

**《软件测试基本技能实践》**

**实 训 指 导 书**

**系 别： 计 算 机**

**专 业 名 称： 软件技术**

**班 级： 18高 职**

**指 导 教 师： 何 元 娇**

**学 号： 2019**

**姓 名： NAME**

目录

[**实训1 英文版Windows7安装2种方法** 1](#_Toc12433450)

[**实训2 Word2003文件名与文件打开方式测试** 3](#_Toc12433451)

[**实训3 三角形问题和电话号码问题测试** 12](#_Toc12433452)

[**实训4 日期问题测试** 22](#_Toc12433453)

[**实训5 逻辑覆盖测试方法** 28](#_Toc12433454)

[**实训6 基本路径测试方法** 40](#_Toc12433455)

[**实训7 web网站测试** 53](#_Toc12433456)

[**实训8 兼容性测试** 58](#_Toc12433457)

**实训1 英文版Windows7安装2种方法**

班级： X班 学号： 2018 姓名：NAME

日期：2019 地点： PLECE

**一、实训目的**

1、能熟练安装系统，并对系统进行相应的配置。

2、软件兼容性，安全性等测试。

3、软件的安装和使用，其中包括系统软件（操作系统）和应用软件。

4、硬盘分区、格式化。

5、系统备份与恢复。

**二、实训环境**

虚拟机VMware Workstation，en\_windows\_7\_ultimate\_x64\_dvd\_x15-65922.iso，

，360浏览器和杀毒软件，输入法（如：搜狗拼音输入法\_8.5.0.1244），QQ，1KG\_20170217\_HD ，Office2010，Microsoft Visual C++ 6.0等软件。

**三、背景知识**

1. 虚拟机中默认安装-》不可以分区，不可以装一键GHOST硬盘版-》需要在虚拟机开始设置密码
2. 虚拟机中设置后安装-》可以分区

**四、实训任务**

1. 根据系统是否分区，用2种方法安装英文win7系统。
2. 系统装好后，测试系统。系统配置：桌面图标、分辨率，分区等；中文识别测试及解决方法。
3. 识别汉字；安装VMtools，安装搜狗输入法等。
4. 网络测试：百度，QQ等。
5. 系统分区、备份、还原。
6. 一键ghost备份和恢复。并对备份和恢复进行测试。

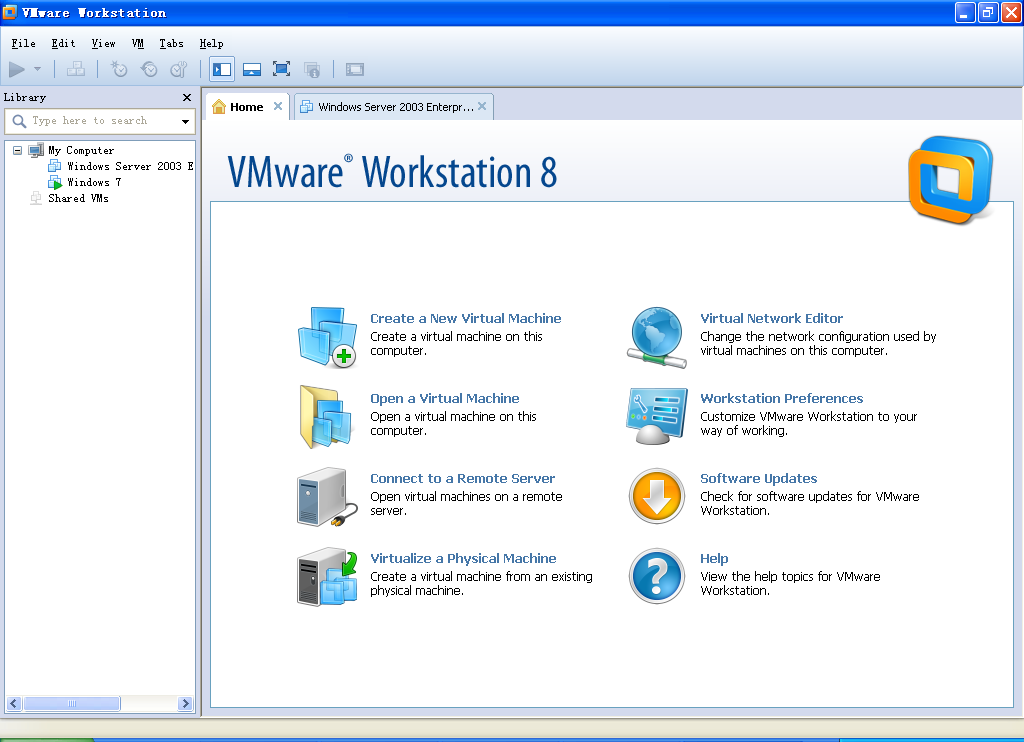
* http://doshome.com/yj/



1. 安装软件：360浏览器和杀毒软件等。测试软件兼容性等性能。
2. 火狐浏览器与seleniumIDE安装，录制脚本测试
3. 安装Microsoft Visual C++ 6.0，测试安装是否成功。
4. 安装Microsoft Office2010，测试安装是否成功；再安装一级考试系统，测试安装是否成功。（附加题）
5. 中文桌面（附加题）

