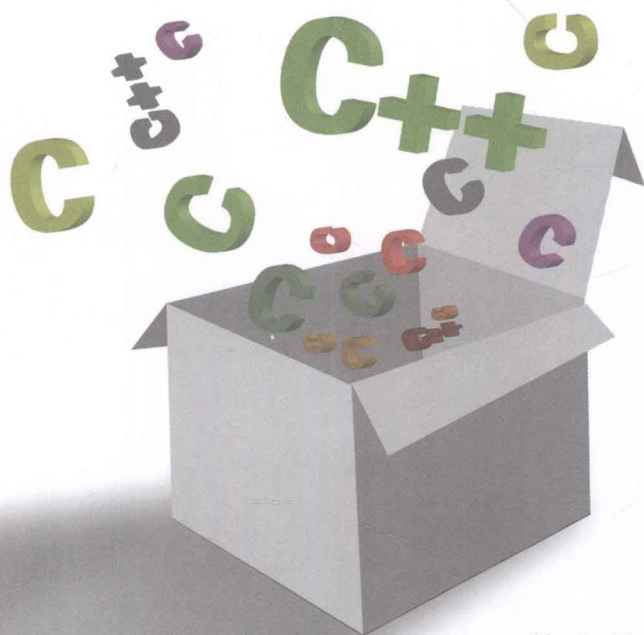


Parasoft认证指定用书系列

白盒测试之道

——C++test



徐宏革 郭 庆 编著
涛 杨 坤



北京航空航天大学出版社
BEIHANG UNIVERSITY PRESS

Parasoft 认证指定用书系列

白盒测试之道——C++test

徐宏革 郭 庆 编著
雷 涛 杨 坤

北京航空航天大学出版社

内 容 简 介

本书对当前软件行业中容易忽视的白盒测试通过结合专业化的测试工具 C++test 进行了深入的讨论, 主要包括软件测试语言基础、软件测试基础、软件测试环境、测试计划、静态测试、数据流分析、逻辑覆盖、CppUnit、单元测试、测试插桩、回归测试、黑盒测试、团队协同开发测试及 C++test 常见问题汇总等。这些技术和方法是任何一位合格的测试工程师所必备的知识, 也是软件测试培训、软件测试交流等所涉及的知识。阅读本书, 不仅可以获取软件测试中白盒测试的精髓, 还可以得到专业的 C++test 的项目实践。

本书可供软件测试工程师、测试主管、研发主管、项目经理等人员阅读, 也可作为初入测试领域者的引导教材, 还可作为相关职业教育的软件测试培训教材。

图书在版编目(CIP)数据

白盒测试之道: C++test/徐宏革等编著. —北京
:北京航空航天大学出版社, 2011. 4
ISBN 978-7-5124-0386-4

I. ①白… II. ①徐… III. ①软件—测试 IV.
①TP311.5

中国版本图书馆 CIP 数据核字(2011)第 046723 号

版权所有, 侵权必究。

白盒测试之道——C++test

徐宏革 郭 庆 编著
雷 涛 杨 坤
责任编辑 王慕冰 龚荣桂 王平豪

*

北京航空航天大学出版社出版发行

北京市海淀区学院路 37 号(邮编 100191) <http://www.buaapress.com.cn>

发行部电话:(010)82317024 传真:(010)82328026

读者信箱: emsbook@gmail.com 邮购电话:(010)82316936

涿州市新华印刷有限公司印装 各地书店经销

*

开本: 787×1092 1/16 印张: 21.5 字数: 550 千字

2011 年 4 月第 1 版 2011 年 4 月第 1 次印刷 印数: 4 000 册

ISBN 978-7-5124-0386-4 定价: 39.00 元

前言

本书是 Parasoft 认证指定用书,主要应用于 C/C++ 测试领域。几位作者,近几年来,一直投身于软件项目实施和软件测试,具有丰富的项目管理和软件测试经验。我们通过提供一些实际经典的例程,并结合专业的测试理论,将白盒测试之道——C++test 全新地呈现到各位读者面前,希望能够让读者学着轻松,领悟深刻,为各位读者走向成功做些绵薄的铺垫。

