HZ BOOKS
华章教育

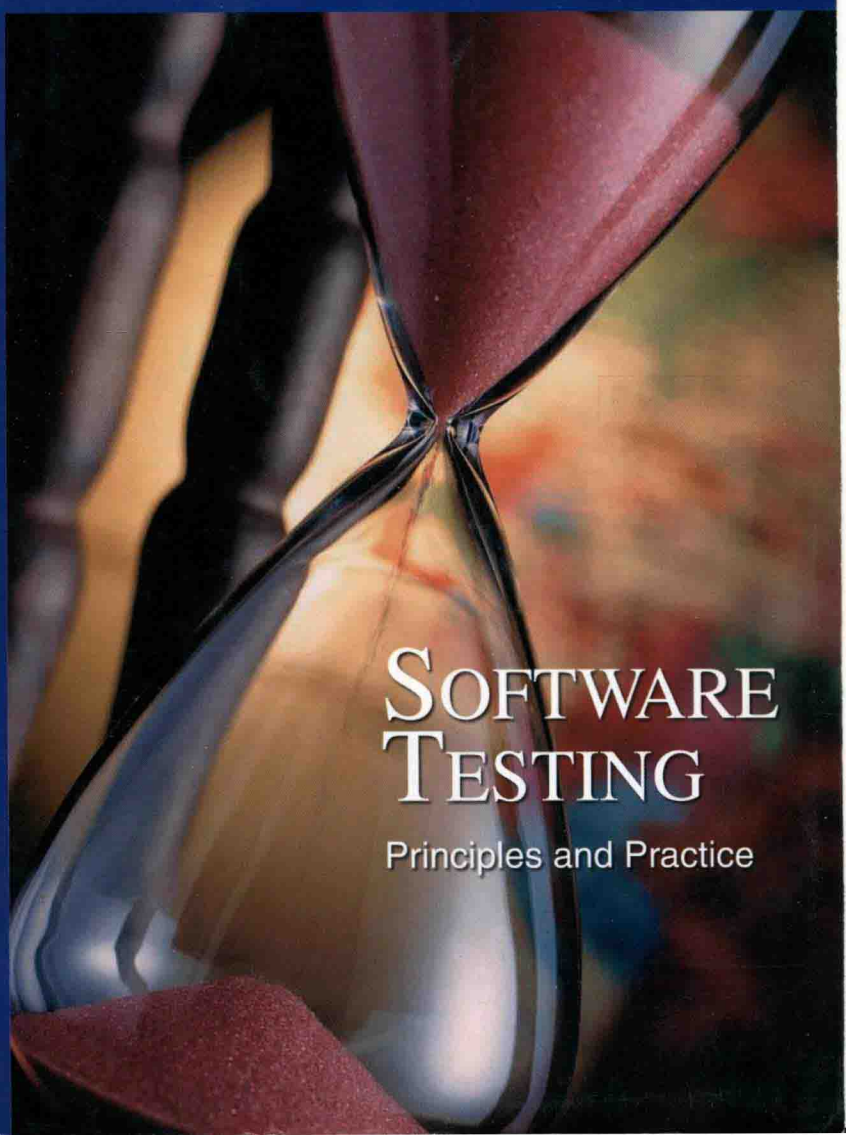
经 典 原 版 书 库

软件测试

原理与实践

（美国）Stephen Brown Joe Timoney Tom Lysaght
（中 国）Deshi Ye（叶德仕） 著

Software Testing
Principles and Practice
（英文版）



机械工业出版社
China Machine Press

这本软件测试教材基于爱尔兰国立大学梅努斯和中国浙江大学的一系列课程讲稿编写而成。全书一开始简单回顾了软件产业的发展历程，进而引出接下来要讲述的本书主要内容——软件测试的基础知识、原理以及相关技术。然后，在详细阐述静态和动态软件测试原理之前给出了一些关键定义。接着，针对过程式编程和面向对象编程（使用Java语言）提供了两个完整的单元测试实例。随后，本书讨论了系统测试和软件测试自动化的方法学。在这之后，本书采用常用的软件过程来解释软件测试在软件开发生命周期中的作用。最后，作者讨论了多个测试方面的高级主题，这些开放性问题需要深入的思考和研究，更进一步的资料在研究生课程中提供。

