****

**软件测试**

**软件测试课程作业说明文档**

|  |  |  |
| --- | --- | --- |
| 组 长 | 严祯颋 | 1750063 |
| 组 员 | 浦家瑞 |  |
| 王笑天 |  |
| 周宇东 | 1751659 |
| 指导老师 | 杜 庆 峰 | |
| 所在院系 | 软件学院 | |
| 专 业 | 软件工程 | |
|  |  | |

二〇二〇年六月

# 练习1—三角形问题及万年历问题

判定三角形类型、万年历分别用边界值和等价类设计测试用例，需求描述如下：

三角形问题：

输入3个正数a、b和c，作为三角形的3条边。通过程序判断出由这3条边所构成的三角形的类型是等边三角形、等腰三角形还是一般三角形，并打印出相应的信息。

万年历问题：

输入3个整数，分别代表年月日，通过程序输出下一个天的日期

## 三角形问题分析

在本题中，我们需要根据输入的三个正数变量a、b、c，判断以这三个整数为边的三角形类型。

在这里我们对程序的输出格式加以规定，以方便之后的测试，对于不同类型的三角形，程序分别输出“等边三角形”、“等腰三角形”和“一般三角形”，同时针对可能存在的a、b、c三条边无法构成三角形的异常情况，输出“无法构成三角形”。

同时，由于在这道题中，三条边的长度只需要满足正数即可，不存在上界，因此在这里人工规定三角形各边取值的定义域为(0,100]。对于超出范围的输入，则增加输出“变量超出取值范围”。

### 边界值法

对于三角形问题来说，我们人工规定各边取值范围为(0,100]，因此，在这里，我们可以规定各边正常值为50，同时假定我们固定住a，b的值，由于构成三角形的条件存在，对于第三条边c，我们可以进一步得出c的一个限制条件，也就是|a-b|<c<a+b，因此此时c的取值范围为max(0,|a-b|)<c<min(100,a+b)。同样，我们可以构造出a，b的取值范围。同时，由于a，b，c的取值依赖于其他两项的取值，因此在本问题中，我们采用健壮性边界测试设计设计用例。

根据上述分析，我们得到以下测试用例：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | a | b | c | 预计输出 |
|  | 0 | 50 | 50 | 变量超出取值范围 |
|  | 2.2 | 50 | 50 | 等腰三角形 |
|  | -0.1 | 50 | 50 | 变量超出取值范围 |
|  | 90.1 | 50 | 50 | 等腰三角形 |
|  | 100 | 50 | 50 | 无法构成三角形 |
|  | 102 | 50 | 50 | 变量超出取值范围 |
|  | 50 | 0 | 50 | 变量超出取值范围 |
|  | 50 | 2.2 | 50 | 等腰三角形 |
|  | 50 | -0.1 | 50 | 变量超出取值范围 |
|  | 50 | 90.1 | 50 | 等腰三角形 |
|  | 50 | 100 | 50 | 无法构成三角形 |
|  | 50 | 102 | 50 | 变量超出取值范围 |
|  | 50 | 50 | 0 | 变量超出取值范围 |
|  | 50 | 50 | 2.2 | 等腰三角形 |
|  | 50 | 50 | -0.1 | 变量超出取值范围 |
|  | 50 | 50 | 90.1 | 等腰三角形 |
|  | 50 | 50 | 100 | 无法构成三角形 |
|  | 50 | 50 | 102 | 变量超出取值范围 |
|  | 50 | 50 | 50 | 等边三角形 |