全书分为五篇,共 14 章。

第一篇主要以测试基础为重点,包含第 1~4 章,共 4 个章节。第 1 章引入 C 和 C++ 语言基础(若读者是专注于 C 或 C++ 领域的工程师,则可以跳过)。第 2 章介绍软件测试基础,通过软件测试的发展历史和现状让读者认清测试领域的形势,并对测试的相关基本知识进行介绍,尤其是引入了 Parasoft 的自动化错误预防理论。第 3 章介绍软件测试环境的搭建,以及 C++test 环境的搭建、工程的导入与创建。第 4 章介绍测试计划的编写,让读者对测试领域工作有一个全面的认识。

第二篇主要介绍白盒测试,包含第 5~11 章,共 7 个章节。第 5 章介绍静态测试的基本知识,并引入 C++test 专业的静态测试。第 6 章介绍数据流分析和 C++test 的专利测试技术 BugDetective 技术。第 7 章介绍白盒测试的核心逻辑覆盖、C++test 自动化测试及其测试覆盖分析。第 8 章介绍 CppUnit 相关内容,并解析 C++test 与 CppUnit 的关系。第 9 章介绍单元测试的相关理论,以及 C++test 如何开展自动化的单元测试工作。第 10 章重点介绍测试插桩技术以及 C++test 如何进行插桩。第 11 章分析回归测试以及 C++test 如何进行自动化回归测试。

第三篇主要介绍黑盒测试,即第 12 章。主要介绍黑盒测试技术与 C++test 专业的数据源测试,以及其独有的相关功能黑盒测试。

第四篇主要介绍团队配置,即第 13 章。介绍 C++test 团队协同测试管理方

案的实施。

第五篇即第 14 章是 C++test 执行测试过程时常见问题的汇总。

本书出现的相关例程,读者请登录 <http://www.edukit.com.cn/down-class.asp?lx=big&anid=62> 进行检索下载。

另外,读者还可以在该网站上下载最新的软件以及相关的视频教程。

在本书的编写过程中得到了各方面的支持和帮助。首先本书得到了 Parasoft 公司、Embest 公司和 Beyondtest 公司的大力支持。Parasoft 公司在 C++test 使用方面给予了全面的、完整的技术支持,并提供了相关的技术资料;Embest 公司团队在第一时间对相关文献进行修正并提出建设性建议;Beyondtest 公司在相关测试理论上给予了无私的帮助。在此要对 Parasoft 公司的陈兆仁、刘岳、伍宏益和李成等资深工程师表示感谢,对 Embest 公司的刘炽、尹锐、卢斯、余朝华、屈银和高之详等资深工程师表示感谢,对 Beyondtest 公司的李东辉、李毅飞和李昂等资深工程师表示感谢。另外,还要感谢 Pararsoft 公司亚太区总经理陈兆仁先生及其技术团队的大力支持以及北京博测科技有限公司团队的大力帮助。最后对北京航空航天大学出版社为本书审阅的老师们表示衷心的感谢。