本书主要是面向软件测试的本科课程来设计相关内容，但书中讲述的理论和技术辅以许多实际的例子来说明，这些实用的内容对于软件专业人员也很有吸引力，因为软件测试是现代软件工程项目管理的一个非常关键的要素，其重要性日渐显现，正在引起软件产业乃至IT产业的充分重视……

本书特色

- 凝练作者多年的行业经验和教学经验，以丰富的实例，系统阐释软件测试的基本理论和技术。
- 附录中给出大量与实际应用相关的练习，便于读者理解和巩固所学知识。
- 涵盖一些高级主题，如测试技术选择、用浮点数测试、用复杂数据结构测试、GUI组件的单元测试、嵌入式软件测试等，需要读者在接下来的学习和工作实践中不断深入研究和思考。

限中国大陆地区销售

客服热线：(010) 88378991, 88361066
购书热线：(010) 68326294, 88379649, 68995259
投稿热线：(010) 88379604
读者信箱：hzjsj@hzbook.com

华章网站 http://www.hzbook.com

网上购书：www.china-pub.com

封面设计 · 杨宇梅

上架指导：计算机 软件测试

ISBN 978-7-111-38863-0

9 787111 388630

定价：35.00元

HZ BOOKS
华章教育

软件测试

原理与实践

Software Testing
Principles and Practice

Stephen Brown
（爱尔兰）Joe Timoney
Tom Lysaght
（中国）Deshi Ye 著

英文版

机械工业出版社
China Machine Press

经 典 原 版 书 库

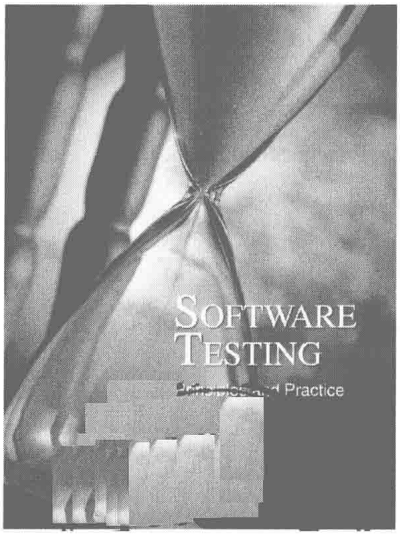
软件测试

原理与实践

（英文版）

Software Testing

Principles and Practice



（爱尔兰）Stephen Brown Jo[illegible]mon Lysaght
（中　国）Deshi Ye 叶德仕
著

机械工业出版社
China Machine Press

Authorized English language edition from the Original edition, entitled Software Testing: Principles and Practice by Stephen Brown, Joe Timoney, Tom Lysaght, Deshi Ye.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanic, including photocopying, recording, or by any information storage retrieval system, without the prior permission of by Stephen Brown, Joe Timoney, Tom Lysaght, Deshi Ye.

English language edition published by China Machine Press.

Copyright © 2012 by China Machine Press.

This edition is manufactured in the People's Republic of China, and is authorized for sale and distribution in the Worldwide.

本书英文版由 Stephen Brown, Joe Timoney, Tom Lysaght, Deshi Ye 授权机械工业出版社在全球独家出版发行。未经 Stephen Brown, Joe Timoney, Tom Lysaght, Deshi Ye 预先许可，不得以任何方式抄袭、复制或节录本书的任何部分。

