理解迭代，应该首先理解Scrum，Scrum是一个敏捷开发框架，是一个增量迭代的开发过程。在这个框架的整个开发周期由若干个小的迭代周期（Sprint）组成，每个Spirit长度2周~4周。

区域，可以是项目中的产品功能模块，或者团队项目组。

网络搜索比较好的解释：

区域和迭代 MSDN解释

区域：将工作项分组为产品功能或团队区域。 区域必须是项目层次结构中的有效节点。

迭代：将工作项按照指定的冲刺 (sprint) 或时间段分组。 迭代必须是项目层次结构中的有效节点。

网上解释 区域：细分项目的不同功能模块； 迭代：渐进螺旋型的开发模式

心得体会： 一般说来，我们要想完整的把握项目，就会从好几个角度去观察和控制它。也就是说我们把握项目是从角度出发，即把项目分成几块，每一块每一块的去看。我们把这种角度称做“视点”，而在VSTS中这个视点被叫做“区域”。因此我们需要规划出我们感兴趣的视点，以便能将为数不多的精力投入到这些对项目来说非常重要的区域里去。 在项目中可以规划出许多种视点来。例如三部项目可以定区域：计划制订、计划结项、查询统计、在线、权限控制这些区域。 再来看软件迭代。假设现在要发布一个新版本的软件系统。我们计划通过三次迭代过程，完成整个软件的发布。每一个迭代过程都包括一个完整的发布流程。每一个迭代过程也都生成一个完整的软件版本。在三次迭代过程完成之后，我们就得到一个可以发布的最终版本。例如：三部项目开发过程的sprint1、sprint2、sprint3就是迭代，既是一个开发阶段，一个冲刺阶段。 在定义好区域后，我们通过区域与迭代的组合，就能帮我们定位某个工作项所处的位置和性重要程度。(转)

# 比较负载测试，容量测试和强度测试的区别有哪些

#### [性能测试](http://www.51testing.com/html/37/javascript:;" \t "http://www.51testing.com/html/37/_self)

　　负载测试、容量测试、[压力测试](http://www.51testing.com/html/37/javascript:;" \t "http://www.51testing.com/html/37/_self)、强度测试都属于性能测试，  
　　性能测试是指在给定条件基准的前提下能达到的运行程度，测试软件在系统中的运行性能，度量系统与预定义目标的差距。

#### 负载测试

　　负载测试是模拟在超负 荷环境中运行，通过不断加载（如逐渐增加模拟用户的数量）或其它加载方式来观察不同负载下系统的响应时间和数据吞吐量、系统占用的资源（如CPU、内存）等，以检验系统的行为和特性，以发现系统可能存在的性能瓶颈、内存泄漏、不能实时同步等问题。负载测试更多地体现了一种方法或一种[技术](http://www.51testing.com/html/37/javascript:;" \t "http://www.51testing.com/html/37/_self)。

#### 压力测试（强度测试）

　　压力测试（强度测试）：压力测试是在强负载（大数据量、大量并发用户等）下的测试，查看应用系统在峰值使用情况下操作行为，从而有效地发现系统的某项功能隐患、系统是否具有良好的容错能力和可恢复能力。压力测试分为高负载下的长时间（如24小时以上）的稳定性压力测试和极限负载情况下导致系统崩溃的破坏性压力测试。

#### 容量测试

　　容量测试目的是通过测试预先分析出反映软件系统应用特征的某项指标的极限值（如最大并发用户数、[数据库记录](http://www.51testing.com/html/37/javascript:;" \t "http://www.51testing.com/html/37/_self)数等），系统在其极限值状态下没有出现任何软件故障或还能保持主要功能正常运行。容量测试是面向数据的，并且它的目的是显示系统可以处理目标内确定的数据容量。  
　　针对上述 负载测试、压力测试、容量测试 举个例子  
　　例：一个人背X斤  
　　负载测试：200斤情况下，是否能坚持5分钟。  
　　压力测试：200,300,400... 斤情况下，他的表现，什么时候失败，失败之后什么表现，重新扛200是否正常。  
　　容量测试：在坚持5分钟的情况下，他一次最多能扛多少斤。（转）

L10N 是 localization 的缩写形式，意即在 l 和 n 之间有 10 个字母，本意是指软件的“本地化”。

**[I18n](https://baike.baidu.com/item/I18n" \t "https://baike.baidu.com/item/l18n/_blank)**即 Internationalization，[软件国际化](https://baike.baidu.com/item/%E8%BD%AF%E4%BB%B6%E5%9B%BD%E9%99%85%E5%8C%96" \t "https://baike.baidu.com/item/l18n/_blank)，一个过程，通过修改代码来实现代码的文化独立性，即代码不依赖与任何特定文化的信息。I18n 是一个代码修改的过程，目的是实现代码完全独立与任何特定的文化信息。这类信息保存在外部文件中，在程序运行的时候被装入。