### 等价类法

通过对题目的分析，我们可以看到，问题的输出有“等边三角形”、“等腰三角形”、“一般三角形”、“无法构成三角形”以及针对无效输入的输出“变量超出取值范围”。

因此我们根据问题输出构造等价类，可以构造出如下等价类：

D1={{a,b,c}|a,b,c构成等边三角形}

D2={{a,b,c}|a,b,c构成非等边等腰三角形}

D3={{a,b,c}|a,b,c构成非等边非等腰的一般三角形}

D4={{a,b,c}|a,b,c不构成三角形}

同时，我们考虑各边的取值范围(0,100]，对于每个参数的无效取值分别有低于最小值和高于最大值两种情况（其中各边取0已经包含于低于最小值这一情况）。

综上所述，在基于“单缺陷”假设的情况下，我们采用弱健壮性等价类测试，可以构造出如下10种测试用例：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | a | b | c | 预计输出 |
|  | 10 | 10 | 10 | 等边三角形 |
|  | 2.5 | 2.5 | 2 | 等腰三角形 |
|  | 60 | 50 | 32.5 | 一般三角形 |
|  | 10 | 20 | 10 | 无法构成三角形 |
|  | 0 | 2 | 2 | 变量超出取值范围 |
|  | 2 | 0 | 2 | 变量超出取值范围 |
|  | 2 | 2 | 0 | 变量超出取值范围 |
|  | 105 | 70 | 70 | 变量超出取值范围 |
|  | 70 | 70 | 105 | 变量超出取值范围 |
|  | 70 | 105 | 70 | 变量超出取值范围 |

