**《软件测试》**

**实 验 报 告**

实 验 题 目： 实验一 白盒测试

学 院（系）： 信息数理学院 专 业 班 级： 计算机162

学 号： 16430121 姓 名： 林锦雄

指 导 老 师： 刘锁兰 实 验 时 间： 2019.4.23

2018-2019-2

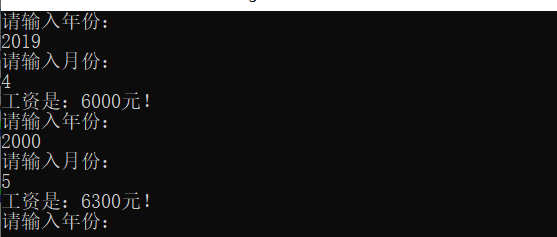
# 实验一 白盒测试

一、 实验目的  
1、 掌握白盒测试的基本方法；  
2、 掌握白盒测试用例的编写。  
二、 实验要求  
1、 根据给出的程序分别使用语句覆盖、判定覆盖（也称为分支覆盖）、条件覆盖、路径测试设计测试用例。  
2、 输入数据进行测试，填写测试用例。

三、 实验内容

1、测试环境搭建

用C# 等熟悉的语言编写职工工资计算程序。该程序的基本功能是：根据输入的日期，计算出当月职工的工资。运行界面如下：



输入当前年月，计算工作天数及相应的工资：

月基本工资为 200元/天 \* 30天=6000元；

如果工作31天：工资为 6000元+300元=6300元；

如果工作29天：工资为 6000元-100元=5900元；

如果工作28天：工资为 6000元-150\* 2元=5700元。

2、 页面控件说明

（1）年： 输入框

输入类型： 数字

位数： 最大4位

默认：空

范围：西历 1950---2050

（2）月： 输入框

输入类型： 数字

位数： 最大2位

默认：空

范围：1---12

（3）按钮

操作：按钮按下时计算当月工资

（4）显示框：显示当月工资。

**测试用例**设计**要求：**

1. 分析各种输入情况，结合程序输出结果，并给出详细测试用例；

(2)根据(1)所划分的等价类，进行边界值分析，并给出具体的测试用例；

(3)决策表测试法；

①列出输入变量month、 year的有效等价类；（条件桩）

②分析程序的规格说明，给出问题规定的可能采取操作；(动作桩)

③画出决策表（简化）；

④根据决策表，给出详细测试用例。

等价类划分表：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输入条件 | 有效等价类 | 编号 | 无效等价类 | 编号 |
| month | month=1,3,5,7,8,10,12  month=2  month=4,6,9,11 | 1  2  3 | month<1  month>12  month为非整数 | 7  8  9 |
| year | year为闰年  year不为闰年  1950<=year<=2050 | 4  5  6 | year为非整数  year为非正数  year>2050  0<year<1950 | 10  11  12  13 |

测试用例表：

1.覆盖有效等价类的测试用例

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例ID | 输入数据 | | 预期输出 | 覆盖等价类号码 |
| year | month |
| 1  2  3  4  5  6  7 | 2017  2018  2019  2020  2020  2024  2025 | 1  2  3  2  7  6  9 | 工资是：6300元  工资是：5700元  工资是：6300元  工资是：5900元  工资是：6300元  工资是：6000元  工资是：6000元 | 1,5,6  2,5,6  2,5,6  2,4,6  1,4,6  1,4,6  3,5,6 |

1. 覆盖无效等价类的测试用例

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例ID | 输入数据 | | 预期输出 | 覆盖等价类号码 |
| year | month |
| 8  9  10  11 | 1949.5  -1950  3000  2080.6 | 0  -7  13  20.5 | 输入有误,请重新输入!  输入有误,请重新输入!  输入有误,请重新输入!  输入有误,请重新输入! | 13,10,7  11,13,7  12,8  10,12,8,9 |

边界值分析表（1950<=year<=2050）

|  |  |  |  |
| --- | --- | --- | --- |
| 测试用例 | 输入数据 | | 预期输出 |
| year | month |
| 1  2  3  4 | 1949  1951  2049  2051 | 6  6  6  6 | 输入有误,请重新输入!  工资是：6000元  工资是：6000元  输入有误,请重新输入! |
| 5  6  7  8 | 2019  2019  2019  2019 | 0  2  11  13 | 输入有误,请重新输入!  工资是：5700元  工资是：6300元  输入有误,请重新输入! |

决策表

M1:{month=4,6,9,11} M2:{month=1,3,5,7,8,10,12}

M3:{month=2}

Y1:{year为闰年} Y2:{year不为闰年}

动作桩：A1：基本工资 A2：基本工资+300

A3：基本工资-100 A4：基本工资-300

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 规则 | 1 | 2 | 3 | 4 | 5 | 6 |
| 条件 | | | | | | |
| C1：月份在 | M1 | M1 | M2 | M2 | M3 | M3 |
| C2：年份在 | Y1 | Y2 | Y1 | Y2 | Y1 | Y2 |
| 动作 | | | | | | |
| A1：基本工资 | √ | √ |  |  |  |  |
| A2：基本工资+300 |  |  | √ | √ |  |  |
| A3：基本工资-100 |  |  |  |  | √ |  |
| A4：基本工资-300 |  |  |  |  |  | √ |

化简后的决策表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 规则 | 1,2 | 3，4 | 5 | 6 |
| 条件 | | | | |
| C1：月份在 | M1 | M2 | M3 | M3 |
| C2：年份在 | Y1/Y2 | Y1/Y2 | Y1 | Y2 |
| 动作 | A1 |  |  |  |
|  | A 2 |  |  |
|  |  | A3 |  |
|  |  |  | A4 |

根据决策表设计测试用例

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试用例编号 | 月份 | 年份 | 预期输出 | 实际输出 |
| 1 | 1 | 2017 | 工资是：6300元 | 请输入年份：2017  请输入月份：1  工资是：6300元 |
| 2 | 2 | 2018 | 工资是：5700元 | 请输入年份：2018  请输入月份：2  工资是：5700元 |
| 3 | 3 | 2019 | 工资是：6300元 | 请输入年份：2019  请输入月份：3  工资是：6300元 |
| 4 | 2 | 2020 | 工资是：5900元 | 请输入年份：2020  请输入月份：2  工资是：5900元 |
| 5 | 7 | 2020 | 工资是：6300元 | 请输入年份：2020  请输入月份：7  工资是：6300元 |
| 6 | 6 | 2024 | 工资是：6000元 | 请输入年份：2024  请输入月份：6  工资是：6000元 |
| 7 | 9 | 2025 | 工资是：6000元 | 请输入年份：2025  请输入月份：9  工资是：6000元 |

2、用C++或者Java等熟悉的语言编写一个类，完成下面函数的功能，并编写另外一个类，调用该方法：

void DoWork(int x,int y,int z)

{

int k=0,j=0;

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

{

k=x\*y-1; //语句块1

j=sqrt(k);

}

if((x= =4)||(y>5))

{

j=x\*y+10; //语句块2

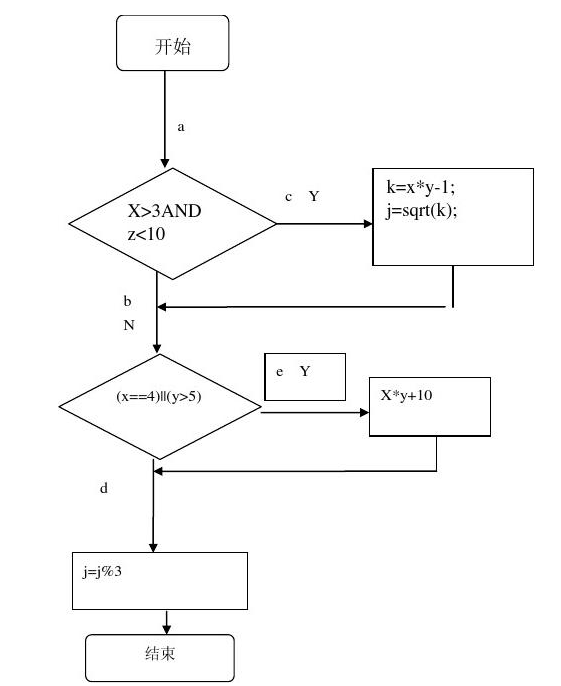
}

j=j%3; //语句块3

}

**要求：**

1. 画出上面函数的流程图。



(2)分别使用语句覆盖、判定覆盖（也称为分支覆盖）、条件覆盖、基本路径覆盖法设计测试用例，并写出每个测试用例的执行路径（注意测试用例的格式）。

**整体代码见附录**

路径(p1代表第1条路径,依此类推)

P1:(a-c-e) P2:(a-c-d) P3:(a-b-e) P4:(a-b-d)

判定条件(用大写字母表示,M,N)

M:x>3&&z<10 N:(x==4)||(y>5)

语句覆盖

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试用例输入 | 预期输出 | 判定M的取值 | 判定N的取值 | 覆盖路径 |
| 8,6,5 | j=1,k=47 | T | T | P1 |

判定覆盖

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试用例输入 | 预期输出 | 判定M的取值 | 判定N的取值 | 覆盖路径 |
| 8,6,5 | j=1,k=47 | T | T | P1 |
| 4,6,6 | j=1,k=23 | T | T | P1 |
| 1,2,11 | j=0,k=0 | F | F | P4 |
| 8,6,5 | j=1,k=47 | T | F | P2 |
| 1,6,11 | j=1,k=0 | F | T | P3 |

条件覆盖

对于M:x>3取真时T1,取假时F1;

z<10取真时T2,取假时F2;

对于N:x=4取真时T3,取假时F3;

y>5取真时T4,取假时F4。

条件:x>3,z<10,x=4,y>5

条件:x<=3,z>=10,x!=4,y<=5

|  |  |  |  |
| --- | --- | --- | --- |
| 测试用例输入 | 预期输出 | 取值条件 | 覆盖路径 |
| 1,4,9 | j=0,k=0 | F1,T2,F3,F4 | P1 |
| 4,1,9 | j=2,k=3 | T1,T2,T3,F4 | P4 |
| 4,5,6 | j=1,k=23 | T1,T2,T3,T4 | P4 |
| 4,10,5 | j=0,k=0 | T1,F2,T3,F4 | P2 |

基本路径覆盖法

|  |  |  |
| --- | --- | --- |
| 测试用例输入 | 预期输出 | 覆盖路径 |
| 4,6,6 | j=1,k=23 | P1 |
| 1,2,11 | j=0,k=0 | P4 |
| 8,6,5 | j=1,k=47 | P2 |
| 1,6,11 | j=1,k=0 | P3 |

(3)执行每个测试用例，执行完毕后填写测试用例表。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 测试用例编号 | x | y | z | 预期输出 | 实际输出 |
| 1 | 3 | 5 | 6 | j=1,k=0 | j=1,k=0 |
| 2 | 3 | 10 | 5 | j=0,k=0 | j=0,k=0 |
| 3 | 4 | 5 | 8 | j=0,k=31 | j=0,k=31 |
| 4 | 5 | 5 | 4 | j=1,k=19 | j=1,k=19 |
| 5 | 4 | 11 | 6 | j=1,k=0 | j=1,k=0 |
| 6 | 2 | 11 | 5 | j=0,k=0 | j=0,k=0 |

四、实验步骤

1、依次按照每种方法的步骤来写出测试用例；

2、测试，实验数据记录，完善测试用例表。

具体的步骤已经包含在实验内容中。

五、实验总结

1.通过本次实验，我掌握了白盒测试的很多种方法，并且对白盒测试用例的编写也有了很大的掌握。

2.首先，我对语句覆盖、条件覆盖、判定覆盖、路径覆盖的相同点和区别都有了更深的理解和掌握，其次，我对等价类划分以及决策表测试也有了更深的理解和掌握。

3.通过实验，我对测试用例的设计有了更准确的定义与设计，并且也巩固了多种白盒测试的方法，对软件测试这块有了更好的理解与把握。

**附录**

**第1题代码：**

1. **import** java.util.Scanner;
3. **public** **class** Main {
4. **public** **static** **void** main(String[] args) {
5. Scanner sc = **new** Scanner(System.in);
6. System.out.println("输入年份和月份：");
7. **int** year = sc.nextInt(),month = sc.nextInt();
8. **if**(month == 1 || month == 3 || month == 5 || month == 7 || month == 8 || month == 10 || month == 12) {
9. System.out.println("工资是:" + 6300 + "元");
10. } **else** **if**(month == 4 || month == 6 || month == 9 || month == 11) {
11. System.out.println("工资是:" + 6000 + "元");
12. } **else** **if**(month == 2){
13. **if**(year % 400 == 0 || (year % 4 == 0 && year % 100 != 0)) {
14. System.out.println("工资是:" + 5900 + "元");
15. } **else** {
16. System.out.println("工资是:" + 5700 + "元");
17. }
18. } **else** {
19. System.out.println("输入有误，请重新输入！");
20. }
21. }
22. }

**第2题代码：**

1. #include<iostream>
2. #include<math.h>
3. **using** **namespace** std;
4. **void** DoWork (**int** x,**int** y,**int** z) {
5. **int** k=0,j=0;
6. **if** ( (x>3)&&(z<10) ){
7. k=x\*y-1;                //语句块
8. j=**int**(sqrt(**double**(k)));
9. }
10. **if** ( (x==4)||(y>5) ) {
11. j=x\*y+10;           //语句块
12. }
13. j=j%3;                  //语句块
14. cout<<"j="<<j<<endl;
15. cout<<"k="<<k<<endl;
16. }
17. **int** main() {
18. **int** s;
19. DoWork(8, 6, 5);
20. s=getchar();
21. **return** 0;
22. }