

普通高等教育“十三五”规划教材



服务外包产教融合系列教材

主编 迟云平 副主编 宁佳英

# 基于JUnit 单元测试应用技术

JIJU JUNIT DANYUAN CESHI YINGYONG JISHU

主 编 林若钦



华南理工大学出版社  
SOUTH CHINA UNIVERSITY OF TECHNOLOGY PRESS

“十三五”规划教材



服务外包产教融合系列教材

主编 迟云平 副主编 宁佳英

# 基于JUnit 单元测试应用技术

主 编 林若钦



华南理工大学出版社  
SOUTH CHINA UNIVERSITY OF TECHNOLOGY PRESS

· 广州 ·

## 图书在版编目(CIP)数据

基于 JUnit 单元测试应用技术/林若钦主编. —广州: 华南理工大学出版社, 2017. 9  
(服务外包产教融合系列教材/迟云平主编)  
ISBN 978 - 7 - 5623 - 5406 - 2

I. ①基… II. ①林… III. ①JAVA 语言 - 程序设计 - 教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2017)第 228286 号

## 基于 JUnit 单元测试应用技术

林若钦 主编

---

出 版 人: 卢家明

出版发行: 华南理工大学出版社

(广州五山华南理工大学 17 号楼, 邮编 510640)

<http://www.scutpress.com.cn> E-mail:scutc13@scut.edu.cn

营销部电话: 020 - 87113487 87111048 (传真)

总 策 划: 卢家明 潘宜玲

执行策划: 詹志青

责任编辑: 刘 锋 詹志青

印 刷 者: 佛山市浩文彩色印刷有限公司

开 本: 787mm × 1092mm 1/16 印张: 15 字数: 378 千

版 次: 2017 年 9 月第 1 版 2017 年 9 月第 1 次印刷

印 数: 1 ~ 1 000 册

定 价: 36.00 元

---

版权所有 盗版必究 印装差错 负责调换

# “服务外包产教融合系列教材”

## 编审委员会

顾问：曹文炼(国家发展和改革委员会国际合作中心主任，研究员、教授、博士生导师)

主任：何大进

副主任：徐元平 迟云平 徐 祥 孙维平 张高峰 康忠理

主编：迟云平

副主编：宁佳英

编委(按姓氏拼音排序)：

|     |     |     |     |     |     |     |
|-----|-----|-----|-----|-----|-----|-----|
| 蔡木生 | 曹陆军 | 陈翔磊 | 迟云平 | 杜 剑 | 高云雁 | 何大进 |
| 胡伟挺 | 胡治芳 | 黄小平 | 焦幸安 | 金 晖 | 康忠理 | 李俊琴 |
| 李舟明 | 廖唐勇 | 林若钦 | 刘洪舟 | 刘志伟 | 罗 林 | 马彩祝 |
| 聂 锋 | 宁佳英 | 孙维平 | 谭瑞枝 | 谭 湘 | 田晓燕 | 王传霞 |
| 王丽娜 | 王佩锋 | 吴伟生 | 吴宇驹 | 肖 雷 | 徐 祥 | 徐元平 |
| 杨清延 | 叶小艳 | 袁 志 | 曾思师 | 查俊峰 | 张高峰 | 张 芒 |
| 张文莉 | 张香玉 | 张 屹 | 周 化 | 周 伟 | 周 璇 | 宗建华 |

评审专家：

周树伟(广东省产业发展研究院)

孟 霖(广东省服务外包产业促进会)

黄燕玲(广东省服务外包产业促进会)

欧健维(广东省服务外包产业促进会)

梁 茹(广州服务外包行业协会)

刘劲松(广东新华南方软件外包有限公司)

王庆元(西艾软件开发有限公司)

迟洪涛(国家发展和改革委员会国际合作中心)

李 澍(国家发展和改革委员会国际合作中心)