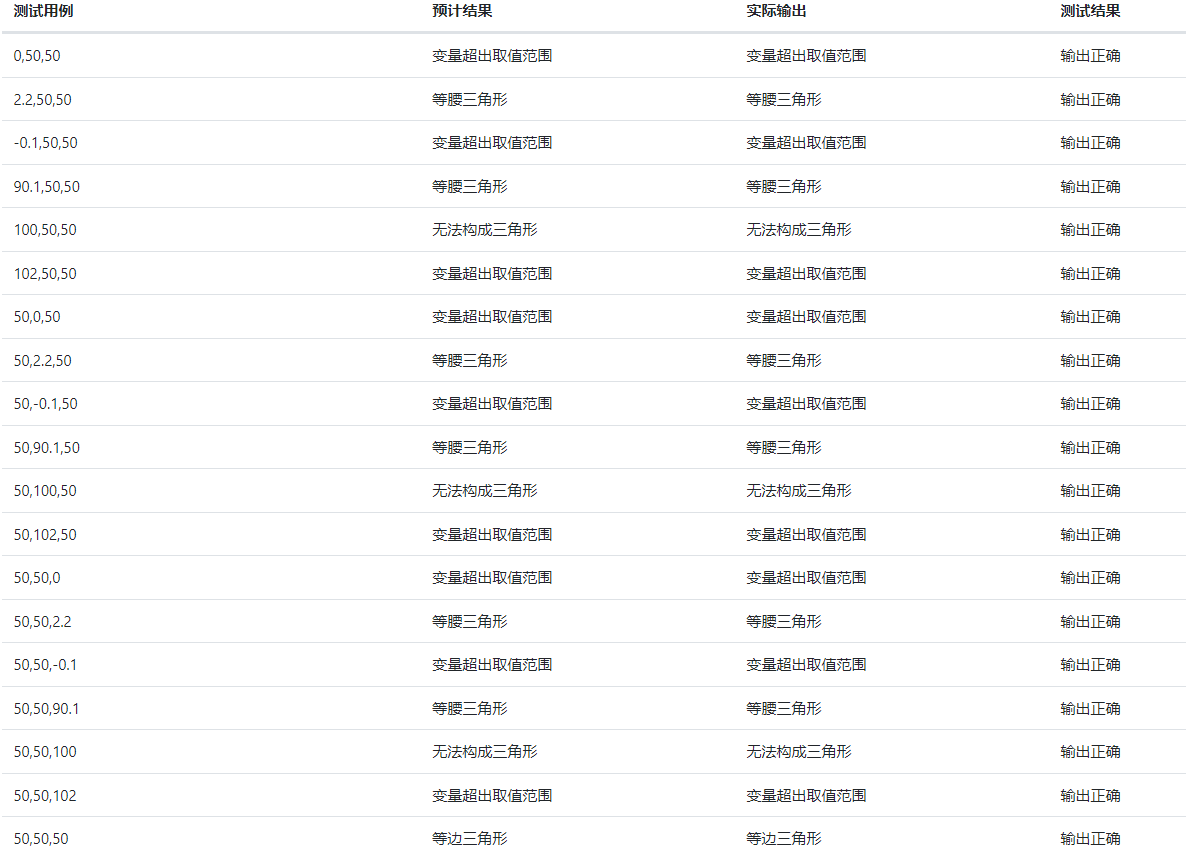
## 三角形问题测试代码设计

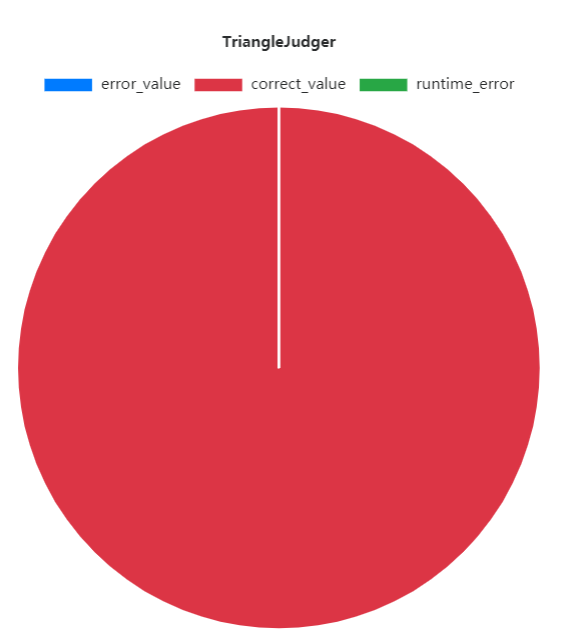
1. **public** **class** TriangleJudger{
2. **private** **static** **final** **double** MinV=0;
3. **private** **static** **final** **double** MaxV=100;
4. **private** **static** **boolean** comp(**double** a,**double** b){
5. **return** Math.abs(a-b)<1e-6;
6. }
7. **public** **static** String getTriangleType(**double** a,**double** b,**double** c){
8. **double** tmp;
9. **if**(a>b){
10. tmp=a;
11. a=b;
12. b=tmp;
13. }
14. **if**(b>c){
15. tmp=b;
16. b=c;
17. c=tmp;
18. }
19. **if**(a>b){
20. tmp=a;
21. a=b;
22. b=tmp;
23. }
24. **if**(a<=MinV||c>MaxV){
25. **return** "变量超出取值范围";
26. }
27. **if**(a+b<c||comp(a+b,c)){
28. **return** "无法构成三角形";
29. }
30. **if**(comp(a,b)&&comp(b,c)){
31. **return** "等边三角形";
32. }
33. **if**(comp(a,b)||comp(a,c)||comp(b,c)){
34. **return** "等腰三角形";
35. }
36. **return** "一般三角形";
37. }
38. }

## 三角形问题测试用例执行情况

使用自制代码测试工具对编写代码加以测试，得到如下结果：

### 边界值法

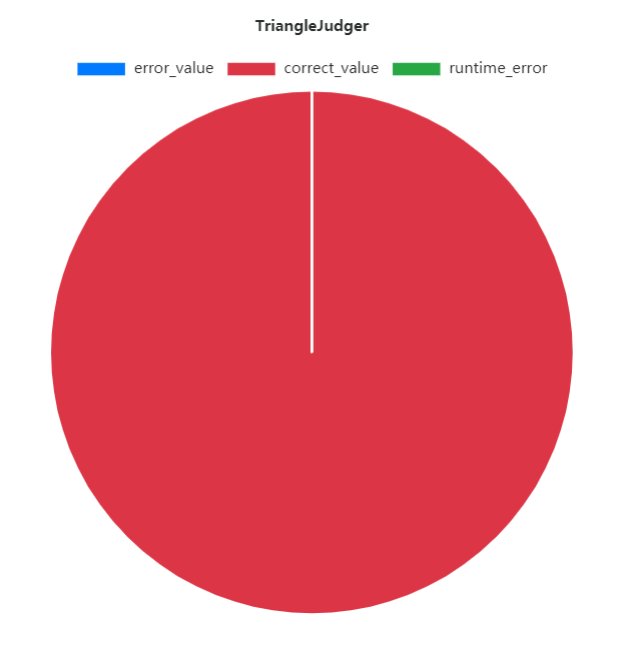




代码测试全部通过

### 等价类法





代码测试全部通过

## 万年历问题分析

在本题中，我们需要根据输入的年月日三个正整数变量，得到下一天的日期输出，在这里，我们规定程序输出格式为形如“yyyy-mm-dd”形式的的字符串，同时，对于非法的输入，程序应输出“非法输入”。

