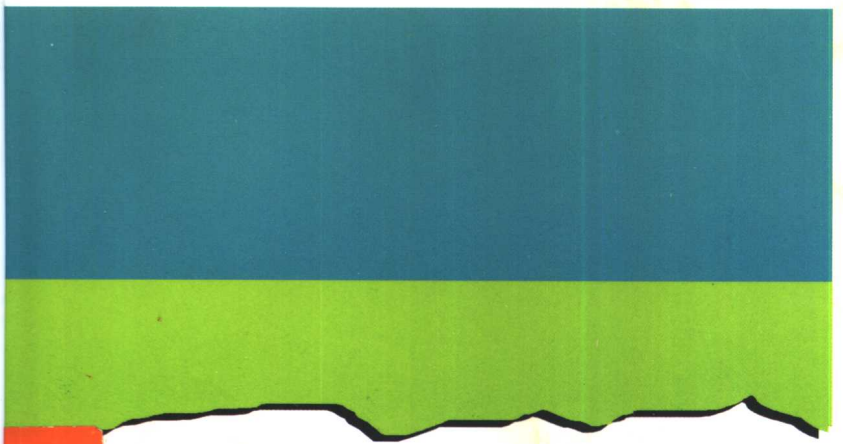
Addison Wesley

面向对象系统的测试

[美] Robert V.Binder 著
华庆一 王斌君 陈 莉 译



OBJECT TECHNOLOGY SERIES

BOOCH
JACOBSON
RUMBAUGH

ADDISON-WESLEY SERIES EDITORS

人民邮电出版社
www.pptph.com.cn

面向对象系统的测试

[美] Robert V. Binder 著

华庆一 王斌君 陈 莉 译

人民邮电出版社

图书在版编目(CIP)数据

面向对象系统的测试:模型、样式和工具/(美)宾德(Binder)著;华庆一,王斌君,陈莉译.
—北京:人民邮电出版社,2001.4

ISBN 7-115-09152-8

Ⅰ.面… Ⅱ.①华…②王…③陈… Ⅲ.数据库系统 Ⅳ.TP311.13

中国版本图书馆 CIP 数据核字(2001)第 16804 号

面向对象系统的测试

◆ 著　[美] Robert V.Binder
译　华庆一　王斌君　陈　莉
责任编辑　陈冀康　万东旭

◆ 人民邮电出版社出版发行　北京市崇文区夕照寺街 14 号
邮编　100061　电子函件　315@ pptph.com.cn
网址　http://www.pptph.com.cn
读者热线　010-67129212　010-67129211(传真)
北京汉魂图文设计有限公司制作
北京顺义振华印刷厂印刷
新华书店总店北京发行所经销

◆ 开本:787×1092　1/16
印张:49.75
字数:1 243 千字　2001 年 4 月第 1 版
印数:1－4 000 册　2001 年 4 月北京第 1 次印刷

著作权合同登记　图字:01－2000－0019 号
ISBN 7-115-09152-8/TP·2107

定价:87.00 元

内容提要

本书是对面向对象应用进行测试设计和实现自动化的权威性的指南。

全书分预备知识、模型、样式和工具四部分，共 19 章，阐明了为什么测试必须是基于模型的，并给出了建立可测模型的技术，引入了测试样式的概念并介绍了 37 种样式，以及如何设计和实现面向对象测试的自动化。

作者力求以易于理解的方式阐述面向对象测试，使读者无需花费太多精力即可以理解并加以利用。本书既可以作为教材，也可以作为面向对象测试的手册，供在软件测试第一线工作的人员以及对面向对象开发和测试有兴趣的人员使用。

版权声明

译者的话

面向对象测试是面向对象技术的重要方面。随着面向对象开发方法和面向对象程序设计语言的广泛使用，用户采用对象技术开发的软件系统和产品也日益增多。测试是保证软件系统和产品达到高质量和高可靠性的重要方面，它代表了对规约、设计和编码的最终审查。有关面向对象测试研究也受到软件界的重视。但是这些研究多散见于各种杂志、会议文集和未正式发表的资料中，而系统地指导人们进行面向对象软件测试的著作还不多。

