

What every programmer should know about OBJECT-ORIENTED DESIGN

面向对象设计 ——程序员必读

[澳] Meilir Page-Jones 著

申玉强 曹 济 程筱芳 等译

北京 SPIN 审校



電子工業出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

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

软件工程丛书

面向对象设计

——程序员必读

What every programmer should know about
OBJECT-ORIENTED DESIGN

[澳] Meilir Page-Jones 著

申玉强 曹济 程筱芳 等译

北京 SPIN 审校

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书是一本介绍关于面向对象软件设计的基本观点、符号、术语、准则和原理的书。运用大量的实例、图表和练习，本书详细地阐述了面向对象设计的各个方面，融入了作者在面向对象开发中的丰富的实践经验，使得面向对象设计中的一些复杂和容易混淆的概念变得明了和简单易懂。

本书适合于作为程序员、设计师、系统工程师或使用面向对象技术的项目管理人员的案头必备之书，也适合于初学者阅读。本书也可作为软件工程相关专业的教科书。

Original edition copyright 1995 by Meilir page-Jones. All rights reserved. Translation published by arrangement with Dorset House Publishing Co., Inc. (www.dorsethouse.com) through the Chinese Connection Agency, a division of The Yao Enterprise, LLC.

本书简体中文专有翻译出版权由 Dorset House Publishing 授予电子工业出版社。未经许可，不得以任何方式复制或抄袭本书的任何部分。

版权贸易合同登记号 图字：01-2002-5199

图书在版编目（CIP）数据

面向对象的设计程序员必读 / （澳）琼斯（Jones,M.）著；申玉强，曹济，程筱芳译.

北京：电子工业出版社，2004.3

（软件工程丛书）

书名原文：What Every Programmer Should Know About Object-Oriented Design

ISBN 7-5053-9608-0

I.面... II.①琼...②申...③曹...④程... 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（2004）第 002781 号

责任编辑：孙学瑛

印 刷：北京增富印刷有限公司

出版发行：电子工业出版社

北京市海淀区万寿路 173 信箱 邮编：100036

经 销：各地新华书店

开 本：787×980 1/16 印张：19.75 字数：300 千字

印 次：2004 年 3 月第 1 次印刷

印 数：5 000 册 定价：38.00 元

凡购买电子工业出版社的图书，如有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系。联系电话：（010）68279077。质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

译者序

面向对象的概念现已深入人心，但是面向对象的本质是什么呢？如何使用面向对象方法进行设计？什么样的设计才是完美的设计？设计完美与否的标准是什么？等等，这些问题相信很多程序员都不能完整地回答出来。值得庆幸的是，本书详尽地回答了这些问题。

本书本着循序渐进的原则，在第 1 部分中首先深入地分析了面向对象的概念，详细分析了面向对象的本质特征，并介绍了面向对象的来源，从而分析面向对象带来的好处。第 2 部分介绍了面向对象的具体设计，包括类继承图、对象聚合图、对象通信图、对象交互/调配图和状态转换图，等等。第 3 部分比较了面向对象软件的封装结构和传统软件的封装结构，利用域和相关性的概念对类的设计做了评估，并在此基础上提出了面向对象设计的一般原则。

本书对采用面向对象方法的设计人员有很好的参考价值，对于那些急于提升设计水平的中级程序员尤其意义重大。

本书由北京 SPIN 组织翻译。作为软件过程改进领域的专业组织，北京 SPIN 在举办 IT 行业公开沙龙，倡导正确舆论，沉淀行业知识，建立行业过程改进数据库，发起国际会议等方面开展了很多工作。

参加本书翻译工作的人员有：申玉强（北京工业大学）、程筱芳（北京邮电大学）、于洁（北方烽火）。此外，曹济（北京 SPIN）负责本书的统稿审核和翻译工作的整体管理。

曹济

北京 SPIN (www.pkspin.org) 简介

北京软件过程改进沙龙（SPIN）是一个促进理论和实践结合的开放交流平台。

SPIN 现有会员 2 100 多名，分为软件过程改进、项目管理、软件工程和项目监理等若干特别兴趣组开展活动。

组织宗旨是：非商业性、开放性、实践性

组织目标是：构造学习型组织，推动企业软件管理能力提升！

致 谢

如果要向所有我应该感谢的人一一致谢的话，那么答谢名单将会长达 94 页。因此，我只能向所有在这些年曾经激励、指点和纠正我的人们一并说一声“谢谢”。但有一些人我想要特别感谢。

