

● 高等学校21世纪计算机教材

# 面向对象 分析与设计

李代平 编著

冶金工业出版社

高等学校 21 世纪计算机教材

# 面向对象分析与设计

李代平 编著



北 京

冶金工业出版社

## 内 容 简 介

本书是论述面向对象技术的专著。全书共分十九章，系统地介绍了面向对象的基本理论和方法，主要对面向对象分析与设计的基本概念、主要步骤、典型特点、关键问题等进行了较为详细的介绍和阐述，同时还介绍了面向对象的表示法、面向对象的开发过程、面向对象分析与设计在各个领域的应用。

本书内容详实、立论严谨、实例丰富、图文并茂，适合作为高等学校相关专业的教材及工程技术人员的参考用书。

### 图书在版编目 (CIP) 数据

面向对象分析与设计 / 李代平编著. —北京: 冶金工业出版社, 2005.7

ISBN 7-5024-3775-4

I. 面... II. 李... III. 面向对象语言—程序设计  
IV. TP312

中国版本图书馆 CIP 数据核字 (2005) 第 065359 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 戈兰

佛山市新粤中印刷有限公司印刷; 冶金工业出版社发行; 各地新华书店经销

2005 年 7 月第 1 版第 1 次印刷

787mm × 1092mm 1/16; 26.75 印张; 617 千字; 416 页

39.00 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、本书背景

随着科学技术的进步，软件的理论及开发方法不断涌现。面向对象技术是指导计算机软件开发的工程科学技术。面向对象技术的概念、原理、技术与方法已成为计算机科学与技术的一项重要内容。

用面向对象技术进行软件设计与开发的先进性是众所周知的，它在计算机科学技术领域占据了无可争议的主流地位。作为软件技术人员，接受面向对象技术的概念并不难，但是要真正理解、掌握和运用这门先进的技术并完整地进行系统开发，是有一定难度的。鉴于此我们编写了本书，其目的是向读者提供一本关于面向对象技术系统分析、设计和实施的教科书，以使更多同行受益。

## 二、本书结构

本书由六部分组成：

第一部分：概念。包括：面向对象方法概论，面向对象方法论。

第二部分：结构。包括：面向对象建模，面向对象体系结构，动态模型，功能模型。

第三部分：分析。包括：发现对象、建立对象类，定义属性与服务，定义结构与连接。

第四部分：设计原则。包括：面向对象方法的转换，面向对象设计原则。

第五部分：设计方法。包括：控制驱动部分的设计，设计软件构件，对象设计，数据库及其接口设计，人机交互部分的设计。

第六部分：分析与设计应用。包括：面向对象分析与软件复用，实时系统，模糊面向对象数据模型。

## 三、本书特点

本书对每章的概念都进行了严格的论述，每一个概念都有相应的例子解释，同时每章都配有习题，使读者巩固所学知识。

## 四、适用范围

面向对象分析与设计是软件系统开发课程的教科书。讲授时间为 44~54 学时。本书适合开设有面向对象系统开发课程或者软件工程课程的大学高年级和低年级研究生作教材，也可作为工程技术人员的参考用书。

在选修本课程之前，读者应该具有计算机的基础知识，同时具有 Visual 类语言或者 C/C++ 的编程经验，会有助于深入理解信息系统开发过程。

## 五、编写方法

本书是作者根据近十年来对软件工程学、面向对象方法等教学与研究，以及作者领导或参与的二十项软件项目开发的实际应用经验，并结合软件开发新技术编写而成。根据过

去的教学经验，读者学习一门新技术，教材是非常重要的。因此，在编写之前，在各方面进行了充分的准备。

## 六、如何使用本书

面向对象系统分析和设计分为六部分。根据读者的实际情况，教师在教授本书时，可以按照自己的风格和喜好删除章节，也可以根据教学目标灵活调整章节顺序。另外，前面带\*为选学内容。

- 第1章 面向对象方法概论（2学时）
- 第2章 面向对象方法论（4学时）
- 第3章 面向对象建模（2学时）
- \*第4章 面向对象体系结构（2学时）
- 第5章 动态模型（2学时）
- 第6章 功能模型（2学时）
- 第7章 发现对象、建立对象类（6学时）
- 第8章 定义属性与服务（4学时）
- 第9章 定义结构与连接（4学时）
- 第10章 面向对象方法的转换（2学时）
- 第11章 面向对象设计原则（4学时）
- 第12章 控制驱动部分的设计（4学时）
- \*第13章 设计软件构件（2学时）
- 第14章 对象设计（4学时）
- 第15章 数据库及其接口设计（2学时）
- 第16章 人机交互部分的设计（2学时）
- \*第17章 面向对象分析与软件复用（2学时）
- \*第18章 实时系统（2学时）
- \*第19章 模糊面向对象数据模型（2学时）

参加本书编写的除作者外，还有张信一、罗伟刚。

由于面向对象分析与设计知识面广，在介绍中不能面面俱到。加上时间仓促，作者水平有限，书中的不足之处在所难免，恳请读者批评指正。

虽然经过严格的审核、精细的编辑，本书在质量上有了一定的保障，但我们的目标是力求尽善尽美，欢迎广大读者和专家对我们的工作提出宝贵建议，联系方法如下：

电子邮件：[service@cnbook.net](mailto:service@cnbook.net)

