面向对象的设计

[美] PETER COAD EDWARD YOURDON 著

邵维忠 廖钢城 苏渭珍 译 杨芙清 校

面向对象的设计

[美] PETER COAD EDWARD YOURDON 著
邵维忠 廖钢城 苏渭珍 译
杨芙清 校

北京大学出版社
北 京

图书在版编目(CIP)数据

面向对象的设计/(美)(Coad,P.),(Yourdon,E.)著;邵维忠等译. —北京:北京大学出版社,1994.11

书名原文:Object-Oriented Design

ISBN 7-301-02695-1

Ⅰ.面… Ⅱ.①C… ②Y… ③邵… Ⅲ.面向对象语言程序设计

Ⅳ.TP311

书　　名:面向对象的设计

著作责任者:[美]PETER COAD EDWARD YOURDON 著
邵维忠等译

责任编辑:沈承凤　胡美香

标准书号:ISBN 7-301-02695-1/TP.245

出 版 者:北京大学出版社

地　　址:北京市海淀区中关村北京大学校内　100871

网　　址:http://cbs.pku.edu.cn/cbs.htm

电　　话:出版部 62752015　发行部 62754140　编辑室 62752037

电子信箱:zpup@pup.pku.edu.cn

排 版 者:兴盛达激光照排中心

印 刷 者:北京大学印刷厂

发 行 者:北京大学出版社

经 销 者:新华书店

850×1168 毫米　32 开本　6.125 印张　159 千字

1994 年 12 月第一版　1999 年 1 月第三次印刷

定　　价:12.00 元

内 容 简 介

本书全面而深入地介绍面向对象的软件设计技术。即：在系统分析员完成了面向对象的分析(OOA)之后，设计人员如何针对系统的具体实现进行面向对象的设计(OOD)。首先介绍了贯穿于 OOA 和 OOD 的基本概念、原则、定义及表示法，然后详细地阐述了 OOD 的四项主要工作——问题域部分的设计、人机交互部分的设计、任务管理部分的设计和数据管理部分的设计。书中还分析比较了多种面向对象的编程语言和非 OO 语言对 OOA 和 OOD 模型语义的表达能力，从而为编程语言的选择提供了依据。最后，本书讨论了 OOD 评价标准及 OOD 对 CASE 的要求等问题。

本书内容充实，论述精辟，图文并茂，实例丰富。它在内容上是独立的，同时它又是我社出版的《面向对象的分析》的姐妹篇；二者构成了一个更为完整的体系——面向对象的分析与设计。

本书可供从事计算机软件开发的工程技术人员，计算机专业的教师、学生和研究生阅读。

Peter Coad and Edward Yourdon
Object-Oriented Design
Yourdon Press 1991

译 者 序

自 80 年代以来，面向对象的方法与技术已受到计算机领域的专家、学者、研究人员和工程技术人员越来越广泛的重视。80 年代中期相继出现了一系列描述能力较强、执行效率较高的面向对象的编程语言，标志着面向对象的方法与技术开始走向实用。自 80 年代末期到 90 年代，一个意义更为深远的动向是，面向对象的方法与技术向着软件生命期的前期阶段发展。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论和系统化的技术体系。

在计算机软件领域，很多新的方法与技术都是从编程阶段首先出现，进而发展到软件生命期的前期阶段。例如大家熟悉的结构化方法，开始提出时叫做“结构化编程”，它引出了一系列的结构化编程语言，最后发展到“结构化分析”和“结构化设计”。又如形式化方法，最先是成功地运用于编程语言的形式化描述，后来着眼于研究对需求分析和设计的形式化描述。软件重用技术也是从程序（可执行程序或源化码）的重用发展到分析结果和设计结果的重用。上述现象的出现并不奇怪。说到底，人们对软件本身的认识，开始也只是着眼于编程，后来才形成了从分析、设计，到编程、测试及维护一整套的软件工程体系。

