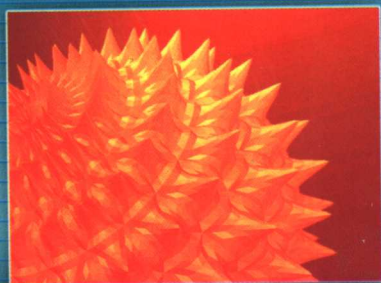
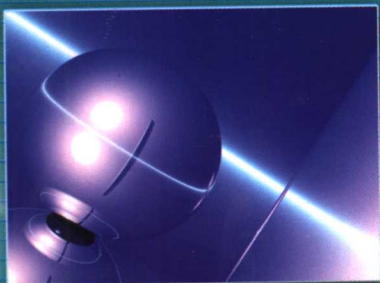




普通高等教育“十一五”规划教材  
高等院校计算机技术系列教材

# 软件组建技术

刘怀亮 陈荣征 吕国华 编著



冶金工业出版社

普通高等教育“十一五”规划教材  
高等院校计算机技术系列教材

# 软件组建技术

刘怀亮 陈荣征 吕国华 编著

北 京

冶金工业出版社

# 内 容 简 介

本书是根据普通高等教育“十一五”国家级规划教材的指导精神而编写的。

本书是一本全面介绍软件组建技术的教材，主要介绍了几种当今最为流行的软件组建开发技术，这些技术包括结构化技术、面向对象技术、软件组件技术、中间件技术、软件复用技术与构件技术、软件 Agent 技术以及软件应用集成技术。编写本书的目的是通过对相关技术的介绍，使广大读者能够对当今的软件组建技术有一个整体的认识，并通过对本书的学习掌握软件组建开发的常用方法和技术。

本书条理清晰、深入浅出，采用图文并茂的阐述手法，具有很强的趣味性和实用性。全书按照组建技术出现的先后顺序和它们之间的相互关系，逐一展开讲述，为了让读者巩固所学知识，培养动手能力，本书课后配有相应的习题与上机实训。同时，为了拓展广大读者的知识面，使大家对当今兴起的新技术、新方法有所了解，本书在论述传统技术的同时，适当引入了对一些新技术、新方法的介绍，从而使本书能够紧跟时代的发展，具有很强的时代性和先进性。

本书资料丰富，内容详实，涵盖了软件组建技术的重点、难点和热点。本书既可作为高等院校计算机相关专业的教材，也可供广大软件开发设计爱好者参考。

## 图书在版编目 (CIP) 数据

软件组建技术 / 刘怀亮, 陈荣征, 吕国华编著. —北京:  
冶金工业出版社, 2007.6

普通高等教育“十一五”规划教材  
ISBN 978-7-5024-4305-4

I. 软… II. ①刘…②陈…③吕… III. 软件开发—  
高等学校—教材 IV. TP311.52

中国版本图书馆 CIP 数据核字 (2007) 第 076322 号

出版人 曹胜利 (北京沙滩嵩祝院北巷 39 号, 邮编 100009)

责任编辑 程志宏

ISBN 978-7-5024-4305-4

广州锦昌印务有限公司印刷; 冶金工业出版社发行; 各地新华书店经销

2007 年 6 月第 1 版第 1 次印刷

787mm × 1092mm 1/16; 18.5 印张; 424 千字; 286 页

35.00 元

冶金工业出版社发行部 电话: (010) 64044283 传真: (010) 64027893

冶金书店 地址: 北京东四西大街 46 号 (100711) 电话: (010) 65289081

(本社图书如有印装质量问题, 本社发行部负责退换)

# 前 言

## 一、关于本书

本书是根据普通高等教育“十一五”国家级规划教材的指导精神而编写的。

计算机软件的发展从面向过程到面向对象、面向组件、面向构件，再到面向 Agent，无论从开发方法、开发技术还是开发工具上都发生了巨大的变化，特别是 20 世纪 90 年代，随着计算机网络技术的快速发展以及计算机资源的网络化，广大 PC 或工作站用户迫切需要共享分布于网络上的信息资源，希望通过计算机之间的联网获得超出局部计算机能力的优质服务。这样大量的计算转移到了网络环境中进行，这就是分布式计算模式。分布式计算技术成为影响当今计算机技术发展的关键技术。当今人们所说的分布式计算技术通常是指在网络环境下的计算机平台开发、部署、管理和维护以资源共享和协同工作为主要应用目标的分布式技术。

在这种情况下，结构化技术、面向对象技术、组件技术、中间件技术、构件技术和复用技术、软件 Agent 技术、软件应用集成技术和 Web 技术的融合彻底改变了软件系统的组建方式。本书正是介绍当今流行的几种软件组建技术及其发展应用，以供读者用最短的时间了解并迅速掌握软件组建的常用方法与技术。

## 二、本书结构

本书共分为 8 章，具体结构如下：

第 1 章：主要介绍了传统的软件组建技术即结构化技术，内容包括结构化分析与设计的步骤、工具以及常用方法等。

第 2 章：详细介绍了当今最流行的面向对象技术，内容涉及采用面向对象技术的原因，面向对象技术的基本概念、基本特征以及软件生命周期与开发模型，并详细介绍了统一建模语言 UML 以及 UML 建模工具 Rational Rose。

第 3 章：讲述了软件组件技术，内容包括面向组件编程思想、组件的创建与组成、组件开发原理和步骤、典型组件技术 COM 概述以及如何使用 Delphi 组件进行应用程序的开发。

