

# 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： 计算机软件楼

实验日期： 2015. 10. 21

学生姓名： 陈建蓉

学生学号： 71113106

东南大学 软件学院 制

# 一、实验目的

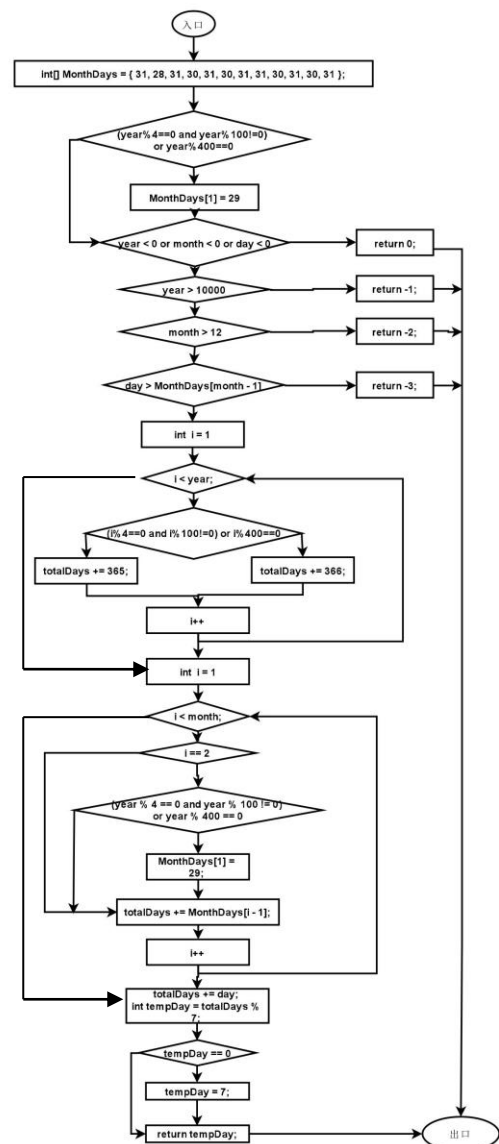
- (1) 巩固白盒测试知识，对于给定的待测程序，能熟练应用基本控制流覆盖方法设计测试用例；
- (2) 通过绘制程序流程图，实现对程序源代码的逻辑描述；
- (3) 培养严谨的测试精神，学习测试用例的设计和分析。

## 二、实验内容

### (一) 实验 1: 控制流测试技术实验

1. 给出 getWeekday() 的流程图:

1	int[] MonthDays
2	((year % 4 == 0) && (year % 100 != 0))    year % 400 == 0
3	MonthDays[1] = 29;
4	year < 0    month < 0    day < 0
5	return 0;
6	year > 10000
7	return -1;
8	month > 12
9	return -2;
10	day > MonthDays[month - 1]
11	return -3;
12	int totalDays = 0;
13	((i % 4 == 0) && (i % 100 != 0))    i % 400 == 0
14	totalDays += 366;
15	totalDays += 365;
16	i == 2
17	((year % 4 == 0) && (year % 100 != 0))    year % 400 == 0
18	MonthDays[1] = 29;
19	totalDays += MonthDays[i - 1];
20	totalDays += day; tempDay = totalDays % 7;
21	tempDay == 0
22	tempDay = 7;
23	return tempDay



2. 设计测试用例

2.1. 语句覆盖测试用例

编号	执行条件	输入	期望输出	实际输出	判定取值								执行路径	
		yyyy/mm/d			2	4	6	8	10	13	16	17		21
1	判定覆盖	400/-1/-1	0	0	T	T								1-2-3-4-5
2	判定覆盖	100001/1/1	-1	-1	F	F	T							1-2-4-6-7
3	判定覆盖	1/13/1	-2	-2	F	F	F	T						1-2-4-6-8-9
4	判定覆盖	1/12/41	-3	-3	F	F	F	F	T					1-2-4-6-8-10-11
5	判定覆盖	2016-2-7	7	7	T	F	F	F	F	T/ F	T/ F	T	T	1-2-4-6-8-10-12-13-14-15-16-17-18-19-20-21-22-23
6	判定覆盖	2015-2-2	1	1	T	F	F	F	F	T/ F	T/ F	F	F	1-2-4-6-8-10-12-13-14-15-16-17-19-20-21-23

编号	执行条件	输入	期望输出	实际输出	执行路径
		yyyy/mm/dd			
1	条件覆盖	400/-1/-1	0	0	1-2-3-4-5
2	条件覆盖	-401/-1/-1	0	0	1-2-4-5
3	条件覆盖	401/1/-1	0	0	1-2-4-5
4	条件覆盖	100001/1/1	-1	-1	1-2-4-6-7
5	条件覆盖	1/13/1	-2	-2	1-2-4-6-8-9
6	条件覆盖	1/12/41	-3	-3	1-2-4-6-8-10-11
7	条件覆盖	2015-2-2	1	1	1-2-4-6-8-10-12-13-14-15-16-17-19-20-21-23
8	条件覆盖	2000-2-6	7	7	1-2-4-6-8-10-12-13-14-15-16-17-18-19-20-21-22-23

[illegible]

3	401/1/-1	F	T	F	F	F	T											
4	100001/1/1	F	T	F	F	F	F	T										
5	1/13/1	F	T	F	F	F	F	F	T									
6	1/12/41	F	T	F	F	F	F	F	F	T								
7	2015-2-2	F	T	F	F	F	F	F	F	F	T/F	T/F	T/F	T/F	F	T	F	F
8	2000-2-6	T	F	T	F	F	F	F	F	F	T/F	T/F	T/F	T/F	T	F	T	T

## 2.4. 判定条件覆盖测试用例

编号	执行条件	输入	期望输出	实际输出	执行路径
		yyyy/mm/dd			
1	判定条件覆盖	400/-1/-1	0	0	1-2-3-4-5
2	判定条件覆盖	-401/-1/-1	0	0	1-2-4-5
3	判定条件覆盖	401/1/-1	0	0	1-2-4-5
4	判定条件覆盖	100001/1/1	-1	-1	1-2-4-6-7
5	判定条件覆盖	1/13/1	-2	-2	1-2-4-6-8-9
6	判定条件覆盖	1/12/41	-3	-3	1-2-4-6-8-10-11
7	判定条件覆盖	2015-2-2	1	1	1-2-4-6-8-10-12-13-14-15-16-17-19-20-21-23
8	判定条件覆盖	2000-2-6	7	7	1-2-4-6-8-10-12-13-14-15-16-17-18-19-20-21-22-23

编号	判定条件取值																				
	2			2	4			4	6	8	10	13			13	16	17			17	21
	a	b	c		d	e	f					g	h	i			j	k	l		
1	T	F	T	T	F	T		T													
2	F	T	F	F	T			T													
3	F	T	F	F	F	F	T	T													
4	F	T	F	F	F	F	F	F	T												
5	F	T	F	F	F	F	F	F	F	T											
6	F	T	F	F	F	F	F	F	F	F	T										
7	F	T	F	F	F	F	F	F	F	F	F	T/F	T/F	T/F	T/F	T/F	F	T	F	F	F
8	T	F	T	T	F	F	F	F	F	F	F	T/F	T/F	T/F	T/F	T/F	T	F	T	T	T

## 2.5. 条件组合覆盖测试用例

编号	执行条件	输入	期望输出	实际输出	执行路径
		yyyy/mm/dd			
1	条件组合覆盖	400/-1/-1	0	0	1-2-3-4-5
2	条件组合覆盖	-401/-1/-1	0	0	1-2-4-5
3	条件组合覆盖	401/1/-1	0	0	1-2-4-5
4	条件组合覆盖	100001/1/1	-1	-1	1-2-4-6-7
5	条件组合覆盖	1/13/1	-2	-2	1-2-4-6-8-9
6	条件组合覆盖	1/12/41	-3	-3	1-2-4-6-8-10-11
7	条件组合覆盖	2016-2-7	7	7	1-2-4-6-8-10-12-13-14-15-16-17-18-19-20-21-22-23
8	条件组合覆盖	1900-2-1	4	4	1-2-4-6-8-10-12-13-15-16-17-18-19-20-21-23

9	条件组合覆盖	2015-2-2	1	1	1-2-4-6-8-10-12-13-14-15-16-17-19-20-21-23
10	条件组合覆盖	2000-2-6	7	7	1-2-4-6-8-10-12-13-14-15-16-17-18-19-20-21-22-23

编号	条件组合取值																				
	2			2	4			4	6	8	10	13			13	16	17			17	21
	a	b	c		d	e	f					g	h	i			j	k	l		
1	T	F	T	T	F	T		T													
2	F	T	F	F	T			T													
3	F	T	F	F	F	F	T	T													
4	F	T	F	F	F	F	F	F	T												
5	F	T	F	F	F	F	F	F	F	T											
6	F	T	F	F	F	F	F	F	F	F	T										
7	T	T		T	F	F	F	F	F	F	F	T/F	T/F	T/F	T/F	T/F	T	T		T	T
8	T	F	F	F	F	F	F	F	F	F	F	T/F	T/F	T/F	T/F	T/F	T	F	F	F	F
9	F	T	F	F	F	F	F	F	F	F	F	T/F	T/F	T/F	T/F	T/F	F	T	F	F	F
10	T	F	T	T	F	F	F	F	F	F	F	T/F	T/F	T/F	T/F	T/F	T	F	T	T	F

### 3. 路径覆盖测试 getWeekday()

可能的路径有 56 条。

2【二月天数可能情况】\*（4【返回 0, -1, -2, -3 情况】+3【只有平年，有闰年平年，跳过循环情况】\*4【只有平年一月，平年二月及以后，闰年二月及以后，跳过循环情况】\*2【星期日，非星期日情况】）

若不考虑循环内部，则可能有 24 条路径

2【二月天数可能情况】\*（4【返回 0, -1, -2, -3 情况】+2【执行循环，跳过循环情况】\*2【执行循环，跳过循环情况】\*2【星期日，非星期日情况】）

### 4. MC/DC(修订的判定条件覆盖)

```

...
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;
}
...

```

编号	执行条件	输入	期望输出	实际输出	条件			判定
		yyyy/mm/dd			1	2	3	
1	MC/DC 覆盖	2016-2-7	7	7	T	T		T
2	MC/DC 覆盖	1900-2-1	4	4	T	F	F	F
3	MC/DC 覆盖	2015-2-2	1	1	F	T	F	F
4	MC/DC 覆盖	2000-2-6	7	7	T	F	T	T

	True	False
1	1	3
2	1	2
3	2	4

```

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

```

编号	执行条件	输入	期望输出	实际输出	条件			判定
		yyyy/mm/dd			1	2	3	
1	MC/DC 覆盖	2016-2-7	7	7	T	T		T
					T	F	F	F
					F	T	F	F
					T	F	T	T

由于循环 `i++`，一个用例即可满足。

### 三、实验体会

1. 通过路径覆盖测试，发现程序中缺陷即未对 `year==0`, `month==0` 情况。`Month` 为 0 时，`MonthDays[month - 1]` 存在 `Exception`。而 `year` 为 0，现实不存在公元 0 年。
2. 程序存在动态白盒测试不能发现的错误：
  - 1) 当 `day` 为 0，现实不存在 0 日。
  - 2) 当输入不为数字时，异常没有处理。
3. 白盒测试的过程中，找出关键有效的测试用例是关键，例如条件覆盖和判定覆盖，找到关键的测试用例可以有效控制测试用例数目。
4. 白盒测试的过程中，在判断语句和循环语句理清逻辑也很重要。