

**软件测试**

**实验报告**

|  |  |
| --- | --- |
| **课程名称：** | **软件测试** |
| **学生姓名：** | **彭晓轩** |
| **学生学号：** | **201730683321** |
| **学生专业：** | **软件工程** |
| **开课学期：** | **2019-2020学年第一学期** |

**软件学院**

**2019年11月20日**

一．实验目的

1. 对被测试模块Shopping系统的index.jsp文件进行单元测试用例设计
2. 完成黑盒测试用例设计，白盒测试用例设计以及静态测试
3. 实验分析
4. 黑盒测试：黑盒测试又称功能测试、数据驱动测试或基于规格说明书的测试，是一种从用户观点出发的测试。黑盒测试用例设计技术主要有等价类划分法、边界值分析法、错误推测方法、因果图方法、判定表方法等。
5. 白盒测试：白盒测试也称结构测试或逻辑驱动测试，是一种测试用例设计方法，它从程序的控制结构导出测试用例。白盒测试使用被测单元内部如何工作的信息，允许测试人员对程序内部逻辑结构及有关信息来设计和选择测试用例，对程序进行测试。主要分为逻辑驱动测试和基本路径测试。
6. 静态测试：静态测试：通过检查和评审软件而不是运行软件对软件进行测试的方法。静态测试可以手工进行，也可以借助软件工具自动进行。静态测试的方法主要有代码审查、代码走查和桌面检查。静态测试的内容主要分需求定义的静态测试、文档的静态测试、源代码的静态测试。
7. 实验过程
8. 黑盒测试用例设计
9. 等价类划分法

|  |  |  |
| --- | --- | --- |
| 输入等价类 | 有效等价类 | 无效等价类 |
| 商品编号数字字符位数 | ①6位数字字符 | ②有非数字字符  ③小于6位数字字符 |
| 商品编号数字范围 | ④数字字符在000001-999999之间 | ⑤000000  ⑥负数 |

为上述等价类设计用例如下所示：

|  |  |  |
| --- | --- | --- |
| 测试用例 | 期望结果 | 覆盖的等价类 |
| 123456 | 有效输入 | ①、④ |
| 1#3456 | 无效输入 | ② |
| 245 | 无效输入 | ③ |
| 000000 | 无效输入 | ⑤ |
| -12345 | 无效输入 | ⑥ |

1. 边界值分析法

根据商品编号设置范围在000001-999999之间，可以通过边界值分析法设置出以下用例：000000、000001、000002、999998、999999

1. 错误推测法

考虑商品名称为空或商品编号为空或两者同时为空时，按下go按钮的情况。

考虑商品名称和商品编号都有输入值，但两者并不匹配时，按下go按钮的情况。

1. 白盒测试用例设计

function MM\_preloadImages()

{ //v3.0

var d=document;

if(d.images)

{ if(!d.MM\_p)

d.MM\_p=new Array();

var i,j=d.MM\_p.length,a=MM\_preloadImages.arguments;

for(i=0; i<a.length; i++)

if (a[i].indexOf("#")!=0)

{

d.MM\_p[j]=new Image; d.MM\_p[j++].src=a[i];

}

}

}

上述程序段的程序流程图如下所示：

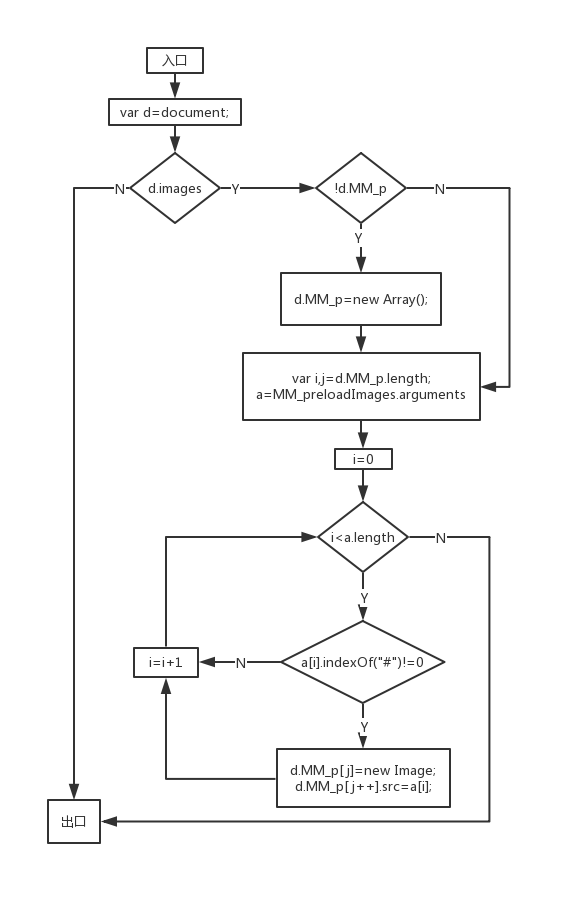


图1 程序流程图

根据图1，找出判断项目包含：①d.images、②!d.MM\_p、③a[i].indexOf(“#”)!=0

1. 分支覆盖（判定覆盖）

执行足够的测试用例，使得程序中的每一个分支至少都通过一次。

设置如下测试用例：

|  |  |
| --- | --- |
| 用例编号 | 用例设计 |
| ① | !d.images |
| ② | d.images、!d.MM\_p、a[i].indexOf(“#”)!=0 |
| ③ | d.images、d.MM\_p、a[i].indexOf(“#”)=0 |

1. 条件覆盖

执行足够的测试用例，使得判定中的每个条件获得各种可能的结果。

设置如下测试用例：

|  |  |
| --- | --- |
| 用例编号 | 用例设计 |
| ① | !d.images |
| ② | d.images、!d.MM\_p、a[i].indexOf(“#”)=0 |
| ③ | d.images、d.MM\_p、a[i].indexOf(“#”)!=0 |

1. 条件组合覆盖

执行足够的例子，使得每个判定中条件的各种可能组合都至少出现一次。

设置如下测试用例：

|  |  |
| --- | --- |
| 用例编号 | 用例设计 |
| ① | !d.images |
| ② | d.images、!d.MM\_p、a[i].indexOf(“#”)=0 |
| ③ | d.images、d.MM\_p、a[i].indexOf(“#”)!=0 |

1. 静态测试
2. 代码审查

内容是检查代码和设计的一致性，检查代码对标准的遵循、可读性，检查代码的逻辑表达的正确性，检查代码结构的合理性。

审查，指调查核实并评定是否正确、妥当,或者说是对某项事情、情况的核实、核查。

**本实验中的Shopping系统的index.jsp文件基本满足了代码的正确性和代码结构的合理性，但这段代码的可读性较差、易修改性不良。例如，这段代码的变量有较多使用了中文拼音而不是有效易读易理解的简单英文。**

1. 代码走查

与代码审查不同，不是读程序和使用代码审查单。而是由被指定的作为测试员的小组成员提供若干测试用例（程序的输入数据和期望的输出结果），让参加会的成员当计算机，在会议上对每个测试用例用头脑来执行程序，也就是用测试用例沿程序逻辑走一遍，并由测试人员讲述程序执行过程，在纸上或黑板上监视程序状态（变量的值）

**设计出上述已经写出的测试用例，让参会人员利用上述各个测试用例在脑中模拟计算机程序逻辑走一遍，确定每个测试用例是否对应于预期的输出（有效或者无效等）。**

1. 桌面检查

程序员阅读自己所编的程序。

**由于该程序并非本人所编写，因而本实验中将略去这一方法。**

1. 实验总结

在这次实验当中，我完成了对某代码段的黑盒测试用例设计、白盒测试用例设计和静态测试分析，进一步理解和巩固了课堂上所学的各种软件测试方法。