了便于程序设计而建立的程序基本单元,也是模块的具体体现,而顺序性、并发性、并行性和分布性反映了程序的内在特性。

描述工具主要是指编写程序的语言和为了便于调试程序而提供的各种语言开发环境。从某种意义上讲,它们决定了应用系统的最终功效,直接影响着软件产品的可靠性、易读性、易维护性以及开发效率。

程序设计方法是用于指导程序设计工作的思想方法,它主要包括程序设计的原理和所

应遵循的基本原则,帮助人们从不同的角度描述问题域。选用合适的程序设计方法,对于开发满足用户需求的高质量应用软件至关重要。这是本课程阐述的核心内容。

至今为止,软件开发具有多种程序设计方法,其中结构化程序设计方法和面向对象程序设计方法最具代表性。本节介绍从20世纪70年代开始广泛应用的结构化程序设计方法。

1.1.1 结构化程序设计方法的产生背景

自从1946年第一台计算机诞生以来,计算机以其惊人的速度迅猛发展。从最初庞大的机体到今天的掌中之物;从单纯的数值计算到今天的文字、图形、视频、音频等各种媒体数据的处理及海量数据的管理;从只能由极少数专家作为科学研究的辅助工具到今天步入家庭成为家庭消费品,计算机的每一步发展都倾注着无数科学家的心血,展示着人类的聪明才智。今天,计算机已经成为人类不可缺少的“伙伴”,这是人类渴望高质量生活的需求,是广大计算机科技工作者长期不懈努力的结果。

回首计算机的发展历程,计算机软件的发展速度始终滞后于计算机硬件的发展,它已经成为制约计算机产业整体发展的瓶颈。究其原因可能有很多方面,但下面两点不容忽视。

1. 个体化

早期的计算机软件主要指程序,软件开发就是指编写程序。由于计算机初期只能识别机器指令,为计算机编写程序是很专业、很复杂的事情,所以具有这种能力的人少之甚少。当时计算机的价格非常昂贵,处理能力也很有限,使得计算机的应用范围窄,处理问题的规模小,复杂度低。这就造成了编写程序的人员往往以个体的身份出现,即一个人接受任务、分析设计、编写程序、调试程序,甚至包括最终的维护程序。这种行业机制使得软件开发过程缺乏可遵循的规范,极大地限制了发展规模,是软件成为计算机发展瓶颈的主要原因。今天,软件已经被“工程化”,从事软件行业的人士已经不再属于“个体”户,而是工业化社会大生产中的一分子。

2. 受限于程序设计语言

程序设计语言是用于书写计算机程序的语言。自从计算机诞生以来,程序设计语言的发展一直伴随着整个计算机行业的成长,影响着计算机应用领域的不断扩展。

20世纪50年代是低级语言的发展时期。计算机只能够识别由0、1组成的机器指令,程序中的每一条指令都是一串0、1序列,复杂、易错、难维护这三大难题长期困扰着每一位编写程序的人,能够调试成功、真正投入使用的程序很少。后来,人们将机器指令符号化,形成了汇编语言。尽管汇编语言改进了编写程序的方法,但仍旧没有摆脱程序难以移植的困惑,这是因为每一种计算机指令系统中的指令格式不一样,汇编语言只是将某种计算机指令系统中的机器指令符号化,并没有解决不同计算机系统之间的程序移植问题。

20世纪60年代是高级语言发展时期,出现了大量的高级程序设计语言,例如,FORTRAN、COBOL、ALGOL60、LIST等语言。所谓高级语言是指将各种计算机系统的指令抽象化,形成一种与指令系统无关的描述形式。显而易见,它们更加贴近人们习惯使用的自然语言描述形式,克服了低级语言的许多弊病,带来了易学、易用、易移植等优势。但在这

个时期,绝大多数程序设计语言还只停留在强调其处理功能上,没有考虑从语言的角度制约程序设计的整个过程,致使编写出来的程序普遍缺乏结构性。

步入 20 世纪 70 年代后,计算机应用领域的迅猛发展,程序规模的不断增大,复杂度的不断提高,原始的编程方式越来越力不从心,低效率、低成功率的软件开发过程很难适应社会对计算机的需求。其关键问题表现在两个方面:一是由于软件行业的“个体化”,人们在编写程序时,只是凭借个人以往的经验,按照自己的习惯,随意地编写出自认为能够解决问题的语句序列,这样得到的程序没有任何章法,大都是缺乏书写规范的语句罗列;二是由于早期计算机硬件的处理能力有限,在编写程序时过分强调减少时间和空间的消耗,使得程序的可读性较差。一旦出现问题,很难确定出错的位置,更难以改正。

在这种背景下,人们开始意识到应该将程序设计过程纳入科学化、规范化的轨道,并提出了结构化程序设计方法的概念,出现了一批支持结构化的程序设计语言,例如, Pascal、C、Ada 等。从此,程序设计语言向着模块化、简明化、形式化的方向迈进,成为支持各种程序设计方法的有力工具。计算机发展到这个时期,随着处理能力的不断提高,支持结构化程序设计方法的语言不断出现,软件需要工程化的认识日益形成。人们越来越重视程序的结构化、可读性,采用结构化程序设计方法指导设计程序过程逐步变为编程人员自觉自愿的行为,编写出结构化强、可读性好的程序迅速成为软件开发行业的时尚。

尽管结构化程序设计方法还存在着诸多缺憾,并逐步被面向对象程序设计方法所取代,但它在程序设计发展的历史长河中,发挥过重要的作用,有着不可磨灭的功绩。

1.1.2 结构化程序设计方法

结构化程序设计方法是一种在 20 世纪七八十年代十分流行的程序设计方法。所谓结构化主要体现在以下 3 个方面:

(1) 自顶向下、逐步求精。即将编写程序看成是一个逐步演化的过程。所谓自顶向下是指将分析问题的过程划分成若干个层次,每一个新的层次都是上一个层次的细化,即步步深入,逐层细分。例如,一个简单的户籍管理系统,可以被划分为户籍迁入迁出、人员迁入迁出、户籍注销、人员注销等几个子系统,而每个子系统又可以被进一步地划分为接收用户信息、实施处理、提供返回信息等几个部分。

(2) 模块化。即将整个系统分解成若干个模块,每个模块实现特定的功能,最终的系统将由这些模块组装而成。模块之间通过接口传递信息,力求模块具有良好的独立性。实际上,可以将模块看作是对欲解决的应用系统实施自顶向下、逐步求精后形成的各子系统的具体实现。即每个模块实现一个子系统的功能,如果一个子系统被进一步地划分为更加具体的几个子系统,它们之间将形成上下层的关系,上层模块的功能需要通过调用下层模块实现。例如,上述提到的户籍迁入迁出子系统所对应的模块需要调用接收用户信息、迁入、迁出、提供反馈信息等几个模块最终完成其功能。

(3) 语句结构化。支持结构化程序设计方法的语言都应该提供过程(函数是过程的一种表现形式)实现模块概念。结构化程序设计要求,在每一个模块中只允许出现顺序、分支和循环 3 种流程结构的语句。如图 1-1 所示,这 3 种流程结构的语句的共同特点是:每种语句只有一个入口和一个出口,这对于保证程序的良好结构、检验程序的正确性十分重要。