在这里，我们对各变量的代表符号定义如下：

Y：输入的年份

M：输入的月份

D：输入的日期

我们很容易得出，1<=M<=12，1<=D<=31，对于年份的范围，我们人为规定1900<=Y<=2500。

### 边界值法

通过对上述问题的分析，我们使用健壮边界分析，对于Y、M、D的普通值，分别取2020，6，15，得到以下测试用例：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | Y | M | D | 预计输出 |
|  | 2020 | 6 | 1 | 2020-06-02 |
|  | 2020 | 6 | 0 | 非法输入 |
|  | 2020 | 6 | 2 | 2020-06-03 |
|  | 2020 | 6 | 31 | 非法输入 |
|  | 2020 | 6 | 32 | 非法输入 |
|  | 2020 | 6 | 30 | 2020-07-01 |
|  | 2020 | 1 | 15 | 2020-01-16 |
|  | 2020 | 0 | 15 | 非法输入 |
|  | 2020 | 2 | 15 | 2020-02-16 |
|  | 2020 | 12 | 15 | 2020-12-16 |
|  | 2020 | 13 | 15 | 非法输入 |
|  | 2020 | 11 | 15 | 2020-11-16 |
|  | 1900 | 6 | 15 | 1900-06-16 |
|  | 1899 | 6 | 15 | 非法输入 |
|  | 1901 | 6 | 15 | 1901-06-16 |
|  | 2500 | 6 | 15 | 2500-06-16 |
|  | 2501 | 6 | 15 | 非法输入 |
|  | 2499 | 6 | 15 | 2499-06-16 |

### 等价类法

由于本问题中，年月日三个变量并非相互独立的，而在使用边界值的情况下并未考虑到这一点，导致效果不好，下面，我们使用等价类法对测试用例进行分析。

通过常识，我们可以发现，对于等价类的划分，我们主要应当考虑闰年问题以及加一天后就将跳转到下一月或下一年的那部分日期，因此我们可以划分出以下等价类

|  |  |  |
| --- | --- | --- |
| 输入数据 | 有效等价类 | 无效等价类 |
| Y | 1:[1900~2500]中的实际闰年 2:[1900~2500]中的普通闰年 3:[1900~2500]中的平年 | 1:(-∞~1900) 2:(2500~+∞) |
| M | 1:2月 2:4,6,9,11月 3:1,3,5,7,8,10月 4:12月 | 1:(-∞~1) 2:(12~+∞) |
| D | 1:[1~27] 2:28 3:29 3:30 4:31 | 1:(-∞~1) 2:(31~+∞) 3:当月最大天数不超过30时输入31  4:当月最大天数为29时输入30  5:当月最大天数为28时输入29 |