由于编者水平和经验的限制,本书难免存在一些错误和不足,欢迎大家来信指正。

E-mail:qingguo_01@sina.com

作 者

2010 年 12 月

目 录

第一篇 基础知识

第 1 章 软件测试语言基础	3
1.1 C 语言基础	3
1.1.1 C 语言简介	3
1.1.2 C 语言特点	3
1.1.3 C 语言的数据类型	3
1.1.4 算术运算符与算术表达式	7
1.1.5 数据类型转换规则	9
1.1.6 程序结构	10
1.2 C++ 基础	12
1.2.1 C++ 介绍	12
1.2.2 C++ 的特点	13
1.2.3 C++ 程序的构成元素	13
1.2.4 C++ 程序的开发步骤	15
1.2.5 对象和类	16
1.2.6 成员函数的定义	17
1.2.7 对象的定义	17
1.2.8 构造函数与析构函数	17
1.2.9 静态成员	21
1.2.10 类的继承与派生	23
1.2.11 运算符重载的意义与规则	38
第 2 章 软件测试基础	41
2.1 软件测试现状	41
2.1.1 国外软件测试现状	41
2.1.2 国内软件测试现状与发展趋势	42
2.2 软件质量与开发效率的博弈	43
2.3 软件测试概念	45
2.3.1 软件测试的哲学思考	45

2.3.2	软件测试	47
2.3.3	软件测试的目的	49
2.3.4	测试和调试的区别	49
2.4	软件测试心理学	50
2.4.1	程序测试的过程具有破坏性	50
2.4.2	程序员应避免测试自己的程序	50
2.4.3	程序设计机构不应测试自己的程序	51
2.5	测试工程师应具备的素质	52
2.6	软件测试的分类	53
2.7	软件测试的流程	56
2.8	软件测试的原则	58
2.9	AEP 自动化错误预防	59
2.9.1	自动错误预防五大法则	59
2.9.2	自动化错误预防的标准流程	60
2.9.3	质量=可靠性+安全性	62
2.10	Parasoft C++test 简介	63
第3章	软件测试环境	65
3.1	测试环境	65
3.1.1	测试环境的主要构成	65
3.1.2	规划测试环境	67
3.1.3	管理测试环境	67
3.2	搭建 C++test 测试环境	69
3.2.1	支持的主机系统与编译器	69
3.2.2	C++test 基于 Windows 单机安装	72
3.2.3	C++test 基于 Windows 插件安装	73
3.2.4	C++test 基于 Linux /Solaris 单机安装	74
3.2.5	C++test 基于 Linux /Solaris 插件安装	74
3.2.6	许可授权	75
3.3	C++test 界面	78
3.3.1	C++test 透视图	78
3.3.2	视图	79
3.3.3	C++test 常用菜单命令	84
3.4	简单的测试工程实例	85
3.5	C++test 中工程的建立	86
3.5.1	通过链接打开工程	86
3.5.2	使用 CVS 和源目录	87
3.5.3	Microsoft Visual C++ 6.0 项目选项	88
3.6	C++test 中项目的导入	90
3.6.1	导入 Visual Studio 6.0 项目	90

3.6.2	命令行导入 Visual Studio 6.0	92
3.6.3	从构建数据文件导入项目	94
第 4 章	测试计划	96
4.1	测试计划的目标	96
4.2	测试计划主题	97
4.2.1	测试目标	97
4.2.2	资源环境	98
4.2.3	定 义	98
4.2.4	团队之间的责任	99
4.2.5	测试范围	101
4.2.6	测试阶段	101
4.2.7	测试策略	101
4.2.8	资源需求	101
4.2.9	测试员的任务分配	102
4.2.10	测试进度	102
4.2.11	测试用例	103
4.2.12	软件缺陷报告	103
4.2.13	度量和统计	104
4.2.14	风险和问题	104
4.3	编写软件测试计划需注意的问题	104
4.4	C++test 测试计划安排	105

第二篇 白盒测试

第 5 章	静态测试	109
5.1	词法分析与语法分析	109
5.2	静态错误分析	110
5.2.1	代码检查	111
5.2.2	静态结构分析法	121
5.2.3	静态质量度量法	123
5.3	C++test 静态测试	128
5.3.1	规则类别	128
5.3.2	严重性等级	129
5.3.3	C++规范	129
5.3.4	C++test 测试配置	137
5.3.5	使用 C++test	142
5.3.6	抑 制	148
5.3.7	查看结果报表	151
5.3.8	创建规则	154

