

教育部高等学校软件工程专业教学指导委员会规划教材
高等学校软件工程专业系列教材



软件测试实验教程

◎ 朱少民 等 主编

清华大学出版社



教育部高等学校软件工程专业教学指导委员会规划教材
高等学校软件工程专业系列教材



软件测试实验教程

◎ 朱少民 马海霞 王新颖 刘冉 蒋琦 吴振宇 蔡秋晃 等 主编

清华大学出版社
北京

内 容 简 介

这本实验教程是对教材《软件测试方法和技术》的有力补充,指导教学过程中所需要的实验,含实验目的、环境、内容和详细的过程指导。全书共有 19 个实验,覆盖单元测试、集成测试、功能测试、性能测试、安全性测试和验收测试等,主要以当今流行的 Web 应用、移动 App 应用为测试对象,并增加了一些深度的实验,如反编译安全测试、开源测试框架 Fitnessse 的解析等,是软件测试教学不可多得的实验教材。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

软件测试实验教程/朱少民等主编. —北京:清华大学出版社,2019

(高等学校软件工程专业系列教材)

ISBN 978-7-302-52373-4

I. ①软… II. ①朱… III. ①软件—测试—高等学校—教材 IV. ①TP311.55

中国版本图书馆 CIP 数据核字(2019)第 039023 号

责任编辑:黄 芝 薛 阳

封面设计:刘 健

责任校对:徐俊伟

责任印制:丛怀宇

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课件下载: <http://www.tup.com.cn>, 010-62795954

印 装 者:北京鑫海金澳胶印有限公司

经 销:全国新华书店

开 本:185mm×260mm 印 张:14.75

字 数:362 千字

版 次:2019 年 6 月第 1 版

印 次:2019 年 6 月第 1 次印刷

印 数:1~1500

定 价:39.50 元

产品编号:078329-01

前 言

《软件测试方法和技术》已经出版整整十年了,从第1版到现在的第3版,深受几百所大学教师的喜欢,也获得不少殊荣,如被评为“十二五”普通高等教育本科国家级规划教材、上海市普通高等学校优秀教材。但在《软件测试方法和技术》作为教材使用的过程中,教师们总感觉实验的辅导不够,缺少一本实验辅助教材,毕竟软件测试是一门实践性很强的专业课程。软件测试的教学需要加强对学生动手能力的培养,而这恰恰需要借助课程相关的实验来实现。通过实验使学生更好地理解所学的测试方法和技术,将来在工作中也可以更好地应用这些方法和技术。为此,我们组织业界工程师来编写这本实验教材,作为《软件测试方法和技术》教材的有力补充,从而使软件测试教学达到更佳的效果。

如今,软件开发模式从传统的瀑布模式已转向敏捷开发模式,软件开发和软件测试越来越趋于融合,这也意味着不仅专职的测试人员要开展软件测试工作,而且开发人员也要从事测试相关的工作。从这个角度看,单元测试就显得更为重要,在软件测试教学中需要进一步加强。况且,在校的大学生对业务的感受比较少,但他们对代码更熟悉、更感兴趣,更容易接受单元测试,这和业界的需求也正好一致。为此,本实验教材重视单元测试,为单元测试共设计了7个实验,不仅包括逻辑覆盖(如语句覆盖、判定覆盖、条件覆盖、MCDC等)的测试设计、动态测试等实验,而且包括静态测试分析工具的实验。考虑到大多数学校开设了C/C++、Java编程的课程,动态测试工具选择了JUnit和CppUnit。在敏捷开发中,持续集成是最重要的、优秀的开发实践之一,为此增加了基于Jenkins的集成测试实验作为集成测试的关键实验。所以,在第1篇单元测试与集成测试实验中共设计了8个实验,分别是:

- ◇ 实验1: 语句和判定覆盖测试设计
- ◇ 实验2: 条件覆盖和条件组合覆盖测试设计
- ◇ 实验3: 修正条件/判定覆盖测试设计
- ◇ 实验4: 基于JUnit的单元测试
- ◇ 实验5: 基于CppUnit的单元测试
- ◇ 实验6: 基于JavaScript的单元测试
- ◇ 实验7: 基于PMD的静态测试
- ◇ 实验8: 基于Jenkins的集成测试