从划分的等价类来说，我们可以看到，本问题的无效等价类是“多缺陷”依赖的，因此采用强健壮性等价类分析，构造出以下用例：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | Y | M | D | 预计输出 |
|  | 2000 | 2 | 27 | 2000-02-28 |
|  | 2000 | 2 | 28 | 2000-02-29 |
|  | 2000 | 2 | 29 | 2000-03-01 |
|  | 2000 | 2 | 30 | 非法输入 |
|  | 2000 | 2 | 31 | 非法输入 |
|  | 2000 | 2 | 0 | 非法输入 |
|  | 2000 | 2 | 32 | 非法输入 |
|  | 2000 | 6 | 27 | 2000-06-28 |
|  | 2000 | 6 | 28 | 2000-06-29 |
|  | 2000 | 6 | 29 | 2000-06-30 |
|  | 2000 | 6 | 30 | 2000-07-01 |
|  | 2000 | 6 | 31 | 非法输入 |
|  | 2000 | 6 | 0 | 非法输入 |
|  | 2000 | 6 | 32 | 非法输入 |
|  | 2000 | 3 | 27 | 2000-03-28 |
|  | 2000 | 3 | 28 | 2000-03-29 |
|  | 2000 | 3 | 29 | 2000-03-30 |
|  | 2000 | 3 | 30 | 2000-03-31 |
|  | 2000 | 3 | 31 | 2000-04-01 |
|  | 2000 | 3 | 0 | 非法输入 |
|  | 2000 | 3 | 32 | 非法输入 |
|  | 2000 | 12 | 27 | 2000-12-28 |
|  | 2000 | 12 | 28 | 2000-12-29 |
|  | 2000 | 12 | 29 | 2000-12-30 |
|  | 2000 | 12 | 30 | 2000-12-31 |
|  | 2000 | 12 | 31 | 2001-01-01 |
|  | 2000 | 12 | 0 | 非法输入 |
|  | 2000 | 12 | 32 | 非法输入 |
|  | 2000 | 0 | 27 | 非法输入 |
|  | 2000 | 0 | 28 | 非法输入 |
|  | 2000 | 0 | 29 | 非法输入 |
|  | 2000 | 0 | 30 | 非法输入 |
|  | 2000 | 0 | 31 | 非法输入 |
|  | 2000 | 0 | 0 | 非法输入 |
|  | 2000 | 0 | 32 | 非法输入 |
|  | 2000 | 13 | 27 | 非法输入 |
|  | 2000 | 13 | 28 | 非法输入 |
|  | 2000 | 13 | 29 | 非法输入 |
|  | 2000 | 13 | 30 | 非法输入 |
|  | 2000 | 13 | 31 | 非法输入 |
|  | 2000 | 13 | 0 | 非法输入 |
|  | 2000 | 13 | 32 | 非法输入 |
|  | 2008 | 2 | 27 | 2008-02-28 |
|  | 2008 | 2 | 28 | 2008-02-29 |
|  | 2008 | 2 | 29 | 2008-03-01 |
|  | 2008 | 2 | 30 | 非法输入 |
|  | 2008 | 2 | 31 | 非法输入 |
|  | 2008 | 2 | 0 | 非法输入 |
|  | 2008 | 2 | 32 | 非法输入 |
|  | 2008 | 6 | 27 | 2008-06-28 |
|  | 2008 | 6 | 28 | 2008-06-29 |
|  | 2008 | 6 | 29 | 2008-06-30 |
|  | 2008 | 6 | 30 | 2008-07-01 |
|  | 2008 | 6 | 31 | 非法输入 |
|  | 2008 | 6 | 0 | 非法输入 |
|  | 2008 | 6 | 32 | 非法输入 |
|  | 2008 | 3 | 27 | 2008-03-28 |
|  | 2008 | 3 | 28 | 2008-03-29 |
|  | 2008 | 3 | 29 | 2008-03-30 |
|  | 2008 | 3 | 30 | 2008-03-31 |
|  | 2008 | 3 | 31 | 2008-04-01 |
|  | 2008 | 3 | 0 | 非法输入 |
|  | 2008 | 3 | 32 | 非法输入 |
|  | 2008 | 12 | 27 | 2008-12-28 |
