

实验一 白盒测试实验一

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、实验环境

硬件环境：PC 机一台

软件环境：Java 编程环境：Java SDK + Eclipse

C/C++编程环境：Visual Studio

程序流程图绘制：Visio 环境下手工绘制；其它环境：ProcessOn

自动生成工具：Soot (<http://sable.github.io/soot/>)，Visustin v7 等

待测程序：**WeekA**

实验指导书、Eclipse 和待测程序可从 <FTP://223.3.68.135> 或课程主页下载：

<http://cse.seu.edu.cn/PersonalPage/pwang/course/st.html>

Java 实验环境配置：

1. 在制定服务器或网站下载 Java SDK 和 Eclipse
2. 配置 Java 运行环境
3. 启动 Eclipse

有精力和兴趣的同学可尝试用 Soot 辅助生成流程图，soot 的下载和安装如下

1. 命令行方式：soot-trunk.jar (参考 <https://github.com/Sable/soot/wiki/Tutorials>)
2. Eclipse 插件方式：soot.zip (参考 <https://github.com/Sable/soot/wiki/Eclipse-Plugin-Installation>)

生成流程图的方法参考 Mooc 课程录像以及 soot 帮助文档

三、实验内容

实验背景：

已知公元 1 年 1 月 1 日是星期一，我们可以实现一个程序，使得只要输入年月日，程序就能自动回答当天是星期几。

现有 07 级同学实现上述功能的程序 **WeekA**，该程序用于计算输入日期是星期几。

实验 1：控制流测试技术实验

运用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**，这是进行基于控制流动态白盒测试的基础。

(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条共有多少条？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？

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(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;
}
...
```

实验过程注意事项：

- 1. 程序流程图只有 1 个入口，1 个出口；
- 2. 根据使用的测试方法，把判定和条件在测试用例执行时的取值给出来；
- 3. 没有执行的判定和条件没有取值；
- 4. 条件组合测试时，对一个判定中的条件取值进行组合；
- 5. 复杂的条件组合要优先进行约简，然后再设计测试用例；
- 6. 注意判定中条件取值的逻辑短路问题；
- 7. MC/DC 覆盖也要注意逻辑短路问题。

四、实验要求

(1) 根据题目要求编写测试用例.

一些测试用例书写示例:

要点: 根据测试方法的特点, 把需要明确给出的判定或条件取值, 以及路径等必要信息给出来

| 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 | x = 4, y = 7, z = 3 | 2 | 2 | 1->2->3->4->5->6->7->8->9->10 |
| 002 | 判定覆盖 | x = 5, y = 7, z = 6 | 0 | 0 | 1->2->3->4->5->6->7->8->9->10 |

| 编号 | 执行条件 | 输入 | 期望输出 | 实际输出 | 条件判断 | | | | | | | 基本路径 |
|----|--------|----------|------|------|------|---|---|---|---|---|----|----------------|
| | | yy/mm/dd | | | 1 | 3 | 4 | 5 | 7 | 8 | 10 | |
| P1 | 基本路径测试 | 0/1/1 | 输入错误 | 输入错误 | T | | | | | | | 1—2—13 |
| P2 | 基本路径测试 | 2/-1/1 | 输入错误 | 输入错误 | F | F | | F | T | | | 1—3—5—7—9—13 |
| P3 | 基本路径测试 | 200/-1/1 | 输入错误 | 输入错误 | F | T | F | F | T | | | 1—3—4—5—7—9—13 |

| 编号 | 执行条件 | 输入 | 期望输出 | 实际输出 | 条件判断 | | | | | | 基本路径 |
|----|--------|----------|------|------|------|---|---|---|---|----|--------------------------------------|
| | | yy/mm/dd | | | 3 | 4 | 5 | 8 | 9 | 10 | |
| P1 | 基本路径测试 | 5/1/1 | 星期六 | 星期六 | T | T | | F | | F | 1—2—3—4—6—2—8—10—12—13—15—16—17—18 |
| P2 | 基本路径测试 | 无测试用例 | | | F | | T | F | | F | 1—2—3—5—6—2—8—10—12—13—15—16—17—18 |
| P3 | 基本路径测试 | 401/1/1 | 星期一 | 星期一 | T | F | T | F | | F | 1—2—3—4—5—6—2—8—10—12—13—15—16—17—18 |

(2) 撰写实验报告 (参照实验报告模板)

(3) 实验报告命名规则: 71108129-11.doc OR 71108129-11.pdf

(4) 实验报告提交: 提交到 FTP 服务器的 homework 目录下; 截止时间为周日

五、实验思考

- (1) 通过测试, 是否发现程序中存在的缺陷?
- (2) 程序中是否还存在采用动态白盒测试技术未能发现的缺陷?
- (3) 在存在逻辑短路的情况下, 请分析条件覆盖、判定条件覆盖、条件组合覆盖、MC/DC 覆盖的联系和区别。
- (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。

六、附录

一、Java 运行环境配置

1. 下载 Java SDK，然后进行安装；
2. 为你的机器配置 Java 运行环境

通过设置系统变量，使系统能随时调用 Java 开发包中的文件。具体方法如下：

假设你的 Java 安装路径为：c:\jdk1.6.0，则你需要配置以下三个环境变量：

分别设置如下三个环境变量：

PATH=C:\jdk1.6.0\bin

CLASSPATH=.;%JAVA_HOME%\lib\tools.jar;%JAVA_HOME%\lib\dt.jar

（注意，CLASSPATH 最前面是有个“.”的，表示当前目录）

JAVA_HOME=C:\jdk1.6.0

详细的安装和配置过程可参见文档“Java 环境配置”

二、Eclipse 安装和使用

1. 下载 Eclipse
2. 安装 Eclipse
3. 具体使用方法请自行学习