第 6 章 数据流分析	167
6.1 数据流	167
6.2 Parasoft 数据流分析特点	168
6.3 BugDetective	169
6.3.1 BugDetective 规则内容	169
6.3.2 执行 BugDetective	169
6.3.3 BugDetective 的优越性	174
第 7 章 逻辑覆盖	176
7.1 实施逻辑覆盖原因	176
7.2 逻辑覆盖标准	177
7.2.1 语句覆盖	178
7.2.2 判定覆盖	178
7.2.3 条件覆盖	178
7.2.4 条件判定组合覆盖	179
7.2.5 多条件覆盖	179
7.2.6 修正条件判定覆盖	180
7.3 基本路径测试法	181
7.3.1 程序的控制流图	181
7.3.2 程序的环路复杂性	182
7.3.3 基本路径测试法步骤	183
7.4 其他白盒测试方法	185
7.4.1 域测试	185
7.4.2 符号测试	186
7.4.3 Z 路径覆盖	187
7.4.4 程序变异	188
7.5 白盒测试综合策略	188
7.5.1 最少测试用例数计算	188
7.5.2 测试覆盖准则	190
7.6 C++test 的覆盖类型	192
第 8 章 CppUnit	194
8.1 CppUnit 概述	194
8.2 基本原则	194
8.2.1 CppUnit 的原理	195
8.2.2 手动使用步骤	197
8.2.3 常用使用方式	199
8.2.4 其他实际问题	201
8.3 CppUnit	201
8.3.1 CppUnit 测试环境	202
8.3.2 CppUnit 开发环境设置	204

8.4 C++test 中使用 CppUnit	208
8.4.1 导入测试用例	209
8.4.2 条件/限制	211
第 9 章 单元测试	212
9.1 单元测试概念	212
9.2 单元测试设计	213
9.2.1 单元测试模型的设计	213
9.2.2 测试项目的设计	213
9.3 单元测试用例设计	214
9.3.1 单元测试用例编写原则	214
9.3.2 单元测试用例设计方法	215
9.4 单元测试的执行	220
9.4.1 测试的结论	220
9.4.2 自动测试	220
9.5 单元测试的评估	220
9.6 实施时遵循的原则和步骤	221
9.7 单元测试的一些考虑和定义	222
9.7.1 单元的定义与测试的定义	222
9.7.2 单元测试	222
9.7.3 测试模块	224
9.8 单元测试中的白盒测试设计	225
9.8.1 一个程序模块的例子	225
9.8.2 语句覆盖	225
9.8.3 判断覆盖	226
9.8.4 条件覆盖	226
9.8.5 判断-条件覆盖	227
9.8.6 条件组合覆盖	227
9.8.7 路径覆盖	228
9.8.8 测试用例的组合和优化	229
9.9 C++test 单元测试	229
9.10 基于 C++test 自动化测试	230
9.10.1 C++test 动态测试概述	230
9.10.2 C++test 自动化动态测试的具体操作流程	230
9.10.3 总 结	234
第 10 章 插桩技术	235
10.1 测试插桩	235
10.1.1 简 介	235
10.1.2 主要内容	237
10.2 C++test 桩函数机制	240

10.2.1	C++ test 实现桩函数	240
10.2.2	自动生成桩函数	241
10.2.3	自定义桩函数	244
第 11 章	回归测试	248
11.1	概 述	248
11.2	回归测试策略	248
11.2.1	测试用例库的维护	249
11.2.2	回归测试包的选择	249
11.2.3	回归测试的基本过程	250
11.3	回归测试实践	251
11.4	C++ test 执行回归测试	251
11.5	总 结	256