第 4 章：讲述了中间件技术，内容包括中间件的定义、特点、功能、分类以及中间件的基本框架模型、工作机理，并介绍了实现中间件的关键技术和支持中间件的平台技术等。

第 5 章：介绍了软件复用与构件技术，内容包括软件复用的关键技术，构件技术的特点、内容以及构件技术与中间件的关系，软件复用密切相关的几个领域以及主流构件模型技术等。

第 6 章：讲述了软件 Agent 技术，内容包括软件 Agent 的分类、基本结构和工作机理，移动 Agent 有关的概念，几种典型的移动 Agent 平台，移动 Agent 的开发方法与步骤等。

第 7 章：介绍了软件应用集成技术，内容包括三种集成模型，多数据库集成技术，XML 技术以及运用 XML 技术进行应用集成的方法，Web Service 有关内容原理和与之相关的三种基础框架平台等。

第8章：数字城市的软件组建模型，给出了一个大型软件组建技术的应用范例——数字城市。

上机实训部分给出了9个实训，每个实训都包含实训概要、实训内容、实训过程和实训总结，使读者在学习本书的过程中能够进一步提高实践能力。

其中，第1章是本书学习的基础，第2~4章是本书学习的重点，第5~8章是本书学习的难点，实训部分是进一步的提高。

### 三、本书特点

本书具有以下特点：

(1) 本书针对初学者基础差、理解能力弱的特点，从基础知识入手，然后逐渐展开，特别强调计算机技能的锻炼和培养，非常符合高等学校计算机相关专业使用。

(2) 本书根据初学者的学习习惯和心理，内容结构清晰明了，做到从零开始，循序渐进，突出重点，讲清难点，以利于学生掌握和应用。

(3) 习题包括选择题、填空题、简答题和上机操作题。上机操作题主要为本章内容相关的操作题，要求读者通过练习提高操作技能和操作技巧。

(4) 本书配备了经过精心制作的电子教案，放在网上供用户下载。

(5) 本书注重内容的先进性与时代的同步性，有机地融合和引入了多种流行的先进技术和理念，引导学生掌握和应用最新的先进技术与成果，激发学生的学习热情和兴趣，深入学习、掌握和应用相关技术。

### 四、适用对象

本书资料丰富，内容详实，涵盖了软件组建技术的重点、难点和热点。本书既可作为高等院校计算机相关专业的教材，也可供广大软件开发设计爱好者参考。

黄健、相洪贵、谷保平、郑壮贤曾为本书的编写提出许多宝贵意见，在此向他们表示最衷心和最诚挚的感谢。此外，本书在编写过程中引用了大量的参考文献，在本书的最后专门列出。

由于编写时间仓促，编者水平有限，书中疏漏和不足之处在所难免，恳请广大读者和专家批评指正。联系方法如下：

电子邮箱：[service@cnbook.net](mailto:service@cnbook.net) 作者邮箱：[great\\_liu@126.com](mailto:great_liu@126.com)

网址：[www.cnbook.net](http://www.cnbook.net)

本书电子教案、代码和习题参考答案可从该网站下载，此外，该网站还有一些其他相关书籍的介绍，可以方便读者选购参考。

编者

2007年5月

# 目 录

<b>第 1 章 结构化技术</b> .....1	
1.1 结构化程序.....1	
1.1.1 结构化程序设计.....1	
1.1.2 控制结构.....2	
1.2 结构化分析与设计的一般步骤.....4	
1.3 结构化分析.....4	
1.3.1 数据流分析.....5	
1.3.2 数据流图.....5	
1.3.3 数据字典.....7	
1.4 结构化设计的图标工具.....9	
1.4.1 IPO 图.....9	
1.4.2 结构图.....11	
1.4.3 程序流程图.....11	
1.4.4 盒图.....12	
1.4.5 PAD 图.....14	
1.5 面向数据结构的设计.....15	
1.5.1 Jackson 图.....15	
1.5.2 Jackson 方法.....16	
1.5.3 设计实例.....16	
1.6 模块设计.....18	
1.6.1 模块与模块化.....18	
1.6.2 模块的特征与独立性.....19	
1.6.3 模块的耦合.....21	
1.6.4 模块的聚合.....22	
1.6.5 模块设计的一般准则.....23	
1.7 一个简单的例子.....24	
小结.....24	
习题一.....25	
一、选择题.....25	
二、填空题.....26	
三、简答题.....26	
四、上机操作题.....26	
<b>第 2 章 面向对象技术</b> .....28	
2.1 为什么讨论面向对象技术.....28	
2.1.1 三种主要方法论简介.....31	
2.1.2 结构化方法存在的问题以及原因.....31	
2.1.3 面向对象分析和设计解决的两大难题.....33	
2.2 面向对象技术的基本概念.....33	
2.2.1 对象.....33	
2.2.2 消息和方法.....34	
2.2.3 类和实例.....35	
2.3 面向对象的基本特征.....35	
2.3.1 继承性.....36	
2.3.2 封装性.....36	
2.3.3 多态性.....37	
2.3.4 应用举例.....38	
2.4 软件生命周期与开发模型.....42	
2.4.1 面向对象的软件生命周期.....42	
2.4.2 面向对象方法与快速原型技术....43	
2.5 几种主要面向对象方法的介绍与比较.....44	
2.5.1 Booch 方法.....44	
2.5.2 OMT 方法.....44	
2.5.3 Coad/Yourdon 方法.....45	
2.5.4 OOSE 方法.....45	
2.6 统一建模语言 UML.....46	
2.6.1 为什么要使用 UML.....46	
2.6.2 UML 的组成.....46	
2.6.3 UML 的结构和图符表示.....48	
2.6.4 UML 的静态建模.....49	
2.6.5 UML 的动态建模机制.....52	
2.7 面向对象的建模工具 Rational Rose....55	
2.7.1 Rational Rose 概况.....56	
2.7.2 Rational Rose 的主要功能.....56	
2.7.3 Rose 模型.....57	
2.7.4 如何使用 Rational Rose.....58	
2.7.5 RUP 的开发过程.....60	