目前,Windows应用越来越少,而Web应用、移动App应用成为主流,所以在系统测试中主要以Web应用、移动App应用作为测试的对象(案例),开展系统的功能测试、性能测试、安全性测试、兼容性测试等。这类实验不仅要求学生能够进行测试分析、测试设计,而且要求学生能够开发自动化测试脚本,借助测试工具来完成测试脚本的执行与结果分析。从测试分析与设计的方法、思路上看,在不同的平台上(Web、移动App、Windows桌面、Mac

OS 桌面等)系统的功能性测试和非功能性测试基本是一致的。如果学生要开展 Windows 或 Mac OS 桌面的系统测试实验,也可以参照 Web 应用、移动 App 应用的相关实验,并利用网络资源,做到举一反三,完成相应的实验。如果确实有困难,可以发邮件到 Kerryzhu@tongji.edu.cn 提出问题,我们会给予解答。根据大家的反映,如果这类需求还比较多,我们将在本书第 2 版增加 Windows 桌面、Mac OS 桌面的相关测试实验。目前,我们在第 2、3 篇共设计了 7 个系统测试的实验,分别是:

- ◇ 实验 9: Web 应用的功能测试
- ◇ 实验 10: Web 应用的性能测试
- ◇ 实验 11: Web 应用的安全性测试
- ◇ 实验 12: 移动 App 功能与兼容性测试
- ◇ 实验 13: 移动 App 功能自动化测试
- ◇ 实验 14: 移动 App 代码反编译安全测试
- ◇ 实验 15: 移动 App 敏感信息安全测试

上述 15 个实验可以被看作软件测试教学的基本实验,可在基础教学计划中安排这些实验。但为了使教材内容相对完整,并照顾某些有测试方向的学校,增加了几个其他实验,覆盖验收测试、利用虚拟技术搭建测试环境等方面的内容。现在开源测试工具或框架很多,是在校学生很好的学习资源。针对开源测试工具的分析能够一举两得,既进一步了解测试工具的实现机制、对测试有更深的探讨与研究,又能学习开源框架的优秀编程实践,提升开发能力,为此特地增加了“开源测试框架 Fitnessse 的解析”实验。总之,在最后一篇,我们设计了 4 个实验,分别是:

- ◇ 实验 16: 基于 Fitnessse 的验收测试实验
- ◇ 实验 17: 开源测试框架 Fitnessse 的解析
- ◇ 实验 18: 搭建虚拟测试环境
- ◇ 实验 19: 系统安装/卸载和兼容性测试实验

本教材的每个实验,首先会说明实验目的、实验前提、实验内容、实验环境,让教师先检查一下是否具备这些条件和环境,明确实验目的和内容,然后再开始实验。如果不具备实验条件或环境,可先做些准备工作。每个实验在简要叙述实验环节之后给出详细的实验操作过程,教师和学生可以按教材的详细过程一步一步进行实验。实验需要安装的文件或文档,统一放在清华大学出版社网站(www.tup.com.cn),大家可以自行下载。

参与本教材编写的(按拼音顺序)有包蕾、蔡秋亮、陈林儿、姜华军、蒋琦、蒋兴、李燕青、林建宇、刘冉、刘涛、马海霞、王新颖、吴振宇、姚煌杰、郑碧娟、朱少民。

由于水平以及大家投入的时间都有限,教材中难免存在一些问题,希望大家不吝赐教,我们将尽力改正,力求不断推出更高质量的教材。

编者 于丁酉年

目 录

第 1 篇 单元测试与集成测试实验

实验 1	语句和判定覆盖测试设计	3
实验 2	条件覆盖和条件组合覆盖测试设计	8
实验 3	修正条件/判定覆盖测试设计	15
实验 4	基于 JUnit 的单元测试	20
实验 5	基于 CppUnit 的单元测试	30
实验 6	基于 JavaScript 的单元测试	40
实验 7	基于 PMD 的静态测试	49
实验 8	基于 Jenkins 的集成测试	54