举个例子：

假设我们写了一个 app，在国内市场获得了巨大成功，准备走出国门，需要做的工作就是 i18n。

假设我们要在国内代理营运一个外国的 app，所做的工作就叫 l10n。

软件测试这个行业，做的好的最后都会是业务专家，不管是自动化测试还是手工测试，除了基本的测试方法，测试理论，测试技术手段和工具外。对于测试产品的业务透彻的理解和定位，分析与远瞻都是灵魂。不管是哪一种测试流程第一步一定是需求分析。所以往往你选择的不仅仅是测试这个职位，而是行业，你到底要在哪个行业进行测试积累，金融，购物，教育等 有的行业壁垒是很高的。

很高兴你在高一文理分科的契机就在思考职业生涯的事情，很有远瞻性。其实软件工程师这个职业名词很大，它是一个广义的概念，包括[软件设计](https://baike.baidu.com/item/%E8%BD%AF%E4%BB%B6%E8%AE%BE%E8%AE%A1" \t "https://baike.baidu.com/item/%E8%BD%AF%E4%BB%B6%E5%B7%A5%E7%A8%8B%E5%B8%88/_blank)人员、[软件架构](https://baike.baidu.com/item/%E8%BD%AF%E4%BB%B6%E6%9E%B6%E6%9E%84" \t "https://baike.baidu.com/item/%E8%BD%AF%E4%BB%B6%E5%B7%A5%E7%A8%8B%E5%B8%88/_blank)人员、软件工程管理人员、程序员等一系列岗位，工作内容都与[软件开发](https://baike.baidu.com/item/%E8%BD%AF%E4%BB%B6%E5%BC%80%E5%8F%91" \t "https://baike.baidu.com/item/%E8%BD%AF%E4%BB%B6%E5%B7%A5%E7%A8%8B%E5%B8%88/_blank)生产相关。从大学到研究生，或者更高的研究学习的时候会更有方向感，会划分的更细。

给你看一个招聘信息：

职位类型：IT/互联网

发布时间：2017-11-19

有效日期：2017-12-19

岗位职责：1.根据需求设计软件架构；2.负责项目或者设备的软件设计，实现；3.完成技术资料和产品文档的编写、维护、归档等工作。 任职要求：1.计算机及相关专业，本科及以上学历 ； 2.熟练掌握C、C++、Java中至少一种编程语言，熟练掌握linux shell语言，会使用python、perl等脚本语言中至少一种； 3.熟悉计算机基本原理，如操作系统，微机原理，编译原理、了解一种或几种微处理器架构；4.熟悉Linux操作系统下编程开发，有完整产品开发经验者，比如安全产品。

最基础的还是要掌握一门语言以及组织语言的逻辑结构，将你要实现的转化成计算机可执行的命令。

这个问题的切入点好大，每一个性能测试计划中第一步都会制定目标和分析系统构成。只有明确目标和了解系统构成才会澄清测试范围，知道在测试中要掌握什么样的技术以及确定哪些性能指标需要度量。换种说法就是就是要懂技术，而且知道衡量哪些指标暴露性能瓶颈，从而优化软件。

性能测试的目的是度量系统相对于预定义目标的差距，同时发现软件系统中存在的性能瓶颈，优化软件。性能测试一般要有专门的工具支持，必要时还是要开发专门的接口工具。测试过程中需要收集系统资源使用情况和系统行为表现，可以采用两种方式，一是采用探针，即在系统代码中插入很多程序指令，通过这些指令记录系统状态，并最终收集数据整理成报告，二是在系统中使用性能监视器，在固定时间内收集系统状态信息。虽然工程不同，采用的评估和度量方法不同，但是还是要懂一些代码和技术，不然都不知道评估那些性能指标，怎么收集系统状态。