2.8 面向对象技术的新应用 .....	62	习题三 .....	90
2.8.1 并发面向对象技术 .....	63	一、选择题 .....	90
2.8.2 面向对象的数据库技术 .....	64	二、填空题 .....	90
小结 .....	64	三、简答题 .....	91
习题二 .....	65	四、上机操作题 .....	91
一、选择题 .....	65	<b>第4章 中间件技术 .....</b>	<b>92</b>
二、填空题 .....	65	4.1 中间件的概述 .....	92
三、简答题 .....	65	4.1.1 中间件的发展动力 .....	92
四、上机操作题 .....	66	4.1.2 中间件的定义 .....	93
<b>第3章 软件组件技术 .....</b>	<b>67</b>	4.1.3 为什么需要中间件 .....	93
3.1 组件技术产生的背景 .....	67	4.1.4 计算模式的演变 .....	94
3.2 组件技术概述 .....	68	4.1.5 中间件与两层、三层 C/S 结构 模式 .....	95
3.2.1 组件技术的编程思想 .....	68	4.2 中间件的特点、功能和分类 .....	96
3.2.2 组件技术简介 .....	68	4.2.1 中间件的特点 .....	96
3.2.3 组件的创建与组成 .....	69	4.2.2 中间件的功能 .....	97
3.3 组件与对象、构件的关系 .....	70	4.2.3 中间件的分类 .....	97
3.3.1 组件与对象 .....	70	4.3 中间件应用现状 .....	98
3.3.2 基于对象的组件体系结构 .....	70	4.3.1 中间件在电子商务中的应用 .....	98
3.3.3 组件与构件 .....	71	4.3.2 中间件在 ERP 系统中的应用 .....	98
3.4 组件开发原理 .....	71	4.4 中间件的基本框架模型和工作机理 .....	99
3.4.1 组件的软件设计模式 .....	71	4.4.1 中间件的基本框架 .....	99
3.4.2 组件的软件体系结构 .....	72	4.4.2 中间件的工作机理 .....	100
3.5 开发组件的一般步骤 .....	72	4.4.3 Web 环境中的中间件 .....	100
3.5.1 确定组件基类 .....	73	4.5 五大中间件的工作机理 .....	100
3.5.2 创建组件单元 .....	73	4.5.1 数据库中间件 .....	101
3.5.3 创建包工程 .....	75	4.5.2 远程过程调用中间件 .....	101
3.5.4 注册组件 .....	76	4.5.3 面向消息的中间件 .....	102
3.5.5 添加组件属性 .....	77	4.5.4 对象请求代理中间件 .....	103
3.5.6 为组件添加事件 .....	78	4.5.5 事务处理监控中间件 .....	104
3.5.7 组件的测试 .....	78	4.6 中间件实现的关键技术 .....	105
3.5.8 组件图标创建与组件的发布 .....	79	4.6.1 远程过程调用 .....	105
3.6 典型组件技术 COM 概述 .....	79	4.6.2 发报文和排队 .....	105
3.6.1 COM 的优点 .....	80	4.7 当前支持服务器端中间件的 平台技术 .....	106
3.6.2 COM 的基本机制 .....	80	4.7.1 SUN 的 J2EE .....	106
3.7 Delphi 组件开发实例 .....	81	4.7.2 Microsoft DNA 2000 .....	107
3.7.1 实例一：聊天程序设计 .....	81	4.7.3 OMG 的 CORBA .....	107
3.7.2 实例二：图形镜像应用 .....	86		
小结 .....	89		