**五、实训步骤（关键步骤截屏，并附加文字加以说明）**

首先打开VMware Workstation点击添加新的虚拟机页面如下图所示。

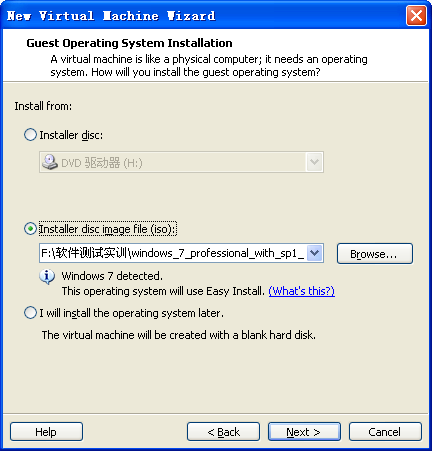


**选择Create a New Virtual Machine创建一个新的虚拟机**

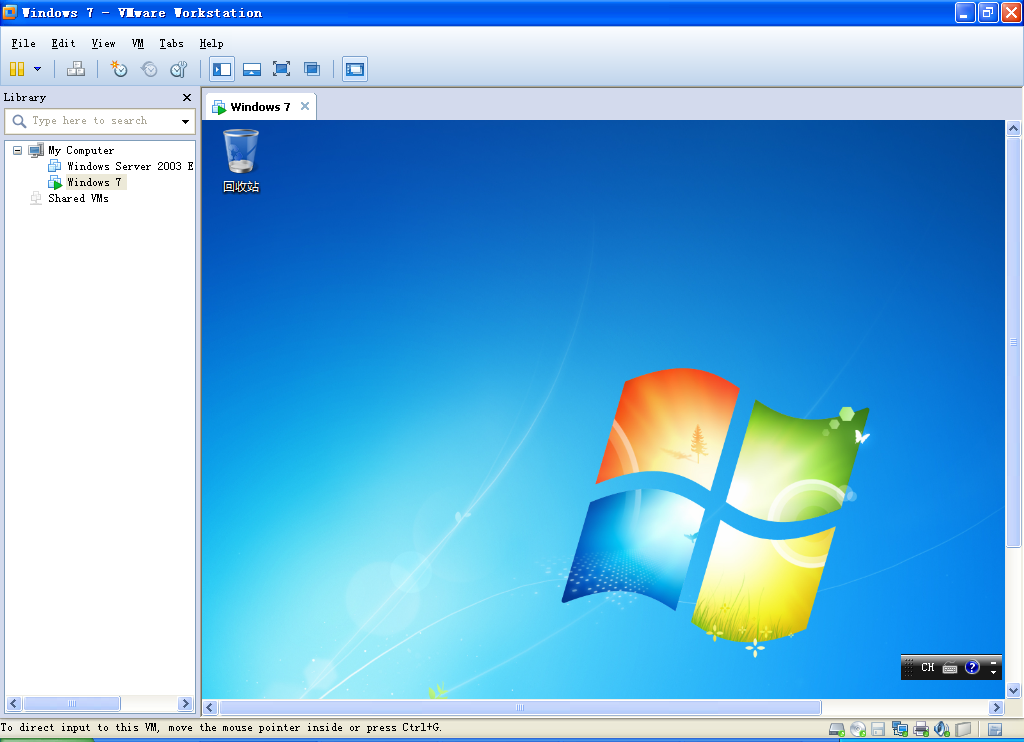


**点击下一步，选择需要安装的虚拟机文件**

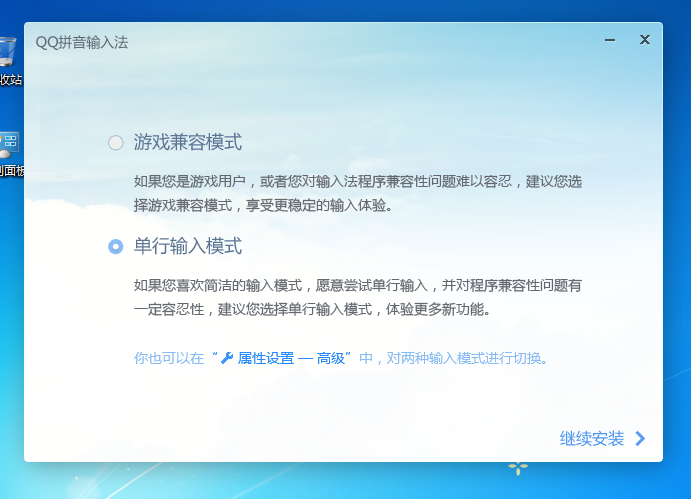
]



**点击Next即可。虚拟机自动安装。完成安装后如下图所示。**



**安装完成后开始安装所需要的软件（可以在虚拟机中直接下载也可以从真机中将安装包直接拖入虚拟机机中进行安装）。一键安装即可。**



**六、实训小结**

在安装虚拟机时需要安装VMtools才可以将虚拟机与真机连通。在安装各种软件时也需要考虑软件与虚拟机的兼容性等问题。

**实训2 Word2003文件名与文件打开方式测试**

班级： X班 学号： 2018 姓名：NAME

日期：2019 地点： PLECE

**一、实训目的**

1、能熟练应用黑盒测试技术（边界值分析法）进行测试用例设计

2、对测试用例进行优化设计

**二、实训环境**

操作系统：Windows XP

应用系统：Office工具

**三、背景知识**

测试用例概念及测试用例设计方法；文件名的命名规则；黑盒测试方法(边界值分析法)。

1. 文件名的命名规则特点：文件名由文件主名和扩展名组成，文件主名由1~8个字符组成，扩展名由1~3个字符组成，主名和扩展名之间由一个小圆点隔开，一般称为8.3规则。其格式如下：

□□□□□□□□. □□□

例如： G9401.DBF，这里G9401是主名，DBF是扩展名。文件主名和扩展名可以使用的字符是：

（1）英文字母：A—z（大小写等价）

（2）数 字：0—9

（3）汉 字

（4）特殊符号：＄＃＆＠（）-[]^～等。

（5）注意点：

① 现在流行的Windows操作系统已经突破了8.3规则，至多255个字符(包括空格)

② 不能包含以下9个字符：

/ \ : \* ? “ < > |

③ 不区分大小写：ＬＯＶＥ与love相同

④ 可以使用汉字和多分隔符：

　 我的小说.my novel .fan.nice.txt

⑤ 查找和显示时可使用通配符“\*”和“？”。

⑥ 文件一般有1~3个字符组成的扩展名，用以标识文件类型；文件夹没有扩展名。

2. 边界值分析是考虑边界条件而选取测试用例的一种功能测试方法。边界值分析关注输入空间的边界，以标识测试用例，因为错误更可能出现在输入变量的极值附近。

**四、实训任务**

1. 文件名测试

1.1 中文文件名

1.2 西文文件名

1.3 中西文混合文件名

1.4 特殊字符文件名

2. 打开方式测试

2.1 菜单方式

2.2 快捷键方式

**五、实训步骤**

**1. 文件名测试**

**1.1 中文文件名**

测试用：CSYL\_ WJM\_中文1，考虑一个普通的中文文件名

前提：软/硬盘或网络资源中有一个中文命名的Word文档。文件名长度为8+3 字节。

操作：在文件打开对话框中找到该文件所在位置，鼠标双击打开该文件。

验证：该文件可在Word2003中打开。

测试用例：CSYL\_ WJM\_中文2，考虑一个最短的中文文件名

前提：软/硬盘或网络资源中有一个中文命名的Word文档。文件名长度为2+3 字节。例如：测.Doc。

操作：在文件打开对话框中找到该文件所在位置，鼠标双击打开该文件。

验证：该文件可在Word2003中打开。

测试用例：CSYL\_ WJM\_中文3，考虑一个中文数字混合的并且以数字开头的文件名。

前提：软/硬盘或网络资源中有一个中文数字混合命名的Word文档。

例如：1测试用例.Doc。

操作：在文件打开对话框中找到该文件所在位置，鼠标双击打开该文件。

验证：该文件可在Word2003中打开。

测试用例：CSYL\_ WJM\_中文4，考虑一个中文数字混合的并且以中文字符开头的文件名。

前提：软/硬盘或网络资源中有一个中文数字混合命名的Word文档。

例如：测试用例1.Doc。

操作：在文件打开对话框中找到该文件所在位置，鼠标双击打开该文件。

验证：该文件可在Word2003中打开。

测试用例：CSYL\_ WJM\_中文5，考虑一个包含空格字符的中文文件名。

前提：软/硬盘或网络资源中有一个包含空格的中文命名的Word文档。

例如：测试 用例.Doc。

操作：在文件打开对话框中找到该文件所在位置，鼠标双击打开该文件。

验证：该文件可在Word2003中打开。

测试用例：CSYL\_ WJM\_中文6，考虑一个包含标点符号的并以符号开头的中文文件名。请考虑如下符号：~`！@#%$^&\*()-+{}[]|\:;”’<>,.?/。