封底无防伪标均为盗版

版权所有，侵权必究

本书法律顾问 北京市展达律师事务所

本书版权登记号：图字：01-2012-3795

图书在版编目（CIP）数据

软件测试：原理与实践（英文版）/（爱尔兰）布朗（Brown，S.）等著．—北京：机械工业出版社，2012.6

（经典原版书库）

书名原文：Software Testing: Principles and Practice

ISBN 978-7-111-38863-0

I. 软… II. 布… III. 软件－测试－英文 IV. TP311.5

中国版本图书馆 CIP 数据核字（2012）第 130806 号

机械工业出版社（北京市西城区百万庄大街 22 号 邮政编码 100037）

责任编辑：迟振春

北京瑞德印刷有限公司印刷

2012 年 9 月第 1 版第 1 次印刷

170mm×242mm • 13 印张

标准书号：ISBN 978-7-111-38863-0

定价：35.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

客服热线：（010）88378991；88361066

购书热线：（010）68326294；88379649；68995259

投稿热线：（010）88379604

读者信箱：hzjsj@hzbook.com

此为试读，需要完整PDF请访问：www.ertongbook.com

出版者的话

文艺复兴以降，源远流长的科学精神和逐步形成的学术规范，使西方国家在自然科学的各个领域取得了垄断性的优势；也正是这样的传统，使美国在信息技术发展的六十多年间名家辈出、独领风骚。在商业化的进程中，美国的产业界与教育界越来越紧密地结合，计算机学科中的许多泰山北斗同时身处科研和教学的最前线，由此而产生的经典科学著作，不仅擘划了研究的范畴，还揭示了学术的源变，既遵循学术规范，又自有学者个性，其价值并不会因年月的流逝而减退。

近年，在全球信息化大潮的推动下，我国的计算机产业发展迅猛，对专业人才的需求日益迫切。这对计算机教育界和出版界都既是机遇，也是挑战；而专业教材的建设在教育战略上显得举足轻重。在我国信息技术发展时间较短的现状下，美国等发达国家在其计算机科学发展的几十年间积淀和发展的经典教材仍有许多值得借鉴之处。因此，引进一批国外优秀计算机教材将对我国计算机教育事业的发展起到积极的推动作用，也是与世界接轨、建设真正的世界一流大学的必由之路。

机械工业出版社华章公司较早意识到“出版要为教育服务”。自 1998 年开始，我们就将工作重点放在了遴选、移译国外优秀教材上。经过多年的不懈努力，我们与 Pearson，McGraw-Hill，Elsevier，MIT，John Wiley & Sons，Cengage 等世界著名出版公司建立了良好的合作关系，从他们现有的数百种教材中甄选出 Andrew S. Tanenbaum，Bjarne Stroustrup，Brain W. Kernighan，Dennis Ritchie，Jim Gray，Afred V. Aho，John E. Hopcroft，Jeffrey D. Ullman，Abraham Silberschatz，William Stallings，Donald E. Knuth，John L. Hennessy，Larry L. Peterson 等大师名家的一批经典作品，以“计算机科学丛书”为总称出版，供读者学习、研究及珍藏。大理石纹理的封面，也正体现了这套丛书的品位和格调。

“计算机科学丛书”的出版工作得到了国内外学者的鼎力襄助，国内的专家不仅提供了中肯的选题指导，还不辞劳苦地担任了翻译和审校的工作；而原书的作者也相当关注其作品在中国的传播，有的还专程为其书的中译本作序。迄今，“计算机科学丛书”已经出版了近两百个品种，这些书籍在读者中树立了良好的口碑，并被许多高校采用为正式教材和参考书籍。其影印版“经典原版书库”作为姊妹篇也被越来越多实施双语教学的学校所采用。

权威的作者、经典的教材、一流的译者、严格的审校、精细的编辑，这些因素使我们的图书有了质量的保证。随着计算机科学与技术专业学科建设的不断完善和教材改革的逐渐深化，教育界对国外计算机教材的需求和应用都将步入一个新的阶段，我们的目标是尽善尽美，而反馈的意见正是我们达到这一终极目标的重要帮助。华章公司欢迎老师和读者对我们的工作提出建议或给予指正，我们的联系方法如下：

华章网站：www.hzbook.com

电子邮件：hzjsj@hzbook.com

联系电话：(010) 88379604

联系地址：北京市西城区百万庄南街 1 号

邮政编码：100037

华章教育

华章科技图书出版中心

Preface

This book is based on a series of lectures given at the National University of Ireland, Maynooth and Zhejiang University. It provides a textbook for a number of courses, describing the fundamentals of software testing. The material has been developed over the past ten years, and reflects both the experiences from 20 years in industry from one of the authors, and the authors' joint experiences in lecturing.