|  | 2008 | 12 | 28 | 2008-12-29 |
|  | 2008 | 12 | 29 | 2008-12-30 |
|  | 2008 | 12 | 30 | 2008-12-31 |
|  | 2008 | 12 | 31 | 2009-01-01 |
|  | 2008 | 12 | 0 | 非法输入 |
|  | 2008 | 12 | 32 | 非法输入 |
|  | 2008 | 0 | 27 | 非法输入 |
|  | 2008 | 0 | 28 | 非法输入 |
|  | 2008 | 0 | 29 | 非法输入 |
|  | 2008 | 0 | 30 | 非法输入 |
|  | 2008 | 0 | 31 | 非法输入 |
|  | 2008 | 0 | 0 | 非法输入 |
|  | 2008 | 0 | 32 | 非法输入 |
|  | 2008 | 13 | 27 | 非法输入 |
|  | 2008 | 13 | 28 | 非法输入 |
|  | 2008 | 13 | 29 | 非法输入 |
|  | 2008 | 13 | 30 | 非法输入 |
|  | 2008 | 13 | 31 | 非法输入 |
|  | 2008 | 13 | 0 | 非法输入 |
|  | 2008 | 13 | 32 | 非法输入 |
|  | 1900 | 2 | 27 | 1900-02-28 |
|  | 1900 | 2 | 28 | 1900-03-01 |
|  | 1900 | 2 | 29 | 非法输入 |
|  | 1900 | 2 | 30 | 非法输入 |
|  | 1900 | 2 | 31 | 非法输入 |
|  | 1900 | 2 | 0 | 非法输入 |
|  | 1900 | 2 | 32 | 非法输入 |
|  | 1900 | 6 | 27 | 1900-06-28 |
|  | 1900 | 6 | 28 | 1900-06-29 |
|  | 1900 | 6 | 29 | 1900-06-30 |
|  | 1900 | 6 | 30 | 1900-07-01 |
|  | 1900 | 6 | 31 | 非法输入 |
|  | 1900 | 6 | 0 | 非法输入 |
|  | 1900 | 6 | 32 | 非法输入 |
|  | 1900 | 3 | 27 | 1900-03-28 |
|  | 1900 | 3 | 28 | 1900-03-29 |
|  | 1900 | 3 | 29 | 1900-03-30 |
|  | 1900 | 3 | 30 | 1900-03-31 |
|  | 1900 | 3 | 31 | 1900-04-01 |
|  | 1900 | 3 | 0 | 非法输入 |
|  | 1900 | 3 | 32 | 非法输入 |
|  | 1900 | 12 | 27 | 1900-12-28 |
|  | 1900 | 12 | 28 | 1900-12-29 |
|  | 1900 | 12 | 29 | 1900-12-30 |
|  | 1900 | 12 | 30 | 1900-12-31 |
|  | 1900 | 12 | 31 | 1901-01-01 |
|  | 1900 | 12 | 0 | 非法输入 |
|  | 1900 | 12 | 32 | 非法输入 |
|  | 1900 | 0 | 27 | 非法输入 |
|  | 1900 | 0 | 28 | 非法输入 |
|  | 1900 | 0 | 29 | 非法输入 |
|  | 1900 | 0 | 30 | 非法输入 |
|  | 1900 | 0 | 31 | 非法输入 |
|  | 1900 | 0 | 0 | 非法输入 |
|  | 1900 | 0 | 32 | 非法输入 |
|  | 1900 | 13 | 27 | 非法输入 |
|  | 1900 | 13 | 28 | 非法输入 |
|  | 1900 | 13 | 29 | 非法输入 |
|  | 1900 | 13 | 30 | 非法输入 |
|  | 1900 | 13 | 31 | 非法输入 |
|  | 1900 | 13 | 0 | 非法输入 |
|  | 1900 | 13 | 32 | 非法输入 |
|  | 1899 | 2 | 27 | 非法输入 |
|  | 1899 | 2 | 28 | 非法输入 |
|  | 1899 | 2 | 29 | 非法输入 |
|  | 1899 | 2 | 30 | 非法输入 |
|  | 1899 | 2 | 31 | 非法输入 |
|  | 1899 | 2 | 0 | 非法输入 |
|  | 1899 | 2 | 32 | 非法输入 |
|  | 1899 | 6 | 27 | 非法输入 |
|  | 1899 | 6 | 28 | 非法输入 |