4.7.4 三种分布式计算平台的特点 .....	108	5.2 软件复用技术 .....	133
4.8 开源 Jdon Framework .....	109	5.2.1 软件复用产生的背景 .....	133
4.8.1 传统的 J2EE 系统面临的问题 .....	109	5.2.2 软件复用的历史 .....	134
4.8.2 Jdon 框架简介 .....	109	5.2.3 软件复用技术的概念 .....	134
4.8.3 Jdon 框架特点 .....	110	5.2.4 复用意义 .....	135
4.8.4 Jdon 两个优势 .....	110	5.2.5 基于构件的软件开发过程 .....	136
4.8.5 Jdon 框架在软件生产中的 作用 .....	111	5.2.6 软件复用的关键因素 .....	137
4.8.6 开发框架和代码生成工具 区别 .....	111	5.3 复用的分类、软件制品以及建议 .....	138
4.9 中间件技术的发展趋势 .....	111	5.3.1 软件复用的分类 .....	138
4.9.1 网格技术 .....	111	5.3.2 可复用的软件制品 .....	140
4.9.2 面向移动计算 .....	112	5.3.3 软件复用途径的一些建议 .....	140
4.9.3 面向服务质量 .....	112	5.4 实现软件复用应考虑的主要问题 .....	140
4.9.4 其他技术对中间件技术的 影响 .....	113	5.4.1 程序设计结构方面 .....	141
4.9.5 中间件研究的最新热点 .....	113	5.4.2 系统的组织结构方面 .....	141
4.10 RMI 概述与编程实例 .....	113	5.5 软件复用的核心技术——软件构件 技术 .....	141
4.10.1 RMI 概述 .....	114	5.5.1 软件构件技术简介 .....	142
4.10.2 RMI 编程步骤 .....	115	5.5.2 软件构件技术的产生 .....	143
4.10.3 使用 RMI 实现回调 .....	120	5.5.3 原则 .....	144
4.11 中间件技术的新应用 .....	124	5.5.4 构件与对象 .....	145
4.11.1 几个相关的概念 .....	124	5.5.5 构件与构件的分类 .....	146
4.11.2 从中间件到 Web Services .....	125	5.5.6 构件开发应该注意的问题 .....	147
4.11.3 中间件的应用级别 .....	125	5.5.7 实现构件技术的条件 .....	147
4.11.4 中间件技术在集成中的应用 .....	126	5.6 构件技术的研究内容 .....	148
4.11.5 中间件技术的新趋势 .....	126	5.6.1 构件模型和构件描述语言 .....	148
小结 .....	127	5.6.2 构件的获取 .....	149
习题四 .....	128	5.6.3 构件的表示和检索 .....	149
一、选择题 .....	128	5.6.4 构件组装 .....	150
二、填空题 .....	129	5.6.5 构件组装技术 .....	150
三、简答题 .....	130	5.6.6 构件及构件库的标准化 .....	150
四、上机操作题 .....	130	5.7 基于构件技术的复用与传统 复用的区别 .....	151
<b>第 5 章 软件复用与构件技术 .....</b>	<b>131</b>	5.8 基于构件复用的软件开发实践 活动 .....	151
5.1 软件复用和构件化 .....	131	5.9 构件技术与中间件的关系 .....	152
5.1.1 软件复用和构件化提出的 背景 .....	131	5.9.1 中间件：构件存在的基础 .....	152
5.1.2 构件化与软件工厂 .....	132	5.9.2 构件思想对中间件的作用 .....	152
		5.10 复用的其他相关技术 .....	153



5.10.1 领域工程 .....	153	6.6 移动 Agent .....	182
5.10.2 软件构架 .....	154	6.6.1 固定 Agent 的定义 .....	182
5.10.3 软件再工程 .....	155	6.6.2 移动 Agent 及相关概念的 区别 .....	182
5.10.4 开放系统技术 .....	156	6.6.3 移动 Agent 的特点 .....	183
5.10.5 CASE 技术 .....	156	6.6.4 移动 Agent 的体系结构 .....	183
5.11 主流构件模型技术 .....	156	6.6.5 基于移动 Agent 的分布计算 模型 .....	184
5.11.1 JavaBean/EJB .....	156	6.6.6 实现移动 Agent 系统的 关键技术 .....	186
5.11.2 CORBA .....	158	6.6.7 移动 Agent 的优点 .....	187
5.11.3 COM/DCOM .....	158	6.6.8 几种典型的移动 Agent 平台 .....	188
5.11.4 EJB 和其他技术的比较 .....	160	6.7 Voyager 的移动代理编程 .....	189
5.12 相关技术 .....	162	6.7.1 启动和终止 Voyager .....	189
5.12.1 .NET (DOTNET) .....	162	6.7.2 使用移动 Agent .....	190
5.12.2 开放网格服务体系结构 (OGSA) .....	163	6.7.3 开发移动 Agent .....	190
5.13 EJB 开发技术与应用实例 .....	163	6.8 移动 Agent 的应用 .....	195
5.13.1 EJB 技术概述 .....	164	6.8.1 电子商务 .....	195
5.13.2 EJB 构件的开发步骤 .....	166	6.8.2 分布式信息检索 .....	196
5.13.3 EJB 开发实例 .....	166	6.9 移动 Agent 存在的问题 .....	197
小结 .....	170	6.9.1 效率问题 .....	197
习题五 .....	171	6.9.2 标准问题 .....	197
一、选择题 .....	171	6.9.3 安全问题 .....	197
二、填空题 .....	172	6.10 移动 Agent 未来的研究方向 .....	197
三、简答题 .....	172	6.11 面向 Agent 的软件开发 .....	198
四、上机操作题 .....	173	6.11.1 面向 Agent 与面向对象 之间的关系 .....	198
<b>第 6 章 软件 Agent 技术 .....</b>	<b>174</b>	6.11.2 几种典型的面向 Agent 软件 开发方法 .....	201
6.1 软件 Agent 技术起源及发展概述 .....	174	6.12 多 Agent 系统开发平台——JADE .....	203
6.1.1 软件 Agent 技术的起源 .....	174	6.12.1 JADE 内容概述 .....	203
6.1.2 多 Agent 系统 .....	175	6.12.2 JADE 的主要功能 .....	204
6.1.3 软件 Agent 技术发展现状 .....	175	6.12.3 多 Agent 系统的创建 .....	204
6.2 软件 Agent 的概述 .....	176	6.12.4 目录服务 (DF) .....	205
6.2.1 软件 Agent 定义 .....	176	6.12.5 Agent 管理系统服务 (AMS) .....	205
6.2.2 软件 Agent 的功能 .....	177	6.13 Agent 类与 JADE Agent 类 .....	205
6.2.3 软件 Agent 技术的标准化 .....	177	6.13.1 Agent 生命周期 .....	206
6.2.4 软件 Agent 的联邦结构 .....	178		
6.3 软件 Agent 的分类 .....	179		
6.4 软件 Agent 的基本结构和工作 机理 .....	180		
6.5 Agent 状态 .....	181		

