# 软件测试基础与实践

# 实验报告

**实验名称： 黑盒测试实验一**

**实验地点： 软件学院机房**

**实验日期： 2020年12月3日**

**学生姓名： 叶宏庭**

**学生学号： 71118415**

# 东南大学 软件学院 制

# 一、实验目的

（1）能熟练应用黑盒测试中的等价类划分方法设计测试用例；

（2）能熟练应用黑盒测试中的边界值分析方法设计测试用例；

（3）能数量综合使用等价类划分和边界值分析解决黑盒测试需求；

（4）能够在黑盒测试用例设计中同时考虑正面测试和负面测试；

（5）学习测试用例的书写。

# 二、实验内容

### （一）题目1: NextDate问题的黑盒测试

**1. 实验背景**

日期是软件中被频繁处理的信息之一，软件开发人员有必要了解的一些公历历法的相关知识。

公历的前身是古罗马凯撒修订的儒略历。 根据儒略历的规定，每 4 年有 1 个闰年，闰年为 366 日，其余 3 年(称为平年)各有 365 日。公元年数能被 4 除得尽的是闰年。儒略历 1 年平均长 365.25 日，比实际公转周期的 365.2422日长 11 分 14 秒，即每 400 年约长 3 日。这样到公元 16 世纪时已经积累了有 10 天误差。可以明显感觉到两至两分提前了。在此情况下，教皇格列高里十三世于 1582 年宣布改历。先是一步到位把儒略历 1582 年 10 月 4 日的下一天定为格列历10 月15 日，中间跳过 10 天。同时修改了儒略历置闰法则。除了保留儒略历年数被 4 除尽的是闰年外。增加了被 100 除得尽而被 400 除不尽的则不是闰年的规定。这样的做法可在 400 年中减少 3 个闰年。在格列高里历历法里，400 年中有 97 个闰年(每年 366 日)及 303 个平年(每年 365 日)，所以每年平均长 365.2425 日，与公转周期的 365.2422 日十分接近。可基本保证到公元 5000 年前误差不超过1天。在软件开发和测试中，我们需要注意以下的一些有用信息：

* 1582 年 10 月 5 日至 10 月 14 日排除在公历外
* 2038 年 1 月 19 日是 BIOS 提供的记时基准时间 1970 年 1 月 1 日的最大值(下一个千年虫问题的根源)
* 英国 1752 年才采用阳历，他们扣除 9/3/1752 到 9/13/1752 年同步以月亮为参照的立法

注意，以上信息中，后两条并不影响我们所进行的测试活动,可不用考虑。

**2. 实验要求**

NextDate 程序中有 3 个输入，分别对应一个日期的年、月、日，程序能输出给定日期的下一天。程序能接收的日期输入范围为 1582 年 1 月 1 日到 3000 年 12 月 31 日。