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

本书电子教案和练习题参考答案可在本网站免费下载，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  
2004年6月

# 目 录

<b>第 1 章 面向对象方法概论</b> .....1	2.1.5 操作和方法	21
1.1 面向对象简介	2.1.6 封装	22
1.1.1 面向对象的思维	2.1.7 继承	23
1.1.2 什么是面向对象	2.1.8 多重继承	24
1.1.3 面向对象方法的历史及现状	2.1.9 消息	26
1.1.4 当前的研究及实践领域	2.1.10 结构与连接	27
1.2 面向对象方法的形成	2.1.11 多态性	28
1.2.1 对事物的认识和描述	2.1.12 永久对象	28
1.2.2 对象的特性	2.1.13 主动对象	29
1.2.3 软件工程学的作用	2.1.14 对象类的表示方法	30
1.3 面向对象开发的概念	2.2 链接与关联	31
1.3.1 面向对象方法学	2.2.1 一般概念	31
1.3.2 三种模型	2.2.2 重数	31
1.3.3 功能方法学上的差异	2.2.3 关联的重要性	32
1.4 面向对象技术要点	2.2.4 三元关联	32
1.4.1 抽象	2.2.5 关联的候选关键字	33
1.4.2 封装	2.2.6 异或关联	33
1.4.3 数据和行为的联合	2.2.7 资格关联	33
1.4.4 共享	2.2.8 链接属性	34
1.4.5 面向对象开发重点	2.2.9 用关联模型化为类	34
1.4.6 协同作用	2.2.10 角色名	35
1.5 面向对象开发的可用性例证	2.2.11 排序	35
1.6 面向对象方法的要点和主要优点	2.2.12 资格符	36
小结	2.3 聚合	36
综合练习一	2.3.1 聚合与关联	36
一、填空题	2.3.2 聚合和概括	37
二、选择题	2.3.3 递归聚合	38
三、简答题	2.3.4 操作的传播	38
	2.3.5 物理聚合与分类聚合	39
	2.3.6 物理聚合的语义扩展	39
	2.3.7 分类聚合的语义扩展	40
<b>第 2 章 面向对象方法论</b> .....18	2.4 概括	40
2.1 面向对象的相关概念	2.4.1 一般概念	40
2.1.1 对象	2.4.2 概括的使用	41
2.1.2 类	2.4.3 重写特征	41
2.1.3 对象图		
2.1.4 属性		

2.4.4 抽象类和具体类.....	41	综合练习三.....	80
2.4.5 概括与其他对象建模结构.....	42	一、填空题.....	80
2.5 构造分组.....	42	二、选择题.....	80
2.5.1 模块.....	42	三、简答题.....	81
2.5.2 表.....	43	<b>第4章 面向对象体系结构.....</b>	<b>82</b>
小结.....	43	4.1 体系结构表示法.....	82
综合练习二.....	43	4.2 流程处理系统.....	83
一、填空题.....	43	4.3 客户机/服务器系统.....	83
二、选择题.....	43	4.3.1 客户机/服务器系统简介.....	83
三、简答题.....	44	4.3.2 基于MVC的网上应用系统.....	84
<b>第3章 面向对象建模.....</b>	<b>45</b>	4.4 层状系统.....	85
3.1 建模.....	45	4.4.1 层状系统简介.....	85
3.1.1 使用模型的目的.....	45	4.4.2 以服务对象分层.....	85
3.1.2 抽象.....	45	4.5 三级和多级系统.....	86
3.2 统一建模语言.....	46	4.5.1 三级系统.....	86
3.2.1 模型的建立.....	47	4.5.2 多级系统.....	86
3.2.2 面向对象的理念.....	48	4.6 代理.....	86
3.2.3 系统观点.....	51	4.7 联邦体系结构和聚合体系结构.....	87
3.2.4 规划分析.....	51	4.7.1 联邦体系结构.....	87
3.2.5 UML的基本图标.....	51	4.7.2 聚合体系结构.....	87
3.2.6 软件开发的传统与现代.....	58	4.8 体系结构的样式.....	88
3.2.7 UML的基本元素.....	60	4.9 案例:自动柜员机系统软件.....	89
3.2.8 UML的语法规则.....	60	4.10 案例:租借公司.....	90
3.2.9 UML的词别.....	60	4.11 案例:共用讨论板.....	91
3.3 对象模型技术.....	61	4.11.1 组的动态组合变更过程.....	91
3.3.1 对象模型.....	61	4.11.2 界面层与控制层.....	92
3.3.2 动态模型.....	61	4.11.3 同步及相关问题.....	94
3.3.3 功能模型.....	62	4.12 案例:EJB的架构.....	94
3.3.4 三种模型的联系.....	62	4.12.1 静态结构.....	94
3.4 其他方法.....	62	4.12.2 资源管理策略.....	97
3.4.1 Rumbaugh面向对象方法论.....	62	4.12.3 实体豆的动态行为.....	97
3.4.2 Shlaer/Mellor面向对象方法论.....	66	小结.....	99
3.4.3 Jacobson面向对象方法论.....	68	综合练习四.....	99
3.4.4 Booch面向对象方法论.....	70	一、填空题.....	99
3.4.5 Yourdon面向对象方法论.....	71	二、选择题.....	99
3.4.6 主流面向对象模型.....	74	三、简答题.....	99
3.4.7 Martin/Odell面向对象方法论.....	79	<b>第5章 动态模型.....</b>	<b>100</b>
小结.....	80		

5.1 事件和状态.....100	6.2.6 控制流.....122
5.1.1 事件.....100	6.3 指定的操作.....122
5.1.2 脚本和事件轨迹.....101	6.4 约束.....123
5.1.3 状态.....102	6.5 数据库应用中的功能模型.....124
5.1.4 状态图.....103	6.5.1 伪码.....124
5.1.5 条件.....104	6.5.2 ONN 的伪代码.....125
5.2 操作.....105	6.5.3 ONN 的构造.....126
5.2.1 控制操作.....105	6.5.4 组合 ONN 构造.....130
5.2.2 操作的状态图小结.....105	6.5.5 添加 ONN 特性.....131
5.3 嵌套状态图.....106	6.5.6 其他范型.....132
5.3.1 平状态图问题.....106	6.5.7 实践技巧.....134
5.3.2 嵌套状态图.....106	6.6 功能模型的实例(飞行模拟机 装置).....135
5.3.3 状态概括.....107	6.7 功能模型与对象模型和 动态模型的关系.....137
5.3.4 事件概括.....108	小结.....138
5.4 并发性.....109	综合练习六.....138
5.4.1 聚合并发性.....109	一、填空题.....138
5.4.2 对象内部并发性.....109	二、选择题.....139
5.4.3 进入和退出动作.....110	三、简答题.....139
5.4.4 内部动作.....111	<b>第7章 发现对象、建立对象类.....140</b>
5.4.5 自动变迁.....111	7.1 对象、主动对象以及它们的类.....140
5.4.6 发送事件.....111	7.2 表示法.....141
5.4.7 并发活动的同步.....112	7.3 研究问题域和用户需求.....142
5.5 动态模型的实例.....113	7.3.1 研究用户需求,明确系统责任..142
5.6 对象模型和动态模型的关系.....115	7.3.2 研究问题域.....143
5.7 实践技巧.....116	7.3.3 确定系统边界.....144
小结.....116	7.4 发现对象.....144
综合练习五.....116	7.4.1 发现对象技术概要.....144
一、填空题.....116	7.4.2 正确地运用抽象原则.....145
二、选择题.....117	7.4.3 策略与启发.....145
三、简答题.....117	7.4.4 审查和筛选.....147
<b>第6章 功能模型.....118</b>	7.4.5 识别主动对象.....149
6.1 功能模型.....118	7.5 对象分类,建立类图的对象层.....149
6.2 数据流图.....119	7.5.1 异常情况的检查和调整.....149
6.2.1 处理.....119	7.5.2 类的命名.....150
6.2.2 数据流.....120	7.5.3 建立类图的对象层.....150
6.2.3 施动者.....120	7.6 电梯例子.....151
6.2.4 数据存储.....121	
6.2.5 嵌套数据流图.....121	



7.6.1 功能需求 .....	151	9.1.1 整体-部分结构及其用途 .....	172
7.6.2 发现对象 .....	151	9.1.2 表示法 .....	174
7.6.3 对象层表示 .....	152	9.1.3 如何发现整体-部分结构 .....	175
小结 .....	152	9.1.4 审查与筛选 .....	175
综合练习七 .....	152	9.1.5 简化对象的定义 .....	176
一、填空题 .....	152	9.1.6 支持软件复用 .....	176
二、选择题 .....	153	9.1.7 整体-部分结构的进一步运用.....	177
三、简答题 .....	153	9.1.8 调整对象层和属性层.....	179
<b>第8章 定义属性与服务 .....</b>	<b>154</b>	9.2 一般-特殊结构 .....	179
8.1 对象的属性和服务 .....	154	9.2.1 一般-特殊结构及其用途 .....	179
8.2 表示法 .....	155	9.2.2 表示法 .....	180
8.3 定义属性 .....	155	9.2.3 如何发现一般-特殊结构 .....	180
8.3.1 策略与启发 .....	155	9.2.4 审查与调整 .....	181
8.3.2 审查与筛选 .....	156	9.2.5 多继承及多态性问题.....	182
8.3.3 推迟到 OOD 考虑的问题.....	157	9.2.6 一般-特殊结构的简化 .....	184
8.3.4 属性的命名和定位.....	158	9.2.7 调整对象层和特征层.....	185
8.3.5 属性的详细说明.....	158	9.3 实例连接 .....	185
8.4 定义服务 .....	158	9.3.1 简单的实例连接 .....	185
8.4.1 对象的状态与状态转换图 .....	158	9.3.2 复杂的实例连接及其表示 .....	186
8.4.2 行为分类 .....	160	9.3.3 三元关联问题 .....	188
8.4.3 发现服务的策略与启发 .....	161	9.3.4 如何建立实例连接 .....	189
8.4.4 审查与调整 .....	161	9.3.5 对象层、特征层的增补及 实例连接说明 .....	190
8.4.5 认识对象的主动行为 .....	162	9.4 消息连接 .....	190
8.4.6 服务的命名和定位.....	162	9.4.1 消息的定义 .....	190
8.4.7 服务的详细说明.....	162	9.4.2 顺序系统中的消息 .....	190
8.5 建立类图的特征层 .....	163	9.4.3 并发系统中的消息 .....	191
8.6 电梯例子 .....	163	9.4.4 消息对 OOA 的意义 .....	194
8.6.1 电梯系统的属性描述 .....	163	9.4.5 OOA 对消息的表示 ——消息连接 .....	194
8.6.2 电梯系统的服务定义.....	165	9.5 如何建立消息连接.....	196
8.6.3 电梯系统的特征层 .....	170	9.5.1 建立控制线程内部的消息连接..	196
小结 .....	171	9.5.2 建立控制线程之间的消息连接..	196
综合练习八 .....	171	9.5.3 对象分布问题及其消息的影响..	197
一、填空题 .....	171	9.6 消息的详细说明 .....	198
二、选择题 .....	171	9.7 电梯例子 .....	198
三、简答题 .....	171	9.7.1 一般-特殊关系 .....	198
<b>第9章 定义结构与连接 .....</b>	<b>172</b>	9.7.2 整体-部分关系 .....	198
9.1 整体-部分结构 .....	172		