前提：软/硬盘或网络资源中有一个包含标点符号的以标点符号开头的中文命名的Word文档。例如：@测试用例.Doc。

操作：在文件打开对话框中找到该文件所在位置，鼠标双击打开该文件。

验证：该文件可在Word2003中打开。

测试用例：CSYL\_ WJM\_中文7，考虑一个包含标点符号的中文文件名。请考虑如下符号：~`！@#%$^&\*()-+{}[]|\:;”’<>,.?/。

前提：软/硬盘或网络资源中有一个包含标点符号的以中文命名的Word文档。

例如：测试用例?.Doc。

操作：在文件打开对话框中找到该文件所在位置，鼠标双击打开该文件。

验证：该文件可在Word2003中打开。

测试用例：CSYL\_ WJM\_中文8，考虑一个中文、数字、标点符号和空格混合的文件名。

前提：软/硬盘或网络资源中有一个中文、数字、标点符号和空格混合命名的Word文档。例如：测试用例1 @ 场所.Doc。

操作：在文件打开对话框中找到该文件所在位置，鼠标双击打开该文件。

验证：该文件可在Word2003中打开。

测试用例：CSYL\_ WJM\_中文9，考虑一个最长的中文文件名。

前提：存在这样一个文件，文件名长度为251+3字节。例如：125汉字+1个数字.Doc。

操作：在文件打开对话框中找到该文件所在位置，鼠标双击打开该文件。

验证：该文件可在Word2003中打开。

测试用例：CSYL\_ WJM\_中文10，考虑一个比最长允许文件长度少一个字节的中文文件名。

前提：存在这样一个文件，文件名长度为250+3字节。

操作：在文件打开对话框中找到该文件所在位置，鼠标双击打开该文件。

验证：该文件可在Word2003中打开。

测试用例：CSYL\_ WJM\_中文11，考虑一个不存在的中文文件名。

操作：在文件打开对话框中填写一个不存在的文件名，点击“打开”。

验证：软件与系统无错误信息提示，同时文件打开对话框保持打开状态。

1.2 西文文件名

将中文文件名替换为英文，重复 1.1 中文文件名的测试用例。注意英文字母长度为单字节。

1.3 中西文混合文件名

考虑中英文混合的文件名称，重复 1.1 中文文件名的测试用例。

1.4 特殊字符文件名

考虑特殊的字符集在文件名中的使用。请查阅特殊的字符在文件名中的使用规定来确定在Word2003文件名中支持的字符。

表1.1 中文文件名测试用例表

|  |  |  |
| --- | --- | --- |
| 测试用例 | 文件名 | 预期输出 |
| CSYL\_ WJM\_中文1 | 测试 | 命名成功且文件可用 |
| CSYL\_ WJM\_中文2 | 测.doc | 命名成功且文件可用 |
| CSYL\_ WJM\_中文3 | 1测试 | 命名成功且文件可用 |
| CSYL\_ WJM\_中文4 | 测试1 | 命名成功且文件可用 |
| CSYL\_ WJM\_中文5 | 测 试 | 命名成功且文件可用 |
| CSYL\_ WJM\_中文6 | ？测试 | 命名成功且文件可用 |
| CSYL\_ WJM\_中文7 | 测试？ | 命名成功且文件可用 |
| CSYL\_ WJM\_中文8 | 测试1？ | 命名成功且文件可用 |
| CSYL\_ WJM\_中文9 |  | 命名成功且文件可用 |
| CSYL\_ WJM\_中文10 |  | 命名成功且文件可用 |
| CSYL\_ WJM\_中文11 |  | 命名成功且文件可用 |

表1.2 西文文件名测试用例表

|  |  |  |
| --- | --- | --- |
| 测试用例 | 文件名 | 预期输出 |
| CSYL\_ WJM\_西文1 | Tc | 命名成功且文件可用 |
| CSYL\_ WJM\_西文2 | c | 命名成功且文件可用 |
| CSYL\_ WJM\_西文3 | 1c | 命名成功且文件可用 |
| CSYL\_ WJM\_西文4 | c1 | 命名成功且文件可用 |
| CSYL\_ WJM\_西文5 | T c | 命名成功且文件可用 |
| CSYL\_ WJM\_西文6 | ？c | 命名成功且文件可用 |
| CSYL\_ WJM\_西文7 | C？ | 命名成功且文件可用 |
| CSYL\_ WJM\_西文8 | C1？ | 命名成功且文件可用 |
| CSYL\_ WJM\_西文9 |  | 命名成功且文件可用 |
| CSYL\_ WJM\_西文10 |  | 命名成功且文件可用 |
| CSYL\_ WJM\_西文11 |  | 命名成功且文件可用 |

表1.3 特殊字符文件名测试用例表

|  |  |  |
| --- | --- | --- |
| 测试用例 | 文件名 | 预期输出 |
| CSYL\_ WJM\_特殊字符1 |  |  |
| CSYL\_ WJM\_特殊字符2 |  |  |
| CSYL\_ WJM\_特殊字符3 |  |  |
| CSYL\_ WJM\_特殊字符4 |  |  |
| CSYL\_ WJM\_特殊字符5 |  |  |
| CSYL\_ WJM\_特殊字符6 |  |  |
| CSYL\_ WJM\_特殊字符7 |  |  |
| CSYL\_ WJM\_特殊字符8 |  |  |
| CSYL\_ WJM\_特殊字符9 |  |  |
| CSYL\_ WJM\_特殊字符10 |  |  |
| CSYL\_ WJM\_特殊字符11 |  |  |

2. 打开方式测试

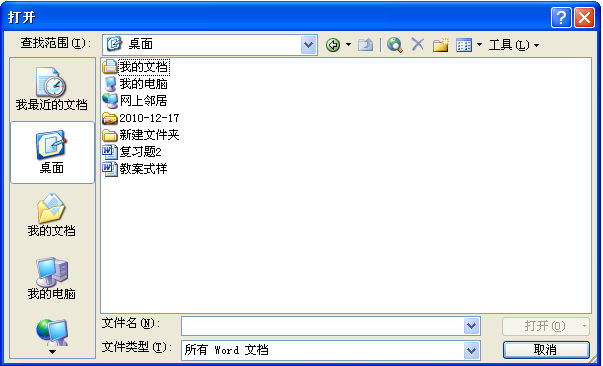


图1.1 文件打开对话框图

2.1 菜单方式

测试用例：CSYL\_ DK\_菜单

操作：在菜单中选择“文件->打开”

验证：“打开”对话框被打开

2.2 快捷键方式

测试用例：CSYL\_ DK\_快捷键

操作：键盘组合“Ctrl+O”

验证：“打开”对话框被打开

表1.4 打开方式测试用例表

|  |  |  |
| --- | --- | --- |
| 测试用例 | 操作 | 预期输出 |
| CSYL\_ DKFS\_菜单 |  |  |
| CSYL\_ DKFS\_快捷键 |  |  |

**六、实训小结**

**实训3 三角形问题和电话号码问题测试**

班级： X班 学号： 2018 姓名：NAME

日期：2019 地点： PLECE

**一、实训目的**

1、能熟练应用黑盒测试技术（等价类划分法和边界值分析法）进行测试用例设计

2、对测试用例进行优化设计

**二、实训环境**

操作系统：Windows XP

应用系统：Visual C++开发工具和Office工具

**三、背景知识**

功能测试是指在对程序进行功能抽象的基础上，将程序划分成功能单元，然后在数据抽象的基础上，对每个功能单元生成测试数据进行测试。进行功能测试时，被测程序被当作打不开的黑盒，因而无法了解其内部构造，因此又称为黑盒测试。

（一）、 等价类测试

等价类测试方法是把所有可能的输入数据，即程序的输入域划分成若干部分，然后从每一部分中选取少数有代表性的数据作为测试用例。使用等价类划分方法设计测试用例要经历划分等价类（列出等价类表）和选取测试用例两步。

等价类的划分有两种不同的情况： ① 有效等价类：是指对于程序的规格说明来说，是合理的，有意义的输入数据构成的集合。 ② 无效等价类：是指对于程序的规格说明来说，是不合理的，无意义的输入数据构成的集合。在设计测试用例时，要同时考虑有效等价类和无效等价类的设计。

用等价类划分法设计测试用例步骤：

(1) 形成等价类表，每一等价类规定一个唯一的编号；

(2) 设计一个新的测试用例，使其尽可能多地覆盖尚未覆盖的有效等价类，重复这一步骤，直到所有有效等价类均被测试用例所覆盖；

(3) 设计一个新测试用例，使其只覆盖一个无效等价类，重复这一步骤直到所有无效等价类均被覆盖。

（二）、 边界值测试

1． 边界值分析

边界值分析是考虑边界条件而选取测试用例的一种功能测试方法。边界值分析关注输入空间的边界，以标识测试用例，因为错误更可能出现在输入变量的极值附近。

边界值分析的基本思想是：使用在最小值、略高于最小值、正常值、略低于最大值和最大值处取输入变量值。

2．健壮性测试

健壮性是指在异常情况下，软件还能正常运行的能力。健壮性考虑的主要部分是预期输出，而不是输入。

健壮性测试是边界值分析的一种简单扩展。除了变量的5 个边界分析取值还要考虑略超过最大值（max）和略小于最小值（min）时的情况。

3． 最坏情况测试

最坏情况测试将意味着更大工作量，n 变量函数的最坏情况测试会产生5 的n 次方个测试用例，而边界值分析只产生4n+1 个测试用例。

**四、实训任务**

1．三角形问题

2．电话号码问题

**五、实训步骤**

**1. 三角形问题测试**

根据下面给出的规格说明，利用等价类划分的方法，给出足够的测试用例。

三角形问题：输入三个整数a、b、c分别作为三角形的三条边，现通过程序判断由三条边构成的三角形的类型为等边三角形、等腰三角形、一般三角形（特殊的还有直角三角形）以及构不成三角形。现在要求输入三个整数a、b、c，必须满足以下条件：