第 2 篇 Web 应用的系列测试实验

实验 9	Web 应用的功能测试	79
实验 10	Web 应用的性能测试	95
实验 11	Web 应用的安全性测试	105

第 3 篇 移动 App 的系列测试实验

实验 12	移动 App 功能与兼容性测试	125
实验 13	移动 App 功能自动化测试	137
实验 14	移动 App 代码反编译安全测试	145
实验 15	移动 App 敏感信息安全测试	150

第 4 篇 验收测试及其框架解析实验

实验 16	基于 Fitnesse 的验收测试实验	159
实验 17	开源测试框架 Fitnesse 的解析	168
实验 18	搭建虚拟测试环境	175
实验 19	系统安装/卸载和兼容性测试实验	182

附加案例.....	186
教材中源代码.....	212
附录 A Java 环境配置	218
附录 B 邮件服务器搭建	219
附录 C SVN 环境安装配置.....	225
附录 D 关于 JeeSite	228

第1篇

单元测试与集成测试实验

单元测试是软件开发和系统测试的基础,只有每个单元模块得到了充分的测试,系统测试才能相对轻松地完成,否则系统测试变得没有止境,缺陷永远找不完。作为开发人员,需要对自己所写的代码负责,也需要做单元测试。单元测试一般和编程同步进行,写完一段代码,就要进行单元测试。

相对来说,软件的单元规模很小,可以精确、有效、完整地进行测试,也比较容易进行测试覆盖率的分析,通过不断改进测试,最终可以达到所需的测试覆盖率。从测试充分性看,单元测试可以更好地帮助我们保证软件产品质量。

单元测试,除了人工的代码评审,其他的测试(代码静态分析、动态测试等)都属于自动化测试范畴,通过工具和脚本自动完成。单元测试的实验,从代码行覆盖/判定覆盖开始,逐步深入到条件覆盖、条件/判定覆盖、组合覆盖和 MC/DC 覆盖、基本路径覆盖等,并结合 PMD、JUnit、CppUnit 等单元测试工具,完成实际代码的测试,其中包括测试覆盖率的度量和分析,并把 TDD/ATDD/BDD、类、包的测试和 Mock 技术的运用等更复杂的单元测试给大家练习与思考。

本篇主要开展单元测试实验。通过这些实验,提高同学们的单元测试能力,并巩固结构化测试方法的应用。

- ◇ 实验 1: 语句和判定覆盖测试设计
- ◇ 实验 2: 条件覆盖和条件组合覆盖测试设计
- ◇ 实验 3: 修正条件/判定覆盖测试设计
- ◇ 实验 4: 基于 JUnit 的单元测试
- ◇ 实验 5: 基于 CppUnit 的单元测试
- ◇ 实验 6: 基于 JavaScript 的单元测试
- ◇ 实验 7: 基于 PMD 的静态测试
- ◇ 实验 8: 基于 Jenkins 的集成测试

实验 1

语句和判定覆盖测试设计

1.1 实验目的

- (1) 巩固所学的语句覆盖和判定覆盖测试方法；
- (2) 提高运用语句覆盖和判定覆盖测试方法的能力。

1.2 实验前提

- (1) 掌握语句覆盖和判定覆盖的基本方法、概念；
- (2) 熟悉程序语言的逻辑结构与基础知识；
- (3) 选择一段程序语言。

1.3 实验内容

以保险产品投保为实例,针对保险产品投保业务逻辑代码进行分析,运用语句覆盖和判定覆盖法进行测试用例设计。

某个人税收优惠型保险产品 A/B1/B2/C 款承保规则:

(1) 凡 16 周岁以上且投保时未达法定退休年龄的(男性为 59 周岁、女性为 54 周岁,后续将随国家相关法规做相应调整),适用商业健康保险税收优惠政策的纳税人,可作为本合同的被保险人。保险公司根据被保险人是否参加公费医疗或基本医疗保险确定适用条款。

(2) 被保人为健康体,或者参加医疗保险的,可选择 A 款、B1 款或 B2 款。

(3) 未参加公费医疗的非健康体(有既往症)只能选择 C 款。

以下为个人税收优惠型保险产品承保的部分伪代码实现:

```
If (性别 = '男' and 16 < 年龄 < 59 or 性别 = '女' and 16 < age < 54)
{
    If (被保人健康属性为正常 or 有医疗保险)
    {可选择险种类型为 A 或 B1 或 B2 的险种、份数为 1}
Else
    {可选择险种类型为 C 的险种、份数为 1}
EndIf
}
```

```
Else  
{提示"不能承保"}  
EndIf
```