与上述种种方法与技术相比，面向对象的方法与技术发展到软件生命期的前期阶段有着更为深刻的意义。因为面向对象方法的本质，就是强调从客观世界中固有的事物出发来构造系统；用人类在现实生活中常用的思维方式来认识、理解和描述客观事物；强调最终建立的系统能够映射问题域，即：系统中的对象以及对象之间的关系能够如实地反映问题域中固有事物及其关系。这恰恰是从分析和设计

阶段入手才能根本解决的问题。试想，在一个软件开发单位，如果只拥有一批会使用某种面向对象的编程语言（例如C++）的程序员，而分析与设计仍然采用传统的方法，则程序员的编程技巧再熟练，也难以透过传统方法产生的分析及设计文档（例如数据流图和模块结构图）而看透现实世界中事物的本来面貌；纵然在程序中定义了一些类及对象，也很难真正地映射现实世界中的事物。这就好比让画家通过一个人物的传记（而不是通过面对面的观察）去为这个人画肖像，绘画技法再熟练也难以画得十分逼真。从这个意义上讲，软件生命期的前期阶段，正是面向对象的方法能从根本上发挥其优势的领域。面向对象的分析（OOA）和面向对象的设计（OOD）的出现是面向对象方法发展的必然结果，也是它走向完善、走向实用的重要标志。

这并不是说，在OOA和OOD出现之前人们对面向对象方法的运用效果可以一笔抹杀。实际上，对一些小型系统而言，开发人员常常是根据面向对象方法的基本原则，首先在头脑中（或纸面上）进行构思，决定系统中要设立哪些类及对象，然后才用面向对象的语言去编程。应该说，这种开发方式确实是面向对象的，也能够产生一些相当不错的小型系统。但是对大系统而言，事情就不那么简单了。一是问题域、系统责任以及与实现有关的因素要复杂得多；二是这样的系统不可能由少数几个人从头到尾地完成，而要有针对不同开发阶段的分工。它的分析和设计不是在头脑中想一想或者在草纸上构画一下就能解决的。它需要一种完整的面向对象的系统模型；需要一整套针对分析和设计阶段的方法、策略和技巧；还需要一套便于分析人员、设计人员、编程人员、管理人员以及用户彼此勾通的一致的基本表示。这些，正是面向对象的分析和面向对象的设计所要解决的问题。

从90年代末期开始，国际上有一批论述面向对象的分析与设计（或面向对象的建模与设计）的专著相继问世。这些著作的共同点是把面向对象的方法在分析与设计阶段的运用提升到理论和工程的高度（而不仅仅是一些可供参考的指导思想），各自提出了一套较为完

整的系统模型、表示法和实施策略。同时，在模型、表示法和策略等方面，彼此又各有差异。相比之下，我们认为 Peter Coad 和 Edward Yourdon 合著的《Object-Oriented Analysis》和《Object-Oriented Design》是同类著作中较为优秀的两部。其主要的优点是：理论体系比较完善，而且具有坚实的实践基础；系统模型结构合理，而且与 OOA 和 OOD 活动紧密对应；表示法和实施策略简明清晰，具有很强的可操作性。此外，一个更为突出的优点是，对 OOA 和 OOD 的责任范围以及彼此之间的关系处理得十分恰当，可以简要地概括如下：