6.13.2 Agent 内的通信 .....	207	7.8 基于 Web 服务的系统集成 .....	234
6.13.3 带有 GUI 的 Agent .....	207	7.9 CORBA 与 Web 服务的集成 .....	235
6.13.4 Agent 任务与 Agent 行为 .....	207	7.9.1 CORBA 简介 .....	235
6.13.5 JADE 对移动 Agent 的支持 .....	208	7.9.2 CORBA 与 Web 服务的集成 .....	235
小结 .....	208	7.9.3 集成网关的设计 .....	236
习题六 .....	209	7.9.4 关键技术 .....	237
一、选择题 .....	209	7.10 Voyager 和 CORBA 集成 .....	237
二、填空题 .....	209	7.10.1 Voyager CORBA 编程基础 .....	237
三、简答题 .....	209	7.10.2 Voyager CORBA 应用程序 举例 .....	238
四、上机操作题 .....	210	小结 .....	240
<b>第 7 章 软件应用集成技术 .....</b>	<b>211</b>	习题七 .....	240
7.1 集成概念的发展与集成模型的分类 ..	211	一、选择题 .....	240
7.1.1 集成概念的发展 .....	211	二、填空题 .....	241
7.1.2 集成模型的分类 .....	212	三、简答题 .....	241
7.2 集成与耦合 .....	213	四、上机操作题 .....	241
7.3 三种集成模型概述 .....	214	<b>第 8 章 数字城市的软件组建模型 .....</b>	<b>242</b>
7.3.1 表示集成模型 .....	214	8.1 数字城市的相关概念 .....	242
7.3.2 数据集成模型 .....	215	8.1.1 数字城市的基本概念 .....	242
7.3.3 功能集成模型 .....	215	8.1.2 数字城市的内容 .....	243
7.4 软件集成的应用模式 .....	216	8.1.3 数字城市意义和作用 .....	244
7.4.1 企业应用集成 .....	216	8.2 基于元数据的数字城市数据组织 模型 .....	244
7.4.2 电子商务 .....	220	8.2.1 数字城市中元数据的内涵 .....	244
7.5 多数据库集成 .....	221	8.2.2 数字城市中元数据的特征 .....	247
7.5.1 CORBA 与多数据库集成的 技术 .....	223	8.2.3 基于元数据的数字城市数据 组织模型 .....	247
7.5.2 基于 CORBA 的多数据库集成 策略 .....	223	8.3 基于软件 Agent 的数字城市软件 构架模型 .....	249
7.5.3 基于 CORBA 的多数据库集成 结构和访问途径 .....	223	8.3.1 软件 Agent 在数字城市中的 适应性 .....	249
7.5.4 基于 ASP 的多源数据库集成 .....	224	8.3.2 基于 CORBA/DCOM 的软件 Agent 数字城市模型 .....	249
7.6 XML 技术以及基于 XML 的应用 集成 .....	225	小结 .....	253
7.6.1 XML 技术简介 .....	225	习题八 .....	253
7.6.2 利用 XML 进行应用集成 .....	227	一、选择题 .....	253
7.7 Web Service 技术与三种开发平台 ..	228	二、填空题 .....	254
7.7.1 Web Service 技术简介 .....	229	三、简答题 .....	254
7.7.2 三种典型的 Web Service 开发 平台 .....	233		

<b>附录 上机实训</b> .....	<b>255</b>	实训 5 用 RMI 实现“城市信息” 服务器的应用程序开发 .....	269
实训 1 用结构化方法验证 “歌德巴赫猜想” .....	255	实训 6 基于 CORBA 的数值计算 应用程序 .....	272
实训 2 用面向对象方法实现学生 信息管理 .....	257	实训 7 基于 CORBA 的银行应用程序 ....	275
实训 3 用 Delphi 7.0 中的组件实现 优化的数据库设计 .....	259	实训 8 使用 Voyager 开发移动代理 程序 .....	279
实训 4 用 CORBA 实现分布式照明 系统 .....	263	实训 9 Voyager 和 CORBA 的集成 .....	282
		<b>参考文献</b> .....	<b>286</b>

# 第 1 章 结构化技术

结构化方法是最早的、最传统的软件开发方法。结构化方法起源于 20 世纪 60 年代末期，它主要强调如何应用一些概念、方法与工具，以提高系统分析与设计、程式设计与测试的效率与功能。在当今的软件开发实践中，它仍然发挥着重要的作用。

本章教学目标：

- (1) 了解结构化设计的几种常用的图标工具。
- (2) 理解模块设计中的耦合和内聚。
- (3) 掌握结构化程序设计的思想和三种基本程序结构。
- (4) 熟悉用于面向数据结构设计的 Jackson 方法。

## 1.1 结构化程序

结构化程序设计方法是最基本的程序设计方法，这种方法简单，设计出来的程序可读性强、容易理解、便于维护，是面向对象程序设计的基础。结构化程序设计方法就是把模块分割作为对大型系统进行分析的手段，使其最终转化为三种基本结构。它的目的是解决由很多人共同开发一个大型软件时，如何更加高效地完成可靠系统开发的问题。

### 1.1.1 结构化程序设计

程序设计的最终目的是能让用户在计算机上得到所需要的结果。作为大脑思维的产物或反映，程序当然不能千篇一律，也绝不是随心所欲的产物。程序设计有它自身的一套基本原理和方法。