1.4 实验环境

- (1) 首先要让学生了解保险产品投保业务场景,能够模拟操作保险产品的承保流程;
- (2) 能够将业务场景与代码逻辑关系对应;
- (3) 根据代码画出程序流程图,并分析各判定节点;
- (4) 根据代码流程图分析出判定条件与真假取值。

1.5 实验过程简述

- (1) 明确被测试对象使用的测试方法;
- (2) 小组讨论业务场景并进行分析;
- (3) 测试实施工作安排;
- (4) 评审程序流程图和测试用例;
- (5) 执行测试,根据测试用例代入各条件测试数据,给出测试结果。

1.6 测试过程实施

1. 测试分析

- (1) 根据保险产品的承保业务描述,分析产品承保流程,包括主流程、分支流程以及正常流程、异常流程。
- (2) 模拟保险产品承保场景:触发允许产品承保的条件,不同条件是否走不同的承保流程。
- (3) 数据项检查:数据项的计算规则,数据项后台判断逻辑。

2. 测试设计

根据产品承保代码,设计出程序流程图,并对程序流程图做节点标记,分析图 1-1 所示的两个判定:

判定 A: (性别 = "男" AND 16 < 年龄 < 59) OR (性别 = "女" AND 16 < 年龄 < 54)
判定 B: 健康体 OR 有医疗保险

3. 测试设计

根据业务场景与流程逻辑判定,运用语句覆盖法进行用例设计。

语句覆盖是一个比较弱的逻辑覆盖标准,通过选择足够多的测试用例,使得被测试程序中的每个语句至少被执行一次。根据如图 1-1 所示的流程图,为使程序中的每个语句至少执行一次,只需设计两个测试用例,覆盖语句 A、B、C、E,即覆盖判定 A“成立”、判定 B“成立”或“不成立”各被覆盖一次,如表 1-1 所示。

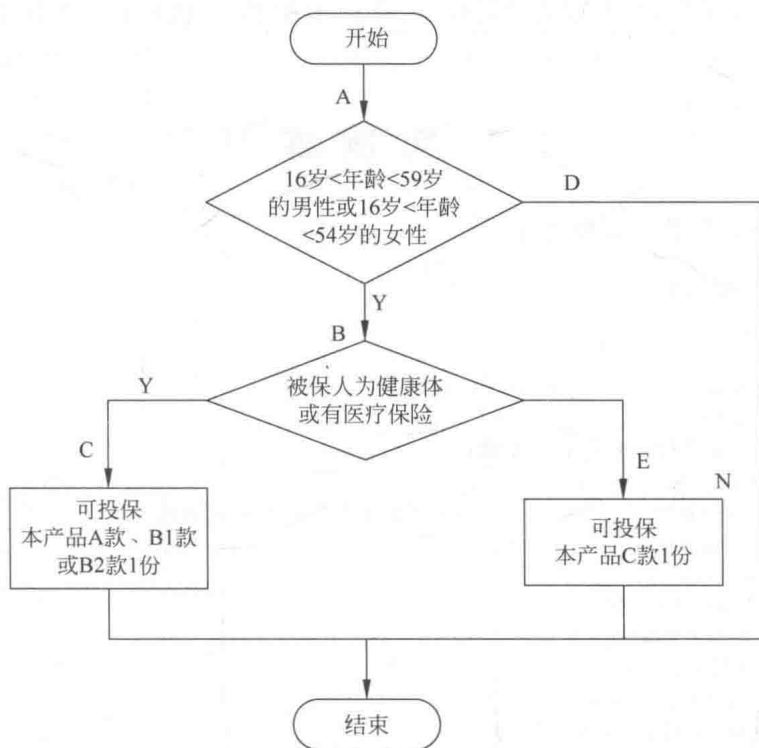


图 1-1 流程图

表 1-1 语句覆盖测试用例设计

测试用例名称	测试用例描述	测试路径
CASE1	投保成功：年龄 20,男性,健康体,有医疗保险	ABC
CASE2	投保成功：年龄 20,男性,非健康体且没有医疗保险	ABE