9.7.3 连接 .....	199	11.4.4 混淆类及其实例 .....	221
9.7.4 电梯控制系统的关系层 .....	199	11.4.5 误用 .....	222
小结 .....	199	11.4.6 多态性的危险性 .....	222
综合练习九 .....	200	11.5 状态空间和行为 .....	224
一、填空题 .....	200	11.5.1 一个类的状态空间和行为 .....	224
二、选择题 .....	200	11.5.2 子类的状态空间 .....	225
三、简答题 .....	200	11.5.3 子类的行为 .....	225
<b>第 10 章 面向对象方法的转换 .....</b>	<b>201</b>	11.5.4 状态空间的一个约束条件： 类的不变式 .....	225
10.1 从布什的面向对象升级到 UML .....	201	11.5.5 前置条件和后置条件 .....	226
10.2 对象模块化技术升级到 UML .....	202	11.5.6 类接口中支持的状态 .....	227
10.3 一般性的升级方式 .....	203	11.5.7 类接口中支持的行为 .....	228
10.4 模块化会议 .....	204	11.5.8 类接口中操作的聚合 .....	229
小结 .....	206	小结 .....	229
综合练习十 .....	206	综合练习十一 .....	229
一、填空题 .....	206	一、填空题 .....	229
二、选择题 .....	206	二、选择题 .....	229
三、简答题 .....	207	三、简答题 .....	230
<b>第 11 章 面向对象设计原则 .....</b>	<b>208</b>	<b>第 12 章 控制驱动部分的设计 .....</b>	<b>231</b>
11.1 类型一致性与闭合行为 .....	208	12.1 什么是控制驱动部分 .....	231
11.1.1 类与类型 .....	208	12.2 相关技术问题 .....	231
11.1.2 类型一致性原则 .....	208	12.2.1 系统总体方案 .....	231
11.1.3 闭合行为原则 .....	209	12.2.2 软件体系结构 .....	232
11.2 封装与共生性 .....	210	12.2.3 分布式系统的体系结构风格 .....	233
11.2.1 封装结构 .....	210	12.2.4 系统的并发性 .....	236
11.2.2 共生性 .....	211	12.3 如何设计控制驱动部分 .....	239
11.2.3 面向对象系统中共生性的 滥用 .....	213	12.3.1 选择软件体系结构风格 .....	239
11.2.4 共生性的术语 .....	214	12.3.2 确定系统分布方案 .....	239
11.3 领域、依附集和内聚 .....	214	12.3.3 识别控制流 .....	243
11.3.1 对象类的领域 .....	214	12.3.4 用主动对象表示控制流 .....	245
11.3.2 依附集 .....	216	12.3.5 把控制驱动部分看作 一个主题 .....	247
11.3.3 类的内聚：一个类和 它的特性 .....	218	小结 .....	247
11.4 继承与多态性的危险性 .....	220	综合练习十二 .....	247
11.4.1 继承的滥用 .....	220	一、填空题 .....	247
11.4.2 错误的聚集 .....	220	二、选择题 .....	247
11.4.3 倒置的层次结构 .....	221	三、简答题 .....	248

**第 13 章 设计软件构件 ..... 249**

13.1 构件的构成与表达方式 .....	249
13.2 构件与对象的相似性和区别 .....	250
13.3 构件技术简介 .....	251
13.3.1 构件技术的概念及发展 .....	251
13.3.2 当前的三种构件标准规范 .....	252
13.3.3 三种分布式构件的比较分析 .....	254
13.4 构件的大小与分类 .....	255
13.4.1 构件的大小 .....	255
13.4.2 构件的分类 .....	256
13.5 如何用好构件 .....	256
13.5.1 原则 .....	257
13.5.2 过程 .....	258
13.5.3 Java 应用构件平台 .....	259
13.6 构件的优点和缺点 .....	260
13.6.1 构件的优点 .....	260
13.6.2 构件的缺点 .....	260
小结 .....	261
综合练习十三 .....	261
一、填空题 .....	261
二、选择题 .....	262
三、简答题 .....	262