There is no one standard textbook on software testing, and this book is the result of many years of extracting and interpreting test techniques from a wide and varied number of sources. These include testing classics such as The Art of Software Testing by Myers, Software Testing by Roper, and Testing Object-Oriented Systems by Binder; standard textbooks such as Software Engineering by Pressman and Ince, and Software Engineering by Sommerville; software process books such as Software Testing in the Real World by Kit and extreme Programming explained by Beck; and ISO and IEEE standards related to software quality and testing.

Software testing is a challenging task – it is as important for businesses and government as it is for research institutions. It is still as much an art as a science: there are no accepted standards or norms for applying the various techniques, and interpretation is required. There is no well established research on the effectiveness of different approaches. The techniques are easy to understand, but generally difficult to apply to real-world software. By providing extensive worked examples, this book aims to provide a solid basis for both understanding, and applying, various test techniques

前　言

本书的内容基于爱尔兰国立大学梅努斯和浙江大学的一系列课程讲稿。该书覆盖了软件测试的基本原理，可以作为许多课程的参考教材。本书的内容历经十年发展，融合了其中一位作者近二十年的工业界经验及全部作者的教学经验。

在软件测试领域，目前还没有统一的标准教科书，而本书是通过对各种不同的软件测试技术进行多年的提炼、阐释而形成的。这些测试技术的来源包括：一些经典的测试书籍，如 Myers 的《软件测试的艺术》、Roper 的《软件测试》、Binder 的《面向对象系统测试》；标准的软件工程教材，如 Pressman 和 Ince 的《软件工程》、Sommerville 的《软件工程》；软件过程类书籍，如 Kit 的《现实世界中的软件测试》、Beck 的《极限编程》；软件质量及测试相关的 ISO 和 IEEE 标准。

软件测试是一项具有挑战性的任务，它对科研机构、企业及政府具有同等的重要性。软件测试既是一门科学也是一门艺术。针对如何应用不同的软件测试技术，目前还没有广泛认同的标准，因此对各种软件测试技术的阐释是必需的。针对各种不同技术的有效性，目前还没有成熟的研究成果。总的来说，各种测试技术容易理解，但如何将其应用到软件产品中是困难的。本书旨在通过广泛丰富的实例，为读者理解和应用各种不同的软件测试技术提供一个坚实的基础。

List of Figures

1.1 Failure Rate for a Software Product over its Lifecycle 5
1.2 Ideal Project Progression using Forward Engineering 6
1.3 Realistic Project Progression including Checks 7
1.4 Verification in the Development Process 7
1.5 Validation in the Development Process 8

2.1 Software Tests Procedure 15
2.2 Specification Model of Testing 18
2.3 Implementation Model of Testing 19
2.4 Natural Ranges 23
2.5 Equivalence Partitions 23
2.6 State Diagram for counter 30
2.7 CFG for Sequence 31
2.8 CFG for Selection (if-then) 32
2.9 CFG for Selection (if-then-else) 32
2.10 CFG for Selection (switch) 33
2.11 CFG for Iteration (while) 33
2.12 CFG for Iteration (do-while) 34
2.13 CFG for Iteration (for) 34
2.14 Paths for condIsNeg() 35
2.15 CFG for Basis Paths Example 37
2.16 Coverage Criteria Ranking 39
2.17 Test Components 40

3.1 CC without BC 55

4.1 CFG for seatsAvailable() 69
4.2 CFG for premium() 84

5.1 Sample Code Inspection Checklist 101

6.1 Object-Oriented Testing Model 102
6.2 State Diagram for CarTax 112
6.3 Class Diagram for Banking System 114

7.1 Test Drivers and Stubs 118
7.2 Drivers and Stubs Example 119
7.3 Top-down Integration Testing 119
7.4 Bottom-Up Integration Testing 120