本书是指导人们进行面向对象测试的一部综合性著作。它阐明了为什么测试必须是基于模型的，并深入地介绍了从状态机、组合逻辑和 UML 模型开发可测模型的技术；强调了测试样式的重要性，并介绍了 37 种测试样式，说明如何设计基于责任的测试包，如何对 OO(面向对象)代码进行集成和回归测试，如何测试构件和框架，如何使用实例开发高效测试包。有效的测试必须是自动的，并使用对象技术的优势。本书也描述了如何设计基于规约的断言和编码，以解决由于继承性和多态性而可能导致的可测试性的损失、测试自动化中十分重要的预测、测试装置设计问题以及 17 种相关样式。

本书由卞雷、华庆一教授组织翻译。陈莉副教授、博士翻译了第一部分，刘晓霞副教授、博士，郭小群硕士翻译了第二部分，王斌君副教授、博士，龚晓庆博士翻译了第三部分，赵慧博士翻译了第四部分，华庆一教授翻译了术语汇编，并由卞雷教授对译稿进行了审校。在本书翻译过程中，西北大学软件工程研究所郝克刚教授、西北大学英语系徐启升教授给予了热情的指导。西北大学计算机科学系刘小宁、韦志勇在译稿的编辑、打印等方面做了大量工作。在此谨向所有关心支持此书出版的人们表示深切的感谢。

由于时间仓促，水平有限，本书肯定会有翻译不妥的地方或没有检查出来的错误。我们诚恳地希望得到读者的帮助。

程序设计行家说："任何程序，无论多么小，都有错误。"

新手不相信行家的话。“如果程序小到只能执行一个单一的功能，也是这样吗?”他问道。

“这样的程序不会有任何意义。”行家说。“假如这样的程序存在，操作系统最终也会由于一个错误而失效。”

新手并不满意。“如果操作系统不失效呢?”他问道。

“没有不失效的操作系统。”行家说。“假如这样的操作系统存在，硬件最终也会由于错误而失效。”

新手仍然感到不满意。“如果硬件不失效呢?”他问道。

行家长叹了一口气。“不存在不失效的硬件。”他说。“假如存在这样的硬件，用户还会要程序做不同的事情。这也是一个错误。”

没有错误的程序是荒唐可笑的，是不可能存在的。假如存在没有任何错误的程序，那么世界也会不复存在。

Geoffirey Jamers

《The Zen of Programming》(程序设计的境界)

原书序

一些热心的、但被误导的早期面向对象程序设计(OOP)的倡导者，不考虑测试，是由于他们错误地相信——OOP 的采用将会大大减少错误的发生率，使得测试将不再需要。我们在两个时代前，在 Cobol 作为标准程序设计语言的时代，就听到过类似的言论。其后的 CASE 尽管有明显的生产率优势，却也仍然不能不考虑测试，无法履行它的承诺。对于所有这三个时代，Cobol 时代、CASE 时代和近来的面向对象时代而言，如果范式(paradigm)的采用可将生产率提高到在代码生成中不再需要任何人力的程度，则剩下的全部工作将是测试和调试—将消耗 100%的工作内容。正如对于过程式程序设计语言，从过去几十年来有时甚至是痛苦的经历中所认识到的，每一个进步都有其代价；就面向对象而言，导致更大灵活性、健壮性、一般性和高生产率的真实东西，也是要求进行测试的东西；如果不是更为困难，至少也更具挑战性。

几乎所有在测试过程式程序设计语言程序中所学到的东西，也适用于测试面向对象实现。如本书所示，面向对象测试也是建立在这样的基础之上的。然而，对面向对象来说，各种测试技术的重要性和有效性是不同的。例如，人们可能没有理由将数据流测试和有限状态机测试，同时用于用过程式程序设计语言写的应用；而对一个利用了 OOP 必须提供的东西的应用，则同时使用这些技术是不可避免的。此外，单元测试和集成测试的相对重要性也有改变。在过程式语言中，单元测试处于首要地位，而集成测试次之。但在 OOP 中，这种相对重要性恰好相反。