**第 14 章 对象设计 ..... 263**

14.1 对象设计综述 .....	263
14.1.1 从分析和系统结构着手 .....	263
14.1.2 对象设计的步骤 .....	264
14.1.3 对象模型工具 .....	264
14.2 组合三种模型 .....	264
14.3 设计算法 .....	265
14.3.1 选择算法 .....	265
14.3.2 选择数据结构 .....	267
14.3.3 定义内部类和操作 .....	267
14.3.4 指定操作的职责 .....	268
14.4 设计优化 .....	268
14.4.1 添加冗余关联获取有效访问 .....	269
14.4.2 重新安排执行次序以 获得效率 .....	270

14.4.3 保存导出属性避免重复计算 .....	270
14.5 控制实现 .....	271
14.5.1 在程序内进行状态设置 .....	271
14.5.2 状态机器引擎 .....	272
14.5.3 控制作为并发任务 .....	272
14.6 继承的调整 .....	272
14.6.1 重新安排类和操作 .....	272
14.6.2 抽象出公共的行为 .....	273
14.6.3 使用授权共享实现 .....	274
14.7 关联设计 .....	275
14.7.1 分析关联遍历 .....	275
14.7.2 单向关联 .....	275
14.7.3 双向关联 .....	276
14.7.4 链接属性 .....	276
14.8 对象的表示 .....	277
14.9 物理打包 .....	277
14.9.1 信息隐藏 .....	278
14.9.2 实体的相关性 .....	278
14.9.3 构造模块 .....	279
14.10 设计决策文档 .....	279
小结 .....	280
综合练习十四 .....	280
一、填空题 .....	280
二、选择题 .....	280
三、简答题 .....	281

**第 15 章 数据库及其接口设计 ..... 282**

15.1 数据管理系统及其选择 .....	282
15.2 数据库系统 .....	289
15.2.1 面向对象技术 .....	293
15.2.2 面向对象数据库的应用 .....	295
15.2.3 应用程序设计程序 .....	296
15.2.4 面向对象数据库的最佳化 .....	297
15.3 技术整合 .....	297
15.4 数据接口 .....	298
15.5 对象存储方案和数据接口的 设计策略 .....	300
15.5.1 针对文件系统的设计 .....	300
15.5.2 针对 RDBMS 的设计 .....	304

15.5.3 使用 OODBMS .....	313	17.1.3 维也纳开发方法 .....	339
小结 .....	313	17.1.4 面向对象方法 .....	340
综合练习十五 .....	313	17.2 软件复用 .....	340
一、填空题 .....	313	17.2.1 软件复用的层次 .....	340
二、选择题 .....	314	17.2.2 软件复用所遭遇的问题 .....	341
三、简答题 .....	314	17.3 对象模型化技术 .....	341
<b>第 16 章 人机交互部分的设计 .....</b>	<b>315</b>	17.3.1 OMT 的三大模型 .....	341
16.1 什么是人机交互部分 .....	315	17.3.2 基本结构技术 .....	343
16.2 人机交互部分的需求分析 .....	315	17.3.3 抽象类技术 .....	343
16.2.1 分析活动者		17.3.4 组件技术 .....	343
——与系统交互的人 .....	316	17.3.5 设计样板技术 .....	344
16.2.2 从 Use Case 分析人机交互 .....	316	17.4 统一建模语言 (UML) .....	344
16.2.3 分析处理异常事件的		17.4.1 UML 的发展 .....	344
人机交互 .....	319	17.4.2 统一建模语言 (UML) 的	
16.2.4 命令的组织 .....	320	内容 .....	345
16.2.5 输出信息的组织结构 .....	323	17.4.3 统一建模语言 (UML) 的	
16.2.6 总结与讨论 .....	324	主要特点 .....	347
16.3 人机界面的设计准则 .....	325	17.4.4 统一建模语言 (UML) 的	
16.4 人机界面 (OO) 设计 .....	326	应用领域 .....	347
16.4.1 界面支持系统 .....	326	17.5 面向对象技术的新应用 .....	347
16.4.2 界面元素 .....	327	17.5.1 可重用软件组件 .....	348
16.4.3 设计过程与策略 .....	328	17.5.2 分布式对象中间件 .....	350
16.5 可视化编程环境下的		17.5.3 并发面向对象技术 .....	351
人机界面设计 .....	332	17.5.4 基于面向对象技术的	
16.5.1 问题的提出 .....	332	应用软件体系结构 .....	352
16.5.2 设计的必要性 .....	332	17.6 面向对象的分布式开发系统 .....	355
16.5.3 基于可视化编程环境的		17.6.1 概述 .....	355
设计策略 .....	334	17.6.2 分布式系统所应具有的	
小结 .....	336	体系结构 .....	355
综合练习十六 .....	337	17.6.3 分布式系统的关键技术 .....	357
一、填空题 .....	337	17.6.4 分布式系统对开发流程的	
二、选择题 .....	337	改进 .....	360
三、简答题 .....	337	小结 .....	361
<b>第 17 章 面向对象分析与软件复用 .....</b>	<b>338</b>	综合练习十七 .....	361
17.1 软件开发方法 .....	338	一、填空题 .....	361
17.1.1 结构化方法 .....	338	二、选择题 .....	361
17.1.2 Jackson 方法 .....	338	三、简答题 .....	361
<b>第 18 章 实时系统 .....</b>	<b>362</b>		

