新一代信息系统

——面向对象信息系统的分析与设计

陈　睿　谢新洲　编著

航空工业出版社

新一代信息系统

——面向对象信息系统的分析与设计

陈　睿　谢新洲　编著

航空工业出版社

1993

（京）新登字161号

内 容 提 要

面向对象的软件技术目前在世界上受到了越来越广泛的重视，已成为软件研究与开发领域中的热点，在CAD、办公室信息系统、图形化用户接口、多介质应用、分布式系统等应用方面发挥了卓有成效的作用。本书详细讨论了面向对象信息系统的分析和设计方法，结合具体实际问题介绍了C++、Smalltalk-80这类面向对象程序设计语言以及面向对象的数据库系统和将数据库与面向对象程序设计有机结合起来的集成化工具OODBPL。本书注重理论与实践的结合，具有较强的实用性和系统性。

读者对象：各类计算机软件开发人员、工程设计人员、大学计算机专业及其相关专业的本科生、研究生和教学科研人员。

新一代信息系统

——面向对象信息系统的分析与设计

陈 睿 谢新洲 编著

航空工业出版社出版发行

（北京市安外小关东里14号）

—邮政编码：100029—

全国各地新华书店经售

航空工业出版社印刷厂印刷

1993年3月第1版 1993年3月第1次印刷

开本：787×1092毫米 1/32 印张：13.72

印数：1—3000 字数：320千字

ISBN 7-80046-477-6/TP·033

定价：9.50元

前　言

面向对象的方法，发端于Simula67语言。经过许多年的发展，得到了愈来愈广泛的重视，80年代中后期，开始成为软件技术研究的一个热门课题。

总的来说，面向对象领域中，实践活动比理论研究要活跃得多。迄今为止，面向对象领域仍然缺乏严格的形式化理论基础，因此，许多关系方法领域的学者对面向对象方法的前途抱怀疑的态度。尽管如此，由于开发出了一些成功的系统，例如Smalltalk-80、C++、Orion GemStone，O_2等，越来越多的研究人员被吸引到这个研究领域中，人们对这个课题充满了信心，甚至有人预言，面向对象方法在90年代将占据软件技术的主流地位。事实上，面向对象的方法已经在一些新的应用领域，例如CAD、OIS（办公室信息系统）、图形化用户接口、多介质应用、分布式系统等非传统应用中发挥了卓有成效的作用。

这本书主要讨论面向对象信息系统的开发方法，包括面向对象分析、设计和实现的方法。面向对象的信息系统的基本构造单位是对象，以对象作为基本构件的系统结构会带来一系列的好处，例如，减低了系统的复杂度，使系统结构清晰、易于理解，提高了系统的可靠性和可维护性。对于一个用惯了过程化程序设计语言，也习惯于以功能分解的方法进行系统分析和设计的人来说，要理解面向对象方法和在实践中运用这种方法并不是一件容易的事，需要一个转变过程。

随着C++等面向对象程序设计语言的流行，越来越多的人将面临这个问题。

全书的结构是这样的：第一章讨论面向对象的基本概念；第二、三、四章讨论面向对象信息系统的分析和设计方法；第五章讨论与系统实现有关的基本理论问题，主要是类型理论；第六、七、八章讨论面向对象程序设计方法和工具；第九、十章讨论面向对象的数据库系统；第十一章讨论OODBPL，这是一种将数据库和面向对象程序设计有机结合起来的集成化工具，能够有效地用于面向对象的系统实现。

本书由陈睿、谢新洲合著，具体分工如下：第一、二、三、四、五、七章为谢新洲撰写，第六、八、九、十、十一、十二章为陈睿撰写。衷心感谢周士林副总编辑的指导与帮助。由于作者水平有限，缺点错误在所难免，敬请读者指正。

目 录

第一章 面向对象的基本概念 …… (1)
1.1 面向对象的基本概念 …… (2)
1.1.1 对象 …… (2)
1.1.2 类和子类、继承性、重置 …… (4)
1.1.3 类间关系 …… (6)
1.1.4 对象的封装性 …… (7)
1.1.5 多态性 …… (7)
1.1.6 传统程序设计与面向对象程序设计的比较 …… (8)
1.2 面向对象方法的优越性 …… (8)
第二章 信息系统的认知模型 …… (10)
2.1 三个世界 …… (10)
2.1.1 观念世界 …… (11)
2.1.2 数据世界 …… (13)
2.2 概念模型 …… (14)
2.2.1 概念间的关系 …… (14)
2.2.2 信息系统的概念模型 …… (17)
2.3 数据模型 …… (21)
2.4 问题域 …… (22)
2.5 小结 …… (24)
第三章 面向对象分析：建立概念模型的方法 …… (26)
3.1 为什么要建立概念模型 …… (26)
3.2 用面向对象的方法建立概念模型的基本思想 …… (29)
3.2.1 概念、对象和类 …… (31)
3.2.2 类描述语言CDL …… (36)