面向对象程序设计也给测试者带来新的问题，这些问题在过程式程序设计中无法发现。在这些问题中，多态、继承和动态绑定(binding)是最易出问题的地方，而它们又是面向对象的核心。某些早期的面向对象测试研究相当悲观，竟然到了说“面向对象有什么用呢?我们完全无法对它进行测试，或许也无法进行真正的调试”的地步。这种说法是 OOP 研究团体和机敏的从业者都不愿接受的。测试面向对象软件的方法正是在这些团体的共同关

心下出现的。这些方法使用了新的技术和/或为了适合于这种新范式而经过改造的老技术。然而，大部分这样的知识从业人员是难于得到的；它们散布在成百上千的研究论文或大量未发表的 OOP 资料之中。Binder 对研究成果精巧的论述消除了这种隔阂。这些研究成果，尽管是建立在以前程序设计范式(经过几十年使用业已证明)的坚实基础之上，但也经过了在给 OOP 提供了方法和技术的大厦中实践的严酷现实锤炼。我相信本书提供了 OOP 丢失的那一半—测试的那一半。

Boris Beizer
阿宾格登　宾夕法尼亚

前　言

有关本书

《面向对象系统的测试》是指导人们设计面向对象软件的测试包及自动测试的一本书。它向人们说明了，针对面向对象编程语言及面向对象的分析/设计方法，如何设计测试实例的过程，类、类簇、框架、子系统和应用系统均在考虑之列，并对许多测试设计问题提供了实用的综合指导。具体包括如下内容：

- 如何利用行为模型、状态—空间覆盖和接口数据流分析，设计用于类和小簇的基于责任能力的测试。
- 如何利用覆盖分析，评估测试的完全性。
- 如何利用依赖分析和层次状态模型，设计用于族和子系统的基于责任的测试。
- 如何利用 00A/D 方法，设计用于应用系统的基于责任的测试。
- 如何用面向对象的驱动器测试、桩测试、框架测试和内置测试，使测试执行自动化。

本书是关于系统工程和软件工程的书，更是一本有关测试面向对象软件的书。用于测试设计的模型(model)是必须的——本书向读者展示了如何开发可测试的模型，用于防止和排除错误。书中讨论的样式(pattern)对于设计测试包是非常有效的。工具(tool)可用于实现测试设计——本书告诉读者如何设计有效的自动测试框架。

这本书适合你吗

本书是为那些想要改善面向对象系统的可靠性的人而写的，涉及了基本方法和高级方法。我一直想使这本书像一个设计得很好的厨房，如果你所需要的是一块三明治和一杯冷饮，而你并不了解高

水平的配方、各种成分的目录等，有了这本书，就足以为 20 个客人准备一套七个系列的午餐。

这里，假设读者已至少了解了面向对象程序设计和面向对象分析/设计。如果你是像大多数面向对象的开发者一样，对某一个语言(大多数可能是 C++或 Java)已经很在行，可能还生成过或使用过某一个面向对象的模型，但是我并不认为你对测试了解得很多；你还需要一些计算机科学和软件工程的背景，以便于理解本书更深入的内容。但是即使没有特殊的理论训练，你也能应用测试设计样式。

如果必须回答下列问题之一，你会发现本书是很有用的：

- 测试过程软件和测试面向对象软件之间的不同是什么?
- 我刚刚写了一个新的子类，它似乎是可运行的，我还需要重新测试所继承的超类的特征吗?
- 需要什么类型的测试，以确信一个类对于所有可能的消息序列具有正确的行为?
- 对于快速的、增量的开发，一种好的集成测试策略是什么?
- 用 UML 表示的模型，如何用于设计测试?
- 为了使类和应用的测试更容易，我能做些什么?
- 如何利用测试获取更多的复用?
- 应该设计测试驱动模块和桩模块吗?
- 如何使测试实例可复用?
- 如何设计一个好的用于面向对象应用系统的系统测试计划?
- 多少测试实例就足够了?