OOA 是针对问题域和系统责任的，不考虑与系统实现有关的因素。OOA 模型由五个层次构成，即类及对象层、结构层、主题层、属性层和服务层，对应着 OOA 工作的五个活动。OOD 则是针对与实现有关的因素继续运用 OOA 的五个活动，它包括问题域部分、人机交互部分、任务管理部分和数据管理部分等四个部分的设计。OOD 模型从横向看是上述四个部分，从纵向看每个部分仍然是五个层次。从 OOA 到 OOD 不存在转换问题，而是同一种表示法在不同范围的运用。OOD 也不是对 OOA 模型进行细化。它的问题域部分是把 OOA 模型拿来，针对实现的要求进行必要的增补和调整而得到的（例如为了能用单继承语言编程，把 OOA 模型中的多继承调整为单继承）。OOD 的其它三个部分则是 OOA 阶段未曾考虑的，全部在 OOD 阶段建立。这样划分 OOA 和 OOD 的责任范围有利于避免在分析和设计阶段重复地描述同一事物，从而明显地减少了工作强度和文档数量。在次序上，OOA 和 OOD 既可以顺序地进行，也可以交叉地进行。因此，无论是瀑布式、螺旋式还是渐进式的开发模型它都能适应。

上述优点使 Caad 和 Yourdon 的这两部著作无论对教学、科研，还是对工程实践都是很有价值的参考书。我们感到这样的好书应该原原本本地翻译出来奉献给我国读者。两年前我们翻译了其中的第一本（《面向对象的分析》，北京大学出版社，1992 年 2 月），受到许多读者的欢迎。其中不少读者在热心地询问：《面向对象的设计》何时出版？这对我们既是鼓励又是鞭策。这一工作早就开始做了，但是为了

解决版权及经费问题耽搁了一些时间。今天，当《面向对象的设计》终于与读者见面之际，我们谨向热心的读者们表示衷心感谢，并请求他们的谅解。稍可告慰的是，在这段时间里我们用这本书的译稿（以及《面向对象的分析》）作为主要教材开设了北大计算机系的研究生课程，在国家"八五"科技攻关项目中进行了大量 OOA 和 OOD 的研究与实践，同时和国内其它一些在大、中型系统的开发中率先采用 OOA 及 OOD 的单位开展了交流。这些教学、科研及工程工作，使我们大大加深了对本书的理解，纠正了早期译稿中的若干不当之处。当然，错误仍可能存在，恳请读者予以指正。

《面向对象的设计》是《面向对象的分析》的姐妹篇，同时它又是一部独立的著作。在本书的引言部分和一、二两章，对于从 OOA 到 OOD 一致采用的基本原则、概念、术语和表示法作了全面的介绍或回顾，并在附录 B 中概括而完整地给出了 OOA 实施策略，从而使本书可以独立地阅读。此外，应该说明的是，《面向对象的分析》是按《Obiect-Oriented Analysis》的第一版翻译的，原书后来出了第二版，对第一版作了一些改进。《Object-Oriented Design》是在此之后出版的，较全面地体现了这些改进。因此阅读《面向对象的设计》可以了解到作者对 OOA 作了哪些改进。

本书的第二章论述了 OOA 和 OOD 的关系并给出了 OOD 的系统模型。第三章至第六章详细地阐述了 OOD 的四个部分（问题域部分，人机交互部分，任务管理部分和数据管理部分）的设计，包括每个部分的基本概念、意义、实施策略和具体例子。第七章讨论了根据 OOA 和 OOD 的要求应该怎样选择编程语言，分析比较了多种面向对象的编程语言和几种非 OO 语言对 OOA 及 OOD 模型语义的表达能力，为编程语言的选择提供了依据。第八章介绍了 OOD 评价标准。其意义不仅是 OOD 完成之后用什么尺度去评价它的优势，而且也是在开展 OOD 时的指导原则和努力目标。第九章讨论 OOD 需要什么样的 CASE 工具。最后一章对开发单位是否应开始采用面向对象的开发方法给出了一些参考意见。

最后，值得一提的是，虽然本书的中心论题是面向对象的设计，但是在讨论到原型开发、图形用户界面、软件重用等相关问题时，作者也有许多很好的论述，有些见解是非常精辟的。希望读者在这些方面也能有所收益。

本书由邵维忠、廖钢城、苏渭珍合译。杨芙清院士审核、校对了全稿。沈璞、徐松青同志为译稿的计算机录入和排版付出了辛勤的劳动，在此谨表感谢。

译　　者

1994 年 7 月于北京大学

引　言

本书是我们所著的《面向对象的分析》一书的姐妹篇，两本书的素材都基于对象与属性、整体与部分、类与成员等概念。

正如《大英百科全书》指出的：

人类在认识现实世界的过程中普遍运用着三个构造法则：

(1) 区分对象及其属性。例如，区分一棵树和树的大小或它与其它对象的空间关系。

(2) 区分整体对象及其组成部分。例如，区分一棵树和树枝。

(3) 形成并区分不同对象的类。例如，形成所有树的类和所有石头的类。并区分它们。

——《大英百科全书》"分类学理论"，1986

OOD 的表示符号及方法就基于这三个常用的构造法则。

本书针对从事实践的软件工程师，即那些每天与现实世界中的系统开发项目打交道的人。我们假设读者关心开发周期的中期阶段即设计活动。一个设计者也可能卷入与用户交谈以确定系统需求的前期活动；也可能卷入编码和系统测试这些后期问题。但是，在本书我们基于上是对负责为系统开发总体软件结构的人们说话的。

管理员、测试员、标准负责人、程序员也可以阅读本书，并从其改善系统设计的综合方法中获益。然而我们并不要求最终用户阅读本书；这部分人读了我们的另一本书《面向对象的分析》可望有所裨益。

0.1　历史

为什么说"对象范型"终于成熟了？为什么现在才成熟？

面向对象的程序设计是在 60 年代末期由 SIMULA 集团开始讨论的。70 年代初，它是 Xerox PARC 开发的 Smalltalk 的重要部分。

与此同时，外界正在沿着诸如 COBOL、FORTRAN 之类的语言蹒跚而行，并用功能分解法来解决设计与实现的问题。极少讨论到面向对象的设计，更没有讨论面向对象的分析。

过去的 10 年发生了四个变化并成了我们在 90 年代继续前进的重要因素：

- 软件领域中面向对象方法的基本概念经历了 20 年的成长道路，人们的注意力逐渐从编码问题转移到设计与分析问题。
- 构造系统的基本技术变得更加强有力。设计思想受预想的如何编码的思想影响；而编码思想受人们可用的程序设计语言的强烈影响。当选用的语言是汇编语言和 FORTRAN 时考虑结构化程序设计是很困难的；使用 Passcal、PL/1 和 ALGOL 时事情就变得容易了。类似地，当选用 COBOL 或 Plain-VanillaC 语言时想以面向对象的方式来编码也是很困难的；用 C++和 Smalltalk 则比较容易。
- 如今系统构造和 10 年、20 年之前大不相同。如今的系统更大、更复杂也更易变。面向对象的分析与设计方法看来将导致比较稳定的系统。另外，现今的联机、交互系统比 70 年代面向正文的批处理系统在用户界面上花费多得多的注意力。对于此类系统的面向对象的方法——从分析到设计到编码——是处理这种面向用户的系统的更自然的途径。
- 现在的系统构造比 70 年代和 80 年代更加“面向领域”。对功能复杂性的关心比以前少；数据建模的优先程度较为适当；问题域模型理解及系统职能处于较高的优先地位。

0.2 方法和工具

在本书我们将继续讨论 OOA 一书中提出的方向，介绍一个方法和一个低成本的绘图和检查工具以进行尝试。本书介绍的工具是 OODTool™。为了向您提供一个微小成本的 OODTool™小项目版本，Peter 在本书中附了一张商业回复卡（如果此卡丢失可来函索

取)。

本书中全部例子都是用 OOATool™和 OODTool™的早期版本开发的。

0.3 未来的工作

OOD 和它的姐妹篇 OOA 所介绍的方法还比较年轻,尚待在实践中评价。我们建议你用此书作为应用 OOD 的起点,剪裁和发展其方法以适合你的组织和项目的特定要求。

目　录