总策划：卢家明 潘宜玲

执行策划：詹志青

# 总 序

发展服务外包，有利于提升我国服务业的技术水平、服务水平，推动出口贸易和服务业的国际化，促进国内现代服务业的发展。在国家和各地方政府的大力支持下，我国服务外包产业经过 10 年快速发展，规模日益扩大，领域逐步拓宽，已经成为中国经济新增长的新引擎、开放型经济的新亮点、结构优化的新标志、绿色共享发展的新动能、信息技术与制造业深度融合的新平台、高学历人才集聚的新产业，基于互联网、物联网、云计算、大数据等一系列新技术的新型商业模式应运而生，服务外包企业的国际竞争力不断提升，逐步进入国际产业链和价值链的高端。服务外包产业以极高的孵化、融合功能，助力我国航天服务、轨道交通、航运、医药、医疗、金融、智慧健康、云生态、智能制造、电商等众多领域的不断创新，通过重组价值链、优化资源配置降低了成本并增强了企业核心竞争力，更好地满足了国家“保增长、扩内需、调结构、促就业”的战略需要。

创新是服务外包发展的核心动力。我国传统产业转型升级，一定要通过新技术、新商业模式和新组织架构来实现，这为服务外包产业释放出更为广阔的发展空间。目前，“众包”方式已被普遍运用，以重塑传统的发包/接包关系，战略合作与协作网络平台作用凸显，从而促使服务外包行业人员的从业方式发生了显著变化，特别是中高端人才和专业人士更需要在人才共享平台上根据项目进行有效整合。从发展趋势看，服务外包企业未来的竞争将是资源整合能力的竞争，谁能最大限度地整合各类资源，谁就能在未来的竞争中脱颖而出。

广州大学华软软件学院是我国华南地区最早介入服务外包人才培养的高等院校，也是广东省和广州市首批认证的服务外包人才培养基地，还是我国

服务外包人才培养示范机构。该院历年毕业生进入服务外包企业从业平均比例高达66.3%以上,并且获得业界高度认同。常务副院长迟云平获评2015年度服务外包杰出贡献人物。该院组织了近百名具有丰富教学实践经验的一线教师,历时一年多,认真负责地编写了软件、网络、游戏、数码、管理、财务等专业的服务外包系列教材30余种,将对各行业发展具有引领作用的服务外包相关知识引入大学学历教育,着力培养学生对产业发展、技术创新、模式创新和产业融合发展的立体视角,同时具有一定的国际视野。

当前,我国正在大力推动“一带一路”建设和创新创业教育。广州大学华软软件学院抓住这一历史性机遇,与国家发展和改革委员会国际合作中心合作成立创新创业学院和服务外包研究院,共建国际合作示范院校。这充分反映了华软软件学院领导层对教育与产业结合的深刻把握,对人才培养与产业促进的高度理解,并愿意不遗余力地付出。我相信这样一套探讨服务外包产教融合的系列教材,一定会受到相关政策制定者和学术研究者的欢迎与重视。

借此,谨祝愿广州大学华软软件学院在国际化服务外包人才培养的路上越走越好!

国家发展和改革委员会国际合作中心主任

2017年1月25日于北京

# 前 言

全球软件外包市场规模已达上千亿美元，中国 IT 外包服务市场一直高速增长。软件服务外包业是以技术密集程度高、技术进步速度快为显著特点的产业。软件外包为中国软件业带来的不仅仅是经济发展的机会，还有先进的软件开发管理流程，以及严格的软件质量控制体系。通过发展软件外包产业，我国的软件产业将逐渐告别手工作坊式的时代，进入工程化、规模化的开发领域。随着敏捷开发方法的普及，测试驱动开发、基于代码单元的软件测试越来越受到重视。

单元测试(unit testing)，是指对软件中的最小可测试单元进行检查和验证。对于单元测试中单元的含义，一般要根据实际情况去判定，如 C 语言中单元指一个函数，Java 里单元指一个类，图形化的软件中可以指一个窗口或一个菜单等。总之，单元就是人为规定的最小的被测功能模块。单元测试是在软件开发过程中要进行的最低级别的测试活动。软件的独立单元将在与程序的其他部分相隔离的情况下进行测试。

单元测试活动包括代码走读(code review)、静态分析(static analysis)和动态分析(dynamic analysis)。静态分析就是对软件的源代码进行研读，查找错误或收集一些度量数据，并不需要对代码进行编译和执行。动态分析则通过观察软件运行时的动作，提供执行跟踪、时间分析以及测试覆盖度方面的信息。

随着一些新的开发方法(如敏捷开发)的普及和应用，软件外包行业对软件质量的要求越来越高，单元测试越来越受到业界的重视。本书主要介绍单元测试的概念，以及一些常用的框架和使用的方法。包括五个部分：

第一部分：单元测试基础。包含第 1 章软件外包与软件测试；第 2 章



JUnit；第3章测试覆盖率。

第二部分：单元测试策略。包含第4章 Stub 与 Mock Object 技术；第5章 EasyMock 与 Jmock 的使用。

第三部分：构建工具的使用。包含第6章 Ant 的使用；第7章 Maven 的使用。

第四部分：单元测试的扩展应用。包含第8章服务器端应用测试；第9章数据库访问测试。

第五部分：商业测试工具应用。包含第10章商业单元测试工具的使用。

本书由实际教学案例整合编写而成，其中借鉴或参考了业界同行、专家的意见或方法。感谢软件测试方向组老师的支持。由于时间和编者学识所限，书中不足之处敬请诸位同行、专家和读者指正。

编 者

2016年11月

# 目 录

|                                  |    |
|----------------------------------|----|
| 1 软件外包与软件测试 .....                | 1  |
| 1.1 软件外包概述 .....                 | 1  |
| 1.2 软件质量控制 .....                 | 2  |
| 1.3 敏捷开发与测试 .....                | 4  |
| 1.4 单元测试 .....                   | 5  |
| 2 JUnit .....                    | 13 |
| 2.1 JUnit 简介 .....               | 13 |
| 2.2 用 JUnit 编写测试代码 .....         | 15 |
| 2.3 用 JUnit 编写测试套件 .....         | 22 |
| 2.4 参数化测试运行器 .....               | 24 |
| 2.5 异常测试 .....                   | 25 |
| 2.6 Hamcrest .....               | 28 |
| 3 测试覆盖率 .....                    | 33 |
| 3.1 覆盖率简介 .....                  | 33 |
| 3.2 代码覆盖率的分类及测试目的 .....          | 34 |
| 3.3 代码覆盖率工具的使用 .....             | 37 |
| 4 Stub 与 Mock Object 技术 .....    | 46 |
| 4.1 使用 Stub 进行粗粒度测试 .....        | 46 |
| 4.2 使用 Mock Object 进行细粒度测试 ..... | 58 |
| 5 EasyMock 与 Jmock 的使用 .....     | 63 |
| 5.1 EasyMock 的使用 .....           | 63 |
| 5.2 JMock 的使用 .....              | 70 |
| 6 Ant 的使用 .....                  | 82 |
| 6.1 Ant 简介 .....                 | 82 |
| 6.2 Ant 的安装与配置 .....             | 82 |
| 6.3 Ant 命令介绍 .....               | 84 |
| 6.4 Ant 目标、项目、属性以及任务 .....       | 85 |
| 6.5 Ant 和 Eclipse 集成 .....       | 97 |
| 6.6 从 Ant 中运行 JUnit 测试 .....     | 99 |



|                                         |     |
|-----------------------------------------|-----|
| 6.7 Ivy 的使用 .....                       | 101 |
| 7 Maven 的使用 .....                       | 104 |
| 7.1 Maven 简介 .....                      | 104 |
| 7.2 Maven 的设计理念 .....                   | 105 |
| 7.3 Maven 的生命周期 .....                   | 108 |
| 7.4 Maven 命令 .....                      | 111 |
| 7.5 Maven 仓库 .....                      | 114 |
| 7.6 settings.xml 配置文件详解 .....           | 120 |
| 7.7 使用 Maven 进行 JUnit 测试 .....          | 128 |
| 8 服务器端应用测试 .....                        | 138 |
| 8.1 Cactus 简介 .....                     | 138 |
| 8.2 用 Cactus 进行测试 .....                 | 141 |
| 9 数据库访问测试 .....                         | 161 |
| 9.1 隔离数据库测试业务逻辑 .....                   | 161 |
| 9.2 HSQLDB 数据库 .....                    | 170 |
| 9.3 DbUnit .....                        | 180 |
| 10 商业单元测试工具的使用 .....                    | 190 |
| 10.1 Jtest 的介绍 .....                    | 190 |
| 10.2 Jtest 的静态测试 .....                  | 194 |
| 10.3 使用 Jest RuleWizard 自定义代码检测规则 ..... | 206 |
| 10.4 BugDetective 静态代码分析 .....          | 218 |
| 10.5 Jtest 自动化动态测试 .....                | 222 |

# 1 软件外包与软件测试

## 1.1 软件外包概述

软件外包是指一些发达国家的软件公司将一些非核心的软件项目通过外包的形式交给人力资源成本相对较低国家的公司开发，以达到降低软件开发成本的目的。众所周知，软件开发的成本中 70% 是人力资源成本，所以，降低人力资源成本将有效地降低软件开发的成本。

中国产业调研网发布的 2016—2020 年《中国软件外包行业现状调研分析与发展趋势预测报告》认为，我国软件外包业目前依然呈高度分散的格局，外包市场集中度较低，外包企业相对规模较小，缺少与世界顶级企业规模相媲美的大型企业，也尚未出现一家服务外包收入超过 10 亿美元的国内企业，这导致我国不能有效地获取大型外包和集成项目。其次，由于我国软件外包中 60% 是对日外包，且以低成本、低利润率取胜，企业主营盈利率都不高。近几年，我国软件外包企业一直试图向回报率较高的欧美市场扩张，突破印度外包企业在这些市场的主导地位，但效果一直不佳。

我国软件业自主创新能力不太强，核心技术受控于美、日等大国，使国内外包软件企业要从“中国制造”到“中国创造”的转变面临较大压力。另外，受工资成本大幅提升、人民币升值、欧美经济发展趋缓等影响，我国软件外包业务出现成本上升、利润下降的趋势。人力成本近几年成为我国服务外包企业最大的支出，已占营业收入的 60%~70%。

软件外包企业主要集中于北京、南京、上海、深圳等大城市。这几个地区的共同特点是拥有良好的城市基础设施建设与产业配套基础，拥有当地政府在政策上的大力支持、良好的市场竞争环境、一大批通晓外语的软件人才，具备较强的创新能力，软件企业在此形成了群体优势，并已形成了较为完整的软件产业链。近年来国家在促进软件出口方面的扶持力度有所增强。经国家发改委、商务部和信息产业部批准，建立了北京、深圳、上海、南京等国家软件出口基地，创造了良好的政策、人才、技术、资金、市场和出口条件，充分发挥集聚效应和规模优势，形成了以国家软件出口基地中的国际化软件企业为龙头，辐射周边地区，带动全国软件出口的产业格局。

## 1.2 软件质量控制

软件外包将为中国软件业带来的，不仅仅是经济发展的机会，还有先进的软件开发管理流程，以及严格的软件质量控制体系。通过发展软件外包产业，中国的软件产业将逐渐告别手工作坊式的时代，进入工程化、规模化的开发领域。软件质量控制则成为软件外包一个核心的竞争力。

### 1.2.1 软件外包与质量管理

质量管理在软件外包中极其重要。软件质量是软件企业的生命，软件质量是我国软件企业进入国际外包市场的前提条件和“通行证”。在软件外包中软件质量管理主要是对承包方质量进行管理，主要包括：

- (1) 承包方对质量、进度、成本控制规划的能力。
- (2) 承包方业务能力、交流能力、承包渠道、商业信誉。
- (3) 要求承包方建立软件质量保证(SQA)组织，如图 1-1 所示。

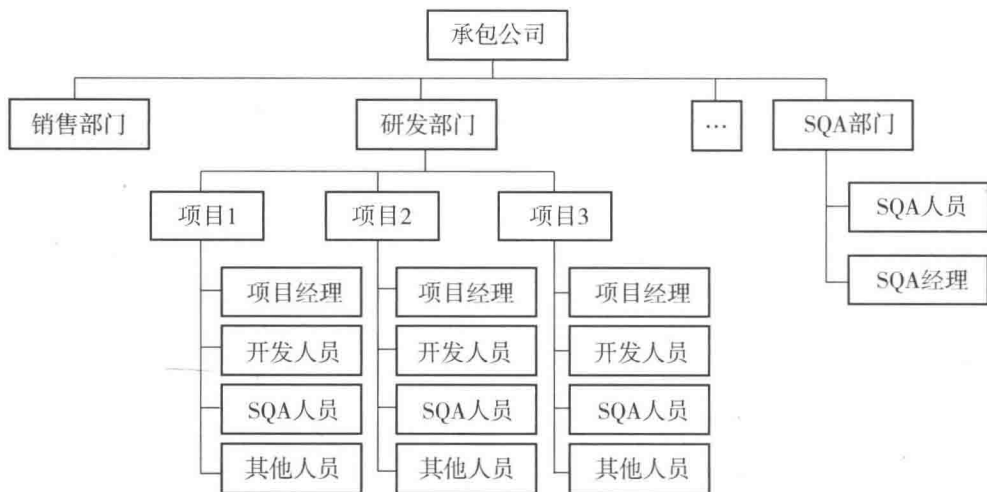


图 1-1 SQA 的组织结构

- (4) 要求承包方建立有效的 SQA 流程，如图 1-2 所示。
- (5) 加强软件质量控制(SQC)工作，要求通过软件测试来实现 SQC。
- (6) 建立健全的文档体系，追求文档的完整性、一致性、连续性、及时性。

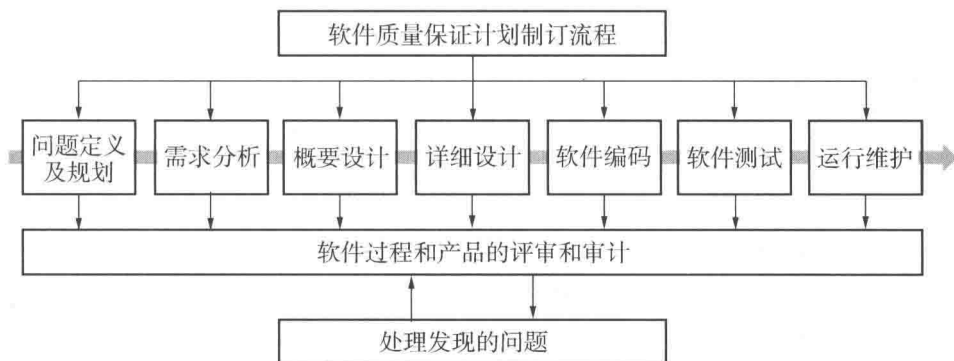


图 1-2 全生命周期的离岸外包软件的 SQA 流程

### 1.2.2 软件质量控制

软件外包的质量指软件产品满足用户需求的程度，包括功能需求、性能需求、稳定性、安全性和技术先进性需求、支持和服务需求等。达到客户的质量要求是外包业务的基本要求。以软件外包开发项目为例，接包方质量控制的常用方法如下：

按软件生命周期，每个阶段结束时实行质量控制。

(1)需求分析：审查需求规格说明书。

需求分析是软件项目正式实施开始的第一个阶段，需求分析应该遵循可行性分析确定的基调，包括技术路线、产品基本功能、产品运行环境及市场定位。需求分析主要应完成对用户应用流程的描述，即完成商业逻辑分析。并根据商业逻辑的需要确定软件的功能列表及描述。

(2)概要设计：审查软件的结构。

概要设计是一个设计师根据用户交互过程和用户需求形成交互框架和视觉框架的过程，其结果往往以反映交互控件布置、界面元素分组以及界面整体版式的页面框架图的形式来呈现。这是一个在用户研究和设计之间架起桥梁，使用户研究和设计无缝结合，将用户目标与需求转换成具体界面设计解决方案的重要阶段。审查把需求分析得到的系统扩展用例图转换为软件结构和数据结构。

(3)详细设计：审查模块内部的数据结构、算法和接口。

详细设计是总体设计的继续，主要目的是完成总体设计对象内部的商业逻辑的实现设计，在总体设计完成后可以将不同的设计对象交由不同的设计人员来完成。原则上，在开始软件编码之前应完成所有的设计细节，避免在编码中进行设计工作。详细设计是编码及软件模块测试的基础。

(4)编码及代码测试。

编码是软件详细设计的一种再现，编码中最重要的还是要遵从相关开发工具的设计规范及数据库设计规范。另外，养成良好的编程习惯是一个软件公司和软件编程人员最基

本的职业素质。对于软件应用可靠性要求严格的应用软件，所有软件模块必须通过模块测试，对一般应用软件中的重要模块也应进行模块测试。

(5)集成测试：模块间的集成、处理流程、接口。

集成测试是一种正规测试过程，必须精心计划，并与单元测试的完成时间协调起来。在制定测试计划时列出各个模块的编制、测试计划表，标明每个模块单元测试完成的日期、首次集成测试的日期、集成测试全部完成的日期以及需要的测试用例和所期望的测试结果。集成测试是在单元测试的基础上，测试在将所有的软件单元按照概要设计规格说明的要求组装成模块、子系统或系统的过程中各部分工作是否达到或实现相应技术指标及要求的活动。换言之，在集成测试之前，单元测试应该已经完成，集成测试中所使用的对象应该是已经过单元测试的软件单元。这一点很重要，因为如果不经单元测试，那么集成测试的效果将会受到很大影响，并且会大幅增加软件单元代码纠错的代价。

(6)确认测试：系统测试，根据验收要求测试。

确认测试的目的是向未来的用户表明系统能够像预定要求的那样工作。经集成测试后，已经按照设计把所有的模块组装成一个完整的软件系统，接口错误也已经基本排除，接着就应该进一步验证软件的有效性，这就是确认测试的任务，即软件的功能和性能如同用户所合理期待的那样。

## 1.3 敏捷开发与测试

随着软件外包市场的竞争越来越激烈，大量的软件公司采用了敏捷开发方法。敏捷开发方法以用户的需求进化为核心，采用迭代、循序渐进的方法进行软件开发。在敏捷开发中，软件项目在构建初期被切分成多个子项目，各个子项目的成果都经过测试，具备可视、可集成和可运行使用的特征。换言之，就是把一个大项目分为多个相互联系但也可独立运行的小项目，并分别完成，在此过程中软件一直处于可使用状态。

敏捷开发是针对传统的瀑布开发模式的弊端，而产生的一种新的开发模式，目标是提高开发效率和响应能力。在建立模型时，就要不断问该如何测试它，如果没办法测试正在开发的软件，就根本不应该开发它。在现代的各种软件开发过程中，测试和质保（quality assurance）活动都贯穿于整个项目生命周期，一些过程更是提出了“在编写软件之前先编写测试”的概念（这是XP的一项实践：测试优先）。

在敏捷开发中，测试为开发过程的一部分，敏捷开发提倡不同层次的自动化测试。图1-3所示为测试金字塔。单元测试/组件测试在金字塔的最下层，是分层自动化测试的基石，成本最低，缺陷容易定位。