本书列举的资料不限于某个特定的面向对象程序设计语言、某个 00A/D 方法、某种应用或某种目标环境。然而，我自始自终使用了统一建模语言(UML)。代码例子是用 Ada 95，C++，Java，Eiffel，Objective-C 和 Smalltalk 写的。

一点看法

我刚刚告诉我的 7 岁的儿子 David 我不得不早点结束同他进行的棒球游戏，回来从事我的写书工作。他又问："爸爸，你的书为什么这么厚?"我要解释我的选择，我试图清楚而诚实地作出回答：

> 测试是复杂的，而我是一个工程师。确保所从事的工作是正确的，对于工程师们是很重要的。假如建筑师因为偷懒不能使我们的房子十分结实，你想将会发生什么情况?房子会倒塌而且我们要受到伤害。假设 GM 的工程师们对用于汽车刹车的软件不做最终的程序测试，当我们需要刹车时，这个软件就可能不能正常工作，就可能出事故。所以当工程师工作或回答一个有关如何工作的问题时，必须确信自己是正确的。必须确信没有忘掉什么。要做到这些，需要大量的工作。

正如我所说的，正是认识到了这一点我才一直努力着。这也解释了我为何要写这本书以及我对测试面向对象软件这一问题的看法。测试是软件工程的一个组成部分。面向对象技术并不减弱测试的作用。与其它程序设计语言范型相比，面向对象技术改变了一些重要的技术细节。所以，从软件工程的观点看，这是一本有关测试应该如何用于面向对象系统开发的厚书。本书之所以厚是因为测试及面向对象系统的开发都是大的主题，具有很大的交叉性。

致　谢

所有对我写书有过帮助的人都不必为书中的错误负责[1]。Dave Bulman，Jim Halon，Pat Loy，Meilir Page-Jones 和 Mark Wallace 总结了有关 FREE 方法学的第一篇技术报告［Binder94］。

1993 年，ACM 的 *Communication* 杂志的编辑 Diane Crawford 接受了我的出版面向对象测试专辑的建议，于 1994 年 9 月出版了面向对象测试的专辑。投稿者帮助我明确了开发过程和测试过程间的关系。Bill Sasso 还有安德森咨询公司倡议举办一个专门回答问题的讲座，从而导致了第 12 章的模式机测试样式(见第 12 章)内容的提出。Chicago 大学的 Bob Ashenhurst，James Weber 和 Regis 研究组的其它人员提出了更多的基本问题：如什么是一个状态(state)？为什么我们应该关心情景(picture)？

后来的一年，作为 *Object Magazine* 的编辑，Marie Lenzie 接受了我的建议，推出了有关测试的双月专栏。从 1995 年以来，每年 6 次写这个专栏，迫使我把常常模糊的概念改变为明确而实用的指导思想。《*Software Testing，Verification and Reliability*》杂志的编辑，Western Reserve 大学 CASE 的 Lee White 和 Liverpool 大学的 Martin Woodward 不断地鼓励我，期望我对所做工作进行全面的论述，并耐心地等待，然后安排这些论著的出版和发行。所写的综述有助于分清什么问题是重要的，为什么必须回答这些问题，以及什么是最有用的思想和不能回答什么问题。

有关面向对象测试的论著、会议指南和专业开发研讨班作为概念上的资料库和试验场，其中的许多资料都做了必要的修改，然后用在了这里。在这一点上，我要感激 RBSC 公司、SIGS 出版社、ACM、IEEE 和 Wiley(U.K)的合作(参见后面资料来源和出处的细节)。由我的咨询客户和研讨班的数千名参与者提出的现实问题，一直使我感到知识的不足并且不断鼓舞我去精益求精。

Carter Shanklin 和他在 Addison Wesly 的前任的耐心支持，　使

[1] 这是 John Le Carre 在《*The Tailor of Panama*》中一句言简意赅的话，我直接引用了这句话。

这个项目一直保持着活力。Boris Beizer 的始终不渝的鼓励、建议和尖锐的批评一直都是我无价的财富。

几个熟练的程序员为我推荐了代码的例子并帮助我改进了有关内容：他们是 Brad Appleton(过滤样式及其它地方的 C++代码)，Steve Donelow(Objective-C 的内置测试)，Dave Hoag(Java 内部类驱动器)，Paul Stachour(Ada95 断言和驱动器)，以及 Peter Vandenberk(Objective-C 断言)。