第三篇 C++ test 使用数据源测试

第 12 章	黑盒测试	259
12.1	什么是测试用例	259
12.2	设计测试用例的目的	260
12.3	等价类划分法	260
12.3.1	划分等价类和列出等价类表	261
12.3.2	确定测试用例	261
12.3.3	案例一	262
12.3.4	等价区间	263
12.3.5	案例二	264
12.3.6	总 结	265
12.4	边界值分析法	266
12.4.1	边界条件	267
12.4.2	次边界条件	268
12.4.3	边界值的选择方法	269
12.5	判定表驱动法	269
12.5.1	判定表组成	270
12.5.2	建立判定表	270
12.5.3	确定测试用例	270
12.5.4	案例一	271
12.5.5	案例二	272
12.5.6	案例三	272
12.6	基于 C++ test 自定义测试用例	273
12.6.1	C++ test 的自定义测试用例概述	273
12.6.2	C++ test 自定义测试用例具体操作流程	273

12.6.3 总 结.....	276
12.7 基于 C++test 的数据源测试用例	276
12.7.1 C++test 的数据源测试用例概述	276
12.7.2 C++test 的数据源测试用例具体操作流程	277
12.7.3 总 结.....	281

第四篇 团队测试

第 13 章 团队协同开发测试	285
13.1 配置团队部署.....	285
13.2 安装和设置.....	286
13.2.1 默认安装.....	286
13.2.2 Windows	286
13.2.3 UNIX	286
13.3 将所有 C++test 安装连接至源码控制器存储库	287
13.3.1 C++test 的源码控制器支持	287
13.3.2 启用源码控制器支持.....	287
13.3.3 测试集成.....	289
13.3.4 CVS 配置	289
13.3.5 Subversion 配置	291
13.3.6 ClearCase 配置	292
13.3.7 StarTeam 配置	292
13.3.8 CM Synergy 配置	292
13.3.9 Perforce 配置	292
13.3.10 Visual SourceSafe 配置	293
13.3.11 从命令行界面指定源码控制器定义	293
13.4 将所有 C++test 安装连接至 TCM	293
13.4.1 关于团队配置管理器 TCM	293
13.4.2 先决条件.....	294
13.4.3 将 C++test 连接至 TCM	294
13.4.4 延长 TCM 超时时限	295
13.5 将 C++test 服务器连接至 GRS	295
13.5.1 关于 GRS	295
13.5.2 先决条件.....	295
13.5.3 配置 C++test 以向 GRS 发送结果	295
13.5.4 配置 GRS 属性	297
13.5.5 将测试用例与 PRs 和 FRs 关联	297
13.5.6 访问 GRS 报告	298
13.6 配置团队测试配置和规则.....	298

13.6.1	关于团队测试配置	298
13.6.2	共享团队测试配置	299
13.6.3	修改团队测试配置	299
13.6.4	设置团队缺省的测试配置	300
13.6.5	共享规则映射	300
13.6.6	共享自定义规则	301
13.6.7	删除 TCM 规则	302
13.7	配置作者设置	302
13.7.1	关于 C++test 中的作者	302
13.7.2	了解 C++test 的作者和任务分配	303
13.7.3	更改作者计算方式	303
13.7.4	使用多个源码控制器系统时的作者处理	304
13.7.5	指定作者和电子邮件映射	304
13.8	运行初始测试和减少噪声	305
13.9	通过源码控制器共享项目和测试资源	306
13.10	配置自动夜间测试	307
13.10.1	配置报告	307
13.10.2	配置自动测试	307
13.11	配置了团队部署后每日用途应涉及方面	307
13.11.1	在添加到源码控制器之前验证新的/修改的代码	308
13.11.2	复审和纠正引入到团队的代码库中的错误	308
13.11.3	访问结果和报告	308
13.11.4	将结果从 TCM 导入到 C++test GUI	309
13.11.5	通过 GUI 访问 TCM 报告	310
13.11.6	通过 Web 浏览器访问 TCM 报告	311
13.11.7	将任务再分配给其他团队成员	311
13.11.8	监视项目范围的测试结果	311
13.12	Concerto	312
13.12.1	概述	312
13.12.2	Concerto 支持工具	313
13.12.3	Concerto 主要功能介绍	314