条件1 1≤a≤100 条件4 a≤b+c

条件2 1≤b≤100 条件5 b≤a+c

条件3 1≤c≤100 条件6 c≤a+b

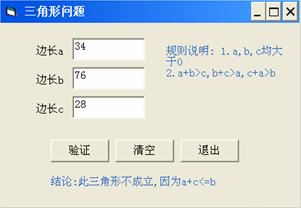


图2.1 三角形示意图

**（1）步骤：**

* 1. 根据功能性测试技术设计测试用例，主要考虑等价类划分和边界值分析测试技术；
  2. 根据所学知识确定优化策略（原则：用最少的用例检测出更多的缺陷、软件测试的充分性与冗余性考虑），设计一套测试用例集；
  3. 根据设计的一套测试用例集进行测试；

**（2）填空**

表2.1 三角形等价类表

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 有效等价类 | 编号 | 无效等价类 |
| 1 | 1≤a≤100 | 7 | a<1 |
| 8 | a>100 |
| 2 | 1≤b≤100 | 9 | b<1 |
| 10 | b>100 |
| 3 | 1≤c≤100 | 11 | c<1 |
| 12 | c>100 |
| 4 | a≤b+c | 13 | a≥b+c |
| 5 | b≤a+c | 14 | b≥a+c |
| 6 | c≤a+b | 15 | c≥a+b |

表2.2 三角形测试用例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test case | a  a | b  b | c  c | 观察记录/预期输出 |
| CSYL\_ SJX\_1 | 50 | 50 | 50 | 等边三角形 |
| CSYL\_ SJX\_2 | 50 | 50 | 80 | 等腰三角形 |
| CSYL\_ SJX\_3 | 30 | 40 | 50 | 直角三角形 |
| CSYL\_ SJX\_4 | -1 | 40 | 30 | a= -1不满足条件1，不能够成三角形 |
| CSYL\_ SJX\_5 | 101 | 40 | 30 | a= 101不满足条件1，不能够成三角形 |
| CSYL\_ SJX\_6 | 40 | -1 | 30 | b= -1不满足条件2，不能够成三角形 |
| CSYL\_ SJX\_7 | 40 | 101 | 30 | b= 101不满足条件2，不能够成三角形 |
| CSYL\_ SJX\_8 | 30 | 40 | -1 | c= -1不满足条件3，不能够成三角形 |
| CSYL\_ SJX\_9 | 30 | 40 | 101 | c= 101不满足条件3，不能够成三角形 |
| CSYL\_ SJX\_10 | 30 | 25 | 5 | 不满足a<b+c，不能构成三角形 |
| CSYL\_ SJX\_11 | 25 | 30 | 5 | 不满足b<a+c，不能构成三角形 |
| CSYL\_ SJX\_12 | 25 | 25 | 60 | 不满足c<a+b，不能构成三角形 |

**（3）界面显示问题**

请指出图2.2三角形问题示意图中，界面设计有哪些错误？（至少三条）

表2.3 界面设计错误表

|  |  |
| --- | --- |
| 错误1 | 不能构成三角形，则就没有右上角的三角形图示 |
| 错误2 | 点击清空后，界面的结论提示信息也应清空 |
| 错误3 | 点击最大/还原按钮后，最大化没有按比例显示 |

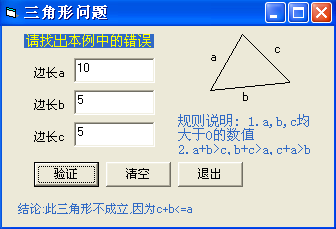


图2.2 三角形问题示意图

**2. 电话号码问题测试**

**问题：**某城市电话号码由三部分组成，它们的名称和内容分别是：

（1）地区码：空白或四位数字；

（2）连接符：“-”；

（3）号码：8 位数字。

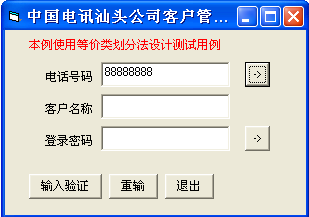


图2.3 电话号码示意图

**填空题：**

表2.4 电话号码等价类表

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 有效等价类 | 编号 | 无效等价类 |
| 1 | 首码不为- | 5 | 首码为- |
| 2 | 位数：8位或13位 | 6 | 位数不为8位或13位 |
| 3 | 无区号8位数字 | 7 | 位数不为8位 |
| 8 | 不全为数字 |
| 4 | 带区号4位数字-8位数字 | 9 | 区号不为4位 |
| 10 | 区号不为数字 |
| 11 | 带区号4位数字-非8位数字 |
| 12 | 带区号4位数字-8位非数字 |

表2.5 电话号码试用例表

|  |  |  |
| --- | --- | --- |
| Test case | 电话号码  a | 观察记录/预期输出 |
| CSYL\_ DHHM\_1 | 00000000 | 您输入的电话是合法的 |
| CSYL\_ DHHM \_2 | 33232880 | 您输入的电话是合法的 |
| CSYL\_ DHHM \_3 | 99999999 | 您输入的电话是合法的 |
| CSYL\_ DHHM \_4 | -7773890 | 电话号码为8位或13位，但不能为负数 |
| CSYL\_ DHHM \_5 | 8232999 | 电话号码应为8位或13位 |
| CSYL\_ DHHM \_6 | 682329997 | 电话号码应为8位或13位 |
| CSYL\_ DHHM \_7 | 12312--3498902 | 电话号码应为8位或13位 |
| CSYL\_ DHHM \_8 | 823299aa | 电话号码必须是数字 |
| CSYL\_ DHHM \_9 | a8232991 | 电话号码必须是数字 |
| CSYL\_DHHM \_10 | 123-49866902 | 电话号码应为8位或13位 |
| CSYL\_DHHM \_11 | 1231-49866902 | 您输入的电话是合法的 |
| CSYL\_DHHM \_12 | 12351-94866902 | 电话号码应为8位或13位 |
| CSYL\_ DHHM \_13 | 123a-94866902 | 区号必须为4位数字 |
| CSYL\_ DHHM \_14 | 1111-1234567 | 电话号码应为8位或13位 |
| CSYL\_ DHHM \_15 | 1111-12345678 | 您输入的电话是合法的 |
| CSYL\_ DHHM \_16 | 1111-123456789 | 电话号码应为8位或13位 |
| CSYL\_ DHHM \_17 | 1111-1234567a | 号码必须为8位数字 |
| CSYL\_ DHHM \_18 | 12312-3498902 | 区号与区号之间必须是-号 |
| CSYL\_ DHHM \_19 | 1298-+8889991 | 您输入的电话是合法的 |

**六、实训小结**

**实训4 日期问题测试**

班级： X班 学号： 2018 姓名：NAME

日期：2019 地点： PLECE

**一、实训目的**

1、能熟练应用黑盒测试技术（因果图法、决策表法）进行测试用例设计。

2、对测试用例进行优化设计。

**二、实训环境**

操作系统：Windows XP

应用系统：Visual C++开发工具和Office工具

**三、背景知识**

基于决策表的测试：决策表适合描述不同条件集合下采取行动的若干组合的情况。使用决策表标识测试用例，则把条件解释为输入，行动解释为输出。有时条件最终引用输入的等价类，行为引用被测试软件的主要功能处理部分，规则解释为测试用例。

对于有限条目决策表，如果有n 个条件，则必须有2 条规则。如果不关心条目实际地表明条件是不相关的，则没有不关心条目的规则统计为1 条规则，规则中每出现一个不关心条目，该规则数乘一次2。

**四、实训任务**

日期问题：用决策表测试法测试以下程序:该程序有三个输入变量month、day、year（month 、 day 和year均为整数值，并且满足：1≤month≤12 和1≤day≤31),分别作为输入日期的月份、日、年份，通过程序可以输出该输入日期在日历上隔一天的日期。例如，输入为 2004 年11 月29 日,则该程序的输出为2004 年12 月1 日。