要求：

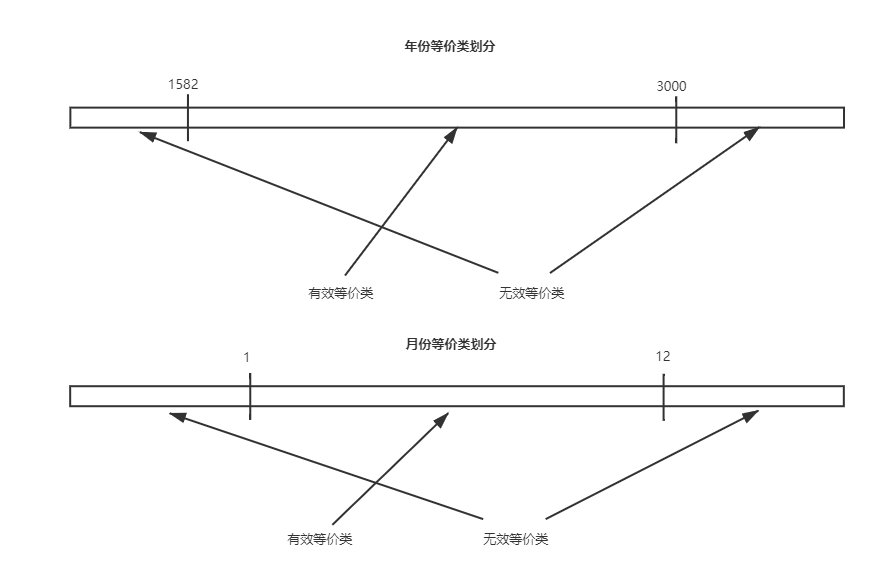
(1) 综合使用等价类划分和边界值分析方法对该程序进行黑盒测试；

(2) 设计的测试用例都要有充分的设计理由。

**3. 实验过程与结果**

**(1) 等价类划分法**

① 等价类划分



首先根据输入的年份，月份进行等价类初步划分，初步划分结果如下表：

|  |  |  |
| --- | --- | --- |
| **输入数据** | **有效等价类** | **无效等价类** |
| 年份yy | 1. 1582 <= yy <= 3000 | 1. yy < 1582 2. yy > 3000 |
| 月份mm | 1. 1 <= mm <= 12 | 1. mm <= 0 2. mm > 12 |
| 日期dd | 参照下方给出的日期划分表 | |

|  |  |  |  |
| --- | --- | --- | --- |
| **输入数据mm** | | **有效等价类** | **无效等价类** |
| 大月（1，3，5，7，8，10，12） | | 1. 1 <= dd <= 31 | 1. dd <= 0 2. dd >= 32 |
| 小月（4，6，9，10） | | 1. 1 <= dd <= 30 | 1. dd <= 0 2. dd >= 31 |
| 平月（2） | yy闰年 | 1. 1 <= dd <= 29 | 1. dd <= 0 2. dd >= 30 |
| yy非闰年 | 1. 1 <= dd <= 28 | 1. dd <= 0 2. dd >= 29 |

注意：特定无效等价类19：1582/10/5 ~ 1582/10/14的日期输入

② 测试用例设计

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **编号** | **测试用例yy/mm/dd** | **等价类覆盖** | **输入类型** | **预期结果yy/mm/dd** |
| 1 | 1588/8/31 | 1，2，7 | 有效 | 1588/9/1 |
| 2 | 1588/9/30 | 1，2，8 | 有效 | 1588/10/1 |
| 3 | 1588/2/29 | 1，2，9 | 有效 | 1588/3/1 |
| 4 | 1589/2/28 | 1，2，10 | 有效 | 1589/3/1 |
| 5 | 1700/2/28 | 1，2，10 | 有效 | 1700/3/1 |
| 6 | 1588/8/0 | 1，2，11 | 无效 | 警告信息 |
| 7 | 1588/8/33 | 1，2，12 | 无效 | 警告信息 |
| 8 | 1588/9/0 | 1，2，13 | 无效 | 警告信息 |
| 9 | 1588/9/33 | 1，2，14 | 无效 | 警告信息 |
| 10 | 1588/2/0 | 1，2，15 | 无效 | 警告信息 |
| 11 | 1588/2/30 | 1，2，16 | 无效 | 警告信息 |
| 12 | 1589/2/0 | 1，2，17 | 无效 | 警告信息 |
| 13 | 1589/2/29 | 1，2，18 | 无效 | 警告信息 |
| 14 | 1588/0/1 | 1，5 | 无效 | 警告信息 |
| 15 | 1588/13/1 | 1，6 | 无效 | 警告信息 |
| 16 | 1581/4/1 | 3 | 无效 | 警告信息 |
| 17 | 3001/4/1 | 4 | 无效 | 警告信息 |
| 18 | 1582/10/5 | 19 | 无效 | 警告信息 |

