

## 《软件测试》期末试题答案(B)

(考试形式：开卷 考试时间：2小时)



《中山大学授予学士学位工作细则》第六条

考试作弊不授予学士学位

方向：\_\_\_\_\_ 姓名：\_\_\_\_\_ 学号：\_\_\_\_\_ 成绩：\_\_\_\_\_

### 一、三角形测试的参考答案

表1是我给出的解答，我使用了等价性划分和边界值分析的组合。

表1 三角形测试的解答

测试行动和数据	期待结果
错误的输入	
对每个参数位置 a、b 或者 c 上各设置一个非数字的一个测试用况	一条错误信息
对于每个参数位置 a、b 或者 c 上各设置一个实数的一个测试用况	一条错误信息
不规则三角形	
2, 3, 4	不规则
maxint-2,maxint-1,maxint(maxint=最大的合法整数)	不规则
1, 2, 3	一条错误信息
5, 3, 2	一条错误信息
1, 2, 4	一条错误信息
-2, 3, 4	一条错误信息
0, 1, 2	一条错误信息
maxint-1,maxint,maxint+1	一条错误信息
等腰三角形型	
2, 2, 1	等腰
9, 5, 5	等腰
maxint, maxint-1, maxint	等腰
1, 2, 1	一条错误信息
2, 5, 2	一条错误信息
3, 1, 1	一条错误信息
2, -2, 1	一条错误信息
2, 2, 0	一条错误信息
等边三角形	
1, 1, 1	等边

maxint, maxint, maxint	等边
maxint+1, maxint+1, maxint+1	一条错误信息
1, 1, -1	一条错误信息
0, 0, 0	一条错误信息
用户界面测试	
每个输入参数位置 a、b 和 c 上的溢出测试	一条错误信息
在 a、b 和 c 输入前试着加一个引导空格	一条错误信息或者忽略空格
在 a、b 和 c 输入后试着加一个后继空格	一条错误信息或者忽略空格
将 a、b 或 c 输入留为空（一次留空一个）	一条错误信息

注意：我的解答只是大量正确可能答案中的一个。

## 二、复审三角形需求首页所找到的需求缺陷

表 2 问题描述

有问题的措辞	Wieggers 问题编号
缺少处理错误的讨论	1. 省略了关键需求
“以用户界面友好的方式”的含义是什么	3. 含混和不明确性
程序如何接受输入？通过文件？图形用户界面？命令行？Java 的 LAF 界面	3. 含混和不明确性
“很快画出”一个三角形是什么意思？多长时间算很快	3. 含混和不明确性
缺少对所支持平台和界面的任何讨论，这是重要的	1. 省略了关键需求
我们真的需要画出三角形么	5. 不需要的功能
不需要角度，我们真的能画出三角形么	7. 规模蔓延
画出三角形和返回类别信息的相对优先级是怎样的	4. 为划分优先级的需求
“输出”到哪里？文件？图形用户界面？基于字符的界面	3. 含混和不明确性
程序如何开始和结束	1. 省略了关键需求
除了含糊的可用性和性能需求，我们只有功能需求	1. 省略了关键需求
我们没有对客户有任何指示	2. 没有引入客户或用户
有文档或帮助么？	1. 省略了关键需求

表 3 问题解决

改善的措辞	解决的问题编号
高优先级需求	
程序在 Linux 系统的命令行上运行	1. 提供关键需求
用户可以输入 “triangle[-h]” a b c.”来启动程序	1. 提供关键需求
程序将使用过滤器的 Linux 脚本输出来归类三角形	2. 标识客户
三角形将被归类为“等边”、“等腰”、“不规则”或者“不是一个三角形”	3. 消除不明确性
标准的 Linux I/O 重定向会用于直接 I/O 定向到文件读写	1. 提供关键需求
中等优先级需求	
程序将接受三个整型输入（例如 a、b、c）。以空格（空格符合制	3. 消除不明确性

表符) 间隔	
如果整型数是在文件中的, 每一行应该包含以空格分隔的恰好三个整型数	3. 消除不明确性
程序应该在接受输入后的 100 秒返回归类信息	3. 消除不明确性
如果一个文件中制定了多个三角形, 每条归类信息各占一行	3. 消除不明确性
如果用户指定了“-h”标识, 程序会打印“man”页面作为帮助	1. 提供关键需求
如果用户指定了“-h”标识, 程序会忽略所提供的其他输入	3. 消除不明确性
如果程序接受了有效的三角形边长, 以状态 0 退出	1. 提供关键需求
如果程序接受了非整型数的边长, 总结处理后以状态 1 退出	1. 提供关键需求
如果程序接受了无效的边长, 总结处理后以状态 2 退出	1. 提供关键需求
如果程序遇到了边长缺失, 立刻以状态 3 退出	1. 提供关键需求
程序可以接受上至 4MB 的输入	1. 提供关键需求
低优先级需求	
扩充程序使之能接受“triangle a b c AB BC AC”, AB、BC 和 AC 是角度	1. 提供关键需求
创建一个 web 站点 (HTML/GIF) 显示三角形, 每页上显示一个三角形, 处理同前所述	1. 提供关键需求
一旦绘制完成第一个三角形, 则应该启动浏览器, 指向该网页	1. 提供关键需求
在白色背景上一黑线显示三角形, 并标出边长和角度	1. 提供关键需求

### 三、关于 ATM 状态转换测试的参考答案

首先, 我画了如图 1 所示的状态转换图。

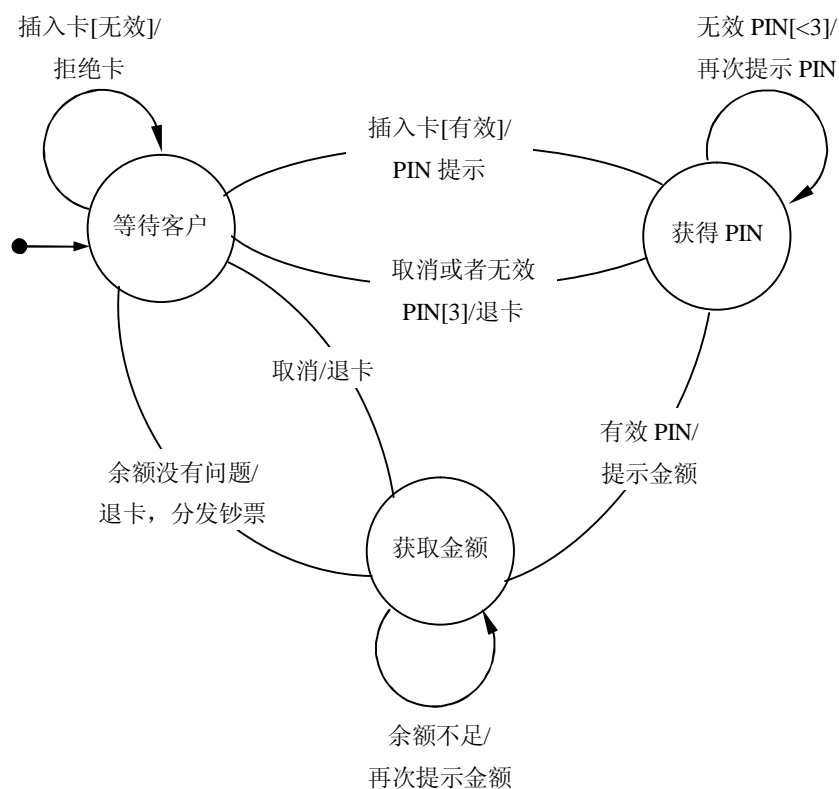


图 1 ATM 状态转换图

当然这是一个简化的状态图。它没有包括涉及初始化和错误处理的状态。然而，对于测试低风险的状况（例如，在后端进行一个微小的变动后进行的回归测试中），它可能就是足够的。

接着，我创建的了如表 4 所示的测试用况。对这个测试而言，你需要一个道具，即一个有一些钱的有效银行卡，但是卡中的钱不需太多。这个账号应该有一些可用的余额，金额小到足够让一个取款可以超过可用的余额。

表 4 ATM 测试情况

	测试步骤/子步骤	预期结果
#	在所有的用况中，ATM 机从等待用户开始	
1	1. 插入无效卡	1. ATM 机拒绝卡
	2. 插入有效卡	2. ATM 提示输入 PIN
	3. 输入无效的 PIN 两次	3. ATM 再次提示输入 PIN
	4. 第三次输入无效 PIN	4. ATM 机退还卡，回到等待用户状态
2	1. 插入有效卡	1. ATM 机提示输入 PIN
	2. 按“取消”按钮	2. ATM 机退还卡，回到等待用户状态
3	1. 插入有效卡	1. ATM 机提示输入 PIN
	2. 输入有效 PIN	2. ATM 机提示输入金额
	3. 输入超过可用余额的金额	3. ATM 再次提示输入一个较低金额
	4. 输入“取消”按钮	4. ATM 机退还卡，回到等待用户状态
4	1. 重复测试用况 3 中的前两个子步骤	1. 如 3.1 和 3.2 指定的那样验证正确的行为
	2. 输入有效金额	2. ATM 机退还卡，分发现钞，回到等待用户状态

#### 四、关于十六进制字符输入转换的控制流测试的参考答案

##### 1. 理解潜在的测试用况的数量

潜在的测试集是无限的。在数学意识上精确地说，潜在测试的集合在测试个数上是无限的。尝试对此程序构造一个你认为是完整并且穷尽的测试集合，看看如何。

我能在你的测试集中找到一个具有最多十六进制和非十六进制输入字符的测试。然后我可在这测试中再加一个字符。于是我就创造了一个新的测试。这个测试和你测试集中所有的其他测试不同。

我可以对任何给定的测试集重复这一过程。这包括我刚从你的测试集中构造出来的集合

再加上一个测试。这样，我们就有了一个无限的循环，从来不会缺乏可构造的新测试。既然我每次都创建新测试，就可以知道测试的数量并进行计数，这一切都可趋于无限。

显然，为这样一个小程序运行个数无限的测试说不上高效的。它也不是非常有效，因为许多测试在他们所揭示的缺陷方面是雷同的。

2. 为语句、分支、条件和循环的全覆盖创建一套测试用例集

我所创建的测试集如表 5 所示。

表 5 语句、分支、条件和循环覆盖

测试编号	测试者动作和数据	预期结果
语句覆盖		
1	输入 “0”	输出"Got 1 hex digits: 0" 返回 0
2	输入 “a”	输出"Got 1 hex digits: a" 返回 0
3	输入 “F”	输出"Got 1 hex digits: f" 返回 0
4	输入 “g”	输出"Got 0 hex digits: 0" 返回 0
分支覆盖		
5	输入 “”（空输入）	输出"Got 0 hex digits: 0" 返回 0
条件覆盖		
6	输入 “0123456789”	输出"Got 0 hex digits: 123456789" 返回 0
7	输入 “abcdef”	输出"Got 6 hex digits: abcdef" 返回 0
8	输入 “ABCDEF”	输出"Got 6 hex digits: abcdef" 返回 0
循环		
9	一个很大的输入，具有已知数量(n) 和已知值(x)的十六进制数	输出"Got n hex digits: x" 返回 0