7.5 Sideways Integration Testing 121
7.6 System Test Model 124
7.7 GUI Test Model 125
7.8 Network Test Model 125
7.9 Browser Web Test Model 125
7.10 Direct Web Test Model 126
7.11 Calcint Main Window 127
7.12 Calcint Confirm Window 128
7.13 Calcint GUI State Diagram 128

9.1 Timeline of the stages of incremental testing 142
9.2 Key Documents in IEEE Standard 892-1998 143
9.3 Waterfall Model 144
9.4 V-Model 145
9.5 Documentation in the V-Model 146
9.6 Incremental Development 148
9.7 Testing in eXtreme Programming 149
9.8 The SCRUM Process 150
9.9 Synch and Stabilize 152
9.10 Levels of The Capability Maturity Model 153

10.1 IEEE 754 Floating Point 159
10.2 Overlapping Partitions 163
10.3 Relative Partitions 164
10.4 Invalid Triangles 166
10.5 Boundary Crossings 168

B.1 Seats.java 181
B.2 SeatsTest.java 181

List of Tables

1.1 Software Quality Attributes ISO 9126 . . . 3
1.2 Classification of Software Failures . . . 5

2.1 Comparison of Black-Box and White-Box Testing . . . 16
2.2 Truth Table for isNegative() . . . 27
2.3 Truth Table for isZero() . . . 27
2.4 Truth Table for largest() . . . 28
2.5 Truth Table for condIsNeg(int,boolean) . . . 29
2.6 Test Data Specification Template . . . 41

4.1 Input Partitions for seatsAvailable() . . . 63
4.2 Output Partitions for seatsAvailable() . . . 63
4.3 Test Cases for seatsAvailable() . . . 64
4.4 EP Tests for seatsAvailable() . . . 64
4.5 Input Boundary Values for seatsAvailable() . . . 65
4.6 Output Boundary Values for seatsAvailable() . . . 66
4.7 BV Tests for seatsAvailable() . . . 66
4.8 Truth Table for seatsAvailable() . . . 67
4.9 TT Tests for seatsAvailable() . . . 68
4.10 Random Test Cases for seatsAvailable() . . . 68
4.11 Random Data Tests for seatsAvailable() . . . 69
4.12 SC Test Cases for seatsAvailable() . . . 70
4.13 SC Tests for seatsAvailable() . . . 70
4.14 BC Test Cases for seatsAvailable() . . . 70
4.15 BC Tests for seatsAvailable() . . . 71
4.16 CC Test Cases for seatsAvailable() . . . 71
4.17 CC Tests for seatsAvailable() . . . 72
4.18 DCC Test Cases for seatsAvailable() . . . 72
4.19 DCC Tests for seatsAvailable() . . . 73
4.20 MCC Test Cases for seatsAvailable() . . . 73
4.21 MCC Tests for seatsAvailable() . . . 74
4.22 Path Test Cases for seatsAvailable() . . . 74
4.23 DU Pairs for freeSeats . . . 75
4.24 DU Pairs for seatsRequired . . . 75
4.25 DU Pairs for rv . . . 75
4.26 DUP Test Cases for seatsAvailable() . . . 76
4.27 DUP Tests for seatsAvailable() . . . 76
4.28 Initial SMT Test Results for seatsAvailable() . . . 77
4.29 Improved SMT Test Results for seatsAvailable() . . . 77

此为试读，需要完整PDF请访问：www.ertongbook.com

4.30 EP Input Test Cases for premium() 79
4.31 EP Output Test Cases for premium() 80
4.32 EP Tests for premium() 80
4.33 BV Input Test Cases for premium() 81
4.34 BV Output Test Cases for premium() 81
4.35 BV Tests for premium() 82
4.36 Truth Table for premium() 83
4.37 TT Tests for premium() 84
4.38 SC Test Cases for premium() 85
4.39 SC Tests for premium() 85
4.40 BC Test Cases for premium() 86
4.41 BC Tests for premium() 86
4.42 CC Test Cases for premium() 87
4.43 CC Tests for premium() 88
4.44 DCC Test Cases for premium() 89
4.45 DCC Tests for premium() 90
4.46 MCC Test Cases for premium()/Decision 1 91
4.47 MCC Test Cases for premium()/Decision 2 91
4.48 MCC Test Cases for premium()/Decision 3 92
4.49 MCC Test Cases for premium()/Decision 4 92
4.50 MCC Tests for premium() 92
4.51 Path Test Cases for premium() 93
4.52 Path Tests for premium() 94
4.53 DU Pairs for age 94
4.54 DU Pairs for gender 95
4.55 DU Pairs for married 95
4.56 DU Pairs for premium 95
4.57 DUP Tests for premium() 96
4.58 SC Tests for premium() 96
4.59 SMT SC Test Results for premium()/Mutation(1) 97
4.60 BC Tests for premium() 97
4.61 SMT BC Test Results for premium()/Mutation(1) 97

6.1 EP Test Cases in Class Context for CarTax 107
6.2 EP Tests in Class Context for CarTax 108
6.3 Method-level d-u pairs for co2Pollution 109
6.4 DUP Tests for co2Pollution 109
6.5 Truth Table for CarTax 110
6.6 TT Tests for CarTax 111
6.7 State Transitions for CarTax 112
6.8 State Tests for CarTax 113

7.1 Navigation Tests for Calcint 129
7.2 Appearance Tests for Calcint 130
7.3 Functional Tests for Calcint 130

10.1 Truth Table for 4 Booleans 166
10.2 Pairwise Test Cases for 4 Booleans 167
10.3 BX Tests for negProduct() 167

Contents

Preface iv
前言 v

1 Introduction 1
1.1 The Software Industry . . . 1
1.2 Software Testing and Quality . . . 2
1.3 Errors, Faults and Failures . . . 3
1.3.1 Software Faults . . . 3
1.3.2 Software Failures . . . 4
1.3.3 Need for Testing . . . 6
1.4 The Role of Specifications . . . 8
1.5 Overview of Testing . . . 9
1.5.1 Testing in the Development Process . . . 9
1.5.2 Test Automation . . . 9
1.6 The Theory of Testing . . . 10
1.6.1 Exhaustive Testing Example . . . 10
1.6.2 Implications . . . 11
1.7 When To Finish Testing . . . 12
1.8 Notes on Book Structure . . . 12

2 Principles of Software Testing 14
2.1 Dynamic and Static Verification . . . 14
2.1.1 Static Verification . . . 14
2.1.2 Dynamic Verification . . . 14
2.2 Black-Box and White-Box Testing . . . 15
2.2.1 Errors of "Omission" and "Commission" . . . 16
2.3 Test Approaches . . . 17
2.3.1 Black-Box Testing . . . 18
2.3.2 White-Box Testing . . . 18
2.3.3 Fault Insertion . . . 20
2.4 Test Activities . . . 20
2.4.1 Analysis Outputs . . . 20
2.4.2 Test Cases . . . 20
2.4.3 Test Data . . . 21
2.4.4 Test Code (or Test Procedures) . . . 22
2.5 Analysis of Software Specifications . . . 22
2.5.1 Parameters . . . 22
2.5.2 Parameter Ranges . . . 22
2.5.3 Equivalence Partitions . . . 23

2.5.4 Boundary Values . . . 24
2.5.5 Combinations of Values . . . 25
2.5.6 Sequences of Values . . . 29
2.6 Analysis of Software Components . . . 31
2.6.1 Control Flow Graphs . . . 31
2.6.2 Decisions and Conditions . . . 34
2.6.3 Paths . . . 35
2.6.4 Data-Flow Testing . . . 38
2.6.5 Ranking . . . 38
2.7 Analysis of Targets for Fault Insertion . . . 39
2.7.1 Offutt's 5 Sufficient Mutations . . . 40
2.8 Test Artefacts . . . 40