程序设计的发展史大体上可分为以下几个时期。

20 世纪 50 年代，程序都是用指令代码或汇编语言编写的，这种程序的设计相当麻烦，编制和调试一个稍大一点的程序常常要花费很长时间，培养一个熟练的程序员更需要经过长期的训练，这种局面严重影响了计算机的普及应用。

20 世纪 60 年代高级语言的出现大大简化了程序设计，缩短了解题周期，因此显示出强大的生命力。此后，编制程序已不再是只有软件专业人员才能做的事了，一般工程技术人员花上较短的时间学习，也可以使用计算机解题。这个时期，随着计算机的应用日益广泛地渗透到各学科和技术领域，发展了一系列不同风格的、为不同目标服务的程序设计语言。其中较为著名的有 FORTRAN、COBOL、ALGOL、LISP、PL/1、PASCAL 等十几种语言。高级语言的蓬勃兴起，使得编译和形式语言理论相应日趋完善，这是该时期的主要特征。但就整个程序设计方法而言，并无实质性的改进。

自 60 年代末到 70 年代初，出现了大型软件系统，如操作系统、数据库，这给程序设计带来了新的问题。大型系统的研发需要花费大量的资金和人力，可是研制出来的产品可靠性差、错误多，并且不易维护和修改。一个大型操作系统有时需要几千人/年的工作量，而所获得的系统又常常会隐藏着成百上千个错误。当时，人们称这种现象为“软件危机”。

为了克服软件危机，1968 年北约组织提出“软件工程”的概念。对程序设计语言的认

识从强调表达能力为重点转向以结构化和简明性为重点，将程序从语句序列转向相互作用的模块集合。这也就引出了结构化程序设计。

结构化程序设计是荷兰学者 E.W.Dijkstra 等人在研究人的智力局限性随着程序规模的增大而表现出来的不适应之后，于 1969 年提出的一种程序设计方法，这是一种完成复杂任务时避免混乱的技术。这种思想现在已经成为软件界的普遍共识。结构化程序设计者认为，在初始阶段，一般情况下，不能触及到问题的细节和实质方面。因此，首先应从全局上把握系统所应具备的功能是什么，并将系统划分成几个相对独立的子系统，而每个子系统只涉及局部的环境和条件。按照这种方法再将子系统细分成更小的子系统。这样，层层细分，逐步求精，最后，把一个大系统自顶而下逐步分割成可以理解的、容易实现的、功能简单的小片段。然后，进入第二步的实现阶段，这时就需要编写代码。在第一阶段，只要知道“做什么”，而在实现阶段，才考虑“如何做”。

结构化程序设计的思想规定了一套方法，这样便使得程序具有合理的结构，以保证和验证程序的正确性，这种方法要求程序设计者不能随心所欲地编写程序，而要按照一定的结构形式来设计程序和编写代码，它的一个重要目的是使程序具有良好的结构，使程序易于设计，易于理解，易于调试和修改，从而提高程序设计和维护的效率。

说到结构化程序设计就不能不提 GOTO 语句，因为任何一个结构化程序在编译以后都是需要用机器语言中的直接转移指令语句（同 GOTO 完全是一回事）来实现其结构的。然而，GOTO 语句却是破坏结构化原则最根本的因素。E.W.Dijkstra 就是使用 GOTO 语句的反对者，他曾在文章“GOTO Statement considered harmful”中说到：“GOTO 语句太原始，是构成程序混乱的祸根，应该从所有高级语言中消失”。当时，很多人支持他的观点。可是，也有相当一部分人认为 GOTO 语句简单易懂，可以提高程序设计的效率。在 20 世纪 60 年代末和 70 年代，关于 GOTO 语句的争论是比较激烈的。直到 1974 年，D.E.Knuth 对于 GOTO 语句的争论作了全面的公正的评述，并得到了普遍认可。他的基本观点是：不加限制地使用 GOTO 语句，特别是使用往回跳的 GOTO 语句，会使程序的结构难于理解，这种情形应该尽量避免使用 GOTO 语句；另外，为了提高程序的效率，同时又不破坏程序的良好结构，有控制地使用一些 GOTO 语句是有必要的。用他的话来说：“有些情形，我主张废除转向语句，有些情形我主张引进转向语句”。关于 GOTO 语句的争论实质上是关于要好的结构还是高的效率的问题。

Bohm 和 Jacopini 于 1966 年证明了只用三种最基本的控制结构便能实现任何单入口单出口程序，这三种控制结构是顺序、分支和循环，这一证明为结构化程序设计奠定了理论基础。并且“一个入口，一个出口”的原则成为结构化程序设计中的一个非常重要的原则。

那么，什么是结构化程序设计？到现在还没有一个为人所普遍接受的定义，一个常见的定义是：结构化程序设计是一种程序设计技术，它采用自顶向下、逐步求精的设计方法和单入口单出口的控制结构。

### 1.1.2 控制结构

为了方便描述控制结构和指令执行情况，首先来介绍程序流程图。程序流程图被定义为人们对解决问题的方法、思路或算法的一种描述。它有如下优点：

- (1) 采用的符号简单规范，画法简单；

(2) 结构清晰, 逻辑性强;

(3) 便于描述, 容易理解。

程序流程图是一个有向图, 通常由以下三种结点组成:

(1) 函数结点: 这种结点有一个入口线和一个出口线, 如图 1-1 (a) 所示。其中, F 是函数结点的名字。因为一般情况下, 函数结点和赋值语句对应, 故 F 也表示了这个结点所对应的函数关系。