- 感谢 Bob Binder, Tom Bragg, Tom Bruce, Mike Frankel, David McClintock, Jim Odell, David Ruble, Ben Sano 和 Becky Winant。他们曾经审核过我的十分凌乱的早期手稿。
- 感谢 Steve Weiss，早在面向对象技术流行之前，他就同 Walter Beck 一起劝说我转向面向对象的领域。如果不是他们，我恐怕到现在也不会写这本书。
- 感谢 Larry Constantine，他是第一个提出了软件设计原则的人。
- 感谢 Ed Yourdon 和 Tim Lister 对我的信任。
- 感谢 Bertrand Meyer 给了我持久不衰的灵感。
- 感谢 Paul Becker 在这么多年给我的鼓励。
- 感谢 Roland Racko，我从他那儿分享了许多观点。
- 感谢多达 62 位面向对象书籍的作者，他们的著作我大部分都拜读过。
- 感谢 Steve McConnell，他是《完整代码》一书的作者。他曾说他喜欢我书中的技术部分，但并不认为书中的笑话有多幽默。（Steve：你将会很高兴地看到我已经遵从了你的建议，在这本书中我已把笑话、轶事减到最少的地步。）
- 感谢多塞特书社的工作人员们，他们堪称运用最新技术勤奋工作的散文大师。他们花费大量精力剔除错误的用法，如双重否定，分开的不定词，模棱两可的词汇，令人讨厌的冗长句子（它们从动名词开始，并且有着毫无理由的自我引用），毫无必要的说明，不一致的时态，以及早期手稿中作为专业书所不需要的过多的形容词。
- 感谢我的其他著作的热心读者们，感谢他们给我写信。我十分感谢你们的

评论。

- 感谢我的网友们，他们的 E-mail 所包含的博学和机智的信息在这些年一直启发和激励着我。
- 感谢全世界的客户和学生们与我分享他们的经验，并且尽最大努力真诚地帮助我。
- 感谢我的家人，他们一直保持着极大的耐心与一个整天写书的隐居者一起生活。

最后，让我向 Sid Dijkstra 教授表示崇高的敬意，因为他对软件工程的贡献是无法衡量的。

序

一本有关面向对象设计的新书的问世，即将面临的无非就是久经沙场的程序员的哈欠，以及那些已经双眼红肿的评论家们面无表情的眼神。但这是一本由 Meilir Page-Jones 写成的关于面向对象的范例的书，其作者是一位头脑非常清晰的思想家，他在软件开发方面的作品值得一读。因此，这本书对于那些想要彻底了解面向对象的读者，以及那些认为已经非常了解面向对象技术的评论家和程序员来说，都极具价值。

本书也许不是第一本关于面向对象的书籍，但是在它上面花费时间是值得的，这体现在它所包含的丰富的设计经验。这本书具有面向对象开发的最新见解，从继承的使用和滥用，到如何在软件中对小狗的主人和熊猫进行建模。这本书体现了 Page-Jones 的最新的成就和永恒的智慧。

我第一次遇见 Meilir，是由于他写了一本关于结构化程序设计的普及读本，而这本书的原版是我与 Ed Yourdon 合著的。Meilir 的书，《结构和系统设计实践指南》（Your don press/Prentice Hall 出版，1980），获得了巨大的成功，其销量超过了我的书好几倍，原因很简单——它的结构组织得更好而且更易读懂。1989 年，当得知我们两人都将在同一个会议上发言时，我非常急切地想要见到这位作家，要知道他可是非常透彻地领会了结构化程序设计的精髓，并从中提炼了许多对程序员实战非常有用的东西。在那次会议与我会面时，他起初有点拘谨，但当我告诉他我非常喜欢他的书，并且很高兴他能把结构化程序设计的思想带给那么多读者的时候，他和我便结识了。从那时起，我们开始了合作和竞争。因为他曾给我的《康士坦丁的人件》（Prentice Hall，1995）写过序言，所以我给他也写一个，我们这就扯平了。

事实上，Meilir 是一位天才的教师，他特别擅长将复杂和容易混淆的东西讲得非常明了简单。他能够把一车问题用一个原型包装起来，使它们变得如此易懂，以至于我们所有其他的人都不禁会问：“我们怎么没想到？”。在面向对象的“奶厂”