18.1 实时系统的简介及其分类 .....	362	三、简答题 .....	386
18.1.1 实时系统简介 .....	362	<b>第 19 章 模糊面向对象数据模型 .....</b>	<b>387</b>
18.1.2 实时系统的分类 .....	363	19.1 模糊集合理论 .....	387
18.2 实时系统的相关概念 .....	363	19.1.1 模糊理论简介 .....	387
18.2.1 实时系统的概念 .....	364	19.1.2 模糊集合及逻辑 .....	388
18.2.2 前后台系统 .....	364	19.2 数据系统模型和结构 .....	390
18.2.3 任务与多任务 .....	364	19.2.1 层次模型 .....	391
18.2.4 共享资源与临界区 .....	366	19.2.2 网状模型 .....	392
18.2.5 内核 .....	366	19.2.3 关系模型 .....	393
18.2.6 调度 .....	368	19.3 模糊数据模型 .....	394
18.2.7 同步 .....	370	19.3.1 模糊网状数据模型 .....	394
18.2.8 互斥 .....	371	19.3.2 模糊层次数据模型 .....	397
18.2.9 中断 .....	376	19.3.3 模糊实体-联系数据模型 .....	398
18.2.10 使用实时内核的优缺点 .....	380	19.3.4 模糊关系数据模型 .....	400
18.3 实时系统与通用系统的区别 .....	380	19.3.5 面向对象的模糊数据模型 .....	407
18.4 多种实时系统的比较 .....	381	19.4 模糊数据库语言 .....	411
18.4.1 基本特征概述 .....	381	19.4.1 语言的模糊模型 .....	411
18.4.2 体系结构异同 .....	382	19.4.2 面向对象的模糊数据库语言 .....	412
18.4.3 调度策略分析 .....	382	小结 .....	414
18.4.4 系统开放性对比 .....	383	综合练习十九 .....	414
18.4.5 系统服务比较 .....	384	一、填空题 .....	414
18.5 实时系统与面向对象 .....	385	二、选择题 .....	415
小结 .....	386	三、简答题 .....	415
综合练习十八 .....	386	<b>参考文献 .....</b>	<b>416</b>
一、填空题 .....	386		
二、选择题 .....	386		

# 第 1 章 面向对象方法概论

面向对象方法的基本思想是从现实世界中客观存在的事物（即对象）出发来构造软件系统，并在系统构造中尽可能地运用人类的自然思维方式。面向对象开发是现实世界中以抽象为基础的软件思维的一种崭新方式。“开发”的含义延伸至软件生命周期的前期部分：分析、设计和实现。面向对象开发的本质是应用领域概念的鉴别和组织，而不是它们在一种程序设计语言中的最终表示。是面向对象还是非面向对象，其软件开发的硬指标是问题固有本质上的操纵，而不是对特定语言的操纵。本章将详细介绍面向对象的相关知识。

## 1.1 面向对象简介

“面向对象”是把一组相互之间没有联系的对象有效地集成在一起，这些对象都是将数据结构和行为紧密地结合在一起的。这和传统的程序设计不同，传统的思想是将数据结构和行为松散地连接在一起。当然，这里存在着哪些特性是真正的面向对象方法的争论。但不管怎么说，通常面向对象方法至少包含四个方面：标识、分类、多态和继承。

“面向对象”是把一组对象中的数据结构和行为紧密结合在一起组织系统的一种策略。它应用于许多技术领域，包括硬件、编程语言、数据库、软件工程和用户接口。软件开发的面向对象方法超越了面向过程的方法。譬如，面向对象方法是围绕操作相同的类型来组织数据库结构和程序的，而面向对象的程序更加接近于问题描述的实际而使开发与需求更加吻合。

通过将抽象、封装和模块结合在一起的方法，从而使面向对象的思想贯穿于整个软件开发生命周期中。抽象是一个重要概念。为了能使用计算机处理来自现实世界的事物，人们必须提取出事物的本质特征。在研究现实世界事物的过程中，忽略问题中与当前目标无关的部分，以便更充分地集中与当前目标有关的部分，这种方法就是抽象。现实世界中的事物是复杂的，所以不能试图一步到位地理解完整的事物，而仅选择它的一部分作为研究重点，这是一种行之有效的方法。抽象是体现这一方法的有效工具。抽象是去除一个事物中对当前目标不重要的细节，仅保留它的可描述的特征，从而形成一个抽象数据。抽象是面向对象程序设计的起点和基础。

封装的意义在于将该数据抽象类型及其操作紧密结合在一起，成为一个整体，从而使每个模块的界面尽量少地暴露它的内部工作。封装将外部规格说明从内部实现中分离出来，外部规格说明必须在开发早期提出，而内部实现则可以推迟。模块化促进了逻辑一致性、可理解性和对称性。

软件工程中特别需要面向对象方法的抽象能力。软件工程是软件开发的系统性方法，它强调在设计和编码之前透彻地进行概念化理解。问题的全面理解能改善软件可靠性、增强灵活性、缩短开发时间并降低开销。

### 1.1.1 面向对象的思维

面向对象领域中的焦点是语言编程问题。当前文献的重点是在实现上而不是在分析和

设计上。面向对象编程语言消除传统编程语言的非灵活性限制是相当有益的。但是，在某种意义上，这个重点又返回到软件工程过多地考虑实现机制上，而不是支持本质的思维过程。要真正获得高效的软件产品，首先是概念问题，而不是实现问题。在实现期间企图修改设计的缺陷，要比早期发现设计的缺陷进行修改付出更高的代价。实践经验表明：过早集中在实现问题上，限制了设计的选择，从而经常导致拙劣的软件产品。面向对象开发方法是根据大多数软件工程生命周期的应用领域来思考的，所以它能有效地促进软件开发工作。只有数据结构和功能的细节有效地确定后，这个应用的本质概念才被标识、组织和理解。面向对象开发是一个独立于编程语言的概念化过程。面向对象开发本质上是一种新的思维方式，而不是一种编程技术。它的最大好处是能够帮助规格说明者、开发者和用户清晰地表达抽象概念和对象之间的相互交流。它能为规格说明、分析、文档、接口以及程序设计服务，甚至可以作为一种程序设计工具。面向对象开发还可以有不同的目标，包括常规编程语言和数据库以及面向对象语言。

### 1.1.2 什么是面向对象

由于面向对象方法已经发展到计算机科学技术的许多领域，所以若想从一般意义上给出“面向对象方法”严格而清晰的定义，使之对这些领域都能适用，是一件很困难的事。20世纪80年代初期以前的定义是：面向对象是一种新兴的程序设计方法，或者是一种新的程序设计范型（Paradigm），其基本思想是使用对象、类、继承、封装、消息等基本概念来进行程序设计。自20世纪80年代以来，面向对象方法已深入计算机软件领域的几乎所有分支，远远超出了程序设计语言和编程技术的范畴。但是，即使在“计算机软件”范围内定义什么是面向对象也仍然是不完整的，因为面向对象方法还发展到计算机软件以外的一些领域，如计算机体系结构和人工智能等。

由于本书是讨论软件开发方面的问题的，而面向对象方法已在其中产生巨大影响并发展为较为完整的理论与技术体系，所以就在这个范围内讨论什么是面向对象。面向对象（Object-oriented 或 Object-orientation）不仅是一些具体的软件开发技术与策略，而且是一套关于如何看待软件系统与现实世界的关系，同时以什么观点来研究问题并进行求解，以及如何进行系统构造的软件方法学。

概括地说，面向对象方法的基本思想是从现实世界中客观存在的事物（即对象）出发来构造软件系统，并在系统构造中尽可能地运用人类的自然思维方式。开发一个软件是为了解决某些问题，这些问题所涉及的业务范围称作该软件的问题域。面向对象方法强调直接以问题域（客观世界）中的事物为中心来思考问题、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，保持问题域中的事物及其相互关系的本来面貌。另外，软件开发方法不应该是超脱人类日常的思维方式，并与人类在长期进化过程中形成的各种行之有效的思想方法迥然不同的思想理论体系。如果说，在某些历史阶段出现的软件开发方法没有从人类的思想宝库中汲取较多的营养，只是建立在自身独有的概念、符号、规则、策略的基础之上，那只能说明软件科学本身尚处于比较幼稚的时期。结构化方法采用了许多符合人类思维习惯的原则与策略。面向对象方法更加强调运用人类在日常逻辑思维中经常采用的思想方法与原则，例如抽象、分类、继承、聚合、封装等等。这使得软件开发者能更有效地思考问

题，并以其他人也能明白的方式把自己的观点表达出来。

具体地讲，面向对象方法有以下一些主要特点：

(1) 从问题域中客观存在的事物出发来构造软件系统，用对象作为对这些事物的抽象表示，并以此作为系统的基本构成单位。

(2) 事物的静态特征（即可能用一些数据来表达的特征）用对象的属性表示，事物的动态特征（即事物的行为）用对象的服务表示。

(3) 对象的属性与服务结合为一体，成为一个独立的实体，对外屏蔽其内部细节（称作封装）。

(4) 对事物进行分类。把具有相同属性和服务的对象归为一类，类是这些对象的抽象描述，每个对象是它的类的一个实例。

(5) 通过在不同程度上运用抽象的原则（较多或较少地忽略事物之间的差异），可以得到较一般的类和较特殊的类。特殊类继承一般类的属性与服务，面向对象方法支持对这种继承关系的描述与实现，从而简化系统的构造过程及其文档。