**(2) 边界条件法**

① 确定边界条件

1. 每次只考虑一个参数的边界，固定其他参数
2. 补充确定的关联边界值

边界条件设计

1. 固定dd、yy的月边界条件

mm：1，12

dd：1-28

yy：1582-3000

|  |  |  |  |
| --- | --- | --- | --- |
| **边界条件** | **mm** | **dd** | **yy** |
| 1 | 1 | 1-28 | 1582-3000 |
| 2 | 12 | 1-28 | 1582-3000 |

1. 固定mm、yy的日边界条件

dd：1，30，31

dd：大/小月

yy：1582-3000

|  |  |  |  |
| --- | --- | --- | --- |
| **边界条件** | **mm** | **dd** | **yy** |
| 3 | 小月 | 1 | 1582-3000 |
| 4 | 小月 | 30 | 1582-3000 |
| 5 | 大月 | 1 | 1582-3000 |
| 6 | 大月 | 31 | 1582-3000 |
| 7 | 2月 | 1 | 闰年 |
| 8 | 2月 | 29 | 闰年 |
| 9 | 2月 | 1 | 非闰年 |
| 10 | 2月 | 28 | 非闰年 |

1. 固定mm、dd的年边界条件

yy：1582，3000，闰年，非闰年

mm：1-12

dd：1-28

|  |  |  |  |
| --- | --- | --- | --- |
| **边界条件** | **mm** | **dd** | **yy** |
| 11 | 1-12 | 1-28 | 1582 |
| 12 | 1-12 | 1-28 | 3000 |

1. 补充确定的关联边界条件

|  |  |  |  |
| --- | --- | --- | --- |
| **边界条件** | **mm** | **dd** | **yy** |
| 13 | 1 | 1 | 1582 |
| 14 | 12 | 31 | 3000 |
| 15 | 10 | 5 | 1582 |
| 16 | 10 | 14 | 1582 |

②测试用例设计

总测试用例数 = 3 \* 16 = 48

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **编号** | **测试用例yy/mm/dd** | **边界条件** | **输入类型** | **预期结果yy/mm/dd** |
| 1 | 1588/0/28 | 1 | 无效 | 警告信息 |
| 2 | 1588/1/28 | 有效 | 1588/1/29 |
| 3 | 1588/2/28 | 有效 | 1588/2/29 |
| 4 | 1588/11/28 | 2 | 有效 | 1588/11/29 |
| 5 | 1588/12/28 | 有效 | 1588/12/29 |
| 6 | 1588/13/28 | 无效 | 警告信息 |
| 7 | 1588/9/0 | 3 | 无效 | 警告信息 |
| 8 | 1588/9/1 | 有效 | 1588/9/2 |
| 9 | 1588/9/2 | 有效 | 1588/9/3 |
| 10 | 1588/9/29 | 4 | 有效 | 1588/9/30 |
| 11 | 1588/9/30 | 有效 | 1588/10/1 |
| 12 | 1588/9/31 | 无效 | 警告信息 |
| 13 | 1588/8/0 | 5 | 无效 | 警告信息 |
| 14 | 1588/8/1 | 有效 | 1588/8/2 |
| 15 | 1588/8/2 | 有效 | 1588/8/3 |
| 16 | 1588/8/30 | 6 | 有效 | 1588/8/31 |
| 17 | 1588/8/31 | 有效 | 1588/9/1 |
| 18 | 1588/8/32 | 无效 | 警告信息 |
| 19 | 1588/2/0 | 7 | 无效 | 警告信息 |
| 20 | 1588/2/1 | 有效 | 1588/2/2 |
| 21 | 1588/2/2 | 有效 | 1588/2/3 |
| 22 | 1588/2/28 | 8 | 有效 | 1588/2/29 |
| 23 | 1588/2/29 | 有效 | 1588/3/1 |
| 24 | 1588/2/30 | 无效 | 警告信息 |
| 25 | 1589/2/0 | 9 | 无效 | 警告信息 |
| 26 | 1589/2/1 | 有效 | 1589/2/2 |
| 27 | 1589/2/2 | 有效 | 1589/2/3 |
| 28 | 1589/2/27 | 10 | 有效 | 1589/2/28 |
| 29 | 1589/2/28 | 有效 | 1589/3/1 |
| 30 | 1589/2/29 | 无效 | 警告信息 |
| 31 | 1581/1/1 | 11 | 无效 | 警告信息 |
| 32 | 1582/1/1 | 有效 | 1582/1/2 |
| 33 | 1583/1/1 | 有效 | 1583/1/2 |
| 34 | 2999/1/1 | 12 | 有效 | 2999/1/2 |
| 35 | 3000/1/1 | 有效 | 3000/1/2 |
| 36 | 3001/1/1 | 无效 | 警告信息 |
| 37 | 1581/12/31 | 13 | 无效 | 警告信息 |
| 38 | 1582/1/1 | 有效 | 1582/1/2 |
| 39 | 1582/1/2 | 有效 | 1582/1/3 |
| 40 | 3000/12/30 | 14 | 有效 | 3000/12/31 |
| 41 | 3000/12/31 | 有效 | 3001/1/1 |
| 42 | 3001/1/1 | 无效 | 警告信息 |
| 43 | 1582/10/4 | 15 | 有效 | 1582/10/15 |
| 44 | 1582/10/5 | 无效 | 警告信息 |
| 45 | 1582/10/6 | 无效 | 警告信息 |
| 46 | 1582/10/13 | 16 | 无效 | 警告信息 |
| 47 | 1582/10/14 | 无效 | 警告信息 |
| 48 | 1582/10/15 | 有效 | 1582/10/16 |

### （二）题目2:四边形覆盖问题的黑盒测试

**1. 实验背景**

四边形覆盖问题描述：

（1）程序输入：2 个四边形：（X1Coord, Y1Coord, Width1, Height1）和 （X2Coord, Y2Coord, Width2, Height2），其中前 2 个参数是四边形左上角坐标，后 2 个参数指四边形的宽和高；

（2）程序输出：两个四边形的覆盖关系。

（3）四边形覆盖：判断 2 个四边形在平面上的覆盖关系。

**2. 实验要求**

（1）利用等价类划分和边界值分析方法，设计四边形覆盖问题的测试用例。请给出测试用例的具体设计思路。

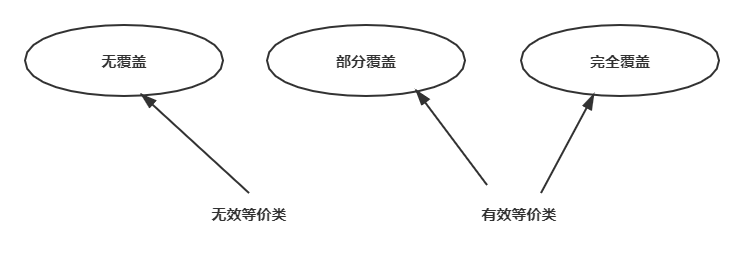
（2）github 上有一个少有人关注的项目 https://github.com/cuthullu/box-black-box，（可下载该项目的源码box-black-box-gh-pages.zip，解压后可运行 index.html）。这个项目中，给出了四边形问题的可视化测试界面，其中还包含 5 种判断四边形关系的函数。

（3）请利用（1）中设计的测试用例来对 box-black-box 项目进行黑盒测试，通过黑盒测试，分析该项目给出的 6种函数中是否存在 BUG。

**3. 实验过程与结果**

（一）等价类划分法

① 等价类划分



|  |  |
| --- | --- |
| **等价类编号** | **覆盖情况** |
| 1 | 无覆盖 |
| 2 | 部分覆盖 |
| 3 | 完全覆盖 |