3 Unit Testing **42**
3.1 Introduction . . . 42
3.2 Usage . . . 43
3.3 Black-Box Techniques . . . 44
3.3.1 Equivalence Partitioning (EP) . . . 44
3.3.2 Boundary Value Analysis (BVA) . . . 45
3.3.3 Testing Combinations of Inputs (CI) . . . 46
3.3.4 Testing Sequences of Inputs (SI) or State . . . 47
3.3.5 Random Input Data (RID) . . . 48
3.3.6 Error Guessing (EG) . . . 50
3.4 White-Box Techniques . . . 51
3.4.1 Statement Coverage (SC) . . . 51
3.4.2 Branch Coverage (BC) . . . 52
3.4.3 Condition Coverage (CC) . . . 54
3.4.4 Decision Condition Coverage (DCC) . . . 55
3.4.5 Multiple Condition Coverage (MCC) . . . 56
3.4.6 Path Coverage (PC) . . . 57
3.4.7 d-u pair Coverage (DUP) . . . 58
3.5 Fault Insertion . . . 59
3.5.1 Strong Mutation Testing (SMT) . . . 59

4 Unit Testing Examples **61**
4.1 Example One: seatsAvailable() . . . 61
4.1.1 Description . . . 61
4.1.2 Specification . . . 62
4.1.3 Source Code . . . 62
4.1.4 Equivalence Partitioning . . . 62
4.1.5 Boundary Value Analysis . . . 65
4.1.6 Combinational Testing . . . 66
4.1.7 Tests . . . 67
4.1.8 Using Random Test Data . . . 68
4.1.9 Statement Coverage . . . 69
4.1.10 Branch Coverage . . . 70
4.1.11 Condition Coverage . . . 71
4.1.12 Decision/Condition Coverage . . . 72
4.1.13 Multiple Condition Coverage . . . 73

4.1.14 Path Coverage 74
4.1.15 D-U pair Coverage 75
4.1.16 Strong Mutation Testing 76
4.2 Example Two: premium() 78
4.2.1 Description 78
4.2.2 Specification 78
4.2.3 Source Code 79
4.2.4 Equivalence Partitioning 79
4.2.5 Boundary Value Analysis 80
4.2.6 Combinational Testing 82
4.2.7 Statement Coverage 84
4.2.8 Branch Testing 85
4.2.9 Condition Coverage 86
4.2.10 Multiple Condition Coverage 90
4.2.11 Path Testing 93
4.2.12 d-u pair Testing 94
4.2.13 Strong Mutation Testing 96

5 Static Verification **98**
5.1 Design Reviews 98
5.1.1 Informal Walk-through 98
5.1.2 Formal Design Review 99
5.2 Static Code Analysis 99
5.2.1 Walk-throughs 99
5.2.2 Code Inspections 100

6 Testing Object-Oriented Software **102**
6.1 Characteristics Of Object-Oriented Software 102
6.2 Effects Of OO On Testing 103
6.3 Object Oriented Testing Models 103
6.3.1 Conventional Models 104
6.3.2 Combinational Models 104
6.3.3 State Machine Models 104
6.3.4 Specification & Design Models 104
6.3.5 Built-In-Test 105
6.4 Example 105
6.4.1 Class CarTax 105
6.4.2 Black-Box Testing in Class Context 107
6.4.3 White-Box Testing in Class Context 108
6.4.4 Combinational Testing 110
6.4.5 State-Machine Testing 111
6.4.6 Specification/Design Testing 113
6.4.7 Built-In Testing 115

7 Integration and System Testing **117**
7.1 Integration Testing 117
7.1.1 Drivers and Stubs 117
7.1.2 Top-Down Integration 119
7.1.3 Bottom-Up Integration 120