3.2.3 类图……(42)
3.3 一个实例：图书馆流通管理系统的概念模型……(47)
3.4 面向对象分析：建立概念模型的方法……(52)
3.4.1 识别对象和类……(52)
3.4.2 定义类和组织类间关系……(66)
第四章 面向对象的信息系统开发方法……(74)
4.1 面向对象设计的优越性……(76)
4.2 面向对象设计方法……(78)
第五章 现代程序设计语言的抽象机制与面向对象程序设计的关系……(89)
5.1 抽象对现代程序设计的影响……(89)
5.2 数据类型的概念和数据类型间的关系……(93)
5.2.1 数据类型和类型系统……(94)
5.2.2 数据类型间的关系……(100)
5.3 数据抽象和抽象数据类型……(102)
5.3.1 抽象数据类型的概念……(103)
5.3.2 抽象数据类型的规范描述……(105)
5.3.3 程序设计语言中支持抽象数据类型的机制……(111)
5.4 多态性……(126)
5.4.1 过载多态性(overloading polymorphism)和强制多态性 (coercing polymorphism)……(128)
5.4.2 参数化多态性(parametric polymorphism)……(131)
5.4.3 包含多态性(inclusion polymorphism)和继承性(136)
5.4.4 动态联编 (dynamic binding)……(141)
5.5 面向对象程序设计语言的抽象机制……(145)
第六章 面向对象程序设计语言概论……(149)
6.1 面向对象程序设计语言的发展历史和背景……(149)
6.2 面向对象程序设计语言的基本特性……(152)
6.2.1 对象和类……(153)

6.2.2 方法和消息……(155)
6.2.3 继承性……(156)
6.2.4 类库……(160)
6.3 面向对象程序设计语言的分类……(162)
6.4 几种面向对象程序设计语言评述……(164)
6.4.1 Objective-C 语言……(164)
6.4.2 面向对象的Pascal语言……(167)
6.4.3 Eiffel 语言……(169)
6.4.4 CLOS(Common Lisp Object System)语言……(172)
第七章 Smalltalk-80……(175)
7.1 Smalltalk-80的语言特性……(175)
7.1.1 对象、类、消息、方法……(176)
7.1.2 Smalltalk-80的表达式和表达式语法……(178)
7.1.3 Smalltalk处理消息传递的语义行为……(195)
7.1.4 类层次结构和继承性……(198)
7.1.5 多态性和动态联编……(203)
7.1.6 Smalltalk-80的类型检查……(206)
7.2 Smalltalk-80的程序设计环境和程序设计过程……(210)
7.2.1 类的定义……(211)
7.2.2 创建实例和发送消息……(224)
7.2.3 Smalltalk-80的浏览窗口 (browser)……(229)
7.3 Smalltalk-80系统评价……(237)
第八章 C++……(239)
8.1 C++的基本语言特性……(240)
8.1.1 类……(241)
8.1.2 C++对C的改进……(277)
8.1.3 C++语言的多态性……(282)
8.1.4 C++的类型检查……(304)
8.2 C++ 程序设计过程……(305)

8.2.1 定义类……(305)
8.2.2 利用已定义类来编写程序……(311)
8.2.3 C++的程序设计环境……(312)
8.3 C++系统评价……(313)
第九章 面向对象数据库概论……(316)
9.1 什么是面向对象的数据库（OODB）……(318)
9.2 面向对象的数据模型……(321)
9.2.1 类……(323)
9.2.2 对象和对象标识（object identity）……(323)
9.2.3 方法和消息传递……(326)
9.2.4 类层次结构和类组合结构……(327)
9.3 OODB系统的数据定义语言（DDL）……(328)
9.4 OODB系统的数据操纵语言(DML)……(333)
9.4.1 用PDML进行实例化……(333)
9.4.2 对象查询……(337)
9.4.3 对象修改……(344)
9.4.4 对象删除……(344)
9.4.5 循环……(345)
9.5 OODB系统在面向对象信息系统实现中的作用……(346)
9.6 小结……(349)
第十章 几个面向对象数据库系统评述……(350)
10.1 Orion……(350)
10.1.1 Orion的数据模型……(350)
10.1.2 Orion的DDL……(353)
10.1.3 Orion的DML……(363)
10.1.4 Orion系统评价……(375)
10.2 POSTGRES……(377)
10.2.1 Postgres的数据模型……(378)
10.2.2 POSTQUEL……(380)