(6) 复杂的对象可以用简单的对象作为其构成部分（称作聚合）。

(7) 对象之间通过消息进行通信，以实现对象之间的动态联系。

(8) 通过关联表达对象之间的静态关系。

从以上几点可以看到，在用面向对象方法开发的系统中，以类的形式进行描述并通过对类的引用而创建的对象是系统的基本构成单位。这些对象对应着问题域中的各个事物，它们内部的属性与服务刻画了事物的静态特征和动态特征。对象类之间的继承关系、聚合关系、消息和关联如实地表达了问题域中事物之间实际存在的各种关系。因此，无论系统的构成成分，还是通过这些成分之间的关系而体现的系统结构，都可直接地映射问题域。

通过以上的介绍，读者可以对“面向对象”有一个大致的了解。现在摘录《对象技术语典》对这个术语的几种定义。对于“面向对象”的名词条目（Object-orientation），该词典中收集的定义有：

(1) 一种使用对象（它将属性与操作封装为一体）、消息传送、类、继承、多态和动态绑定来开发问题域模型之解的范型。

(2) 一种基于对象、类、实例和继承等概念的技术。

(3) 用对象作为建模的原子。

对于“面向对象”的形容词条目（Object-oriented），该词典收集的定义有：

(1) 用来描述一些基于下述概念的东西：封装、对象（对象的标识、属性和操作）、消息传送、类、继承、多态、动态绑定。

(2) 用来描述一种把软件组织成对象集合的软件开发策略，对象中既包括数据也包括操作。

然而在这本内容相当丰富的《对象技术词典》中，却没有对“面向对象方法”（Object-oriented Methods 或 Object-oriented Methodology）的术语给出一个定义。其原因或许是由于它所涉及的领域非常广泛，很难简单而确切地界定它的作用范围并指出它是一种关于什么的方法。它早已超过“程序设计方法”的范畴，说它是一种“软件开发方法”也不够全面，因为它涉及计算机软件以外的一些领域。不过从目前看，面向对象方法最主要的应用范围仍是软件开发，对软件生命周期的各个阶段（包括分析、设计、编程、测试



与维护)以及它所涉及各个领域(如人机界面、数据库、软件复用、形式化方法、CASE工具与环境等)都已形成或正在形成面向对象的理论与技术体系。所以,如果在这个范围内讨论问题,则可对“面向对象方法”作如下定义:面向对象方法是一种运用对象、类、继承、封装、聚合、消息传送、多态性等概念来构造系统的软件开发方法。

面向对象方法的基本概念与原则、发展历史与现状以及它对改进软件开发的重要意义等,将在以后各节中详细介绍。

### 1.1.3 面向对象方法的历史及现状

面向对象方法起源于面向对象的编程语言。在编程语言这个领域,它的诞生与发展经历了下述主要阶段。

#### 1. 雏形阶段

面向对象方法的某些概念,如“对象”、“对象属性”等可以追溯到20世纪50年代人工智能的早期研究。但是人们一般把20世纪60年代由挪威计算中心开发的Simula-67语言看作面向对象语言发展史上的第一个里程碑。正如徐家福教授等指出的,“这是由于Simula语言首先引入了类的概念和继承机制”。对于Simula-67是不是面向对象的语言,评论者意见不一,但共同的看法是确定它具有一些面向对象语言的重要特征,被称作面向对象语言的先驱。

#### 2. 完善阶段

但是直到20世纪80年代后期,Smalltalk的应用尚不够广泛,这种情况与它在学术上的重大影响相比显得很不相称,其原因有以下几点:

- (1)面向对象作为一种崭新的软件方法学被广泛接受需要一定的时间。
- (2)Smalltalk的商品化软件开发工作到1987年才开始进行。
- (3)追求纯OO的宗旨(例如严格的封装)使许多讲究实效的软件开发人员感到不便。

#### 3. 繁荣阶段

自20世纪80年代中期到90年代,是面向对象语言走向繁荣的阶段。其主要表现是大批比较实用的OOPL的涌现,例如C++、Objective-C、Object Pascal、CLOS(Common Lisp Object System)、Eiffel、Actor等。

从最近的动态看,C++的销售总量仍居首位,但Smalltalk的销售增长率却一度超过C++。这说明,许多单位为了在软件开发中更彻底地运用OO方法而宁愿承受一些效率上的损失。

面向对象编程语言的繁荣是面向对象方法走向实用的重要标志,也是面向对象方法在计算机学术界、产业界和教育界受到越来越高重视的推动力。20世纪90年代以来仍不断有OOPL问世,许多非OO语言增加了OO概念与机制而发展为OO语言。这表明OOPL的繁荣仍在继续,也表明面向对象是大势所趋。

自20世纪80年代末期到90年代以来,一个意义颇为深远的动向是面向对象的方法与技术向着软件生命周期的前期阶段发展,即:人们对面向对象方法的研究与运用不再局限于编程阶段,而是从系统分析和系统设计阶段就开始采用面向对象方法。这标志着面向对象方法已经发展成一种完整的方法和系统化的技术体系。

正如Northrop L.m在《软件工程百科全书》中所指出的:“尽管面向对象语言正取得