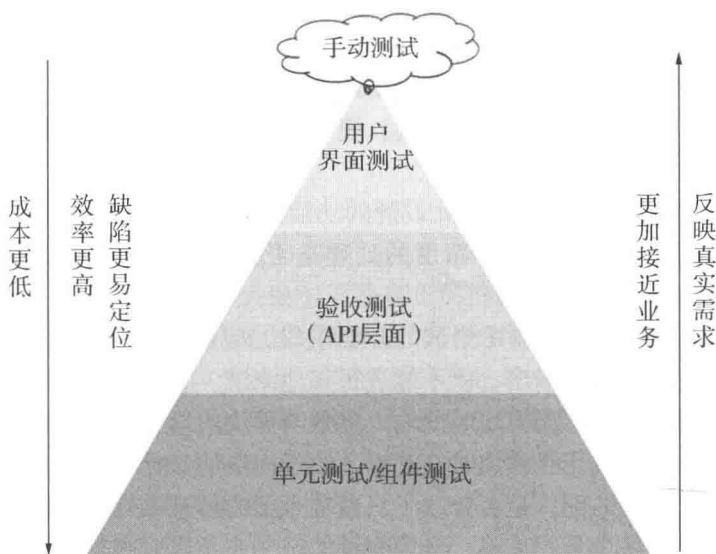


图 1-3 测试金字塔

## 1.4 单元测试

单元测试是对软件基本组成单元/模块进行的测试，又称为模块测试。基本单元/模块可以是函数、类实例、方法、存储过程，也可以是任何具有明确功能、规格定义、明确接口定义并且其规模一般比较小的程序代码模块的组合物。单元测试的重点在于发现程序设计或实现的逻辑错误，使问题及早暴露，便于问题的定位解决。

### 1.4.1 单元测试概述

单元测试(模块测试)是编写的一小段代码，用于检验被测代码的一个很小的、很明确的功能是否正确。通常而言，一个单元测试是用于判断某个特定条件(或者场景)下某个特定函数的行为。

单元测试中多采用白盒测试和黑盒测试相结合的方法，既关注单元功能，也关注程序模块的逻辑结构。两者结合起来，既可以避免由于过多关注路径而导致测试工作量很大的问题，又可以避免因从外部设计测试用例而可能丢失一些路径的问题。单元测试的重点应该以功能测试为主，同时统计测试的覆盖率，并且测试模块的输入/输出接口、内部的数据流是否正确等。

功能测试主要测试模块是否正确实现了设计要求的功能，以及有无遗漏的功能。这里有一个功能覆盖的概念。因为被测试模块可能包括多个功能点，在做测试时，设计测试用例要覆盖这些功能点，以保证这些功能点能通过测试。一般要求功能 100% 覆盖。覆盖率一般包括语句覆盖率和分支覆盖率(又称判断覆盖率)，同时要求测试所有的关

键路径。关键路径的表达不是很明确,但是如果要求做基本路径集覆盖测试,即使是一个比较小的模块也是很难做到的。如果使用单元测试工具,则可以统计覆盖率。单元测试结束后,如果有些分支无法覆盖,则需要给出原因说明。

单元测试的内容包括单元功能测试、模块接口测试、数据流测试、逻辑路径测试等。

(1)单元功能测试的内容:①单元设计的功能点是否全部实现。②运算的优先级和先后执行顺序是否正确。③计算中精度的处理是否正确。④计算中的误差是否会无限放大。

(2)模块接口测试:①在单元测试的开始阶段,应对所测模块的数据流进行测试。如果数据不能正确地输入和输出,就不能进行其他测试。②对模块接口可能需要进行下面的测试外包项目:调用所测模块时的输入参数与模块的形式参数在个数、属性、顺序上是否匹配;所测模块调用子模块时,它输入给子模块的参数与子模块的形式参数在个数、属性、顺序上是否匹配;是否修改了只做输入用的形式参数;输出给标准函数的参数在个数、属性、顺序上是否正确;全局变量的定义在各模块中是否一致。

(3)限制是否通过形式参数来传送。模块对外部文件、数据库进行输入/输出时,必须对文件操作进行测试。例如,缓冲区的大小、是否在读写文件前打开文件,在结束前关闭文件等。

(4)内部数据流测试包括以下几点:不正确或不一致的数据类型说明;使用尚未赋值或尚未初始化的变量;错误的初始值或错误的默认值;变量名拼写错误或书写错误;不一致的数据类型;全局变量对模块是否产生影响。

(5)逻辑路径测试包括以下几点:是否到达重要的功能点路径;逻辑判断的边界点是否正确;异常/错误处理。

比较完善的模块设计要求能预见异常或出错的条件,并设置适当的异常处理和出错处理机制,以便在程序出现异常或错误时,能对出错程序重新安排,保证逻辑上的正确性。重点应该考虑下面几个问题:①异常或出错的描述是否可以理解;②异常处理是否合理、出错后对错误的定位是否准确;③提示的错误与实际的错误是否一致;④对错误条件的处理是否正确。

以上提到的很多内容在编码规范或代码检查单中大多已经包含,如果模块在进入单元测试之前,已经进行了桌前检查(desk checking)和同行评审,则模块中潜在的缺陷就可能比较少。

单元测试过程包括计划、设计、执行、评审等几个步骤,分述如下:

- 计划:确定测试需求,制订测试策略,确定测试所用资源(如人员、设备等),创建测试任务的时间表。这部分工作可以简单描述。

- 设计:设计单元测试模型,制订测试方案,制订具体的测试用例,创建可重用的测试脚本。

- 执行:执行测试用例,对单元模块进行测试,验证测试的结果并记录测试过程中出现的缺陷。

- 评审:对单元测试的结果进行评审,主要进行测试完备性评估。

由于单元模块往往不是一个独立的程序,在设计时,要考虑单元模块与其他模块的联系,用桩模块和驱动模块模拟所测模块相联系的其他模块。由被测试模块、驱动模块和桩模块共同构成可运行的程序。

### 1.4.2 单元测试用例设计

模块单元设计完毕,下一个开发阶段就是设计单元测试。值得注意的是,如果在书写代码之前设计测试,测试设计就会显得更加灵活。一旦代码完成,对软件的测试可能会倾向于测试该段代码在做什么(这根本不是真正的测试),而不是测试其应该做什么。单元测试说明实际上由一系列单元测试用例组成,每个测试用例应该包含4个关键元素。

- 被测单元模块初始状态声明,即测试用例的开始状态。
- 被测单元的输入,包含由被测单元读入的任何外部数据值。
- 该测试用例实际测试的代码,用被测单元的功能和测试用例设计中使用的分析来说明,如:单元中哪一个决策条件被测试。

- 测试用例的期望输出结果总是应该在测试进行之前在测试说明中定义。

进行测试用例设计有以下几个过程。

#### 1. 运行被测单元

任何单元测试说明的第一个测试用例应该是以一种可能的简单方法执行被测单元。看到被测单元第一个测试用例运行成功可以增强人的自信心。如果不能正确执行,最好选择一个尽可能简单的输入对被测单元进行测试/调试。

这个阶段适合的技术有:

- 模块设计导出的测试;
- 对等区间划分。

#### 2. 正面测试(positive testing)

正面测试的测试用例用于验证被测单元能够执行应该完成的工作。测试设计者应该查阅相关的设计说明;每个测试用例应该测试模块设计说明中一项或多项陈述。如果涉及多个设计说明,最好使测试用例的序列对应一个模块单元的主设计说明。

适合的技术:

- 设计说明导出的测试;
- 对等区间划分;
- 状态转换测试。

#### 3. 负面测试(negative testing)

负面测试用于验证软件是否执行其不应完成的工作。这一步骤主要依赖于错误猜测,需要依靠测试设计者的经验判断可能出现问题的位置。

适合的技术有:

- 错误猜测;
- 边界值分析;
- 内部边界值测试;
- 状态转换测试。