② 测试用例设计

1. 无覆盖与部分覆盖考虑八个方向的覆盖情况；
2. 完全覆盖考虑互相包含两种情况
3. 部分覆盖考虑两个四边形形成“+”形状。

|  |  |  |  |
| --- | --- | --- | --- |
| **编号** | **四边形1** | **四边形2** | **覆盖等价类** |
| 1 | （5，5，2，2） | （5，3，1，1） | 1 |
| 2 | （5，5，2，2） | （3，3，1，1） | 1 |
| 3 | （5，5，2，2） | （3，5，1，1） | 1 |
| 4 | （5，5，2，2） | （3，7，1，1） | 1 |
| 5 | （5，5，2，2） | （5，8，1，1） | 1 |
| 6 | （5，5，2，2） | （7，8，1，1） | 1 |
| 7 | （5，5，2，2） | （8，5，1，1） | 1 |
| 8 | （5，5，2，2） | （8，3，1，1） | 1 |
| 9 | （5，5，2，2） | （5，4，2，2） | 2 |
| 10 | （5，5，2，2） | （4，4，2，2） | 2 |
| 11 | （5，5，2，2） | （4，5，2，2） | 2 |
| 12 | （5，5，2，2） | （4，6，2，2） | 2 |
| 13 | （5，5，2，2） | （5，6，2，2） | 2 |
| 14 | （5，5，2，2） | （6，6，2，2） | 2 |
| 15 | （5，5，2，2） | （6，5，2，2） | 2 |
| 16 | （5，5，2，2） | （6，4，2，2） | 2 |
| 17 | （5，5，1，3） | （4，6，3，1） | 2 |
| 18 | （4，6，3，1） | （5，5，1，3） | 2 |
| 19 | （5，5，3，3） | （6，6，1，1） | 3 |
| 20 | （5，5，3，3） | （4，4，5，5） | 3 |

（二）边界条件法

① 确定边界条件

1. 无覆盖情况下，存在边重合，分为上下左右四边
2. 无覆盖情况下，存在顶点重合，分四个顶点
3. 完全覆盖情况下，存在边重合，分为上下左右四边
4. 完全覆盖情况下，存在顶点重合，分四个顶点

② 设计测试用例

|  |  |  |  |
| --- | --- | --- | --- |
| **编号** | **四边形1** | **四边形2** | **边界条件** |
| 1 | （5，5，2，2） | （5，4，1，1） | I |
| 2 | （5，5，2，2） | （4，5，1，1） |
| 3 | （5，5，2，2） | （5，7，1，1） |
| 4 | （5，5，2，2） | （7，5，1，1） |
| 5 | （5，5，2，2） | （4，4，1，1） | II |
| 6 | （5，5，2，2） | （4，7，1，1） |
| 7 | （5，5，2，2） | （7，7，1，1） |
| 8 | （5，5，2，2） | （7，4，1，1） |
| 9 | （5，5，2，2） | （5，5，1，1） | Ⅲ，Ⅳ |
| 10 | （5，5，2，2） | （5，6，1，1） |
| 11 | （5，5，2，2） | （6，6，1，1） |
| 12 | （5，5，2，2） | （6，5，1，1） |