10.2.3 Postgres 评价……(385)
10.3 O_2 系统……(386)
10.3.1 O_2 的数据模型……(387)
10.3.2 O_2语言 ……(393)
10.3.3 O_2系统评价 ……(400)
10.4 小结……(401)
第十一章 面向对象的数据库程序设计语言……(402)
11.1 数据库程序设计语言的基本特性 ……(403)
11.1.1 Pascal/R的类型系统……(405)
11.1.2 持久性 ……(408)
11.1.3 DBPL的目标和要求……(410)
11.2 面向对象的DBPL (OODBPL) ……(411)
11.2.1 O_2 ……(412)
11.2.2 O^{++} ……(414)
11.3 小结 ……(419)
第十二章 结束语 ……(420)
参考文献 ……(422)

符号说明

＊代用＊

∷代用∷

∷＝代用 ∷＝

第一章　面向对象的基本概念

面向对象并不是一个新的概念。现在普遍认为，面向对象的概念最早发源于挪威的 K.Nygaard 等人开发的模拟离散事件的程序设计语言Simula67。然而，事实上，对象的概念在计算机科学的多个分支领域中几乎是在60年代后半期到70年代初这段时间中相互独立地同时提出的，尽管形式不同，但实质是一样的。表 1.1 列举了几种不同形式的对象概念。

表 1.1　不同形式的对象概念

K.Nygaard等提出的Simula语言	对象
A.Kay提出的Smalltalk	对象
C.Hewitt提出的并行计算模型	演员
M.Minsky提出的知识表示的框架方法	框架
J.Dennis等人提出的操作系统中资源保护方法	资源可保护性

随着时间的推进，面向对象的概念在越来越多的领域得

到了应用和发展。软件工程中的数据抽象和信息隐蔽思想、概念模型化研究领域中的语义数据模型、知识表示、CAD和多介质应用领域中的复杂对象模型等等研究成果都包含了面向对象的基本思想，或者说，殊路同归，多方面的研究都归结到面向对象方法上来。可见，面向对象方法是一个应用广泛、很有发展前途的研究课题。尤其是进入80年代以来，面向对象方法得到了日益广泛的重视，甚至有人预言，面向对象技术将成为90年代计算技术的主流。

在这本书里，我们要讨论一种以对象为基本构造单位组成的信息系统——面向对象的信息系统，讨论面向对象信息系统设计、开发和实现的基本理论、方法和技术。在这一章中，我们首先讨论面向对象的基本机制、基本概念和相关的技术问题。面向对象方法是一个发展中的研究领域，术语混乱，充满争论，我们将依据被普遍接受的观点来展开讨论。

1.1 面向对象的基本概念

首先，要搞清楚什么是对象，然后再讨论对象的基本特性。

1.1.1 对象

简言之，对象是由一组数据和施加于这些数据上的一组操作构成。这是对计算机中的对象的解释（即数据世界中的对象）。要理解对象的实质，最方便的办法就是运用拟人化的思维方式。人是一个自主的物体，由许多器官构成。一个具体的人，总有一个状态，由身体器官的状态构成；一个具体的

人，对自己的行为负责，他的行为由他的大脑支配，其他人不能根据自己的意志支配他的行为。对象就象人一样，是一个自主的实体，它具有以下基本特性：

1. 有一个状态；
2. 有一个名字以区别于其它对象；
3. 有一组操作，每一个操作决定对象的一种行为；
4. 对象的操作可以分为两类：一类是自身所承受的操作，一类是施加于其它对象的操作；
5. 对象的状态只能被自身的行为改变；
6. 对象之间的消息互相通信，当一个对象要求另一个对象作某个动作时，就向它发送一个消息；
7. 一个对象的状态可以由多个其它对象的状态构成。

由此可见，对象对应于自然存在的实体，刻划了实体的结构特征和行为特征。

如果一个对象的状态由其它对象表示，则称构成它的状态的那些对象是它的属性（在某些术语体系中称为实例变量）。对象的结构特征是由它的属性表示的。对象的每个操作称作方法，一个对象的方法构成了其它对象可见的接口。一个对象向另一个对象发送消息，以激活它的某个方法，对象的每个方法都对应且仅对应一条消息。

举一个例子，一个名为John的人，他的状态包括：

1. 性别：男；
2. 身高：180cm；
3. 体重：68kg；

他的方法包括：

4. 回答身高：how—tall；
5. 回答体重：how—heavy；

其它对象向John发送一条消息how—tall，John将执行他的方法how—tall。如果其它对象向John发送一条消息what's—your—name，John不会采取任何行动，因为他不理解这条消息。图1.1表现了对象的基本概念。