第五篇 问题汇总

第 14 章	C++test 常见问题汇总	325
附 录		330
参考文献		332

第一篇 基础知识

第 1 章

软件测试语言基础

1.1 C 语言基础

1.1.1 C 语言简介

C 语言于 1972 年由美国的 Dennis Ritchie 发明,并首次在配备了 UNIX 操作系统的 DEC PDP-11 计算机上实现。它由早期的编程语言 BCPL(Basic Combined Programming Language)发展演变而来。1970 年,AT&T 贝尔实验室的 Ken Thompson 根据 BCPL 语言设计出了较先进并取名为 B 的语言,通过不断修改、完善,更先进的 C 语言问世了。

C 语言是一种功能强大、应用广泛、具有发展前途的计算机语言。它既可用于系统软件的设计,也可用于应用软件的开发。许多著名的系统软件都是采用 C 语言编写的。

1.1.2 C 语言特点

- C 语言既具有低级语言直接操纵硬件的特点,又具有高级语言与自然语言和人的思维逻辑相似的特点,且其程序易编写、易查错,实用性很强;
- C 语言具有丰富的数据类型和运算符,语法结构简单;
- C 语言是一种结构化程序设计语言,提供了完整的程序控制语句;
- C 语言是一种模块化程序设计语言,适合大型软件的开发和研制;
- C 语言还有一个突出的优点,就是既适合于多种操作系统,如 DOS 和 UNIX,也适用于多种机型,其程序移植性好。

1.1.3 C 语言的数据类型

数据是程序处理的对象,数据类型是数据的内在表现形式。例如,学生的年龄和成绩具有一般数值的特点,在 C 语言中称为数值型,其中年龄是整数,称为整型;成绩可以为小数,称为实型。而学生的姓名和性别是文字,在 C 语言中称为字符型数据。

C 语言具有丰富的数据类型,其中基本的数据类型有整型、实型和字符型。

1. 变 量

变量是在程序执行过程中其值可以被改变的量。

(1) 变量命名规则

和人的取名一样,变量的命名也有一定的规则。

- 由字母、数字和下划线组成;
- 必须以字母或下划线开头;
- 字母区分大小写(在系统默认状态下);
- 前 32 个字符有效(在系统默认状态下)。

例如: a、Book、book、_Make_Cipher 都是合法的变量名,且 Book 与 book 是不同的变量名,而 123A、x+y 都不是变量名。

(2) 变量的数据类型

变量可以是任意的一种数据类型,如整型变量、字符型变量和指针变量等。C 语言中的基本数据类型及其特性如表 1-1 所列。

表 1-1 C 语言的基本数据类型及其特性

数据类型名	数据类型描述	数据类型的长度/字节	数据取值范围
char	字符型	1	0~255
int	有符号整型	2	-32 768~32 767
unsigned int	无符号整型	2	0~65 535
short	短整型	2	-32 768~32 767
long	长整型	4	-2 147 483 648~2 147 483 647
unsigned long	无符号长整型	4	0~4 294 967 295
float	单精度实数	4	$ 3.4 \times 10^{-38} \sim 3.4 \times 10^{38} $
double	双精度实数	8	$ 1.7 \times 10^{-308} \sim 1.7 \times 10^{308} $
long double	长双精度实数	10	$ 3.4 \times 10^{-4932} \sim 3.4 \times 10^{4932} $

(3) 变量的定义

每个变量在使用前都必须先定义其数据类型。定义变量数据类型的语法格式如下:

数据类型符 变量名 1, 变量名 2, ...;

例如:

```
int age,score;          /* 定义年龄和成绩为整型 */
char name[20];          /* 定义姓名为至多含 20 个字符的字符数组 */
```

(4) 变量的存储类型

当定义某个变量时,C 语言的编译系统就要给该变量分配若干个存储单元,用来存放该变量的值。而在计算机中寄存器和内存都可以存放数据,内存又可分为临时占用和长期占用。变量的存储类型是指变量在计算机中的存放位置及时间。

定义变量存储类型的语法格式如下: