电子科技大学信息与软件工程学院

**实 验 报 告**

学 号 2018091618008

姓 名 袁昊男

(实验) 课程名称 软件工程基础

理论教师 苏生

实验教师 苏生

**电 子 科 技 大 学**

**实 验 报 告**

**学生姓名：袁昊男 学号：2018091618008 指导教师：苏生**

**实验地点：信软楼303 实验时间：2019.12.01**

1. **实验名称：质量保证**
2. **实验学时：4学时**
3. **实验目的：**
4. 掌握软件编码规范；
5. 掌握单元测试的方法，包括白盒测试和黑盒测试；
6. 能对实际程序进行单元测试，能够撰写简单的测试报告。
7. **实验原理：**

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，一般来说，要根据实际情况去判定其具体含义，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。总的来说，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。

系统测试（System Testing），是将已经确认的软件、计算机硬件、外设、网络等其他元素结合在一起，进行信息系统的各种组装测试和确认测试，系统测试是针对整个产品系统进行的测试，目的是验证系统是否满足了需求规格的定义，找出与需求规格不符或与之矛盾的地方，从而提出更加完善的方案。系统测试发现问题之后要经过调试找出错误原因和位置，然后进行改正。是基于系统整体需求说明书的黑盒类测试，应覆盖系统所有联合的部件。对象不仅仅包括需测试的软件，还要包含软件所依赖的硬件、外设甚至包括某些数据、某些支持软件及其接口等。

1. **实验内容：**

本实验主要涉及对三角形判断程序进行编写及黑/白盒测试，以及对前阶段系统的两个关键模块进行黑/白盒测试。具体实验内容包括：

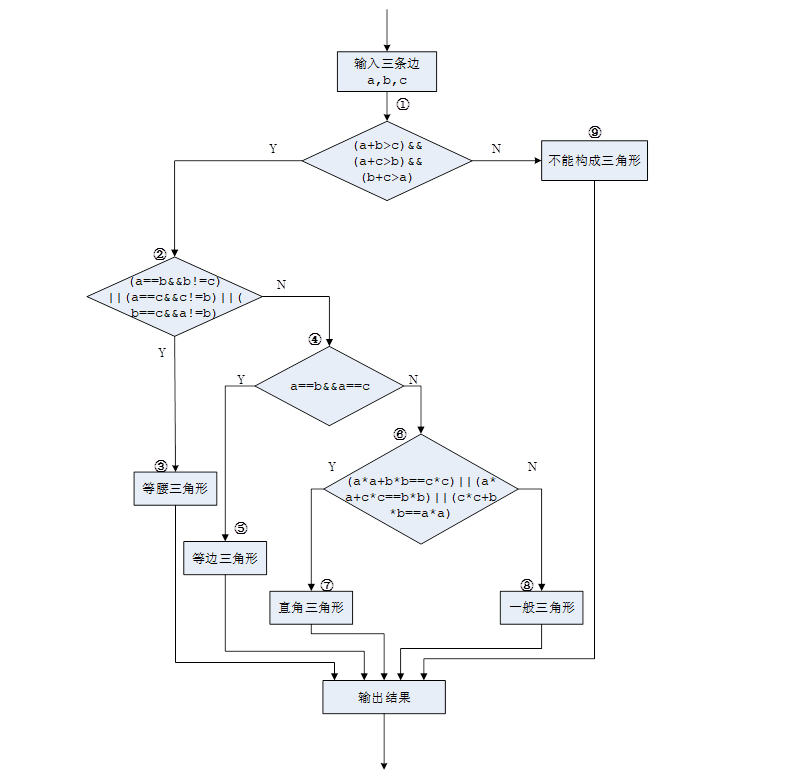
1. 编写三角形判断程序，并进行黑盒测试；
2. 对所编写的三角形判断程序进行白盒测试；
3. 选择系统中的2个关键模块进行编码实现；
4. 使用白盒或黑盒方法测试实现的模块。
5. **实验器材（设备、元器件）：**

安装有Rational Rose、Visio等软件的个人电脑一台。

1. **实验步骤：**
2. 输入三个整数a、b和c分别作为三角形的3条边，判断由这3条边构成的三角形类型是：等边三角形、等腰三角形、一般三角形、或非三角形（即不能构成一个三角形）：
3. 编写该程序，注意编程规范；
4. 按照等价类划分方法设计测试用例，并进行测试；
5. 按照边界值方法设计测试用例，并进行测试。
6. 根据所编写的三角形判断程序，进行白盒测试：
7. 进行语句覆盖测试；
8. 进行判断覆盖、条件覆盖、判断-条件覆盖测试；
9. 体会不同白盒测试的差异。
10. 基于实验2的结果，选择2个关键模块进行编程实现；
11. 使用白盒或黑盒方法测试实现的模块。
12. **实验结果与分析（含重要数据结果分析或核心代码流程分析）**
13. 编写三角形判断程序
14. 程序代码

|  |
| --- |
| 1. #include<stdio.h> 2. **void** main(**void**) 3. { 4. **int** a,b,c,k; 5. printf("请输入三角形的三边长(0-100)："); 6. scanf("%d %d %d",&a,&b,&c); 7. **if**((a>=1&&a<=100)&&(b>=1&&b<=100)&&(c>=1&&c<=100) 8. { 9. **if**((a+b>c)&&(a+c>b)&&(b+c>a)) 10. { 11. **if**((a==b&&b!=c)||(a==c&&c!=b)||(b==c&&a!=b)) 12. k=1; 13. **if**(a==b&&a==c) 14. k=2; 15. **if**((a\*a+b\*b==c\*c)||(a\*a+c\*c==b\*b)||(c\*c+b\*b==a\*a)) 16. k=3; 17. **switch**(k) 18. { 19. **case** 1：printf("等腰三角形\n"); 20. **break**; 21. **case** 2：printf("等边三角形\n"); 22. **break**; 23. **case** 3：printf("直角三角形\n"); 24. **break**; 25. **default**：printf("一般三角形\n"); 26. **break**; 27. } 28. } 29. **else** 30. printf("不能构成三角形\n"); 31. } 32. **else** 33. printf("警告：无效输入！！\n"); 34. } |

1. 程序流程图



1. 对三角形判断程序进行黑/白盒测试
2. 等价类划分及测试用例

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **输入条件** | **有效等价类** | **编号** | **无效等价类** | **编号** |
| 是否构成三角形 | a>=1&&a<=100 | (1) | a<1 | (7) |
| b<1 | (8) |
| b>=1&&b<=100 | (2) | c<1 | (9) |
| a>100 | (10) |
| c>=1&&c<=100 | (3) | c>100 | (11) |
| c>100 | (12) |
| a+b>c | (4) | a+b<=c | (13) |
| a+c>b | (5) | a+c<=b | (14) |
| b+c>a | (6) | b+c<=a | (15) |
| 是否等腰三角形 | a==b && b!=c | (16) | a!=b&&b!=c | (19) |
| a==c && c!=b | (17) |
| b==c && a!=b | (18) |
| 是否等边三角形 | a==b && a==c | (20) | a!=b | (21) |
| b!=c | (22) |
| a!=c | (23) |
| 是否直角三角形 | a\*a+b\*b==c\*c | (24) | a\*a+b\*b!=c\*c &&  a\*a+c\*c!=b\*b &&  c\*c+b\*b!=a\*a | (27) |
| a\*a+c\*c==b\*b | (25) |
| c\*c+b\*b==a\*a | (26) |

测试用例：

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **编号** | **a** | **b** | | **c** | | **覆盖等价类** | **预期输出** |
| 1 | 4 | 5 | | 6 | | (1)(2)(3)(4)(5)(6) | 一般三角形 |
| 2 | -4 | 5 | | 6 | | (7) | 无效输入 |
| 3 | 4 | -5 | | 6 | | (8) | 无效输入 |
| 4 | 4 | 5 | | -6 | | (9) | 无效输入 |
| 5 | 101 | 5 | | 6 | | (10) | 无效输入 |
| 6 | 4 | 101 | | 6 | | (11) | 无效输入 |
| 7 | 4 | 5 | | 101 | | (12) | 无效输入 |
| 8 | 4 | 5 | | 9 | | (13) | 不能构成三角形 |
| 9 | 4 | 13 | | 9 | | (14) | 不能构成三角形 |
| 10 | 14 | 5 | | 9 | | (15) | 不能构成三角形 |
| 11 | 4 | 4 | | 6 | | (1)(2)(3)(4)(5)(6)(16) | 等腰三角形 |
| 12 | 4 | 6 | | 4 | | (1)(2)(3)(4)(5)(6)(17) | 等腰三角形 |
| 13 | 4 | 6 | | 6 | | (1)(2)(3)(4)(5)(6)(18) | 等腰三角形 |
| 14 | 4 | 5 | | 6 | | (1)(2)(3)(4)(5)(6)(19) | 一般三角形 |
| 15 | 4 | 4 | | 4 | | (1)(2)(3)(4)(5)(6)(20) | 等边三角形 |
| 16 | 4 | 5 | | 6 | | (1)(2)(3)(4)(5)(6)(21) | 一般三角形 |
| 17 | 4 | | 5 | | 6 | (1)(2)(3)(4)(5)(6)(22) | 一般三角形 |
| 18 | 4 | | 5 | | 6 | (1)(2)(3)(4)(5)(6)(23) | 一般三角形 |
| 19 | 3 | | 4 | | 5 | (1)(2)(3)(4)(5)(6)(24) | 直角三角形 |
| 20 | 3 | | 5 | | 4 | (1)(2)(3)(4)(5)(6)(25) | 直角三角形 |
| 21 | 5 | | 3 | | 4 | (1)(2)(3)(4)(5)(6)(26) | 直角三角形 |
| 22 | 4 | | 5 | | 6 | (1)(2)(3)(4)(5)(6)(27) | 一般三角形 |

1. 边界值测试用例

边界值分析：每条边的取值范围[0,100]

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **编号** | **a** | **b** | **c** | **预期输出** |
| 1 | 0 | 50 | 50 | 无效输入 |
| 2 | 1 | 50 | 50 | 等腰三角形 |
| 3 | 100 | 50 | 50 | 等腰三角形 |
| 4 | 101 | 50 | 50 | 无效输入 |
| 5 | 50 | 0 | 50 | 无效输入 |
| 6 | 50 | 1 | 50 | 等腰三角形 |
| 7 | 50 | 100 | 50 | 等腰三角形 |
| 8 | 50 | 101 | 50 | 无效输入 |
| 9 | 50 | 50 | 0 | 无效输入 |
| 10 | 50 | 50 | 1 | 等腰三角形 |
| 11 | 50 | 50 | 100 | 等腰三角形 |
| 12 | 50 | 50 | 101 | 无效输入 |
| 13 | 50 | 50 | 50 | 等边三角形 |

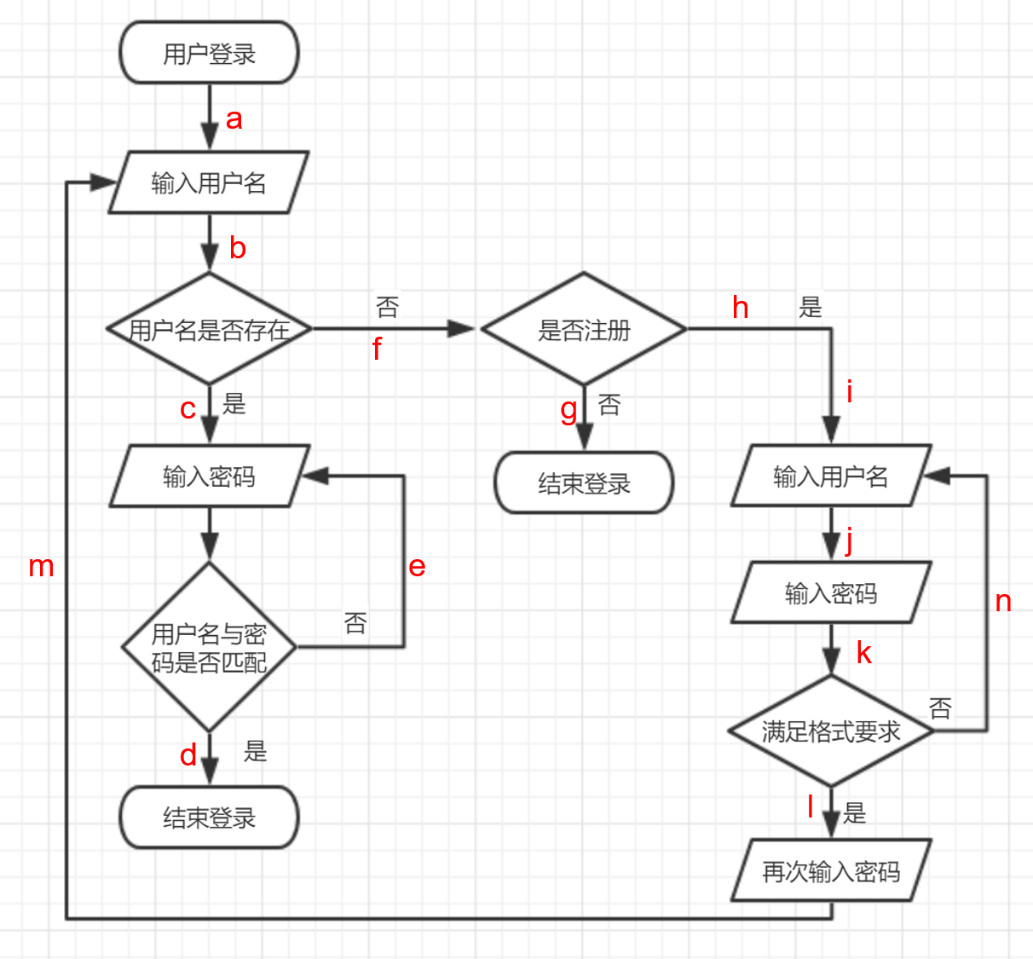
1. 语句覆盖测试用例

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **编号** | **a** | **b** | **c** | **覆盖对象** | **预期输出** |
| 1 | 4 | 4 | 5 | ①②③ | 等腰三角形 |
| 2 | 4 | 4 | 4 | ①②④⑤ | 等边三角形 |
| 3 | 3 | 4 | 5 | ①②④⑥⑦ | 直角三角形 |
| 4 | 4 | 5 | 6 | ①②④⑥⑧ | 一般三角形 |
| 5 | 4 | 5 | 9 | ①⑨ | 不能构成三角形 |

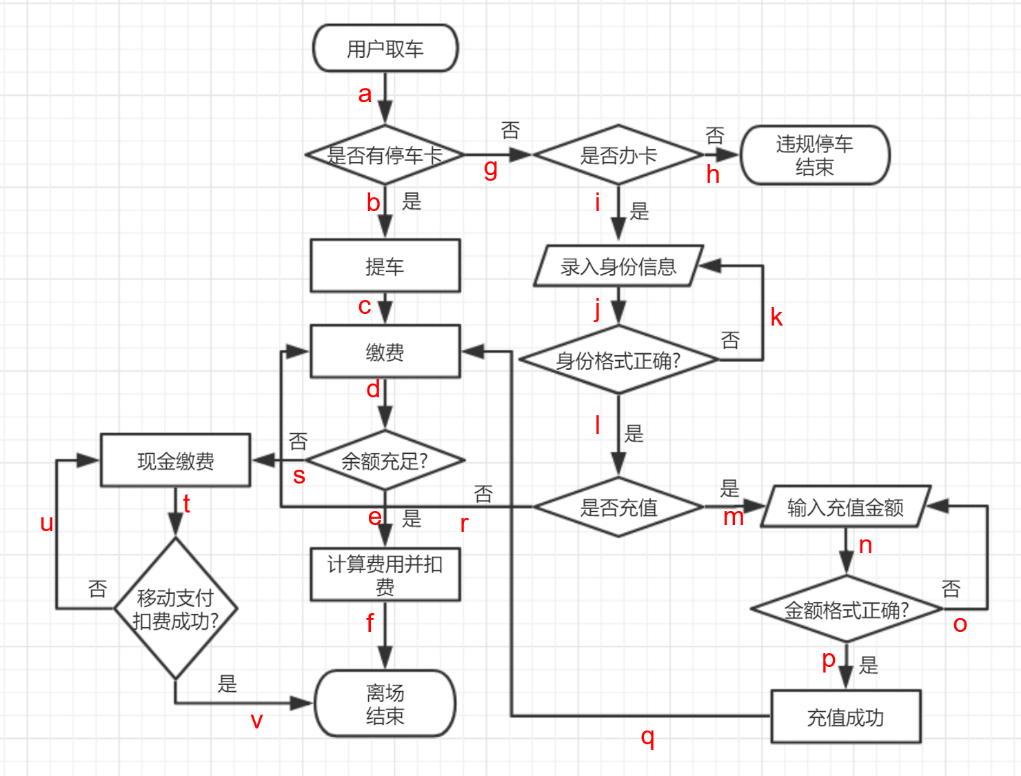
1. 判定覆盖测试用例

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **编号** | **a** | **b** | **c** | **覆盖对象** | **预期输出** |
| 1 | 4 | 4 | 5 | ①②③ | 等腰三角形 |
| 2 | 4 | 4 | 4 | ①②④⑤ | 等边三角形 |
| 3 | 3 | 4 | 5 | ①②④⑥⑦ | 直角三角形 |
| 4 | 4 | 5 | 6 | ①②④⑥⑧ | 一般三角形 |
| 5 | 4 | 5 | 9 | ①⑨ | 不能构成三角形 |

1. 车库管理系统关键模块的程序流程图
2. 用户登录



1. 用户取车



1. 车库管理系统关键模块的黑/白盒测试
2. 用户登录
3. 语句覆盖测试

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **编号** | **测试点** | **前置条件** | **输入** | **预期输出** |
| 1 | 用户登陆，用户名是否存在，输入密码，用户名与密码是否匹配，结束登陆 | 用户名存在，用户名与密码匹配 | 账号：2018001  密码：123546 | 结束登陆 |
| 2 | 用户登陆，输入用户名，用户名是否存在，是否注册，结束登陆 | 用户名存在，不驻车 | 123456 | 结束登陆 |
| 3 | 用户登陆，输入用户名，用户名是否存在，是否注册，输入用户名，输入密码，满足格式要求，再次输入密码，输入用户名，用户是否存在，输入密码，用户名与密码是否匹配，结束登陆 | 用户名不存在，确定注册，满足格式要求，用户名存在，用户名与密码匹配 | 账号：2018002  密码：123456 | 结束登陆 |

1. 分支覆盖测试

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **编号** | **测试分支** | **前置条件** | **输入** | **预期输出** |
| 1 | a-f-g | 用户名不存在，不注册 | 2018002 | 结束登陆 |
| 2 | a-f-h-i-j-k-l-m-b-c-d | 用户名不存在，确定注册，满足格式要求，用户名存在，用户名与密码匹配 | 账号：2018002  密码：123456 | 结束登陆 |

1. 条件组合覆盖测试

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **编号** | **测试分支** | **前置条件** | **输入** | **预期输出** |
| 1 | a-f-h-i-j-k-l-m-b-c-d | 用户名不存在，确定注册，满足格式要求，用户名存在，用户名与密码匹配 | 账号：2018002  密码：123456 | 结束登陆 |
| 2 | a-f-h-i-j-k-n-i-j-k-l-m-b-c-d | 用户名不存在，确定注册，用户名满足格式要求，密码不满足格式要求，重新输入后两者满足，用户名存在，用户名与密码匹配 | 账号：2018002  密码：1234567  重新输入后：账号：2018002  密码：123456 | 结束登陆 |
| 3 | a-f-h-i-j-k-n-i-j-k-l-m-b-c | 用户名不存在，确定注册，用户名不满足格式要求，密码满足格式要求，重新输入后两者满足，用户名存在，用户名与密码匹配 | 账号：asd5  密码：123456  重新输入后：账号：2018002  密码：123456 | 结束登陆 |
| 4 | a-f-h-i-j-k-n-i-j-k-l-m-b-c | 用户名不存在，确定注册，用户名不满足格式要求，密码不满足格式要求，重新输入后两者满足，用户名存在，用户名与密码匹配 | 账号asd5  密码1234567  重新输入后：账号：2018002  密码：123456 | 结束登陆 |

1. 等价类划分及测试用例

|  |  |  |
| --- | --- | --- |
| 输入等价类 | 有效等价类 | 无效等价类 |
| 用户名 | 少于18个字符(1) | 用户名为空(2) |
| 用户名多于18个字符(3) |
| 包含字母、数字、下划线或汉字(4) | 包含特殊字符(5) |
| 密码 | 6~18个字符(6) | 密码少于6个字符(7) |
| 密码多于18个字符(8) |
| 含有数字和字母(9) | 密码全为数字(10) |
| 密码全为字母(11) |
| 密码包含特殊字符(12) |

测试用例：

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 输入 | 覆盖等价类 | 预期输出 |
| 1 | 用户名：Lilly1  密码：zxcvb12345 | (1)(4)(6)(9) | 合法输入 |
| 2 | 用户名：测试\_  密码zxcvb12345 | (1)(4)(6)(9) | 合法输入 |
| 3 | 用户名：  密码： | (2)(7) | 非法输入 |
| 4 | 用户名：123  密码：123456 | (1)(5)(6)(10) | 非法输入 |
| 5 | 用户名：Test  密码：abcdefg | (1)(4)(6)(11) | 非法输入 |
| 6 | 用户名：Test  密码：12345 | (1)(4)(6)(12) | 非法输入 |
| 7 | 用户名：aaaaaaaaaaaaaaaaaaa  密码：bbbbbbbbbbbbbbbbbbb | (3)(8) | 非法输入 |

1. 用户取车
2. 语句覆盖测试

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **编号** | **测试分支** | **前置条件** | **输入** | **预期输出** |
| 1 | 用户取车、是否有停车卡、提车、缴费、余额充足、计算费用并扣费、离场结束 | 数据库内用户名1001，密码888，余额1000 | 停车卡读卡信息1001，确认扣费 | 离场，结束 |
| 2 | 用户取车、是否有停车卡、是否办卡、违规停车结束 | 数据库内用户名1001，密码888，余额1000 | 无输入（读卡失败或无卡）、不办卡 | 违规停车，结束 |
| 3 | 用户取车、是否有停车卡、是否办卡、录入身份信息、身份格式正确、是否充值、输入充值金额、金额格式正确、充值成功、缴费、余额充足、计算费用并扣费、离场结束 | 数据库内用户名1001，密码888，余额1000 | 无输入（读卡失败或无卡）、确认办卡、张三，15810000000，确认充值，-100，1000，确认扣费 | 离场，结束 |
| 4 | 用户取车、是否有停车卡、提车、缴费、余额充足、现金缴费、移动支付扣费成功、离场结束 | 数据库内用户名1001，密码888，余额1000 | 停车卡读卡信息1001、确认扣费、有效付款码 | 离场，结束 |

1. 分支覆盖测试

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **编号** | **测试分支** | **前置条件** | **输入** | **预期输出** |
| 1 | a-g-h | 数据库内用户名1001，密码888，余额1000；用户名1002，余额0 | 无输入（读卡失败或无卡）、不办卡 | 违规停车，结束 |
| 2 | a-b-c-d-e-f | 数据库内用户名1001，密码888，余额1000；用户名1002，余额0 | 停车卡读卡信息1001，确认扣费 | 离场，结束 |
| 3 | a-g-i-j-k-j-l-m-n-o-n-p-q-d-e-f | 数据库内用户名1001，密码888，余额1000；用户名1002，余额0 | 无输入（读卡失败或无卡）、确认办卡、张三，110、张三，15910000000、确认充值、-100、1000，确认扣费 | 离场，结束 |
| 4 | a-g-i-j-l-r-d-s-t-v | 数据库内用户名1001，密码888，余额1000；用户名1002，余额0 | 无输入（读卡失败或无卡）、确认办卡、张三，15910000000、不充值、确认扣费、出示有效付款码 | 离场，结束 |
| 5 | a-b-c-d-s-t-u-t-v | 数据库内用户名1001，密码888，余额1000；用户名1002，余额0 | 停车卡读卡信息1002、确认扣费、出示有效付款码 | 离场，结束 |

1. 条件覆盖测试

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **编号** | **测试分支** | **前置条件** | **输入** | **预期输出** |
| 1 | a-g-h | 数据库内用户名1001，密码888，余额1000；用户名1001，余额0 | 无输入（读卡失败或无卡）、不办卡 | 违规停车，结束 |
| 2 | a-b-c-d-e-f | 数据库内用户名1001，密码888，余额1000；用户名1001，余额0 | 停车卡读卡信息1001，确认扣费 | 离场结束 |
| 3 | a-g-i-j-l-m-n-p-q-d-e-f | 数据库内用户名1001，密码888，余额1000；用户名1001，余额0 | 无输入（读卡失败或无卡）、确认办卡、张三，15810000000，确认充值，1000，确认扣费 | 离场结束 |
| 4 | a-b-c-d-s-t-v | 数据库内用户名1001，密码888，余额1000；用户名1001，余额0 | 停车卡读卡信息1002、确认扣费、出示有效付款码 | 离场结束 |

1. 等价类划分及测试用例

|  |  |  |
| --- | --- | --- |
| **输入等价类** | **有效等价类** | **无效等价类** |
| 是否办卡 | 确认办卡(1) | 取消(3) |
| 不办卡(2) |
| 录入身份信息 | 张三，15810000000(4) | 张三，(5) |
| ，15810000000(6) |
| （David等非中文字符），（%#￥等非11位数字字符串）(7) |
| 是否充值 | 确认充值(8) | 取消(10) |
| 不充值(9) |
| 输入充值金额 | 1000(11) | 小于等于0的数(12) |
| 大于10000的数(13) |
| 付款码 | 有效付款码(14) | 无效二维码或条形码(15) |

测试用例：

|  |  |  |  |
| --- | --- | --- | --- |
| **编号** | **输入** | **覆盖等价类** | **预期输出** |
| 1 | 无输入（读卡失败或无卡）、确认办卡、张三，15910000000、不充值、确认扣费、出示有效付款码 | (1)(4)(9)(14) | 离场结束 |
| 2 | 无输入（读卡失败或无卡）、不办卡 | (1)(2) | 违规停车，结束 |
| 3 | 无输入（读卡失败或无卡）、确认办卡、张三，15910000000、确认充值、1000，确认扣费 | (1)(4)(8)(11) | 离场结束 |
| 4 | 无输入（读卡失败或无卡）、取消 | (1)(3) | 警告提示 |
| 5 | 确认办卡、张三， | (1)(5) | 非法输入 |
| 6 | 确认办卡、，15810000000 | (1)(6) | 非法输入 |
| 7 | 确认办卡、Tom，15810￥%…&\*00 | (1)(7) | 非法输入 |
| 8 | 无输入（读卡失败或无卡）、确认办卡、张三，15910000000、取消 | (1)(4)(10) | 警告提示 |
| 9 | 无输入（读卡失败或无卡）、确认办卡、张三，15910000000、确认充值、-100，确认扣费 | (1)(4)(8)(12) | 非法输入 |
| 10 | 无输入（读卡失败或无卡）、确认办卡、张三，15910000000、确认充值、50000，确认扣费 | (1)(4)(8)(13) | 非法输入 |
| 11 | 无输入（读卡失败或无卡）、确认办卡、张三，15910000000、不充值、确认扣费、出示微信无效二维码 | (1)(4)(9)(15) | 非法输入 |

1. **总结及心得体会：**

软件质量保证一个重要的环节是测试。本实验主要考察对黑/白盒测试的各种方法的了解、掌握与实际应用。对比黑盒、白盒测试的异同，现总结如下表所示：

|  |  |  |  |
| --- | --- | --- | --- |
| **项目** | **黑盒测试法** | **白盒测试法** | |
| 规划 | 功能的测试 | 结构的测试 | |
| 优点 | 能确保从用户的角度出发进行测试。 | 能从程序内部的特定部位进行覆盖测试。 | |
| 缺点 | 无法测定程序内部特定部位；当规格说明有误则不能发现问题。 | | 无法检查程序的外部特性；无法对未实现规格说明的程序内部欠缺部分进行测试。 |
| 应用技术 | 边界分析法，  等价类划分法，  决策表测试。 | | 语句覆盖，判定覆盖，  条件覆盖，判定/条件覆盖，  路径覆盖，循环覆盖，  模块接口测试。 |

经过本次实验，掌握了软件测试的一般方法，如白盒测试中的语句覆盖、条件覆盖和判定覆盖，掌握了黑盒测试中等价类的划分，以及如何根据等价类表设计测试用例覆盖有效、无效等价类。对《软件工程》中软件测试及质量保证的学习有非常好的实践反馈效果。

1. **对本实验过程及方法、手段的改进建议：**

本实验中要求实现系统中两个关键模块的编码和测试对现阶段同学来说在短时间内较难完成，可以考虑将编码环节改换成根据两个关键模块的程序流程图编写用例进行测试，这样能将实验重点放在对软件测试方法的掌握和应用上。

**附：**成员分工

白珈瑞：三角形测试；

蒲龙飞：白盒测试；

吴雨晴：流程及分支设计；

杨晗：黑盒测试；

袁昊男：白盒测试、报告撰写；

**报告评分：**

**指导教师签字：**