基本步骤如下：

1. 确定性能测试需求
2. 学习的相关技术和工具
3. 设计测试用例
4. 运行测试用例
5. 分析测试结果

模拟大量手机调用接口对服务器的压力，所以测试的重点还是在服务器上，你可以用Jmeter模拟接口报文，来并发压服务器，看服务器的响应和处理能力。

一、打开模拟器

运行monkeyrunner之前必须先运行相应的模拟器，不然monkeyrunner无法连接设备。

用Elipse打开Android模拟器或在CMD中用Android命令打开模拟器。

二、模拟器启动成功后，我们仍在CMD环境中操作。现在进入monkeyrunner的shell命令交互模式。

命令：monkeyrunner

进入shell命令交互模式后，首要一件事就是导入monkeyrunner所要使用的模块。直接在shell命令下输入：

from com.android.monkeyrunner import MonkeyRunner,MonkeyDevice 回车

OK，这步完成我们就可以利用monkeyrunner进行测试工作了。

**这里有两种方案，一是直接在shell命令下输入以下命令；**

命令说明

device=MonkeyRunner.waitForConnection() #连接手机设备

device.installPackage("../samples/android-10/ApiDemos/bin/Apidemos.apk") #安装apk包到手机设备。

启动其中的任意activity了，只要传入package和activity名称即可。命令如下：

device.startActivity(component="com.example.android.apis/com.example.android.apis.ApiDemos")

此时模拟器会自动打开ApiDemos这个应用程序的主页。

device.reboot() #手机设备重启

device.touch(300,300,'DOWN\_AND\_UP')

MonkeyRunner.alert("hello")#在emulator上会弹出消息提示

device.press('KEYCODE\_HOME',MonkeyDevice.DOWN\_AND\_UP)

device.type('hello')#向编辑区域输入文本'hello'

**二是将以下命令写到python文件里，例如test.py，然后我们再从命令行直接通过monkeyrunner运行它即可。**比如，我们还是用上面的例子，语法如下：monkeyrunner test.py 接下来monkeyrunner会自动调用test.py，并执行其中的语句，相当方便。

实例：test.py

[python] view plaincopy

from com.android.monkeyrunner import MonkeyRunner,MonkeyDevice

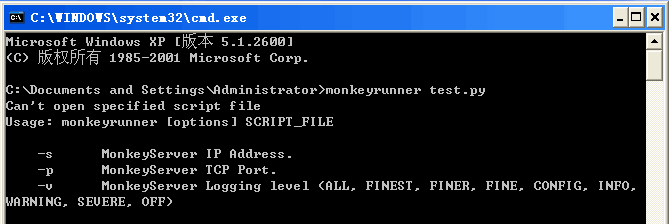
device=MonkeyRunner.waitForConnection()

device.startActivity(component="your.www.com/your.www.com.TestActivity")

在CMD中执行

monkeyrunner test.py

可能出现错误“Can't open specified script file”，如下图所示：



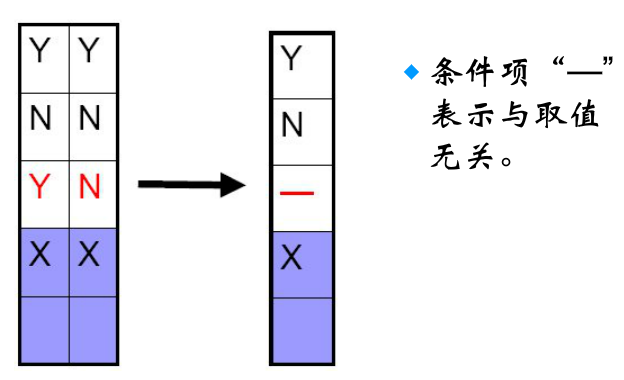
原因在于python脚本文件路径不正确。你可以有以下解决办法：

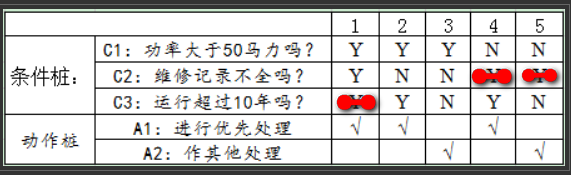
1、将test.py文件存放到monkeyrunner文件同一目录中。可以执行：monkeyrunner test.py 调用