（三）测试结果

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **编号** | **四边形1** | **四边形2** | **预期结果** | **实际输出(T：算法正确；F：算法错误)** | | | | | |
| **a** | **b** | **c** | **d** | **e** | **f** |
| 1 | （5，5，2，2） | （5，3，1，1） | False | T | T | T | F | T | T |
| 2 | （5，5，2，2） | （3，3，1，1） | False | T | T | T | T | T | T |
| 3 | （5，5，2，2） | （3，5，1，1） | False | T | F | T | F | T | T |
| 4 | （5，5，2，2） | （3，7，1，1） | False | T | T | T | T | T | T |
| 5 | （5，5，2，2） | （5，8，1，1） | False | T | T | T | F | T | T |
| 6 | （5，5，2，2） | （7，8，1，1） | False | T | T | T | T | T | T |
| 7 | （5，5，2，2） | （8，5，1，1） | False | T | T | T | F | T | T |
| 8 | （5，5，2，2） | （8，3，1，1） | False | T | T | T | T | T | T |
| 9 | （5，5，2，2） | （5，4，2，2） | True | T | F | T | T | T | T |
| 10 | （5，5，2，2） | （4，4，2，2） | True | T | F | T | T | T | T |
| 11 | （5，5，2，2） | （4，5，2，2） | True | T | F | T | T | T | T |
| 12 | （5，5，2，2） | （4，6，2，2） | True | T | F | T | T | T | T |
| 13 | （5，5，2，2） | （5，6，2，2） | True | T | T | T | T | T | T |
| 14 | （5，5，2，2） | （6，6，2，2） | True | T | T | T | T | T | T |
| 15 | （5，5，2，2） | （6，5，2，2） | True | T | T | T | T | T | T |
| 16 | （5，5，2，2） | （6，4，2，2） | True | T | F | T | T | T | T |
| 17 | （5，5，1，3） | （4，6，3，1） | True | F | F | T | T | T | T |
| 18 | （4，6，3，1） | （5，5，1，3） | True | F | T | T | T | T | T |
| 19 | （5，5，3，3） | （6，6，1，1） | True | T | T | T | T | T | T |
| 20 | （5，5，3，3） | （4，4，5，5） | True | T | F | T | T | T | T |
| 21 | （5，5，2，2） | （5，4，1，1） | False | T | T | F | F | T | T |
| 22 | （5，5，2，2） | （4，5，1，1） | False | T | T | F | F | T | T |
| 23 | （5，5，2，2） | （5，7，1，1） | False | T | T | F | F | T | T |
| 24 | （5，5，2，2） | （7，5，1，1） | False | T | T | F | F | T | T |
| 25 | （5，5，2，2） | （4，4，1，1） | False | T | T | F | T | T | T |
| 26 | （5，5，2，2） | （4，7，1，1） | False | T | T | F | T | T | T |
| 27 | （5，5，2，2） | （7，7，1，1） | False | T | T | F | T | T | T |
| 28 | （5，5，2，2） | （7，4，1，1） | False | T | T | F | T | T | T |
| 29 | （5，5，2，2） | （5，5，1，1） | True | T | T | T | T | T | T |
| 30 | （5，5，2，2） | （5，6，1，1） | True | T | T | T | T | T | T |
| 31 | （5，5，2，2） | （6，6，1，1） | True | T | T | T | T | T | T |
| 32 | （5，5，2，2） | （6，5，1，1） | True | T | T | T | T | T | T |

通过上表测试结果可得知：函数啊a，b，c，d存在BUG。

# 三、实验思考

1. 通过测试，是否发现程序中存在的缺陷？

答：发现缺陷，函数a，b，c，d存在BUG。

2. 在黑盒测试中，测试用例的设计实际上是一件非常具有挑战性的工作。谈谈你在进行黑盒测试过程中所碰到的难题。

答：等价类如何划分，边界值如何选取，两种方法会有部分测试用例重合。

3. 思考为什么现在企业内大量的项目主要采用黑盒测试，而比较少而且有限的使用白盒测试技术？谈谈你对企业这样做的原因的理解和这样做的危害。

答：首先，使用黑盒测试的原因：企业级项目代码量大，如何针对所有代码都进行白盒测试，会耗费很多时间、人力等，因此采用黑盒测试，只要能够满足用户说明的预期输入输出即可。

危害：黑盒测试容易遗漏一些错误的测试用例，不能找出软件的存在BUG，这些BUG也许会造成致命性错误。

# 四、实验体会

通过本次实验，我更加深入理解了黑盒测试的目的与意义，动手实践过程中，对黑盒测试的方法、过程有了初步了解与掌握，希望在未来的学习与工作中，能够继续学习，深入掌握，设计出更好的测试用例，做一个合格的测试工程师。