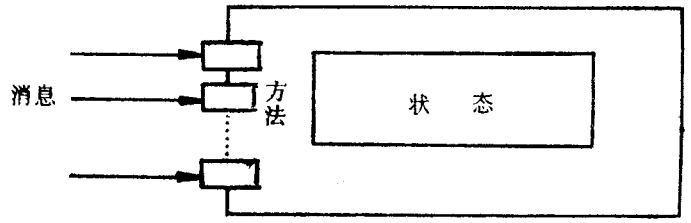

图 1.1　对象的基本机制

1.1.2　类和子类、继承性、重置

一组具有相同结构特征和行为特征的对象构成了一个类，所有这些对象都是类的实例。一个类的不同实例，必定具有：

1.　相同的方法集合；

2.　相同的属性定义，但可以有不同的属性值；

3.　不同的对象名。

因此，类是表现对象结构共性和行为共性的一种机构。类中至少包含了以下两方面的描述：

1.　本类所有实例的属性定义（或实例变量），指明属性值是哪一个类的实例，即属性域；

2.　本类所有实例的方法定义（或实例方法）。

另一方面，一个类也可以表现其它类对象的共性。这就

是类与子类的关系。一个类A是另一类B的子类，当且仅当

1. 类B的所有属性定义包含在类A中；

2. 类B的所有方法定义包含在类A中；

3. 类B中可以包含类A中没有的属性或方法。

若类A是类B的子类，则我们说，类A继承了类B，A类实例将具有B类实例的结构特征和行为特征。若A是多个类的子类，则称为多重继承。

子类A继承自超类B的属性和方法可以在子类A中重新定义，这种情形称为重置（overriding）。如果A中重置了B中的某个属性attribute，或者某个方法method，则：

1. A的属性attribute的属性域与B的attribute属性域不同；

2. A的方法method具有与B的方法method不同的语义。

举一个例子，定义三个类，person、teacher和student：

1.	person	属性	属性域
		name	string
		height	integer
		heavy	integer
		方法	
		how-tall	
		how-heavy	
2.	student	继承 person	
		属性	属性域
		heavy	real
		grade	integer

方法

how—heavy

which—grade

3. teacher 继承 person

属性 属性域

department string

方法

which—department

其中，student和teacher都是person的子类，继承了person类的所有属性定义和方法定义，但在student类中，属性heavy被重置，其属性域由整数变成实数，方法how-heavy也被重置，它回答一个实数值，而不是一个整数值。

1.1.3 类间关系

综上所述，类和类之间存在两种关系，即继承关系和组合关系。设有两个类A和B：

1. A和B之间存在继承关系，当且仅当A是B的子类；

2. A和B之间存在组合关系，当且仅当A是B的某个属性的属性域。

由于存在这两种关系，可以把许多类组织成类层次结构和类组合结构两种结构。类层次结构表现了类的继承关系，类组合结构表现了类的组合结构。我们用图1.2来表现person、student、teacher、integer、string、real类之间的关系。用实线表示类层次结构，用虚线表现类组合结构：

类层次结构和类组合结构是面向对象信息系统分析、设计和实现的基础。

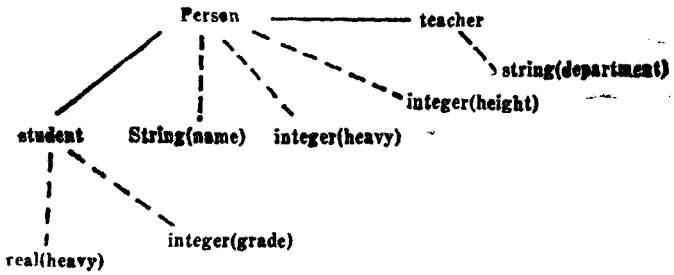

图 1.2 类层次结构和类组合结构

1.1.4 对象的封装性

封装性是一个源自抽象数据类型的概念（见第五章）。对象是具有封装性的，这有两方面的含义：

1. 对象将状态和操作集中起来；
2. 对象的状态只能由自身的操作存取。

例如，对任意一个student类的实例，除非向它发送消息激活 which-grade 方法，否则我们不可能搞清楚这个学生是哪一个年级的。

1.1.5 多态性

多态性是一个与类相关的概念。同一类的所有对象，在收到同一条消息时将采取同样的动作（也有例外的情况，比如O_2对象的例外方法，见第十章）。不同类的对象，在接收到同一条消息时可能采取不同的动作。不同对象对同一消息采取不同动作，这种情形称为多态性。例如，一个student实例和一个person实例在接收到how—heavy消息时采取不