2、指定python文件位置。如果test.py文件在D盘根目录下，可以这样执行：monkeyrunner d:\test.py**（转）**

5）合并相似规则，用以对初始判断表进行简化。

　　有两个或者多条规则具有相同的动作，并且条件项之间存在极为相似的关系就可以进行合并。



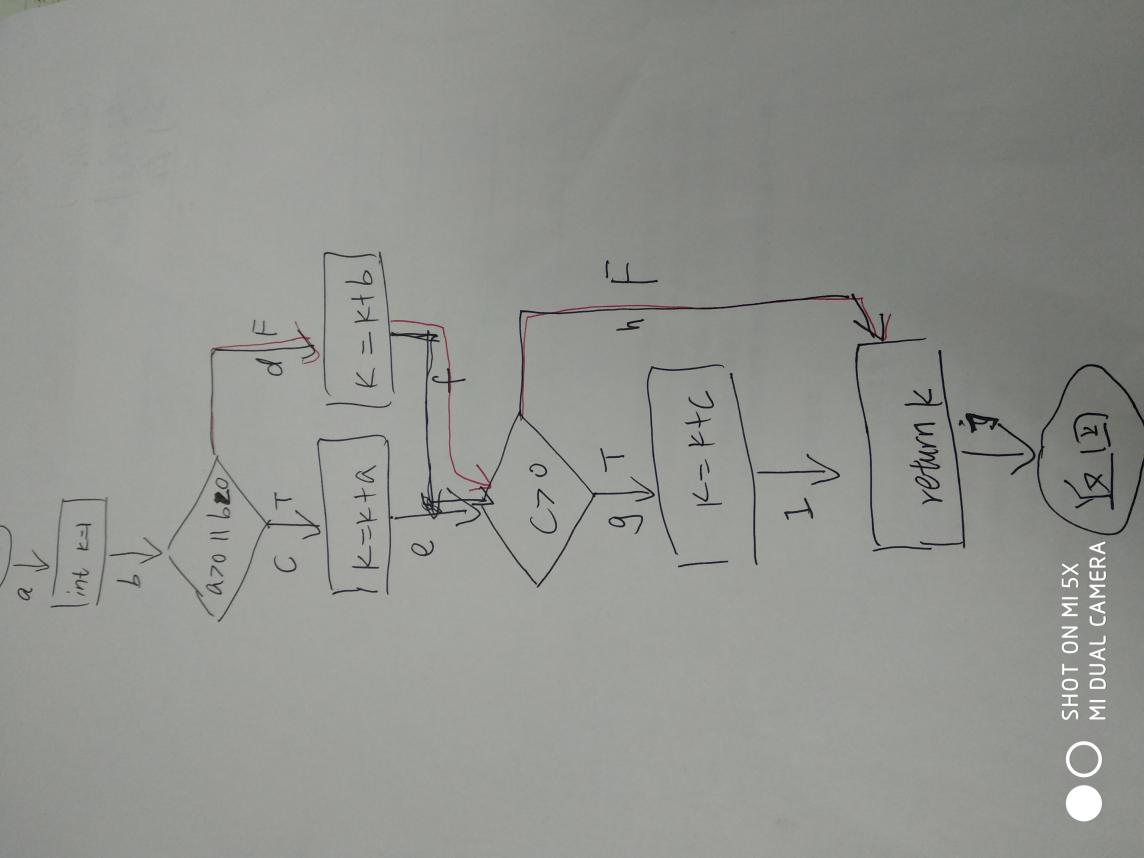


int fun(int a,int b,int c)

{

1. int k=1;
2. if(a>0||b<0)
3. k=k+a;
4. else
5. k=k+b;
6. if(c>0)
7. k=k+c;
8. return k;

}



1. **语句覆盖指的是：**代码中的所有语句都至少执行一遍，用于检测测试用例是否有遗漏。

首先画出程序流程图

为了是每个语句都执行一次，程序的执行路径应该是两条：abcefij ，abdfgij;

设计的测试用例只要覆盖这两条路径，就能将所有语句执行一遍；

a=1 b=-1 c=1

a=-1 b=1 c=1

1. **条件判定覆盖指的是：**设计足够的测试用例，是的判定中的每个条件的所有可能的取值至少出现一次，并且每个判定取到的各种可能的结果也至少出现一次。

由于条件判定覆盖是条件覆盖和判定覆盖的组合，所以只要取条件覆盖和判定覆盖测试用例的并集就可以。

条件覆盖：每个判断的条件的每一种可能至少执行一次。

对于判断语句 a>0 || b<0 :

条件 a>0 : 取真 T1 ，取假 -T1

条件 b<0: 取真 T2 ，取假 -T2

对于判断语句 c>0 :

条件 c>0: 取真 T3 ，取假 -T3.

设计测试用例如表1所示：

|  |  |  |
| --- | --- | --- |
| 输入 | 通过路径 | 条件取值 |
| a=1,b=-1,c=2 | abcegij | T1,T2,T3 |
| a=-1,b=1,c=-2 | abdfhij | -T1,-T2,-T3 |

判定覆盖的死性是使每个判断的真分支和假分支至少执行一次。

对于判断语句 a>0 || b<0 : 取真 M ，取假 -M

对于判断语句 c>0 : 取真 N ，取假 -N.

