东南大学国家示范性软件学院
College of Software Engineering
Southeast University

软件测试基础与实践

实验报告

实验名称：白盒测试实验一

实验地点：计算机楼 268

实验日期：2019 年 10 月 23 日

学生姓名：姜子玥

学生学号：71117201

东南大学 软件学院 制

一、实验目的

(1) 巩固基于控制流白盒测试知识，对于给定的待测程序，能熟练应用基本控制六覆盖方法设计测试用例；
(2) 通过绘制程序控制流程图，实现程序源代码的逻辑描述
(3) 掌握逻辑短路对测试的影响
(4) 掌握严谨和系统的测试精神，学习测试用例的设计和分析

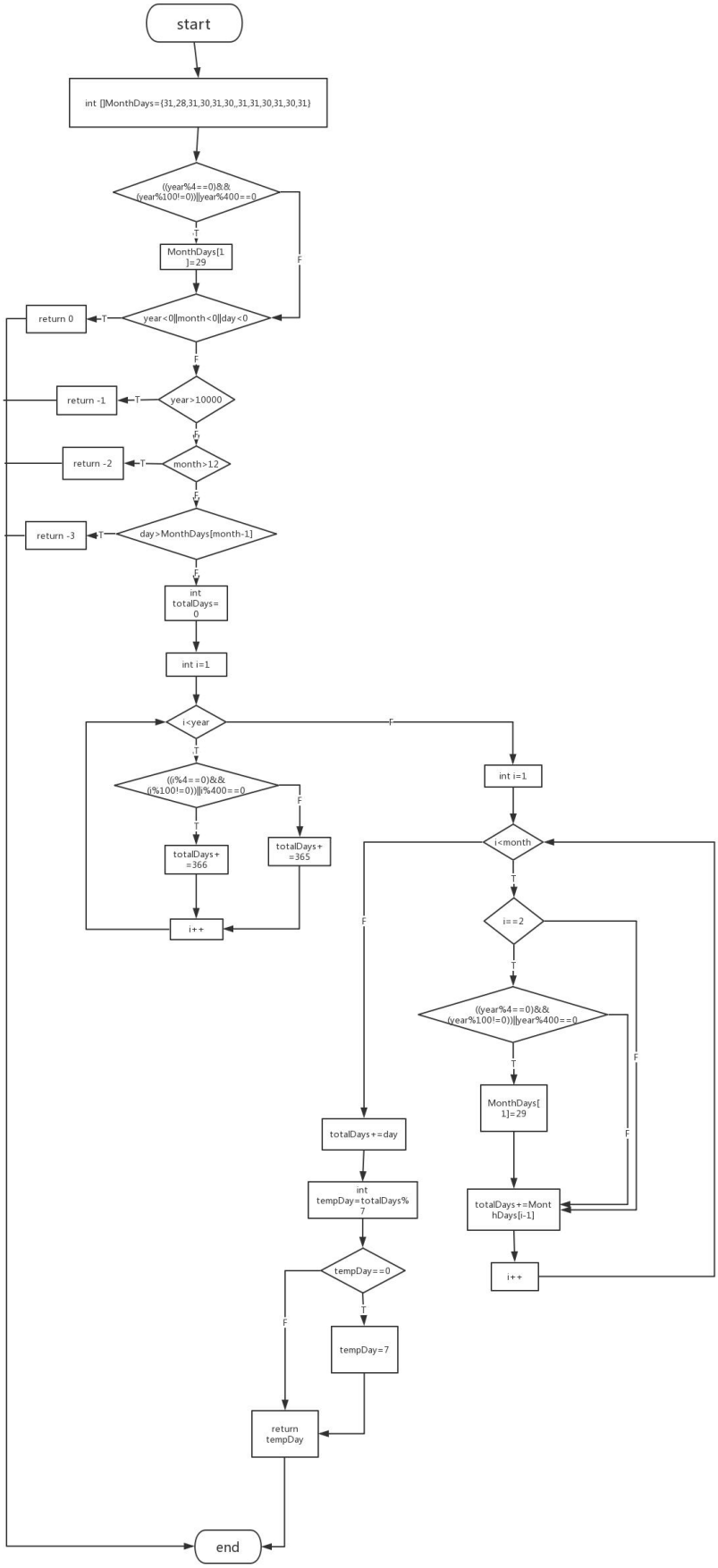
二、实验内容

（一）题目 1：控制流测试技术实验

1. 用于基于控制流的动态白盒测试方法，对 WeekA 程序中的方法 getWeekday()进行测试。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%
要求：
(1) 给出 getWeekday()的程序流程图。
(2) 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
通过这段小程序的执行，加强对逻辑短路现象的理解。
(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，并写出每个测试用例的执行路径
(5) 给出对程序中循环的测试用例，并说明测试用例设计的理由
(6) 如果进一步用路径覆盖准则来测试 getWeekday()，基于程序流程图计算其中可能的路径共有多少条？
注意：
A. 正确分析程序可能的执行路径；
B. 对于涉及循环的路径，统计时可简化为：执行 N>=1 次视为同一条路径，执行 0 次（即跳过）视为另一条路径；
(7) 给出 MC/DC 方法对下列 2 处语句的测试用例

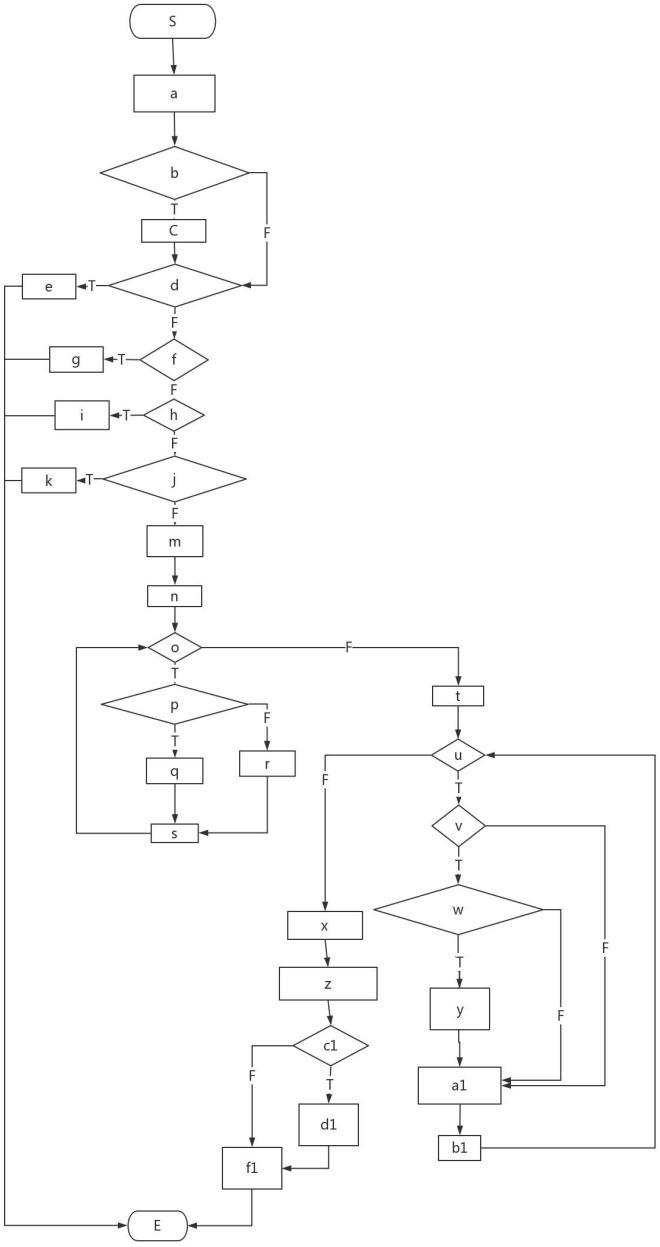
```
    …
    int[] MonthDays = { 31, 28, 31, 30, 31, 30, 31,
31, 30, 31, 30, 31 };
    if (((year % 4 == 0) && (year % 100 != 0)) ||
year % 400 == 0) {
                MonthDays[1] = 29;
    }
    …
```

```
…
if (((i % 4 == 0) && (i % 100 != 0)) || i % 400 == 0) {
                totalDays += 366;
            } else {
                totalDays += 365;
            }
…
```


start
int []MonthDays={31,28,31,30,31,30,,31,31,30,31,30,31}
((year%4==0)&&(year%100!=0))||year%400==0
T
F
MonthDays[1]=29
return 0
year<0||month<0||day<0
T
F
return -1
year>10000
T
F
return -2
month>12
T
F
return -3
day>MonthDays[month-1]
T
F
int totalDays=0
int i=1
i<year
T
F
((i%4==0)&&(i%100!=0))||i%400==0
T
F
totalDays+=366
totalDays+=365
i++
int i=1
i<month
T
F
i==2
T
F
((year%4==0)&&(year%100!=0))||year%400==0
T
F
MonthDays[1]=29
totalDays+=MonthDays[i-1]
i++
totalDays+=day
int tempDay=totalDays%7
tempDay==0
T
F
tempDay=7
return tempDay
end

（1） 程序流程图如上

（2） 为了方便描述测试用例的执行路径，绘制 getWeekday() 的流图如下



以语句覆盖设计的测试用例如下：

编号	执行条件	输入	期望输出	实际输出	执行路径
001	语句覆盖	year=2018 month=3, day=11	7	7	S-a-b-c-d-f-h-j-m-n-o-p-r-s-o-p-r-s-o-p-r-s-o-p-q-s-.. .-t-u-v-a1-b1-u-v-w-y-a1-b1-u-x-z-c1-d1-f1-E

002	语句覆盖	year=-1,month=12,day=1	0	0	S-a-b-d-e-E
003	语句覆盖	year=10001,month=12,day=1	-1	-1	S-a-b-d-f-g-E
004	语句覆盖	year=2019,month=13,day=1	-2	-2	S-a-b-d-f-h-i-E
005	语句覆盖	year=2019,month=2,day=30	-3	-3	S-a-b-d-f-h-j-k-E

以判定覆盖设计的测试实例如下：

编号	执行条件	输入	期望输出	实际输出	条件判断							执行路径
					b	d	f	h	j	w	c1	
001	判定覆盖	year=2018,month=3,day=11	7	7	T	F	F	F	F	T	T	S-a-b-c-d-f-h-j-m-n-o-p-r-s-o-p-r-s-o-p-r-s-o-p-q-s-...-t-u-v-a1-b1-u-v-w-y-a1-b1-u-x-z-c1-d1-f1-E
002	判定覆盖	year=2019,month=3,day=12	2	2	F	F	F	F	F	F	F	S-a-b-c-d-f-h-j-m-n-o-p-r-s-o-p-r-s-o-p-r-s-o-p-q-s-...-t-u-v-a1-b1-u-v-w-a1-b1-u-x-z-c1-f1-E
003	判定覆盖	year=-1,month=1,day=1	0	0	F	T						S-a-b-d-e-E
004	判定覆盖	year=10001,month=12,day=1	-1	-1	F	F	T					S-a-b-d-f-g-E
005	判定覆盖	year=2019,month=13,day=1	-2	-2	F	F	F	T				S-a-b-d-f-h-i-E
006	判定覆盖	year=2019,month=2,day=30	-3	-3	F	F	F	F	T			S-a-b-d-f-h-j-k-E

(3) 设计程序如下：

```
int a ,b,c;
cin>>a>>b>>c;
if(a<0||b>0||c==0){
   cout<<”situation 1”<<endl;
}
else if(a>=0||b<=0||c!=0){
   cout<<”situation 2”<<endl;}
```

经过若干测试用例我发现在我设计的这段程序中其实起决定执行 situation1 或是 situation2 只取

决于 a 的值，当 a<0 时，第一个条件判断直接将后续部分短路，造成逻辑短路；当 a>=0 时，第二个条件短路。

(4) 为了方便在测试实例中表述：

令判定 b 中 year%4==0 为条件 1，year%100!=0 为条件 2，year%400==0 为条件 3；判定 d 中 y<0 为条件 4，month<0 为条件 5,day<0 为条件 6

① 以条件覆盖设计的测试实例如下：

编号	执行条件	输入	期望输出	实际输出	条件										执行路径
					1	2	3	4	5	6	f	h	j	c1	
001	条件覆盖	year=2018, month=3, day=11	7	7	T	T	F	F	F	F	F	F	F	T	S-a-b-c-d-f-h-j-m-n-o-p-r-s-o-p-r-s-o-p-r-s-o-p-q-s-...-t-u-v-a1-b1-u-v-w-y-a1-b1-u-x-z-c1-d1-f1-E
002	条件覆盖	year=2019, month=3, day=12	1	1	F	T	F	F	F	F	F	F	F	F	S-a-b-d-f-h-j-m-n-o-p-r-s-o-p-r-s-o-p-r-s-o-p-q-s-...-t-u-v-a1-b1-u-v-w-a1-b1-u-x-z-c1f1-E
003	条件覆盖	year=2000, month=3, day=12	7	7	T	F	T	F	F	F	F	F	F	T	S-a-b-c-d-f-h-j-m-n-o-p-r-s-o-p-r-s-o-p-r-s-o-p-q-s-...-t-u-v-a1-b1-u-v-w-y-a1-b1-u-x-z-c1-d1-f1-E
004	条件覆盖	year=-2, month=-1, day=-5	0	0	F	T	F	T	T	T					S-a-b-d-E
005	条件覆盖	year=10001 ,month=12, day=1	-1	-1	F	T	F	F	F	F	T				S-a-b-d-f-g-E
006	条件覆盖	year=2019, month=13, day=1	-2	-2	F	T	F	F	F	F	F	T			S-a-b-d-f-h-i-E
007	条件覆盖	year=2019, month=2, day=30	-3	-3	F	T	F	F	F	F	F	F	T		S-a-b-d-f-h-j-k-E

② 以判定条件覆盖设计的测试实例如下：

编号	执行条件	输入	期望输出	实际输出	条件以及相应判定结果												执行路径
					1	2	3	**b**	4	5	6	**d**	**f**	**h**	**j**	**c1**	

001	判定条件覆盖	year=2018, month=3, day=11	7	7	T	T	F	**T**	F	F	F	**F**	**F**	**F**	**F**	**T**	S-a-b-c-d-f-h-j-m-n-o-p-r-s-o-p-r-s-o-p-r-s-o-p-q-s-...-t-u-v-a1-b1-u-v-w-y-a1-b1-u-x-z-c1-d1-f1-E
002	判定条件覆盖	year=2019, month=3, day=12	1	1	F	T	F	**F**	F	F	F	**F**	**F**	**F**	**F**	**F**	S-a-b-d-f-h-j-m-n-o-p-r-s-o-p-r-s-o-p-r-s-o-p-q-s-...-t-u-v-a1-b1-u-v-w-a1-b1-u-x-z-c1f1-E
003	判定条件覆盖	year=2000, month=3, day=12	7	7	T	F	T	**T**	F	F	F	**F**	**F**	**F**	**F**	**T**	S-a-b-c-d-f-h-j-m-n-o-p-r-s-o-p-r-s-o-p-r-s-o-p-q-s-...-t-u-v-a1-b1-u-v-w-y-a1-b1-u-x-z-c1-d1-f1-E
004	判定判定条件覆盖	year=-2, month=-1, day=-5	0	0	F	T	F	**F**	T	T	T	**T**					S-a-b-d-E
005	判定条件覆盖	year=10001 ,month=12, day=1	-1	-1	F	T	F	**F**	F	F	F	**F**	**T**				S-a-b-d-f-g-E
006	判定条件覆盖	year=2019, month=13, day=1	-2	-2	F	T	F	**F**	F	F	F	**F**	**F**	**T**			S-a-b-d-f-h-i-E
007	判定条件覆盖	year=2019, month=2, day=30	-3	-3	F	T	F	**F**	F	F	F	**F**	**F**	**F**	**T**		S-a-b-d-f-h-j-k-E

③ 以条件组合覆盖设计的测试实例如下：（考虑逻辑短路的情况）

编号	执行条件	输入	期待输出	实际输出	条件结果										执行路径
					1	2	3	4	5	6	f	h	j	c1	
001	条件组合覆盖	year=2018 ,month=3, day=11	7	7	T	T	F	F	F	F	F	F	F	T	S-a-b-c-d-f-h-j-m-n-o-p-r-s-o-p-r-s-o-p-r-s-o-p-q-s-...-t-u-v-a1-b1-u-v-w-y-a1-b1-u-x-z-c1-d1-f1-E
002	条件组合覆盖	year=2000 0,month=3 ,day=13	-1	-1	T	F	T	F	F	F	T				S-a-b-d-f-g-E
003	条件组合覆盖	year=1900 ,month=13 ,day=1	-2	-2	T	F	F	F	F	F	F	T			S-a-b-d-f-h-i-E

004	条件组合覆盖	year=2019，month=3,day=32	-3	-3	F	T	F	F	F	F	F	F	T		S-a-b-d-f-h-j-k-E
005	条件组合覆盖	year=-2 ，month=2，day=3	0	0	F	T	F	T							S-a-b-d-E
006	条件组合覆盖	year=2019,month=-2,day=2	0	0	F	T	F	F	T						S-a-b-d-E
007	条件组合覆盖	year=2019,month=2,day=-2	0	0	F	T	F	F	F	T					S-a-b-d-E

(5) 对于循环的测试用例设计如下：

为了理由表述的方便性，将程序中出现的循环按出现的先后顺序分别记为循环 1，循环 2

由于循环 1 与循环 2 构成串接循环，二者之间互不影响，所以可以依据这一特点减少测试用例的个数

编号	执行条件	输入	期望输出	实际输出	设计理由
1	循环测试	year=1,month=1,day=1	1	1	跳过循环 1 跳过循环 2
2	循环测试	year=2,month=2,day=3	7	7	循环 1 执行了一次 循环 2 执行了一次
3	循环测试	year=3,month=3,day=5	5	5	循环 1 执行了 2 次 循环 2 执行了 2 次
4	循环测试	year=2019,month=5,day=5	7	7	循环 1 执行了 2018 次 循环 2 执行了 4 次
5	循环测试	year=9999,month=11,day=3	3	3	循环 1 执行了 9999 次 循环 2 执行了 11 次

(6) 经过对程序流图的分析，可能的路径共有 8+（8*3*2）=56 条，这其中包括不可达路径

按照我正常设计测试用例的速度，每条路径需要 1 分半钟，那么总共需要 84 分钟

(7) 由于两段语句中判断的结构是一样的只有变量不一样，于是只需要设计一次满足 MC/DC 的测试用例即可，设计的测试用例如下：

编号	year%4==0 (i%4==0)	year%100!=0 (i%100!=0)	year%400==0 (i%400==0)	Outcome
001	T	T	F	T
002	T	F	T	T
003	T	F	F	F
004	F	T	F	F

三、实验体会

(1) 经过本次实验，加深了对于语句覆盖、判定覆盖、条件覆盖、判定条件覆盖、条件组合覆盖、路径覆盖、MC/DC 覆盖的理解，更能区别对于这些覆盖之间细微的差别。

(2) 对于如何基于控制流去测试一段程序，如何设计测试用例有了更深刻的体会，提高了动手实验能力。