全书的草稿经过了许多人的审阅。我十分感激审阅者及其详细的反馈意见。Elaine Weyuker 帮助调试了第 6 章提出的变量否定策略的解释。Brad Appleton 和 Chicago 样式研究组举办了两次样式撰写人的研究讨论会，其主要内容是讨论测试设计样式的模板和不变边界与过滤样式的早期版本。Ward Cunningham 评价了测试样式模板的前期草稿。另外有几个人基于他们的工作评审了测试样式：Tom Ostrand(范畴—划分)，John Musa(根据纲要安排测试)以及 Michael Feathers(增量测试框架)。Derek Hatley 审阅了组合逻辑的早期版本(第 6 章)；Lee White 审阅的回归测试(第 15 章)；Doug Hoffman 审阅了第 18 章预测；而 Dave Hoag 审阅了第 19 章测试装置设计。许多无名的审阅者为手稿早期版本的进一步修改提出了很好的建议。Brad Appleton，Boris Beizer，Camille Bell，Jim Hanlon 和 Paul Stachour 审阅了最后完整的手稿并给出了十分中肯的评论。

最后，感谢 Judith，Emily 和 David 多年以来的支持、理解和鼓励。

资料来源和出处

这样的结果。随后，在 1949 年的记者招待会上，由 John Stapp 再次做了陈述。自由引用。

第 7 章：选自 Lewis Carroll 的《*Through the Looking Glass*》(Project Gutenberg etext Edition，1994)。自由引用。

第 8 章："哈哈，不是开玩笑吧"(Ha-ha，only serious)的口头语常常由 Chicago 大学商业学院的 Robert Ashenhurst 教授挂在嘴边。用在这里经过了 Robert Ashenhurst 的许可。Ashenhurst 认为："事实上，我的引证与哲学家 W.V.O.Quine 所说的'没有同一性就没有实体'的话是一致的。虽然他是在存在论(分析哲学中最受人关注的部分)中讲这句话的，但概念'实体'(=对象)和'同一性'(=系统标识符 ID)，不做字面上的任何修改，正像这些词在面向对象上下文中所理解的，对于对象建模也是适合的"。

第 9 章：引用了 Daniel A. Yergin 和 Joseph Stanislaw 的《*The Commanding Height*》(New York:Simon &Schuster，1998)，第 195 页，该引用征得了 Simon & Schuster 的许可。

第 11 章：选自 Brain Marick 的《*The Craft of Software Testing:Subsystem Testing，Including Object-based and Object-oriented Testing*》(Englewood Cliffs，NJ:Prentice Hall，1995)，第 342 页。得到了 Pearson Education 的许可。

第 14 章：选自 H. Tredennick(翻译)，《*Aristotle's Metaphysics*》(Cambridge，MA:Loeb 古典图书馆，Harvard 大学出版社，1933)。Harvard 大学出版社允许引用。

第 15 章：选自 Eric Raymond 的《*The New Hacker 's Dictionary*》(Cambridge，MA:MIT Press，1991)，第 205 页。MIT 出版社允许引用。

第 17 章：1987 年 11 月，在白宫举行的记者招待会上，里根总统说道："尽管我的发音使你听起来很困难，格言是'doveryai，no proveryai'——相信它，但要证实它。"参见 George Schultz 的《*Turmoil and Triumph:Ym Years as Secretary of State*》(New York:Charles Scribner's Sons，1993)。这句俄国格言翻译为命令语气："相信它，但要证实它"，用俄语讲是很压韵的。我要感谢 Nadya Moiseeva，Oksana Deutsch 和 Igor Chudov，因为他们针对 soc.culture.russian.moderated.newsgroup 的疑问证实了拼写和翻译。

第 18 章：选自 Herodotus 所写，George Rawlinson 翻译，Hugh Bowden 编辑的《*The Histories*》(ISBN:0460871706，J.M.Dent)，copyright © 1992，J.M.Dent。这里的引用得到了 Everyman Publishers PLC 的许可。

商标

本书所用的条款不影响任何商标的有效性。

ENVY 是 Object Technology International Inc.的注册商标(OTI)。OTI 是隶属于 IBM 加拿大有限公司的子公司。

NeXT，NeXT logo，NEXTSTEP，NetInfo 和 Objective-C 是 NeXT Software Inc.的注册商标。

Solaris 是 Sun Microsystems 公司的注册商标。

作　者

目　录

第一部分　预备知识