应用中，当到了挤奶的时候，你会怎么做？你会向“奶牛”这个对象发送一条消息，让它自己去挤奶，还是向“挤奶”发送一条消息？一个瞬时的反应和一个对事务管理者发出的协同挤奶的需求就很明白易懂。他所阐述的就像上面的例子，或书中的“人拥有狗”的题目，已经成为面向对象界常见的“段子”之一。

实际上，本书展示了如何将那些已经长期为专业设计人员使用的良好的设计规则转变为开发面向对象系统所用。本书讲述的不仅仅是关于一些大的原则或仅仅是关于一些小的细节，事实上，它两方面都兼顾。本书强调：当将实际经验融入具体例子时，一定要坚持不懈地、务实地注意真实世界的经验，这一点对新手和老手都很重要。

Meilir 无论作为一个咨询者，一个教师，还是一个方法学家，都从面向对象开发中提取了大量的经验。他是协同法的合作开发者，这是一个早期的面向对象分析和开发的方法；他也是创造具有深远影响的单一对象观念的合作者，这个观念的特性在数不清的面向对象方法和理念中（从 OODL 到 MOSES）均有反映。这种多用途建模法的最新版本，现被称为 OODN，用于演示和阐述全书的例子。

本书中用于说明设计的基本技术不但非常简单明了，而且所有的设计技术都有大量的例子，还讨论了在好的面向对象设计中该做和不该做的东西，并附有习题。

Larry I. Constantine

悉尼科技大学计算机科学教授

1995 年 6 月于澳大利亚，悉尼

前言

“所有的人都会对长期以来取悦他们的东西，而产生感激之情，甚至还有一点可敬的偏见。”

——William Wordsworth, *抒情民歌*

在此书还是草稿时，评审它的人们就向我提出了几个问题，也许你也有同样的问题，让我来回答一些吧。

为什么书名叫“程序员必读”？你太夸张了吧。

我相信有这样两类程序员：那些现在正在写面向对象代码的和那些很快就要写面向对象代码的。每个人写代码的同时也设计代码——不论好和坏，不论有意识的还是无意识的。我写这本书的目标是鼓励人们在编码前有意识地创造出好的面向对象的设计。为了这个目标，我介绍了符号表示、原则和术语，便于你和你的同事评价你的设计，进行有意义的讨论。

这本书可以教给我一种面向对象的编程语言吗？

不能。尽管我偶尔会涉及到代码，但这不是一本关于面向对象编程语言的书。

如果我正在学习一门面向对象的语言，这本书对我有用吗？

当然有用。如果你现在不懂一门面向对象的编程语言，你可以通过第 1 章开始学习面向对象的知识。了解一些面向对象的关键概念，本书将使你更快地学习一门面向对象的语言，并且我希望，在你迈向不熟悉的领域时它能鼓励你。本书的后续章节，将以合理的设计帮助你顺利开始你的早期的编程生涯。

如果你已经是经验丰富的面向对象程序员，你可以用本书的第 2, 3 部分来提高设计技巧，这些技巧对于你成为权威的专业软件设计者和程序员是至关重要的。

为什么这本书中的代码没有用 C++ 举例？

我在这本书中写的代码是用一种我自己发明的语言，它混合了三种流行的语言：C++，Eiffel 和 Smalltalk。我这样做是因为——猜猜看！——有两种程序员：那些熟悉 C++ 的和那些不熟悉 C++ 的。如果你是一个 C++ 迷，你可以发现那些代码可以轻而易举地被翻译成 C++；如果你对 C++ 不熟，你就会通过那些例子发现这种语言的看似神秘的语法。不管你的编程语言是什么，我都希望这本书能适合你所熟悉的程序语言。

为什么这本书没有关注设计窗口、图标和菜单？

有两个原因：首先，我不认为面向对象只对图形用户界面的设计有用处；其次，市面上已经有很多专注于面向对象窗口设计的书了，我希望这本书探讨那些其他面向对象书籍没有很好讨论过的话题。