接下来我们运用判定覆盖法来进行用例设计。判定覆盖又称为分支覆盖，判定覆盖语句覆盖的标准稍强一些，它是指通过设计足够多的测试用例，使得被测试程序中的每个判定（即上述判定 A、判定 B）都获得一次“真”“假”值，如表 1-2 所示。

表 1-2 判定覆盖测试用例设计

测试用例名称	测试用例描述	覆盖判定
CASE3	投保成功：年龄 20,男性,健康体,有医疗保险	判定 A =“真”
		判定 B =“真”
CASE4	投保不成功：年龄 15,男性	判定 A =“假”
CASE5	投保成功：年龄 20,男性,健康体,没有医疗保险	判定 A =“真”
		判定 B =“假”

4. 测试结果分析

从本实验可看出，语句覆盖实际上是很弱的，CASE1、CASE2 可以满足语句覆盖，但如果第 2 个条件语句中 OR 写成了 AND，CASE1、CASE2 都不能发现它。

“判定覆盖”比“语句覆盖”严格，因为如果每个分支都执行过了，则每个语句也就执行过

了。但是,“判定覆盖”还是不够的,例如,CASE3~CASE5 未能检查 AB 分支中女性被保人的承保情况。

1.7 实例练习

(1) 程序实例,计算个人所得税。

```
#include <stdio.h>
int main ()
{
    double dSalary, dTax = 0, dNetIncome = 0;
    double dValue;
    printf("请输入您本月的收入总额(元): ");
    scanf("% lf", &dSalary);
    dValue = dSalary - 3500;          //在起征点基础上考虑纳税
    if(dValue>0.0)
    {
        if(dValue<= 1500)
            dTax = dValue * 0.03 - 0.0;
        else if(dValue<= 4500)
            dTax = dValue * 0.10 - 105.0;
        else if(dValue<= 9000)
            dTax = dValue * 0.20 - 555.0;
        else if(dValue<= 35000)
            dTax = dValue * 0.25 - 1005.0;
        else if(dValue<= 55000)
            dTax = dValue * 0.30 - 2755.0;
        else if(dValue<= 80000)
            dTax = dValue * 0.35 - 5505.0;
        else
            dTax = dValue * 0.45 - 13505.0;
    }
    dNetIncome = dSalary - dTax;
    printf("您本月应缴个人所得税 %.2lf 元,税后收入是 %.2lf 元.\n", dTax, dNetIncome);
    return 0;
}
```

(2) 请根据程序实例(表 1-3),设计语句和判定覆盖的测试案例。

表 1-3 条件分析

	条 件	下一步	产品 A 款	产品 B1 款	产品 B2 款	产品 C 款	不承保
A	(性别="男"并且 16<年龄<59)or (性别="女"并且 16<年龄<54)	B					
	年龄小于 16 的男性	F					●
	年龄小于 16 的女性	F					●
	年龄大于 59 的男性	F					●
	年龄大于 54 的女性	F					●

续表

	条 件	下一步	产品 A 款	产品 B1 款	产品 B2 款	产品 C 款	不承保
B	是健康体或者有医疗保险公费	D	1 份				
				1 份			
					1 份		
	有既往症且医疗保险没有	H				1 份	

实验 2

条件覆盖和条件组合覆盖测试设计

2.1 实验目的

- (1) 巩固所学的条件覆盖、条件组合覆盖测试方法；
- (2) 提高运用条件覆盖、条件组合覆盖法的能力。

2.2 实验前提

- (1) 掌握逻辑覆盖的基本方法、概念；
- (2) 熟悉程序语言的逻辑结构与基础知识；
- (3) 选择一段程序语言。

2.3 实验内容

以银行内部转账为实例,针对内部转账业务逻辑代码进行分析,运用条件覆盖进行测试用例设计。

内部转账用于处理发起户口号和接收户口号都是内部账户的系统内资金转账业务,主要用于财务资金的划拨、未实现自动清算业务的清算资金的划拨。

(1) 内部转账发起是指:发起行发出内部资金交易,并换人复核,满足条件时需会计主管授权。

(2) 内部转账接收是指:内部资金交易接收方根据接收方确认方式,对交易进行接收经办,满足条件的需复核或授权。

确定接收方的入账流程,“确认方式”分为以下三种:

(1) 不需接收方确认,即发起方发起后自动记发起方和接收方的一套账务,接收方无须再做接收动作。

(2) 需接收方确认,即接收方接收时不能更改接收信息,只能依据发起方输入的信息入账或退发起方。以目前的处理方式,接收经办→入账(金额小于 100 万元),大于 100 万元时为接收经办+接收授权→入账。

(3) 需接收方经办,即接收方接收时可以更改接收信息,执行入账或退发起行。以目前的处理方式,接收经办+接收复核→入账(金额小于 100 万元),大于 100 万元时为接收经办+接收复核+接收授权→入账。

内部转账权限控制如表 2-1 所示。

表 2-1 内部转账权限控制

操 作	条 件	经办	复核	授权
内部转账发起	100 万元以下	√	√	
	100 万元以上	√	√	√
内部转账接收	“确认方式”为“2”，100 万元以下	√		
	“确认方式”为“2”，100 万元以上	√		√
	“确认方式”为“3”，100 万元以下	√	√	
	“确认方式”为“3”，100 万元以上	√	√	√

以下为银行内部转账控制的部分伪代码实现：

```
If( 判定 1: 转账金额 > 100W) {
    调用"内部转账发起复核";
    调用"内部转账发起授权";
    If( 判定 3: "确认方式" == 1) {
        抛出异常"确认方式不符合业务流程"
    }
    Else If( 判定 3: "确认方式" == 2) {
        调用"内部转账接收经办";
        调用"内部转账接收授权";
        接收确认
    }
    Else If( 判定 3: "确认方式" == 3) {
        调用"内部转账接收经办";
        调用"内部转账接收复核";
        调用"内部转账接收授权";
        接收确认
    }
    Else {
        抛出异常"确认方式不符合业务流程"
    }
    End If
}
Else If (判定 1: 0 < 转账金额 <= 100W) {
    If( 判定 2: "确认方式" == 1) {
        调用"内部转账接收确认";
        接收确认
    }
    Else If( 判定 2: "确认方式" == 2) {
        调用"内部转账接收经办";
        调用"内部转账接收确认";
        接收确认
    }
    Else If( 判定 2: "确认方式" == 3) {
        调用"内部转账接收经办";
        调用"内部转账接收复核";
        调用"内部转账接收确认";
    }
}
```



```
        接收确认
    }
    Else {
        抛出异常"确认方式不符合业务流程"
    }
    End If
}
Else If (判定 1: 转账金额 <= 0) {
    抛出异常"输入金额有误,请重新输入"
}
End if
```

2.4 实验环境

- (1) 首先要让学生了解银行内部转账业务,能够模拟操作转账流程;
- (2) 能够将业务场景与代码逻辑关系对应;
- (3) 根据代码画出程序流程图,并分析各判定节点;
- (4) 根据代码流程图分析出条件覆盖、条件组合覆盖。

2.5 实验过程简述

- (1) 明确被测试对象使用的测试方法;
- (2) 小组讨论业务场景并进行分析;
- (3) 测试实施工作安排;
- (4) 评审程序流程图和测试用例;
- (5) 执行测试,根据测试用例代入各条件测试数据,给出测试结果。

2.6 实验过程实施

1. 测试分析

(1) 根据银行内部转账业务描述,分析内部转账流程,包括主流程、分支流程以及正常流程、异常流程。

(2) 模拟内部转账场景:触发内部转账的条件,不同条件是否走不同的转账流程。

(3) 数据项检查:数据项的计算规则,数据项后台判断逻辑。

2. 测试设计

根据内部转账业务需求,设计出程序流程图,如图 2-1 所示,并对程序流程图做节点标记,分析流程图的判定条件与结果。

3. 测试执行

根据业务场景与流程逻辑判定,运用条件覆盖法进行用例设计。

条件覆盖即设计足够多的测试用例,运行被测程序,使得每一判定语句中每个逻辑条件的可能取值至少满足一次。条件覆盖率的公式是:条件覆盖率=被评价到的条件取值的数