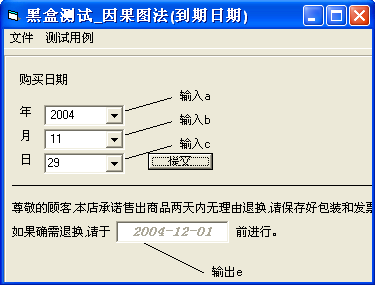


图3.1 日期到期图

**五、实训步骤**

**1. 分析**

(1) 分析各种输入情况，列出为输入变量 month 、 day 、 year 划分的有效等价类。

(2) 分析程序的规格说明，并结合以上等价类划分的情况，给出问题规定的可能采取的操作（即列出所有的动作桩）。

(3) 根据 (1) 和 (2) ，画出简化后的决策表。

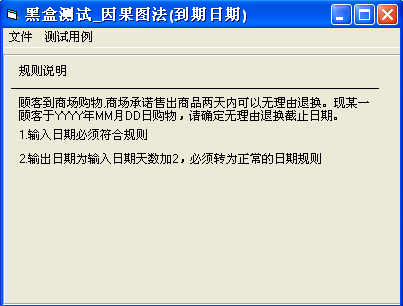


图3.2 日期规则说明图

**2. 步骤：**

（1） 根据功能性测试技术设计测试用例，主要考虑等价类划分和边界值分析测试技术；

（2） 根据所学知识确定优化策略（原则：用最少的用例检测出更多的缺陷、软件测试的充分性与冗余性考虑），设计一套测试用例集；

（3） 根据设计的一套测试用例集进行测试；

**3. 附录：** 因果图 决策表

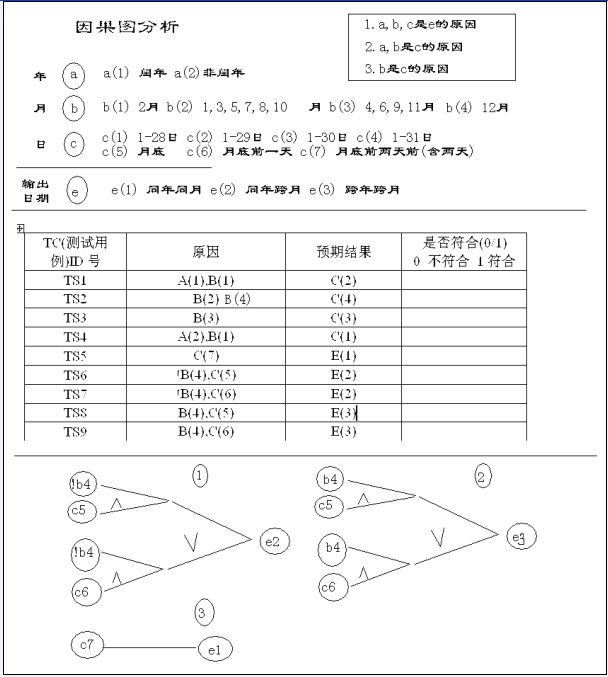


图3.3 日期因果图

**4. 填空题：**只测试输入输出关系，是否符合Y/N(1/0)；参照表3.1设计测试用例表3.2 。

表3.1 日期测试用例表（2009年，2008年）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | A | B | C(result) | E(result) | Y/N(1/0) |
| TS1 | 2009 | 6 | 30 | 2009-7-2 | Y |
| TS2 | 2009 | 6 | 29 | 2009-7-1 | Y |
| TS3 | 2009 | 2 | 28 | 2009-3-2 | Y |
| TS4 | 2009 | 2 | 27 | 2009-3-1 | Y |
| TS5 | 2009 | 6 | 1 | 2009-6-3 | Y |
| TS6 | 2009 | 6 | 28 | 2009-6-30 | Y |
| TS7 | 2009 | 2 | 1 | 2009-2-3 | Y |
| TS8 | 2009 | 2 | 26 | 2009-2-28 | Y |
| TS9 | 2009 | 12 | 1 | 2009-12-3 | Y |
| TS10 | 2009 | 12 | 29 | 2009-12-31 | Y |
| TS11 | 2009 | 12 | 30 | 2010-1-1 | Y |
| TS12 | 2009 | 12 | 31 | 2009-1-2 | Y |
| TS21 | 2008 | 2 | 29 | 2008-3-2 | Y |
| TS22 | 2008 | 2 | 28 | 2008-3-1 | Y |
| TS23 | 2008 | 2 | 1 | 2008-2-3 | Y |
| TS24 | 2008 | 2 | 27 | 2008-2-29 | Y |

表3.1日期测试用例表（2010年）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | A | B | C(result) | E(result) | Y/N(1/0) |
| TS1 | 2010 | 2 | 1 | 2010-2-3 | Y |
| TS2 | 2010 | 2 | 26 | 2010-2-28 | Y |
| TS3 | 2010 | 2 | 27 | 2010-3-1 | Y |
| TS4 | 2010 | 2 | 28 | 2010-3-2 | Y |
| TS5 | 2010 | 6 | 1 | 2010-6-3 | Y |
| TS6 | 2010 | 6 | 28 | 2010-6-30 | Y |
| TS7 | 2010 | 6 | 29 | 2010-7-1 | Y |
| TS8 | 2010 | 6 | 30 | 2010-7-2 | Y |
| TS9 | 2010 | 12 | 1 | 2010-12-3 | Y |
| TS10 | 2010 | 12 | 29 | 2010-12-31 | Y |
| TS11 | 2010 | 12 | 30 | 2010-1-1 | Y |
| TS12 | 2010 | 12 | 31 | 2010-1-2 | Y |

**六、实训小结**

**实训5 逻辑覆盖测试方法**

班级： X班 学号： 2018 姓名：NAME

日期：2019 地点： PLECE

**一、实训目的**

1、能熟练应用白盒测试技术（逻辑覆盖方法）进行测试用例设计。

2、对测试用例进行优化设计。

**二、实训环境**

操作系统：Windows XP

应用系统：Visual C++开发工具和Office工具

**三、背景知识**

结构性测试是知道产品内部工作过程，检测产品内部动作是否按照规格说明书的规定正常进行。

结构性测试允许测试人员利用程序内部的逻辑结构及有关信息，设计或选择测试用例，对程序所有逻辑路径进行测试。通过在不同点检查程序的状态，确定实际的状态是否与预期的状态一致。

一、 逻辑覆盖

结构性测试力求提高测试覆盖率。逻辑覆盖是对一系列测试过程的总称，它是在使用白盒测试法时，选用测试用例执行程序逻辑路径的方法。

逻辑覆盖按覆盖程度由低到高大致分为以下几类：

（1） 语句覆盖：设计若干测试用例，使程序中每一可执行语句至少执行一次；

（2） 判断覆盖：设计用例，使程序中的每个逻辑判断的取真取假分支至少经历一次；

（3） 条件覆盖：设计用例，使判断中的每个条件的可能取值至少满足一次；

（4） 判断/条件覆盖：设计用例，使得判断中的每个条件的所有可能结果至少出现一次，而且判断本身所有可能结果也至少出现一次；

（5） 条件组合覆盖。设计用例，使得每个判断表达式中条件的各种可能组合都至少出现一次；

显然，满足⑤的测试用例也一定是满足②、③、④的测试用例。

（6） 路径覆盖。设计足够的测试用例，使程序的每条可能路径都至少执行一次。

如果把路径覆盖和条件组合覆盖结合起来，可以设计出检错能力更强的测试数据用例。

**四、实训任务**

使用逻辑覆盖测试方法测试以下程序段：

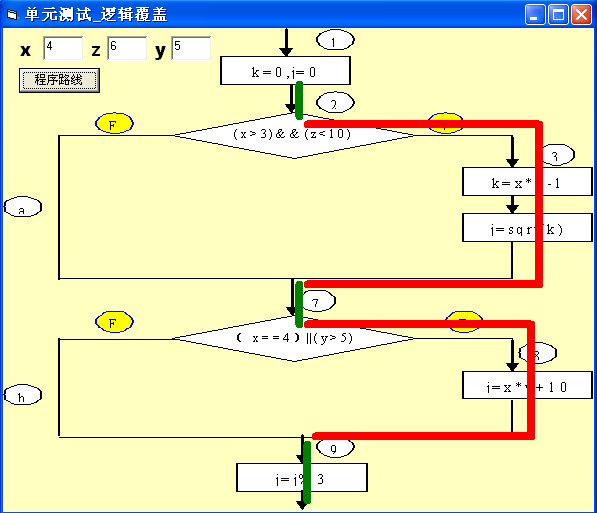


图4.1 逻辑覆盖测试图

**五、实训步骤**

**1. 说明：**

程序段中每行开头的数字（1~10）是对每条语句的编号。

（1）画出程序的控制流图（用题中给出的语句编号表示）。

（2）分别以语句覆盖、判定覆盖、条件覆盖、判定/条件覆盖、组合覆盖和路径覆盖方法设计测试用例，并写出每个测试用例的执行路径（用题中给出的语句编号表示）；决策表。

**2. 步骤：**

（1） 根据逻辑覆盖方法设计测试用例，主要考虑语句覆盖、判定覆盖、条件覆盖、判定/条件覆盖、组合覆盖和路径覆盖方法；

（2） 根据所学知识确定优化策略（原则：用最少的用例检测出更多的缺陷、软件测试的充分性与冗余性考虑），设计一套测试用例集；

（3） 根据设计的一套测试用例集进行测试；

**3. 任务：**

补充完成：流程图；表中内容

（1）、流程图

j=j%3

（2）、决策表

① 语句覆盖

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *序号* | *x* | *z* | *y* | **路径** |
| 1 | 4 | 6 | 7 |  |
|  |  |  |  |  |

② 判断覆盖

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| *序号* | *x>3* | *z<10* | *(x>3)&&(z<10)* | *x==4* | *y>5* | *（x==4）||(y>5)* | **路径** |
| 1 | 4(T) | 6(T) | T | T | 6(T) | T |  |
| 2 |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

③ 条件覆盖

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| *序号* | *x>3* | *z<10* | *(x>3)&&(z<10)* | *x==4* | *y>5* | *（x==4）||(y>5)* | **路径** |
| 1 | 4(T) | 11(F) | F | T | 3(F) | T |  |
| 2 |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

④ 判断/条件覆盖

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| *序号* | *x>3* | *z<10* | *(x>3)&&(z<10)* | *x==4* | *y>5* | *（x==4）||(y>5)* | **路径** |
| 1 |  |  |  |  |  |  |  |
| 2 | 3(F) | 11(F) | F | F | 3(F) | F |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

⑤ 多条件组合覆盖

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| *序号* | *x>3* | *z<10* | *(x>3)&&(z<10)* | *x==4* | *y>5* | *（x==4）||(y>5)* | **路径** |
| 1 | 4(T) | 6(T) | T | T | 6(T) | T |  |
| 2 | 4(T) | 6(T) | T | T | 3(F) | T |  |
| 3 | 5(T) | 6(T) | T | F | 6(T) | T |  |
| 4 | 5(T) | 6(T) | T | F | 3(F) | F |  |
| 5 | 4(T) | 11(F) | F | T | 6(T) | T |  |
| 6 | 4(T) | 11(F) | F | T | 3(F) | T |  |
| 7 | 5(T) | 11(F) | F | F | 6(T) | T |  |
| 8 | 5(T) | 11(F) | F | F | 3(F) | F |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
| 9 |  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

⑥ 路径覆盖

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| *序号* | *x>3* | *z<10* | *(x>3)&&(z<10)* | *x==4* | *y>5* | *（x==4）||(y>5)* | **路径** |
| 1 | 4(T) | 6(T) | T | T | 6(T) | T |  |
| 2 |  |  |  |  |  |  |  |
| 3 |  |  |  |  |  |  |  |
| 4 | 5(T) | 11(F) | T | F | 3(F) | F |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

**六、实训小结**

**参考答案：**

1、流程图

j=sqrt(k)

j=j%3

（x==4）||(y>5)

j=x\*y+10

(x>3)&&(z<10)

k=0,j=0

k=x\*y-1

2、决策表

（1）语句覆盖

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *序号* | *x* | *z* | *y* | **路径** |
| 1 | 4 | 6 | 7 | 23789 |
|  |  |  |  |  |

（2）判断覆盖

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| *序号* | *x>3* | *z<10* | *(x>3)&&(z<10)* | *x==4* | *y>5* | *（x==4）||(y>5)* | **路径** |
| 1 | 4(T) | 6(T) | T | T | 6(T) | T | 23789 |
| 2 | 3(F) | 8(F) | F | F | 3(F) | F | 2a7b9 |
|  |  |  |  |  |  |  |  |

（3）条件覆盖

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| *序号* | *x>3* | *z<10* | *(x>3)&&(z<10)* | *x==4* | *y>5* | *（x==4）||(y>5)* | **路径** |
| 1 | 4(T) | 11(F) | F | T | 3(F) | T | 2a789 |
| 2 | 3(F) | 6(T) | F | F | 6(T) | T | 2a7b9 |
|  |  |  |  |  |  |  |  |

（4）判断/条件覆盖

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| *序号* | *x>3* | *z<10* | *(x>3)&&(z<10)* | *x==4* | *y>5* | *（x==4）||(y>5)* | **路径** |
| 1 | 4(T) | 6(T) | T | T | 6(T) | T | 23789 |
| 2 | 3(F) | 11(F) | F | F | 3(F) | F | 2a7b9 |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

（5）多条件组合覆盖

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| *序号* | *x>3* | *z<10* | *(x>3)&&(z<10)* | *x==4* | *y>5* | *（x==4）||(y>5)* | **路径** |
| 1 | 4(T) | 6(T) | T | T | 6(T) | T | 23789 |
| 2 | 4(T) | 6(T) | T | T | 3(F) | T | 23789 |
| 3 | 5(T) | 6(T) | T | F | 6(T) | T | 23789 |
| 4 | 5(T) | 6(T) | T | F | 3(F) | F | 237b9 |
| 5 | 4(T) | 11(F) | F | T | 6(T) | T | 2a789 |
| 6 | 4(T) | 11(F) | F | T | 3(F) | T | 2a789 |
| 7 | 5(T) | 11(F) | F | F | 6(T) | T | 2a789 |
| 8 | 5(T) | 11(F) | F | F | 3(F) | F | 2a7b9 |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
| 9 | 3(F) | 6(T) | F | F | 6(T) | T | 2a789 |
| 10 | 3(F) | 6(T) | F | F | 3(F) | F | 2a7b9 |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
| 11 | 3(F) | 11(F) | F | F | 6(T) | T | 2a789 |
| 12 | 3(F) | 11(F) | F | F | 3(F) | F | 2a7b9 |
|  |  |  |  |  |  |  |  |

（6）路径覆盖

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| *序号* | *x>3* | *z<10* | *(x>3)&&(z<10)* | *x==4* | *y>5* | *（x==4）||(y>5)* | **路径** |
| 1 | 4(T) | 6(T) | T | T | 6(T) | T | 23789 |
| 2 | 5(T) | 6(T) | T | F | 3(F) | F | 237b9 |
| 3 | 5(T) | 11(F) | F | F | 6(T) | T | 2a789 |
| 4 | 5(T) | 11(F) | T | F | 3(F) | F | 2a7b9 |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

**实训6 基本路径测试方法**

班级： X班 学号： 2018 姓名：NAME

日期：2019 地点： PLECE

**一、实训目的**

1、能熟练应用白盒测试技术（基本路径测试方法）进行测试用例设计。

2、对测试用例进行优化设计。

**二、实训环境**

操作系统：Windows XP

应用系统：Visual C++开发工具和Office工具

**三、背景知识**

基本路径测试：如果把覆盖的路径数压缩到一定限度内，例如，程序中的循环体只执行零次和一次，就成为基本路径测试。它是在程序控制流图的基础上，通过分析控制构造的环路复杂性，导出基本可执行路径集合，从而设计测试用例的方法。

设计出的测试用例要保证在测试中，程序的每一个可执行语句至少要执行一次。

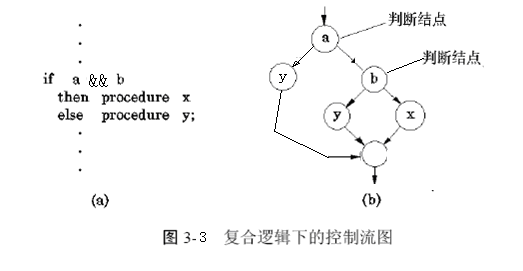
① 程序的控制流图

控制流图是描述程序控制流的一种图示方法。基本控制构造的图形符号如图所示。符号○称为控制流图的一个结点，一组顺序处理框可以映射为一个单一的结点。控制流图中的箭头称为边，它表示了控制流的方向，在选择或多分支结构中分支的汇聚处，即使没有执行语句也应该有一个汇聚结点。边和结点圈定的区域叫做区域，当对区域计数时，图形外的区域也应记为一个区域。



如果判定中的条件表达式是复合条件时，即条件表达式是由一个或多个逻辑运算符（OR，AND，NAND，NOR）连接的逻辑表达式，则需要改复合条件的判定为一系列只有单个条件的嵌套的判定。

例如对应图3-2. (a) 的复合条件的判定，应该画成如图3-2. (b) 所示的控制流图。 条件语句 if a OR b 中条件a 和条件b 各有一个只有单个条件的判定结点。



② 计算程序环路复杂性

进行程序的基本路径测试时，程序的环路复杂性给出了程序基本路径集合中的独立路径条数，这是确保程序中每个可执行语句至少执行一次所必需的测试用例数目的上界。

所谓独立路径，是指包括一组以前没有处理的语句或条件的一条路径。如在图3-3(b)所示的控制流图中，一组独立的路径是：

path1：1 - 11

path2：1 - 2 - 3 - 4 - 5 - 10 - 1 - 11

path3：1 - 2 - 3 - 6 - 8 - 9 - 10 - 1 - 11

path4：1 - 2 - 3 - 6 - 7 - 9 - 10 - 1 - 11

路径path1，path2，path3，path4 组成了图3-3 (b) 所示控制流图的一个基本路径集。只要设计出的测试用例能够确保这些基本路径的执行，就可以使得程序中的每个可执行语句至少执行一次，每个条件的取真分支和取假分支也能得到测试。基本路径集不是唯一的，对于给定的控制流图，可以得到不同的基本路径集。



通常环路复杂性可用以下三种方法求得。

* 将环路复杂性定义为控制流图中的区域数。
* 设E 为控制流图的边数，N 为图的结点数，则定义环路复杂性为 V(G)＝E－N＋2。
* 若设P 为控制流图中的判定结点数，则有 V(G)＝P＋1。

因为图5.14(b)所示控制流图有4 个区域。其环路复杂性为4。 它是构成基本路径集的独立路径数的上界。可以据此得到应该设计的测试用例的数目。

③ 导出测试用例

利用逻辑覆盖方法生成测试用例，确保基本路径集中每条路径的执行。

**四、实训任务**

学生成绩问题：下例程序流程图描述了最多输入50个值（以–1作为输入结束标志），计算其中有效的学生分数（0～100）的个数、总分数和平均值。画出程序控制流程图、计算圈复杂度V(g)，找出基本测试路径。

**五、实训步骤**

**（一）程序流程图如下:**

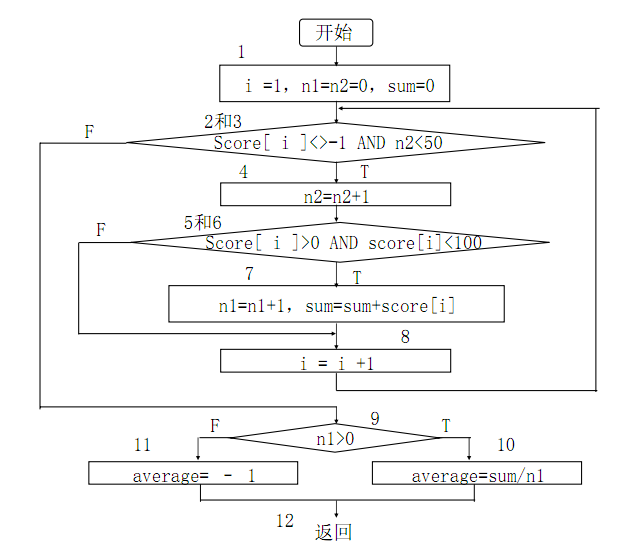


图6.1 程序流程图

**（二）完成以下任务:**

1．画出程序控制流程图

2．填写以下各值

3．写出全部独立路径，并以此设计测试用例

**1．画出程序控制流程图**

**2．填写以下各值**

结点数：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_； 判断结点数：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

边数：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_； 区域数：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

复杂度V(g)= \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. V(g)=
2. V(g)=
3. V(g)=

**3．写出全部独立路径，并以此设计测试用例**

表6.1 独立路径表

|  |  |
| --- | --- |
| Path1 | 1-2-9-10-12 |
| Path2 |  |
| Path3 |  |
| Path4 |  |
| Path5 |  |
| Path6 | 1-2-3-4-5-6-7-8-2… |

**4．设计测试用例**

表6.2 测试用例表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 测试用例 | 输入 | 期望结果 | | | 执行路径 |
| score[i] | n1 | sum | average |
| Test Case1 |  |  |  |  |  |
| Test Case2 |  |  |  |  |  |
| Test Case3 |  |  |  |  |  |
| Test Case4 |  |  |  |  |  |
| Test Case5 |  |  |  |  |  |
| Test Case6 |  |  |  |  |  |

**六、实训小结**

参考答案：

**步骤1：导出过程流程图**

图6.2 程序控制流图

**步骤2:确定环形复杂性度量V(G)：**

1）V(G)= 6 （个区域）

2）V(G)=E–N+2=16–12+2=6

其中E为流图中的边数，N为结点数；

3）V(G)=P+1=5+1=6

其中P为谓词结点的个数。在流图中，结点2、3、5、6、9是谓词结点。

**步骤3：确定基本路径集合**（即独立路径集合）。于是可确定6条独立的路径：

路径1：1-2-9-10-12

路径2：1-2-9-11-12

路径3：1-2-3-9-10-12

路径4：1-2-3-4-5-8-2…

路径5：1-2-3-4-5-6-8-2…

路径6：1-2-3-4-5-6-7-8-2…

**步骤4：为每一条独立路径各设计一组测试用例，**以便强迫程序沿着该路径至少执行一次。

1）路径1(1-2-9-10-12)的测试用例：

score[k]=有效分数值，当k <i ；

score[i]=–1, 2≤i≤50；

期望结果：

根据输入的有效分数算出正确的分数个数n1、总分sum和平均分average。

2）路径2(1-2-9-11-12)的测试用例：

score[ 1 ]= – 1 ；

期望的结果：average = –1 ，其他量保持初值。

3）路径3(1-2-3-9-10-12)的测试用例：

输入多于50个有效分数，即试图处理51个分数，要求前51个为有效分数；

期望结果：n1=50、且算出正确的总分和平均分。

4）路径4(1-2-3-4-5-8-2…)的测试用例：

score[i]=有效分数，当i<50；

score[k]<0， k< i ；

期望结果：根据输入的有效分数算出正确的分数个数n1、总分sum和平均分average。

5）路径5（1-2-3-4-5-6-8-2…）的测试用例：

score[i]=有效分数， 当i<50；

score[k]>100， k< i ；

期望结果：根据输入的有效分数算出正确的分数个数n1、总分sum和平均分average。

6）路径6(1-2-3-4-5-6-7-8-2…)的测试用例：

score[i]=有效分数， 当i<50；

期望结果：根据输入的有效分数算出正确的分数个数n1、总分sum和平均分average。

**实训7 web网站测试**

班级： X班 学号： 2018 姓名：NAME

日期：2019 地点： PLECE

**一、实训目的**

1、Web网站的测试技术

2、对测试用例进行优化设计

**二、实训环境**

操作系统：Windows XP

应用系统：IE浏览器、网络和Office工具

**三、背景知识**

基于Web的系统测试不但需要检查和验证是否按照设计的要求运行，而且还要评价系统在不同用户的浏览器端的显示是否合适。重要的是，还要从最终用户的角度进行安全性和可用性测试，从功能、性能、可用性、客户端兼容性、安全性等方面讨论基于Web的系统测试方法。

Web网站的页面是由文字、图形、音频和超级链接等组成的文档。网络客户端用户通过在浏览器上的操作，搜索查看所需要的信息。

对网站的测试包含多个方面：配置测试、兼容性测试、可用性测试、文档测试、功能测试、压力/负载测试、安全性测试；黑盒测试、白盒测试、静态测试和动态测试等。

**四、实训任务**

1. 网页测试

2. 网站测试

**五、实训步骤**

**1. 网页的测试包括如下几个方面：**

（1）文字测试。网页文字可以看作是软件文档，通常需要注意的是检查用户等级、术语、内容、准确度及内容的时效性。

（2）链接测试。链接是Web页面的一个主要特征，它的功能作用是在页面间进行切换和指导用户访问其它网页页面。链接测试又可分为三方面：

1. 测试所有链接是否能按照指示的那样正确链接到应当链接的页面。
2. 测试所链接的页面是否存在。
3. 确保不存在孤立页面，即没有链接指向的页面。这项测试需要将页面清单与实际网页进行简单的覆盖分析，确定测试的是全部的页面，也就是说没有遗漏，也没有多余的页面。最后，保证Web系统上没有孤立的页面。所谓孤立页面是指没有链接指向的页面。

（3）图形、图像测试。Web应用系统的图形可以包括图片、动画、边框、颜色、字体、背景、按钮等。图形、图像测试的内容有：

1. 确保图形有明确的用途，图片或动画不乱放。Web应用系统的图片尺寸适当，并要能清楚说明某项事物。
2. 背景颜色应与文字颜色协调。
3. 所有图形是否能正确载入和显示。

（4）表单测试。表单是指网页上用于输入和选择信息的文本框、列表框或其它域。目前网页上基本都有表单。表单测试数据接收是否正确、可选项是否可以进行选择等。当用户给Web应用系统提交信息时，需要使用表单操作，如用户注册、登录、信息提交等。因此，必须测试提交操作的完整性，以校验提交给服务器的信息的正确性。例如，用户填写的出生日期是否恰当、填写的城市与省份是否匹配等。如果使用了默认值，还要检验默认值的正确性。如果表单只能接受指定的某些值，也要对其进行测试，以确定正确性。

（5）动态内容测试。动态内容是指根据条件产生变化的图形和文字。大多数动态内容的编程在网站服务器上进行，只有具有Web服务器的访问权限才能查看程序源代码。

**实验练习:(是否有问题（Y/N）)**

表7.1 网页测试表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 测试内容  网站 | 文字  测试 | 超链接  测试 | 图形、  图像测试 | 表单  测试 | 动态内容  测试 |
| <http://www.sohu.com/> |  |  |  |  |  |
| <http://www.51typing.cn/> |  |  |  |  |  |
| <http://mail.163.com/> |  |  |  |  |  |
| <http://www.bjpldx.edu.cn/jsj/index.php> |  |  |  |  |  |
| <http://www.zhaopin.com/> |  |  |  |  |  |
| http://www.51testing.com/html/index.html |  |  |  |  |  |

**2. 网站测试包括如下几个方面：**

（1）数据库测试：数据库为Web系统的管理、运行、查询和实现用户对数据存储的请求提供空间。在使用了数据库的Web应用系统中，一般情况下可能会存在两种故障：第一种是数据的一致性问题，第二种是输出故障。前者主要是由用户提交的表单信息不正确而引起，后者主要是由网络速度或程序设计等问题引起，针对两种情况应分别考虑进行测试。

（2）服务器性能及负载（压力）测试。网站可能每日接受数十万、上百万次的点击，用户的每一次点击都要从网站服务器下载或上传数据。通常对服务器的性能测试和负载测试是通过测试工具实现的。

（3）可用性测试。

1. 整体界面测试。整体界面是指整个Web应用系统的页面结构测试。例如，当用户浏览Web应用系统时是否感到舒适，是否凭直觉就知道要找的信息在什么地方，整个Web应用系统的设计风格是否一致等。对整体界面的测试过程，其实是一个对最终用户进行调查的过程。一般Web应用系统采取在主页上做一个调查问卷的方式，以得到最终用户的反馈信息。因此，此项测试需要有外部人员（与Web应用系统开发无联系或联系很少者）的参与。
2. 导航测试。导航描述了用户在一个页面内操作的方式，在不同的用户接口控制之间，如按钮、对话框、列表和窗口等，或在不同的连接页面之间。通过考虑下列问题，可以决定一个Web应用系统是否易于导航：导航是否直观；Web应用系统的主要部分是否可通过主页存取；Web应用系统是否有站点地图的帮助。Web应用系统的用户趋向于目的驱动，需要很快的扫描一个Web应用系统，看是否有满足自己需要的信息，如果没有，就会很快的离开。极少用户愿意花费精力去熟悉Web应用系统的结构，因此，Web应用系统导航帮助要尽可能的准确。导航的另一个重要方面是Web应用系统的页面结构、导航、菜单、链接的风格是否一致，确保用户凭直觉就知道Web应用系统里面是否还有内容，内容在什么地方。Web应用系统的层次一旦确定，就要着手测试用户导航功能，如果最终用户参与这项工作，效果会更加明显。

（4）安全性测试。网站的安全性是一个非常重要的方面，因此需要进行网站的安全性测试。这项测试涉及的内容有：

1. 对用户的权限进行测试。包括可以尝试登录的次数，可以访问的或浏览的页面等。
2. Web应用系统是否有超时的限制，即用户在登录一段时间内没有点击任何页面，是否需要重新登录才能正常使用。
3. 为保证Web应用系统的安全性，日志文件至关重要。需要对日志文件进行测试，如相关信息是否写入了日志文件，是否可跟踪。
4. 当使用了安全套接字时，需要测试加密是否正确，检查信息的完整性。
5. 服务器端的脚本常常构成安全漏洞，这些漏洞是黑客的主要攻击目标。所以，还需要测试相关安全性的问题。

（5）兼容性测试。需要对Web应用系统在不同浏览器上是否能正常显示进行测试。

**实验练习:(是否有问题（Y/N）)**

表7.2 网站测试表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 测试内容  网站 | 数据库  测试 | 服务器性能及  负载（压力）测试 | 可用性  测试 | 安全性测试 | 兼容性测试 |
| <http://www.sohu.com/> |  |  |  |  |  |
| <http://www.51typing.cn/> |  |  |  |  |  |
| <http://mail.163.com/> |  |  |  |  |  |
| <http://www.bjpldx.edu.cn/jsj/index.php> |  |  |  |  |  |
| <http://www.zhaopin.com/> |  |  |  |  |  |
| http://www.51testing.com/html/index.html |  |  |  |  |  |

**六、实训小结**

**实训8 兼容性测试**

班级： X班 学号： 2018 姓名：NAME

日期：2019 地点： PLECE

**一、实训目的**

1、软件之间的兼容性

2、对测试用例进行优化设计

**二、实训环境**

操作系统：Windows XP

应用系统：IE浏览器、网络和Office工具，pdf文件

**三、背景知识**

1. pdf文件快速转为word文件
2. word文件快速转为pdf文件

**四、实训任务**

1. pdf文件如何快速转为word文件？几种方法？转过后有无bug？
2. word文件快速转为pdf文件？用什么方法？转过后有无bug？

**五、实训步骤**

**六、实训小结**