这是一本关于方法学的书吗？

不。你知道，开发方法学包含的内容远远多于设计，如有需求分析、库管理等，同时，真正的方法学需要去解释不同的开发活动是如何被整合到一起的，太多了！

因此，不像其他很多面向对象的书籍那样罗嗦，我决心集中到一个话题上：面向对象设计。

你说了一大堆关于这本书不是什么的话，那它到底是什么？

它是一本介绍关于面向对象软件设计的基本的观点、符号、术语、准则和原理的书。面向对象软件是包含对象和它们所属的类的软件。一个对象就是一个组件，其中的方法（就像函数或程序）围绕一组变量（就像数据）组织起来。类实现了属于它的对象组一个类型定义。

对于软件设计者和程序员来说，上面的时髦词汇有着令人惊奇的含意，即继承、多态的设计概念和二次定制设计。不过，既然你问了一个具体问题，我就给你一个具体回答。

本书的第1部分（第1~2章）对面向对象做了一个介绍。第1章总结了主要的概念，概述了“多态”、“泛型”和所有其他的O.O.行话。第2章把面向对象放在以前的软件开发框架中，如果你对面向对象已经很熟悉（或许是在用面向对象语言编程），你可以跳过第1部分。

第2部分（第3~7章）介绍了面向对象设计符号，通过举例说明你在面向对象系统中发现的很多结构。第3章介绍了用于描述类的符号，以及它们的外部 and 内部设计。第4章涉及了用于描述子类和父类的继承关系，以及聚合对象的符号。第5章讲述用于消息（同步和异步）的符号，而第6章讨论了如何修改传统的状态转换图以适应于面向对象设计。第7章介绍了用于描述对象持续、系统架构和形成人机界面的窗口的符号。

第3部分（第8~12章）在一定的深度上介绍了面向对象的设计原理，第8章介绍了至关重要的同源和2层封装的符号。第9章探讨了产生类的各种领域，描述了类内聚的不同程度。第10章是第3部分的核心，使用状态空间和行为的概念去评估一个类继承是否合理和可扩展。

第11章进行了轻松的调剂，考察了来自于一些真实项目的设计，有精细的，也有荒谬可笑的（第11章实际上讨论了对于继承和多态滥用带来的危险）。第12章讨论了一个古老的问题：一个好的类是怎样构成的？为了回答这个问题，第12章描述了各种类接口——从那些合适的到那些糟糕的——解释了诸如混合类和方法环等设计技巧，你可以运用它们，让你设计的类尽可能强壮和可维护。

尽管我加进了大量的例子、图表和习题来加固我在书中讲述的内容，但我必须承认，在第3部分中的材料有些难于理解，不过，我不想对它进行普通处理或冲淡这些问题的重要性，因为面向对象设计的一些方面确实难以理解，但不管怎样对你总是有用的，我确信你能从中得到收获。

这本书是否囊括了面向对象设计的全部？

我非常怀疑这点。每天，我都从面向对象中学到更多的东西——我相信你也是！实际上，如果一本书就可以告诉我们关于面向对象设计的全部知识，我们再也不用学习更多的书了，不过这样也够没劲的。并且，这本书的全部内容并不是完全对的！

我写完以前的那些书后，确实有一两件事发生了变化，我变老了也更有智慧了——对，是更老了。

因此，尽管我认为在这本书里已经包括了许多重要的设计原理，如果你真地想对面向对象了解更多，你应该尽可能地继续读更多的书，并且不断地质疑，好好琢磨琢磨我呈现在这里的观点吧。

如他们所言，这是一本为我写成的书吗？

这是一个什么问题？你盼着我说“不”吗？不过说正经的，如果你是——或准备去做——一个程序员、设计师、系统工程师或使用面向对象技术的项目里管理人员，这本书就是为你写的，就算你是一个面向对象的初学者，你可以从本书第 1 部分的阅读中学到很多，练习面向对象编程，再转到第 2 部分、第 3 部分。

如果你是一名在校的大学生，或是已经掌握标准的过程化编程技巧的专业的程序员，正在寻找更宽广的技术领域，你应该阅读此书，本书中的大量材料适合在大学四年级的计算机科学或软件工程的面向对象课程中使用。

但是，不论你是什么角色，我希望你能享受这本书并且感到它对你是有用的，祝你好运！

Meilir Page-Jones

1995 年 4 月于华盛顿，布里韦

CompuServe 76334, 1247

目 录

第 1 部分 绪 论

第 1 章 面向对象究竟是什么	3
1.1 封装	7
1.2 信息/实现隐藏	10
1.3 状态保留	12
1.4 对象标识	12
1.5 消息	15
1.6 类	21
1.7 继承	26
1.8 多态	30
1.9 泛型	34
1.10 小结	37
1.11 习题	39
第 2 章 面向对象——谁定制了它们	41
2.1 面向对象的来源	41
2.2 面向对象的社会背景	43
2.3 作为工程科目的面向对象	47
2.4 面向对象的好处	49
2.5 小结	54
2.6 习题	55

第 2 部分 面向对象符号设计

第 3 章 类和方法的基本符号	59
3.1 OODN 目标	59
3.2 类符号	61
3.3 修改方法和访问方法	62

3.4	类外部接口图表	63
3.5	函数式方法	64
3.6	重载方法	66
3.7	方法的其他符号	67
3.8	包	70
3.9	类内设计图	71
3.10	小结	72
3.11	习题	73
第 4 章	继承和聚合图	75
4.1	类继承图	75
4.2	聚合对象及其组件	77
4.3	小结	81
4.4	习题	82
第 5 章	对象通信图	83
5.1	同步消息	83
5.2	异步消息和并发执行	92
5.3	小结	102
5.4	习题	102
第 6 章	状态转换图	105
6.1	基本状态转换图	105
6.2	嵌套状态	107
6.3	消息参数	109
6.4	连续变量	111
6.5	状态定义	113
6.6	小结	113
6.7	习题	114
第 7 章	其他 OODN 图	115
7.1	描绘数据库存取	115
7.2	描绘系统架构	121

7.3	描绘用户界面	124
7.4	小结	127
7.5	习题	127
第 3 部分 面向对象设计的基本原则		
第 8 章	封装和同源	131
8.1	封装结构	131
8.2	同源	134
8.3	小结	144
8.4	习题	145
第 9 章	域、相关度和内聚	147
9.1	类对象的域	147
9.2	相关度	152
9.3	类内聚: 类和它的方法	156
9.4	小结	162
9.5	习题	163
第 10 章	类和子类的属性	165
10.1	类的状态空间和行为	165
10.2	子类的状态空间	168
10.3	子类的行为	170
10.4	作为状态空间限制的类不变式	171
10.5	前提条件和后续条件	174
10.6	类型一致的原则	176
10.7	作为子类型的子类	177
10.8	闭合行为的原则	183
10.9	小结	185
10.10	习题	186
第 11 章	继承和多态的危险性	189
11.1	继承的滥用	189

11.2	多态的危险性	197
11.3	小结	205
11.4	习题	206
第 12 章	类接口	209
12.1	混合类	209
12.2	方法环	213
12.3	类接口的质量	217
12.4	小结	228
12.5	习题	230
附录 A	面向对象设计过程检查清单	235
附录 B	面向对象设计者手册	241
附录 C	面向对象术语快速导航	245
附录 D	答案	247
	术语表	279

第 1 部分 绪 论

生命是为了维护自身利益而与周围对象进行的斗争。概念则是为了回应攻击而制定的有战略意义的计划。

——Jose Ortega Gasset, *The Revolt of the Masses*

“面向对象”本身没有任何意义。“对象”大概是英语中最普通的一个单词。它在字典中的定义是这样的。

对象：能感觉到或能够被感觉到的东西。

换句话说，对象可以是任何东西！

“面向”也没有任何特别的含义，它的定义是“指向”。在这种定义下，“面向对象”被理解为形容词。因此，面向对象可以定义成这样。

面向对象：指向你可以想到的任何东西。

毫无疑问，软件工业无法为“面向对象”下一个统一的定义。这种模糊性使得任何一个软件零售商都声称他们的商品是“面向对象”的。

几年前，我决定彻底解决这个问题。我召集了 12 个面向对象领域的权威，把他们关在不提供水和食物的房间里，并且告诉他们只有对面向对象下一个确切的定义——一个我可以向软件领域发布的定义——之后才能让他们出来。

屋子里先是一片尖叫和嘈杂，一个小时后恢复了平静。怀着万分的恐惧，我小心地打开一条门缝往里面看，各位权威都还在，只不过彼此分散坐，互不搭理。

接着每一位权威开始讨论，试图通过不断大声的宣称——这是由来已久的科技惯例了吧——来建立对于面向对象的定义。但这种方式没有得出结果，于是他们决