设计测试用例如表2所示：

|  |  |  |
| --- | --- | --- |
| 输入 | 通过路径 | 条件取值 |
| a=1,b=-1,c=2 | abcegij | M，N |
| a=-1,b=1,c=-2 | abdfhij | -M,-N |

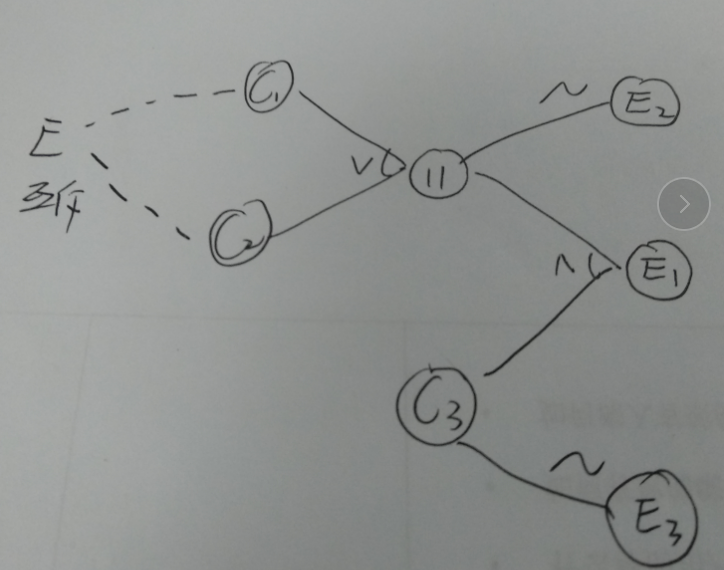
**综合表1和表2，条件判定覆盖的测试用例如表3所示。**

|  |  |
| --- | --- |
| 输入 | 通过路径 |
| a=1,b=-1,c=2 | abcegij |
| a=-1,b=1,c=-2 | abdfhij |

**步骤1：**

分析软件规格说明书，找出原因和结果并进行编号。

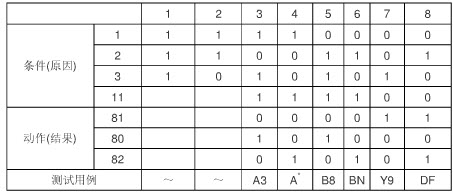
|  |  |  |
| --- | --- | --- |
| 原因 | C1：第一个字符为A C2： 第一个字符为B  C3：第二个字符为数字 |  |
| 中间状态 | 11：确定第一个字符 |  |
| 结果 | E1：修改文件 E2：打印错误信息X12 E3：打印错误信息X13 |  |

找出各原因和结果之间、原因和原因之间的关系，画出因果图，如图所示。  


**步骤2：**

因为有三个条件，所以组合后会有2的三次方，8种情况。将上述因果图转换为判定表，如表所示。

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 原因 | C1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| C2 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| C3 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| 中间状态 | 11 |  |  | 1 | 1 | 1 | 1 | 0 | 0 |
| 结果 | E2 |  |  | 0 | 0 | 0 | 0 | 1 | 1 |
| E1 |  |  | 1 | 0 | 1 | 0 | 0 | 0 |
| E3 |  |  | 0 | 1 | 0 | 1 | 0 | 1 |
| 测试用例输入： | | 不存在 | 不存在 | A3 | AF | B2 | By | T6 | UI |
| 测试用例结果： | |  |  | 修改文件 | 打印错误信息X13 | 修改文件 | 打印错误信息X13 | 打印错误信息X12 | 打印错误信息X12，打印错误信息X13 |

  
根据判定表设计测试用例，如表所示。  


1. 各种来源导致缺陷会广泛分布在软件开发的各个阶段，需求规格说明书、软件设计、代码中都可以看到缺陷的身影。特别是由于需求的变化和人们对文档的轻视，导致**需求规格说明书中的缺陷**通常会占缺陷总数一半还多。
2. 你已经答对了。
3. **黑盒测试**是将被测试程序看成一个黑盒子,不考虑程序内部结构的情况,而只考虑程序的输入与输出之间的关系。
4. 等价类测试的核心难点是**划分等价类，列出等价类列表。**
5. 软件性能测试就是用来测试软件在系统中的运行**性能**的。感觉这个题很搞笑，我觉得重点应该考（在实际的集成系统中运行性能）
6. **恢复性测试**是检验系统从软件或硬件,包括从程序错误或数据错误中恢复的能力恢复测试
7. **缺陷的优先级**指缺陷必须被修复的紧急程度。
8. **缺陷密度**发现缺陷总数与软件规模的比值。
9. 单元测试不考虑每个模块与其他模块之间的关系，为每个模块设计**桩模块和驱动模块**
10. **我也不是非常确定，是发现变量关系缺陷吗**