

**软件测试课程设计报告**

|  |  |
| --- | --- |
| 学 号： | **20171001561** |
| 班级序号： | **01** |
| 学生姓名： | **周麟** |
| 指导教师： | **张剑波** |

**中国地质大学信息工程学院软件工程系**

**2021年 3 月**

# 1、动态测试技术

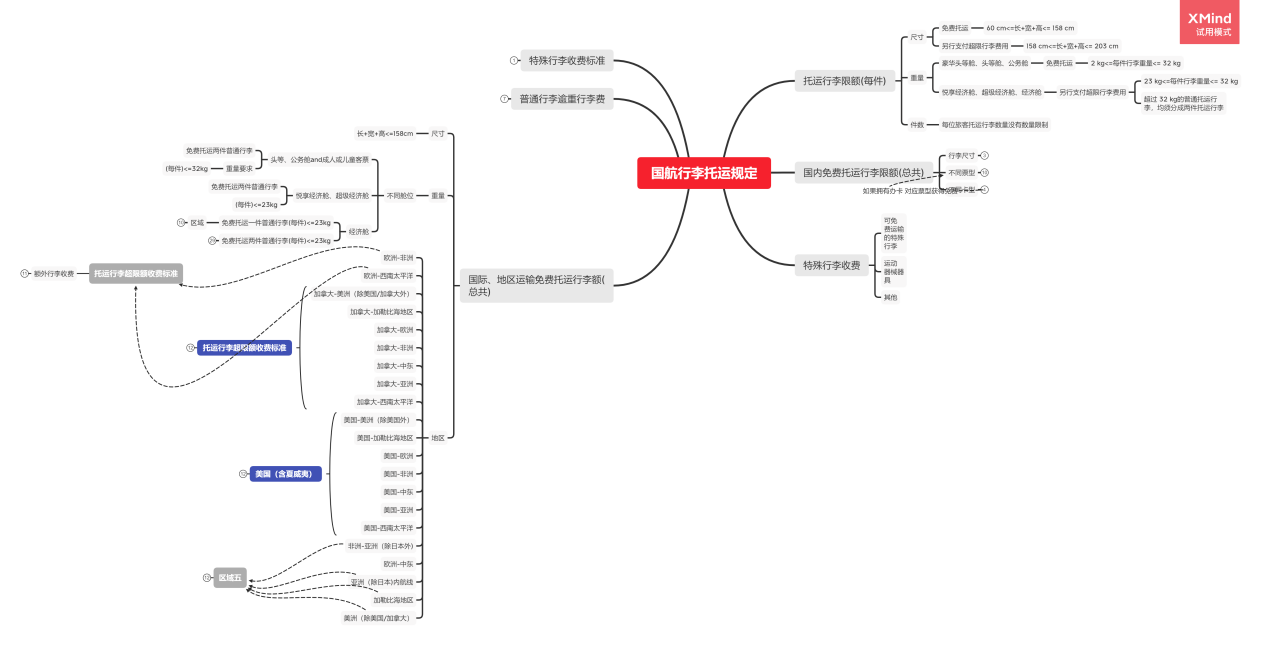
## 1. 软件说明

**程序名称**：《南航国内/国际航班行李托运计算程序》

**程序介绍**：本程序可以根据你携带的行李数量，行李的尺寸和重量，座舱类型，持票类型，旅客类型，航班类型，持卡类型来计算是否需要支付托运行李费用，如果需要支付那么员工需要支付多少行李托运费用，国际、地区航班用不同面值货币显示结果。

**思维导图：（全图见附件）**

**缩略图：**

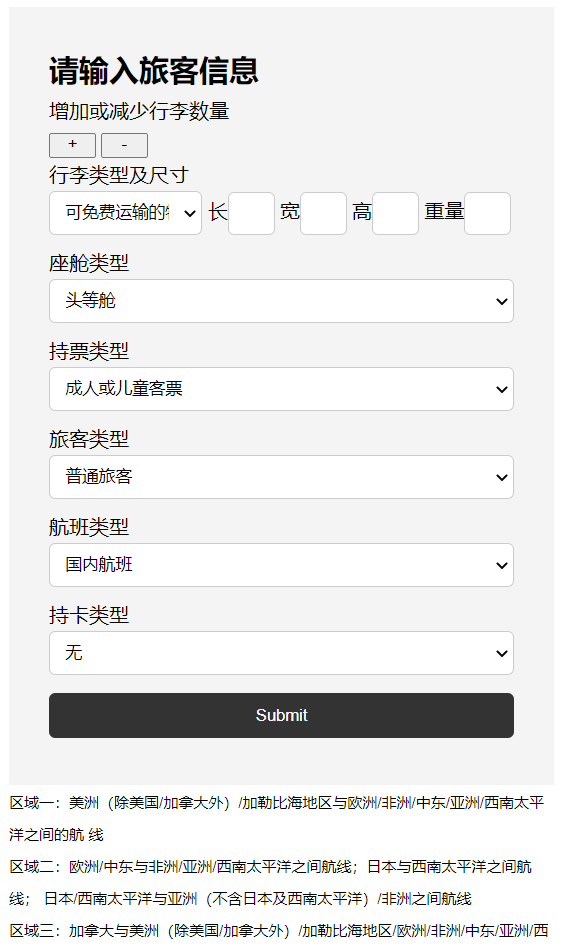
****

**程序说明**：本行李托运计算程序的业务处理和前端显示采用js+html+css实现，测试框架使用jest。

**程序结构：**

|  |  |
| --- | --- |
| Index.html | 前端可视化 |
| Style.css | 前端样式表 |
| Main.js | 逻辑代码 |
| Index.test.js | 单元测试代码 |

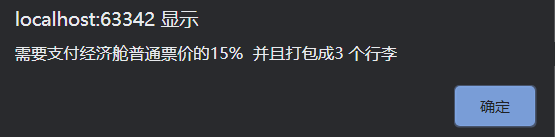
**主界面视图：**



可以根据行李件数增加和减少，行李参数填写完成之后单击submit，如果是国内航班，会显示应该支付多少经济舱普通票价的百分数，并且还有应该打包的件数。如果是国外、地区航班，就会给出不同地区以不同货币应该支付多少行李托运费。

**计算结果：**

任何结果都用alert方式弹出



**测试部分：**

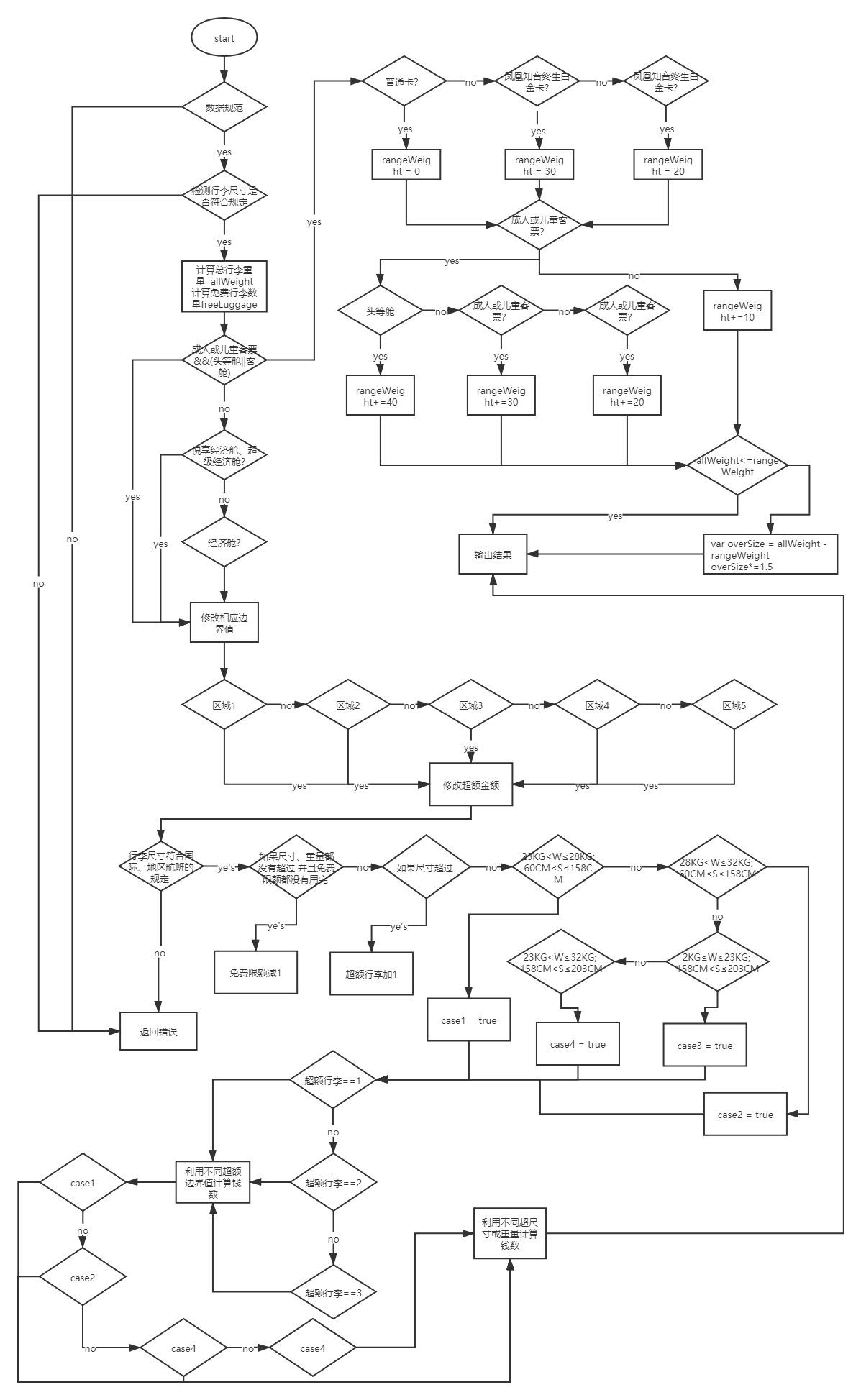
测试部分用jest进行单元测试，并且用指令npx jest --coverage生成覆盖率报告。

**如何运行：**

略

## 2. 测试内容

测试用例根据下列程序运行流程图设计：



## 3. 测试用例设计说明

### 3.1 用例1

### 3.1.1 控制

Kp取空值

### 3.1.2 输入

VluggageList = [‘’,20,20,20,20]

seatIndex = 0

ticketIndex = 0

personIndex = 0

flyIndex = 0

whatGoByIndex = 0

whatAreaIndex = 0

cardIndex = 0

### 3.1.3 输出

return = -1

### 3.1.4 过程

Kp不能取空值

### 3.2 用例2

### 3.2.1 控制

Kp取[0,1,2,3]以外的值

### 3.2.2 输入

VluggageList = [4,20,20,20,20]

seatIndex = 0

ticketIndex = 0

personIndex = 0

flyIndex = 0

whatGoByIndex = 0

whatAreaIndex = 0

cardIndex = 0

### 3.2.3 输出

return = -1

### 3.2.4 过程

Kp不能取[0,1,2,3]以外的值

### 3.3 用例3

### 3.3.1 控制

长取空值

### 3.3.2 输入

VluggageList = [0,’’,20,20,20]

seatIndex = 0

ticketIndex = 0

personIndex = 0

flyIndex = 0

whatGoByIndex = 0

whatAreaIndex = 0

cardIndex = 0

### 3.3.3 输出

return = -1

### 3.3.4 过程

长不能取空值

### 3.4 用例4

### 3.4.1 控制

宽取空值

### 3.4.2 输入

VluggageList = [0,20,’’,20,20]

seatIndex = 0

ticketIndex = 0

personIndex = 0

flyIndex = 0

whatGoByIndex = 0

whatAreaIndex = 0

cardIndex = 0

### 3.4.3 输出

return = -1

### 3.4.4 过程

宽不能取空值

### 3.5 用例5

### 3.5.1 控制

高取空值

### 3.5.2 输入

VluggageList = [0,20,20,’’,20]

seatIndex = 0

ticketIndex = 0

personIndex = 0

flyIndex = 0

whatGoByIndex = 0

whatAreaIndex = 0

cardIndex = 0

### 3.5.3 输出

return = -1

### 3.5.4 过程

高不能取空值

### 3.6 用例6

### 3.6.1 控制

重量取空值

### 3.6.2 输入

VluggageList = [0,20,20,20,’’]

seatIndex = 0

ticketIndex = 0

personIndex = 0

flyIndex = 0

whatGoByIndex = 0

whatAreaIndex = 0

cardIndex = 0

### 3.6.3 输出

return = -1

### 3.6.4 过程

重量不能取空值

### 3.7 用例7

### 3.7.1 控制

Kp取值非数字

### 3.7.2 输入

VluggageList = [‘hhh’,20,20,20,20]

seatIndex = 0

ticketIndex = 0

personIndex = 0

flyIndex = 0

whatGoByIndex = 0

whatAreaIndex = 0

cardIndex = 0

### 3.7.3 输出

return = -1

### 3.7.4 过程

Pk不能取非数字

### 3.8 用例8

### 3.8.1 控制

长取非数字

### 3.8.2 输入

VluggageList = [20,’hhh’,20,20,20]

seatIndex = 0

ticketIndex = 0

personIndex = 0

flyIndex = 0

whatGoByIndex = 0

whatAreaIndex = 0

cardIndex = 0

### 3.8.3 输出

return = -1

### 3.8.4 过程

长不能取非数字

### 3.10 用例10

### 3.10.1 控制

高取非数字

### 3.10.2 输入

VluggageList = [20,20,20,’hhh’,20]

seatIndex = 0

ticketIndex = 0

personIndex = 0

flyIndex = 0

whatGoByIndex = 0

whatAreaIndex = 0

cardIndex = 0

### 3.10.3 输出

return = -1

### 3.10.4 过程

高不能取非数字

### 3.11 用例11

### 3.11.1 控制

重量取非数字

### 3.11.2 输入

VluggageList = [20,20,20,20,’hhh’]

seatIndex = 0

ticketIndex = 0

personIndex = 0

flyIndex = 0

whatGoByIndex = 0

whatAreaIndex = 0

cardIndex = 0

### 3.11.3 输出

return = -1

### 3.11.4 过程

高不能取非数字

## 测试用例表

4.1测试说明

在进行测试时，需要传入luggageList,seatIndex,ticketIndex,personIndex,flyIndex,whatGoByIndex,whatAreaIndex,cardIndex八个参数。

luggageList是一个保存着行李类对象的数组，类属性有kp(行李类型),long(行李长度),width(行李宽度),length(行李高度),weight(行李重量)。必须保持长+宽+高<=203;长<= 100 宽<= 60;高<= 40;

seatIndex 座舱类型 取值：0（头等舱）、1（公务舱）、2（悦享经济舱/超级经济舱）、3（经济舱）

ticketIndex 持票类型 取值：0（成人或儿童客票）、1（婴儿客票）

personIndex 乘客类型 取值：0（普通旅客）、1（残疾、伤、病旅客）

flyIndex 航班类型 取值：0（国内航班）、1（国际、地区航班）

whatGoByIndex 途径地区 取值：0（夏威夷、欧洲、亚洲、非洲）、1（美洲、加勒比地区、中东、日本、巴基斯坦、新加坡、哈萨克斯坦）

whatAreaIndex 区域类型 取值：0（区域1）、1（区域2）、2（区域3）、3（区域4）、4（区域5）

cardIndex 办卡类型 取值：0（无办卡）、1（凤凰知音终生白金卡）、2（凤凰知音金卡、银卡）、3（星空联盟金卡）

**表1-1 等价类划分表**

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 输入等价类 | 有效等价类 | 无效等价类 |
| 1 | luggageList | 1. kp = 0 2. Kp = 1 3. Kp = 2 4. Kp = 3 5. 长+宽+高<=203 6. 长<=100 7. 宽<=60 8. 高<=40 | 1. kp取[0,1,2,3]以外的值 2. Kp空值 3. 长取空值 4. 宽取空值 5. 高取空值 6. 重量取空值 7. Kp取值非数字 8. 长取非数字 9. 宽取非数字 10. 高取非数字 11. 重量取非数字 12. 长>100 13. 宽>60 14. 高>40 |
| 2 | seatIndex | 1. seatIndex = 0 2. seatIndex = 1 3. seatIndex = 2 4. seatIndex = 3 | 27.不输入  28.非数字  29.[0,1,2,3]以外的数字 |
| 3 | ticketIndex | 30.ticketIndex =0  31.ticketIndex =1 | 32.不输入  33.[0,1]以外的数字  34.非数字 |
| 4 | personIndex | 35.personIndex=0  36.personIndex=1 | 37.不输入  38.[0,1]以外的数字  39.非数字 |
| 5 | flyIndex | 40.flyIndex =0  41.flyIndex =1 | 42.不输入  43.[0,1]以外的数字  44.非数字  45.flyIndex =1,whatGoByIndex不输入  46. flyIndex =1,whatGoByIndex为[0,1,2]以外的数字  47. flyIndex =1,whatGoByIndex非数字  48. flyIndex =1,whatAreaIndex不输入  49.flyIndex =1,whatAreaIndex为[0,1,2,3,4]以外的数字  50.flyIndex =1,whatAreaIndex非数字 |
| 8 | cardIndex | 51.cardIndex =0  52.cardIndex =1  53.cardIndex =2  54.cardIndex =3 | 55.不输入  56.[0,1,2,3]以外的数字  57.非数字 |

**表1-2 测试用例设计表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | **用例操作描述** | 对应测试用例 | 期望结果 | 覆盖等价类 |
|  | 设kp为空值 | 测试用例1 | Return -1 | 10 |
|  | Kp取[0,1,2,3]以外的值 | 测试用例2 | Return -1 | 9 |
|  | 长取空值 | 测试用例3 | Return -1 | 11 |
|  | 宽取空值 | 测试用例4 | Return -1 | 12 |
|  | 高取空值 | 测试用例5 | Return -1 | 13 |
|  | 重量取空值 | 测试用例6 | Return -1 | 14 |
|  | Kp取值非数字 | 测试用例7 | Return -1 | 15 |
|  | 长取非数字 | 测试用例8 | Return -1 | 16 |
|  | 宽取非数字 | 测试用例9 | Return -1 | 17 |
|  | 高取非数字 | 测试用例10 | Return -1 | 18 |
|  | 重量取非数字 | 测试用例11 | Return -1 | 19 |
|  | 长>100 | 测试用例12 | Return -1 | 20 |
|  | 宽>60 | 测试用例13 | Return -1 | 21 |
|  | 高>40 | 测试用例14 | Return -1 | 22 |
|  | seatIndex不输入 | 测试用例15 | Return -1 | 27 |
|  | seatIndex非数字 | 测试用例16 | Return -1 | 28 |
|  | seatIndex[0,1,2,3]以外的数字 | 测试用例17 | Return -1 | 29 |
|  | ticketIndex不输入 | 测试用例18 | Return -1 | 32 |
|  | ticketIndex非数字 | 测试用例19 | Return -1 | 33 |
|  | ticketIndex[0,1]以外的数字 | 测试用例20 | Return -1 | 34 |
|  | personIndex不输入 | 测试用例21 | Return -1 | 37 |
|  | personIndex非数字[0,1]以外的数字 | 测试用例23 | Return -1 | 38 |
|  | personIndex非数字 | 测试用例24 | Return -1 | 39 |
|  | flyIndex不输入 | 测试用例25 | Return -1 | 42 |
|  | flyIndex[0,1]以外的数字 | 测试用例26 | Return -1 | 43 |
|  | flyIndex非数字 | 测试用例27 | Return -1 | 44 |
|  | flyIndex =1,whatGoByIndex不输入 | 测试用例28 | Return -1 | 45 |
|  | flyIndex =1,whatGoByIndex为[0,1,2]以外的数字 | 测试用例29 | Return -1 | 46 |
|  | flyIndex =1,whatGoByIndex非数字 | 测试用例30 | Return -1 | 47 |
|  | 48. flyIndex =1,whatAreaIndex不输入 | 测试用例31 | Return -1 | 48 |
|  | 49.flyIndex =1,whatAreaIndex为[0,1,2,3,4]以外的数字 | 测试用例32 | Return -1 | 49 |
|  | 50.flyIndex =1,whatAreaIndex非数字 | 测试用例33 | Return -1 | 50 |
|  | cardIndex不输入 | 测试用例34 | Return -1 | 55 |
|  | cardIndex[0,1,2,3]以外的数字 | 测试用例35 | Return -1 | 56 |
|  | cardIndex非数字 | 测试用例36 | Return -1 | 57 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# 2、单元测试框架

## 1. 单元测试内容

### 1.1 模块接口测试

### 1.2 数据结构测试

### 1.3边界条件测试

### 1.4 独立路径测试

### 1.5 出错处理测试

## 2. 测试对象

每个被测试单元（即测试对象）主要按照以下步骤来写。

### 2.1 测试对象一

**2.1.1 软件单元描述**

简单描述被测试单元或与之相关单元的产品项目名称、所属子系统、单元要完成的功能、需求和设计要求等；

**2.1.2 单元结构**

画出本单元的组织结构，包括本单元包括的属性、方法、输入/输出等；

**2.1.3 单元控制**

根据本单元的控制结构或操作时序，画出其大概过程；

**2.1.4 测试过程**

**表2-1 测试过程表**

|  |  |  |
| --- | --- | --- |
| **测试步骤** | **动作描述** | **预计输出结果** |
| 例如：删除原有计划 | 使用菜单中的编辑功能，选中“计划删除”，删除原有的正常巡检计划  1）程序有确认删除的提示 | 记录程序提示并执行查询 |
|  |  |  |
|  |  |  |

### 2.2 测试对象二

**2.2.1 软件单元描述**

简单描述被测试单元或与之相关单元的产品项目名称、所属子系统、单元要完成的功能、需求和设计要求等；

**2.2.2 单元结构**

画出本单元的组织结构，包括本单元包括的属性、方法、输入/输出等；

**2.2.3 单元控制**

根据本单元的控制结构或操作时序，画出其大概过程；

**2.2.4 测试过程**

**表2-1 测试过程表**

|  |  |  |
| --- | --- | --- |
| **测试步骤** | **动作描述** | **预计输出结果** |
| 例如：删除原有计划 | 使用菜单中的编辑功能，选中“计划删除”，删除原有的正常巡检计划  1）程序有确认删除的提示 | 记录程序提示并执行查询 |
|  |  |  |
|  |  |  |

………

## 3. 测试结果

测试结果包括代码审查结果、测试用例的设计以及测试后的效果图。

**表2-2 测代码审查结果表**

|  |  |  |  |
| --- | --- | --- | --- |
| **Bug ID** | **审查人员** | **审查日期** | **问题描述** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**附表三**

**测试用例执行结果统计表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **测试项** | **测试用例号** | **测试特性** | **用例描述** | **测试结论** | **对应bug ID** |
|  |  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |