# 软件质量保证与测试

# 实验报告

**实验名称： 实验一 白盒测试实验一**

**实验地点：** **C6二区507机房**

**实验日期： 2022/05/01**

**学生姓名： 刘恩嘉、郭天佑、赵熙斌**

**学生学号： 2020015315、2020015274、2020015331**

# 计算机系 制

# 一、实验目的

（1）巩固基于控制流白盒测试知识，熟练应用基本控制流覆盖方法设计测试用例；

（2）通过绘制程序控制流程图，实现对程序源代码的逻辑描述；

（3）掌握逻辑短路对测试的影响；

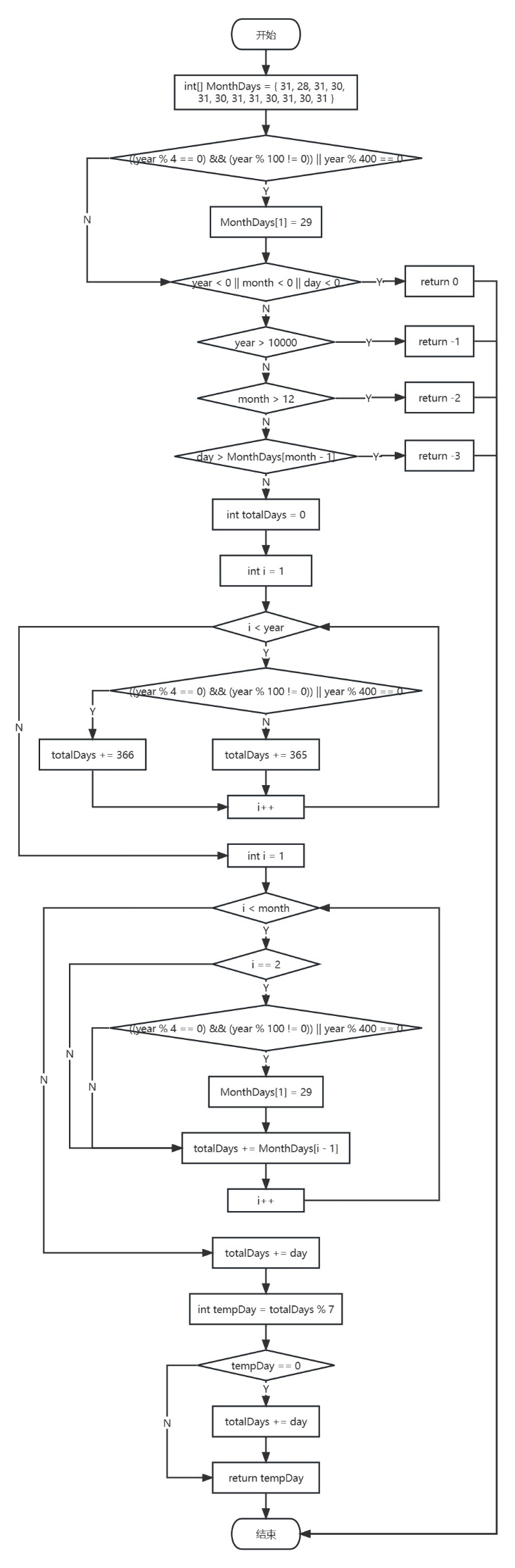
（4）培养严谨和系统的测试精神，学习测试用例的设计和分析；

# 二、实验内容

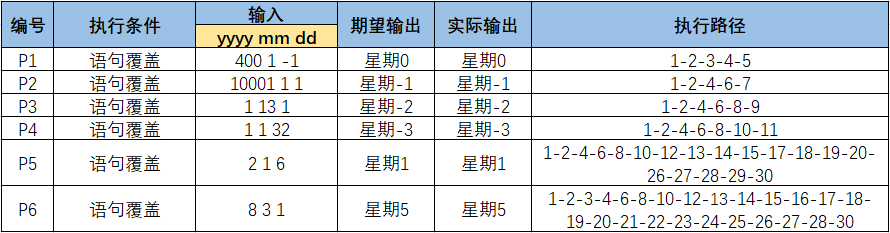
### （一）实验1: 控制流测试技术实验

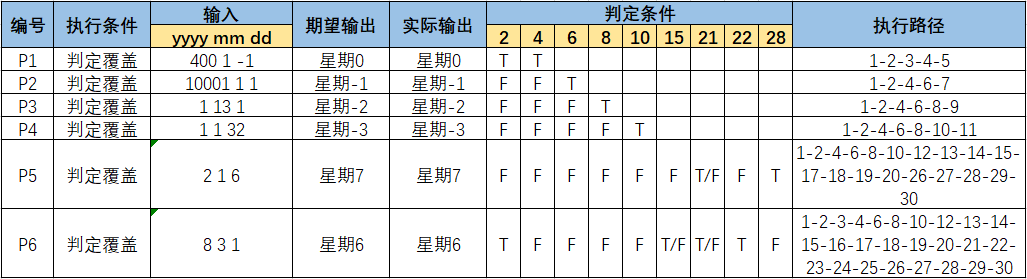
1. 给出getWeekday()的程序流程图。

|  |
| --- |
| 1. int[] MonthDays = { 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 }; 2. if (((year % 4 == 0) && (year % 100 != 0)) || year % 400 == 0) 3. MonthDays[1] = 29; 4. if (year < 0 || month < 0 || day < 0) 5. return 0; 6. if (year > 10000) 7. return -1; 8. else if (month > 12) 9. return -2; 10. else if (day > MonthDays[month - 1]) 11. return -3; 12. int totalDays = 0; 13. int i = 1; 14. for (; i < year;) 15. if (((i % 4 == 0) && (i % 100 != 0)) || i % 400 == 0) 16. totalDays += 366; 17. else totalDays += 365; 18. i++ 19. int i = 1; 20. for (; i < month;) 21. if (i == 2) 22. if (((year % 4 == 0) && (year % 100 != 0)) || year % 400 == 0) 23. MonthDays[1] = 29; 24. totalDays += MonthDays[i - 1]; 25. i++ 26. totalDays += day; 27. int tempDay = totalDays % 7; 28. if (tempDay == 0) 29. tempDay = 7; 30. return tempDay; |



1. 分别以语句覆盖和判定覆盖方法设计测试用例，并写出每个测试用例的执行路径。





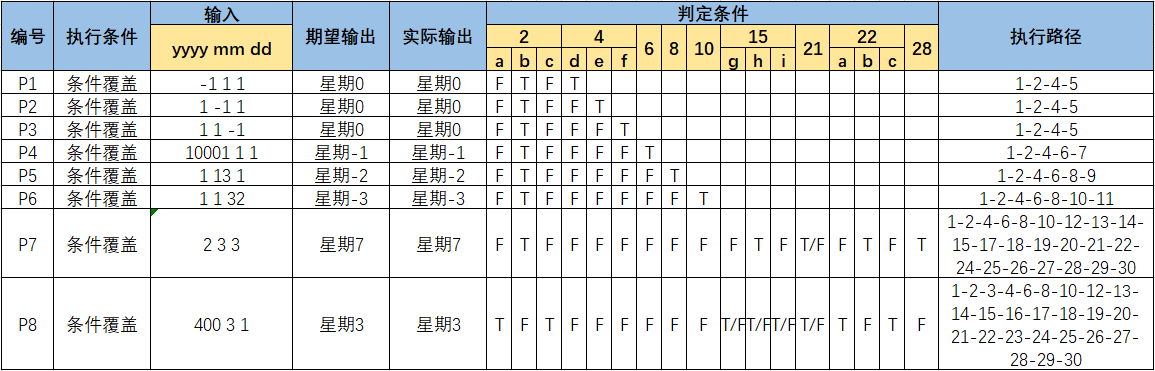
（3）自行写一个小程序，验证当判定中包含多个条件时，条件短路对控制流测试的影响。

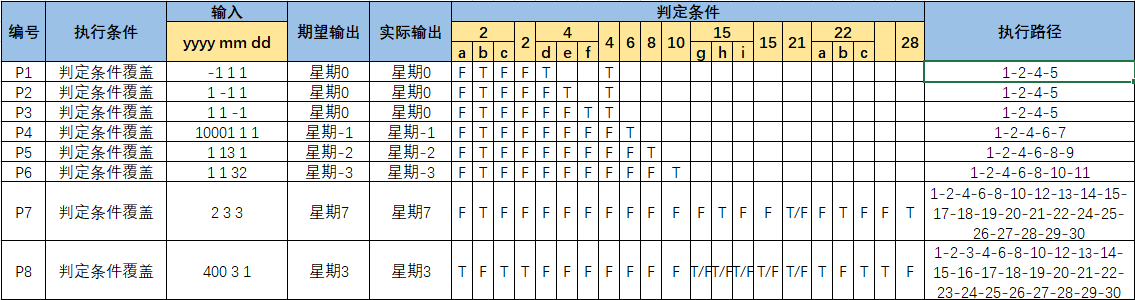
|  |
| --- |
| if (x > 1 && y < 10) {...}  if (y == 5 || z < 10) {...} |



由于短路{2,5,5},{0,11,11}无法使得每个条件各取值至少执行一次，需要额外添加测试用例{2,11,5}

（4）分别以条件覆盖、判定条件覆盖和条件组合覆盖方法设计测试用例，并写出执行路径。







（5）给出程序中循环的测试用例，并说明测试用例设计的理由。

该程序为串接循环结构，且各循环节点相互独立，可使用简单循环方法进行测试。对于简单循环，n表示循环的最大次数，则测试用例如下：

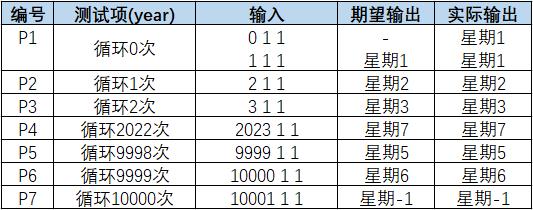
0次循环：直接跳过循环

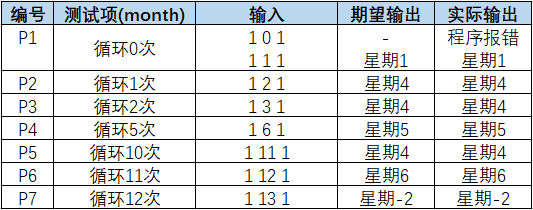
1次循环：检查循环初始值

2次循环：检查多次循环

m次循环：检查多次循环，其中m<n

n-1，n，n+1次循环：检查边界值



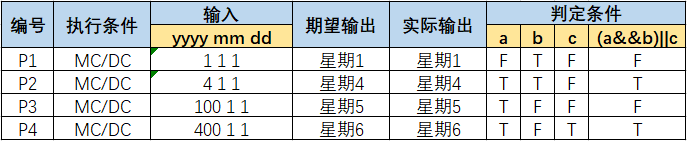


（6）如果要进一步用路径覆盖准则来测试getWeekday()，请基于程序流程图计算其中可能的路径共有多少条？是否包含不可达路径？依照你设计测试用例的速度，完成所有路径的测试需要多少时间？

输入不合法时有2\*4=8条路径。输入合法时经过两层循环，第一层循环有3条路径，第二层循环有4条路径，则有2\*3\*4\*2=48条路径。共有8+48=56条路径。包含不可达路径，程序中出现两次 ((year % 4 == 0) && (year % 100 != 0) 判定，理论上有4条路径但实际只会执行2条路径。完成所有路径用时为测试单条路径用时的56倍。

（7）给出MC/DC（修订的判定条件覆盖）方法对下列2处语句的测试用例。

|  |  |
| --- | --- |
| …  int[] MonthDays = { 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 };  if (((year % 4 == 0) && (year % 100 != 0)) || year % 400 == 0) {  MonthDays[1] = 29;  }  … | …  if (((i % 4 == 0) && (i % 100 != 0)) || i % 400 == 0) {  totalDays += 366;  } else {  totalDays += 365;  }  … |



# 三、实验体会

（1）通过测试，是否发现程序中存在的缺陷？

year，month，day变量可以取0，应该从1开始取值。且month变量取0会造成数组下标越界。

1. 程序中是否存在采用动态白盒测试技术未能发现的缺陷？

原程序中21~23行代码冗余，在程序开头就已经判断闰年并对MonthDays[1]进行赋值。

|  |
| --- |
| if (i == 2)  if (((year % 4 == 0) && (year % 100 != 0)) || year % 400 == 0)  MonthDays[1] = 29; |

（4）在白盒测试实践中，你觉得哪些步骤是关键的？并谈谈你所体会到的白盒测试过程中的难点。

设计测试用例时，尽可 能设计最少的测试用例数，同时保证每种覆盖方法的覆盖率尽可能达到 100%。以及条件短路对控制流测试的影响。