(2) 谓词结点: 这种结点有一个入口线和两个出口线, 并且它不会改变程序的数据项的值, 如图 1-1 (b) 所示。其中, P 是一个谓词, 依据 P 的逻辑值, 结点对应不同的出口。

(3) 汇点: 这种结点有两个入口线和一个出口线, 并且它不执行任何运算, 如图 1-1 (c) 所示。

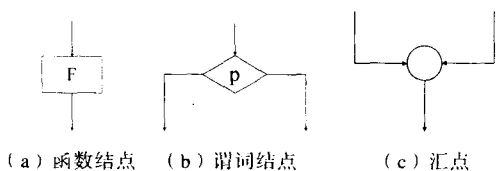


图 1-1 程序流程图的三种节点

下面给出结构化程序设计的三种基本控制结构: 顺序结构、分支结构和循环结构。

(1) 顺序结构。

顺序结构是一种最简单、最基本的结构。在这种结构中, 各程序块按照语句的书写顺序一条一条地执行, 如图 1-2 (a) 所示。一般情况下, 它有一个入口和一个出口, 内部可包含多个程序模块。

(2) 分支结构。

也称为分支选择结构, 是根据给定的条件判断在两条可能的路径中选择其中一条, 这两条路径分别有不同的处理功能, 不妨记作“a 块”和“b 块”。在处理完 a 块或 b 块后, 就从分支结构出来, 然后执行其后的程序。分支结构如图 1-2 (b) 所示。

(3) 循环结构。

循环结构也称为“重复处理结构”, 是在满足给定条件时反复执行某一程序块 a, 否则 (即条件不满足时) 不执行, 如图 1-2 (c) 所示。从图中可以看出, 循环结构也只有一个入口和一个出口。

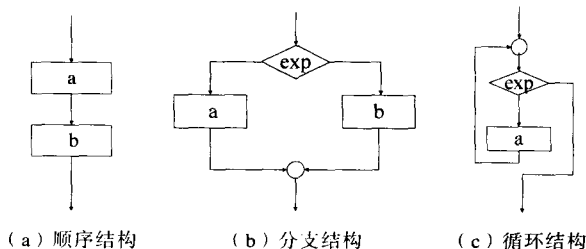


图 1-2 三种基本控制结构

为实际使用方便起见, 通常还有另外两种控制结构: 多分支结构和 UNTIL 结构。实际上, 这两种结构分别是对上面所介绍的分支结构和循环结构的扩充。这里就不再详细介绍。

## 1.2 结构化分析与设计的一般步骤

利用结构化方法开发软件最关键的阶段就是结构化分析与设计。这个阶段的主要任务是分析软件的功能、性能、目标、规模等需求，定义软件的逻辑模型；设计软件的模块结构、数据文件等；给出模块说明和主要算法；为后面的编码实现作算法上和结构上的准备。

在结构化方法的软件生命周期内，软件分析与设计的基本阶段可以分为三个阶段：需求分析、概要设计和详细设计。各阶段的主要文档如图 1-3 所示。

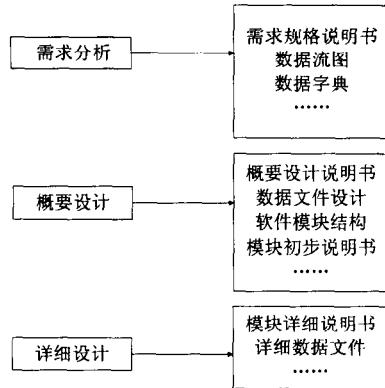


图 1-3 软件分析与设计的基本阶段及主要文档

### 1. 结构化需求分析

结构化分析又称作数据流法，数据流技术是需求分析的结构化方法核心技术。分析过程以数据流为驱动，分析系统的信息流向、信息类型、流量、所需处理的功能和性能，从而推导出系统的逻辑结构。具体体现为需求分析文档、规格说明书、数据流图、数据字典等。

需求规格说明书从总体上来概括软件的目标、功能和性能，并且需求规格说明书应按一定格式编写。

数据流图与数据字典作为结构化需求分析的核心，具体反映软件各种需求和关联。数据流图从宏观上描述软件的信息流向、信息存储要求、数据处理功能等；而数据字典是对数据流图的详细说明。数据流图和数据字典为结构化设计提供了必要基础。

### 2. 结构化概要设计

结构化概要设计以模块化技术为基础，其主要任务是在结构化需求分析的基础上建立软件的总体结构，从而设计具体的数据结构。结构化概要设计基本方法是从数据流图出发推导出软件的模块结构图，从数据字典推导出模块基本要求、数据存储要求、数据结构及数据文件等。

### 3. 结构化详细设计

结构化详细设计是对结构化概要设计的进一步细化，目标是为软件结构图中每个模块提供具体算法，以供程序员编程实现。

## 1.3 结构化分析

结构化分析在有些书上称为数据流法，它的基本思想是跟踪数据流，主要研究问题域



中数据如何流动以及各个环节上如何处理,从而发现数据流和加工。问题域被映射为由数据流、加工以及文件、端点等成分构成的数据流图,然后用处理说明和数据字典对数据流和加工进行详细说明。

### 1.3.1 数据流分析

20世纪70年代,美国的Yourdon公司提出了结构化分析,这是一种简明实用且已经得到广泛使用的方法。这种方法的核心是数据流技术,使其适用于分析大型数据处理系统,特别是管理信息系统,旨在减少分析活动中的错误,产生系统的逻辑模型。

数据流分析方法源于结构化分析,它以数据流技术为基础,是一种自顶向下逐层分解、由粗到细、由复杂到简单的求解方法。通常所说的“结构化分析”就是“数据流分析”。

抽象和分解是结构化分析方法中解决复杂问题的两个基本手段。抽象就是抓住主要问题忽略次要问题,集中精力先解决主要问题;分解就是把大问题分解成若干个小问题,然后逐个解决。

按照以上思想,于是产生了自顶向下、逐层分解的方法。如图1-4所示,X是一个复杂的管理系统,为了更好地理解它,这里将其分解成1、2、3、4四个子系统。若子系统1、3仍然很复杂,可继续将它们分成1.1、1.2、...和3.1、3.2、...等子系统,如此逐层分解直至分解出的各个子系统足够简单、可被清楚理解和准确表达为止。

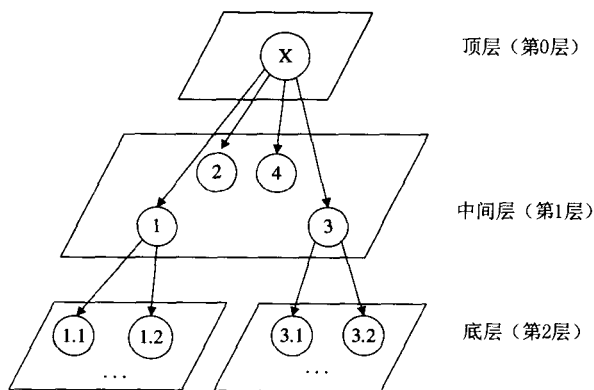


图 1-4 自顶向下逐层分解示意图

按照自顶向下、逐层分解的方法,不管系统是如何的复杂或者规模多么大,分析工作都可以根据这个思想有条不紊地进行。对于大的系统只需多分解几层,而分析的复杂程度并不会随之增大。这正是结构化分析的一大特点。

在需求分析与可行性研究中,结构化方法是最传统且非常有效的一种方法。在结构化方法中,采用数据流图来建立系统的逻辑模型,用数据字典对数据进行说明分析。下面将介绍这两种基本方法和工具。

### 1.3.2 数据流图

数据流图主要用来描述数据处理过程,是结构化分析中最基本的描述工具。数据流图从数据传递和数据加工的角度,以图形的方式刻画数据流从输入到输出的传输变换过程。数据流图作为结构化系统分析的主要工具,它既表示了系统内部信息的流向,也表示了系

统的逻辑处理的功能。由于数据流图是逻辑系统的图形表示，即使非计算机专业的人士也能理解，所以它是很好的通信工具。数据流图以图形的方式描绘数据在系统中流动与处理的过程，并且只反映系统必须完成的逻辑功能，所以说数据流图是一种功能模型。

数据流图有四种基本元素：数据流、数据处理、数据存储和外部实体。当然，在具体软件开发中可以根据具体需要增加其他辅助性元素。

(1) 数据流：是系统内数据传播的路径，它由一组成分固定的数据组成。如订票单由旅客的姓名、年龄、单位、身份证号、日期、目的地等数据项组成。由于数据流是流动中的数据，所以必须有流向，除了与数据存储之间的数据流不用赋予名字外，数据流一般用名词或名词短语命名。

(2) 数据处理（也称作逻辑加工或变换）：对数据流进行某些操作或变换。数据处理也有名字，通常是动词短语，简明地描述完成什么加工。分层的数据流图中，数据处理还应编号。

(3) 数据存储（也称作文件）：指暂时保存的数据，它可以是数据库文件或其他任何形式的数据组织。

(4) 外部实体：是系统外部环境中的实体（包括人员、组织或其他软件系统）。对系统提供数据流的外部实体称作数据源点，用来接收输出数据流的外部实体称作数据终点。需要说明的是，数据源点和数据终点可以是同一外部实体。

数据流图有多种表示方法，图 1-5 是三种常见的符号表示。

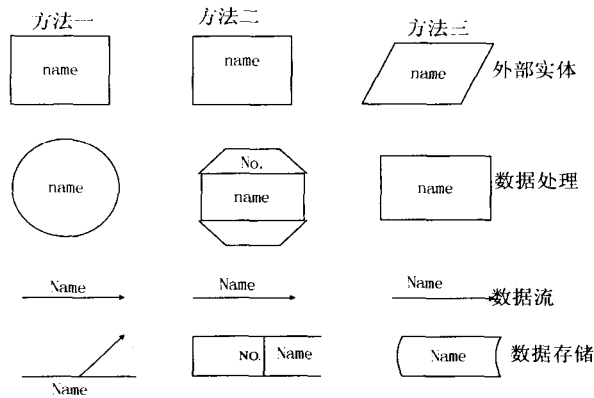


图 1-5 数据流图的常见符号

图 1-6 是一个飞机机票预订系统的数据流图。

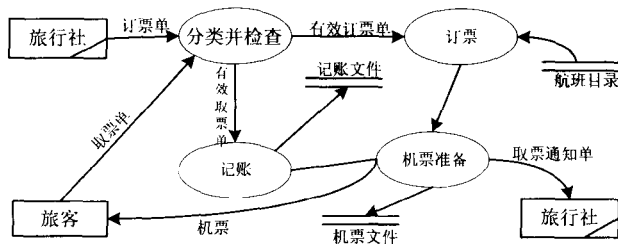


图 1-6 飞机票预订系统的数据流图

其中，旅行社、旅客是外部实体，即数据源点和数据终点；分类并检查、订票、记账、