7.1.4 Sandwich Integration . . . 121
7.1.5 End-to-end User Functionality . . . 121
7.1.6 Test Cases . . . 121
7.1.7 Conclusion . . . 122
7.2 System Testing . . . 122
7.2.1 System Test Categories . . . 122
7.2.2 System Level Functional Testing . . . 124
7.2.3 Test Cases . . . 126
7.2.4 GUI Example . . . 126
7.3 Field Testing and Acceptance Testing . . . 131

8 Software Test Automation 132
8.1 Coverage Measurement . . . 132
8.1.1 Lazy Evaluation . . . 133
8.2 JUnit . . . 133
8.2.1 JUnit Example . . . 134
8.2.2 Test Documentation . . . 135
8.3 JUnit Testing in an IDE . . . 137
8.4 Regression/Inheritance Testing With JUnit . . . 137
8.5 Writing Your Own Test Runner . . . 139
8.6 Mutation Testing . . . 140
8.7 Automated System Testing . . . 140

9 Testing in the Software Process 142
9.1 Test Planning . . . 143
9.2 Software Development Life Cycle . . . 144
9.3 The Waterfall Model . . . 144
9.4 The V-Model . . . 145
9.5 Incremental and Agile Development . . . 146
9.5.1 Incremental Development . . . 147
9.5.2 Extreme Programming . . . 148
9.5.3 SCRUM . . . 150
9.5.4 Synch And Stabilize . . . 151
9.5.5 Process-Related Quality Standards and Models . . . 152
9.6 Repair-Based Testing . . . 155
9.6.1 Specific Repair Test . . . 155
9.6.2 Generic Repair Test . . . 155
9.6.3 Abstracted Repair Test . . . 155
9.6.4 Example . . . 156
9.6.5 Repair-Based Test Suites . . . 156

10 Advanced Testing Issues 157
10.1 Philosophy of Testing . . . 157
10.2 Test Technique Selection . . . 158
10.3 Inserting Data Faults . . . 158
10.4 Design For Testability (DFT) . . . 158
10.5 Testing with Floating Point Numbers . . . 159
10.6 Unit Testing of GUI Components . . . 160
10.7 Testing with Complex Data Structures . . . 161

10.8 Automated Random Testing . . . 161
10.9 Automated Statement and Branch Testing . . . 162
10.10 Overlapping and Discontinuous Partitions . . . 162
10.11 Handling Relative Values . . . 164
10.11.1 Classic Triangle Problem . . . 165
10.12 Pairwise Testing . . . 166
10.13 Testing Boundary Crossings (BX) . . . 167
10.14 Varying Input Parameters . . . 169
10.15 Extended Combinational Testing . . . 169
10.15.1 No "Don't-Care"s . . . 169
10.15.2 Test "Don't-Care"s Individually . . . 169
10.16 Including Testing In The Build Procedure . . . 170
10.17 Testing Concurrent and Parallel Software . . . 170
10.17.1 Unit Testing . . . 171
10.17.2 System Testing . . . 171
10.17.3 Static Analysis . . . 171
10.17.4 Tools . . . 172
10.18 Testing Embedded Software . . . 172
10.19 Testing Network Protocol Processing . . . 174
10.19.1 Text-Based Protocols . . . 174
10.19.2 Binary Protocols . . . 175
10.19.3 Protocol Stacks and DFT . . . 175
10.20 Research Directions . . . 175

APPENDICES

A Terminology **178**

B Exercises **180**
B.1 JUnit and Eclipse . . . 180
B.2 Unit Test - Exercise 1 . . . 182
B.3 Unit Test - Exercise 2 . . . 183
B.4 Unit Test - Exercise 3 . . . 184
B.5 Unit Test - Exercise 4 . . . 185
B.6 Unit Test - Exercise 5 . . . 186
B.7 Unit Test - Exercise 6 . . . 187
B.8 Unit Test - Exercise 7 . . . 188
B.9 Unit Test - Exercise 8 . . . 189
B.10 Unit Test - Exercise 9 . . . 190
B.11 Exercise 10 - Test Projects . . . 191

Select Bibliography **192**

此为试读,需要完整PDF请访问: www.ertongbook.com