|  | 1899 | 6 | 29 | 非法输入 |
|  | 1899 | 6 | 30 | 非法输入 |
|  | 1899 | 6 | 31 | 非法输入 |
|  | 1899 | 6 | 0 | 非法输入 |
|  | 1899 | 6 | 32 | 非法输入 |
|  | 1899 | 3 | 27 | 非法输入 |
|  | 1899 | 3 | 28 | 非法输入 |
|  | 1899 | 3 | 29 | 非法输入 |
|  | 1899 | 3 | 30 | 非法输入 |
|  | 1899 | 3 | 31 | 非法输入 |
|  | 1899 | 3 | 0 | 非法输入 |
|  | 1899 | 3 | 32 | 非法输入 |
|  | 1899 | 12 | 27 | 非法输入 |
|  | 1899 | 12 | 28 | 非法输入 |
|  | 1899 | 12 | 29 | 非法输入 |
|  | 1899 | 12 | 30 | 非法输入 |
|  | 1899 | 12 | 31 | 非法输入 |
|  | 1899 | 12 | 0 | 非法输入 |
|  | 1899 | 12 | 32 | 非法输入 |
|  | 1899 | 0 | 27 | 非法输入 |
|  | 1899 | 0 | 28 | 非法输入 |
|  | 1899 | 0 | 29 | 非法输入 |
|  | 1899 | 0 | 30 | 非法输入 |
|  | 1899 | 0 | 31 | 非法输入 |
|  | 1899 | 0 | 0 | 非法输入 |
|  | 1899 | 0 | 32 | 非法输入 |
|  | 1899 | 13 | 27 | 非法输入 |
|  | 1899 | 13 | 28 | 非法输入 |
|  | 1899 | 13 | 29 | 非法输入 |
|  | 1899 | 13 | 30 | 非法输入 |
|  | 1899 | 13 | 31 | 非法输入 |
|  | 1899 | 13 | 0 | 非法输入 |
|  | 1899 | 13 | 32 | 非法输入 |
|  | 2501 | 2 | 27 | 非法输入 |
|  | 2501 | 2 | 28 | 非法输入 |
|  | 2501 | 2 | 29 | 非法输入 |
|  | 2501 | 2 | 30 | 非法输入 |
|  | 2501 | 2 | 31 | 非法输入 |
|  | 2501 | 2 | 0 | 非法输入 |
|  | 2501 | 2 | 32 | 非法输入 |
|  | 2501 | 6 | 27 | 非法输入 |
|  | 2501 | 6 | 28 | 非法输入 |
|  | 2501 | 6 | 29 | 非法输入 |
|  | 2501 | 6 | 30 | 非法输入 |
|  | 2501 | 6 | 31 | 非法输入 |
|  | 2501 | 6 | 0 | 非法输入 |
|  | 2501 | 6 | 32 | 非法输入 |
|  | 2501 | 3 | 27 | 非法输入 |
|  | 2501 | 3 | 28 | 非法输入 |
|  | 2501 | 3 | 29 | 非法输入 |
|  | 2501 | 3 | 30 | 非法输入 |
|  | 2501 | 3 | 31 | 非法输入 |
|  | 2501 | 3 | 0 | 非法输入 |
|  | 2501 | 3 | 32 | 非法输入 |
|  | 2501 | 12 | 27 | 非法输入 |
|  | 2501 | 12 | 28 | 非法输入 |
|  | 2501 | 12 | 29 | 非法输入 |
|  | 2501 | 12 | 30 | 非法输入 |
|  | 2501 | 12 | 31 | 非法输入 |
|  | 2501 | 12 | 0 | 非法输入 |
|  | 2501 | 12 | 32 | 非法输入 |
|  | 2501 | 0 | 27 | 非法输入 |
|  | 2501 | 0 | 28 | 非法输入 |
|  | 2501 | 0 | 29 | 非法输入 |
|  | 2501 | 0 | 30 | 非法输入 |
|  | 2501 | 0 | 31 | 非法输入 |
|  | 2501 | 0 | 0 | 非法输入 |
|  | 2501 | 0 | 32 | 非法输入 |
|  | 2501 | 13 | 27 | 非法输入 |
|  | 2501 | 13 | 28 | 非法输入 |
|  | 2501 | 13 | 29 | 非法输入 |
|  | 2501 | 13 | 30 | 非法输入 |
|  | 2501 | 13 | 31 | 非法输入 |
|  | 2501 | 13 | 0 | 非法输入 |
|  | 2501 | 13 | 32 | 非法输入 |

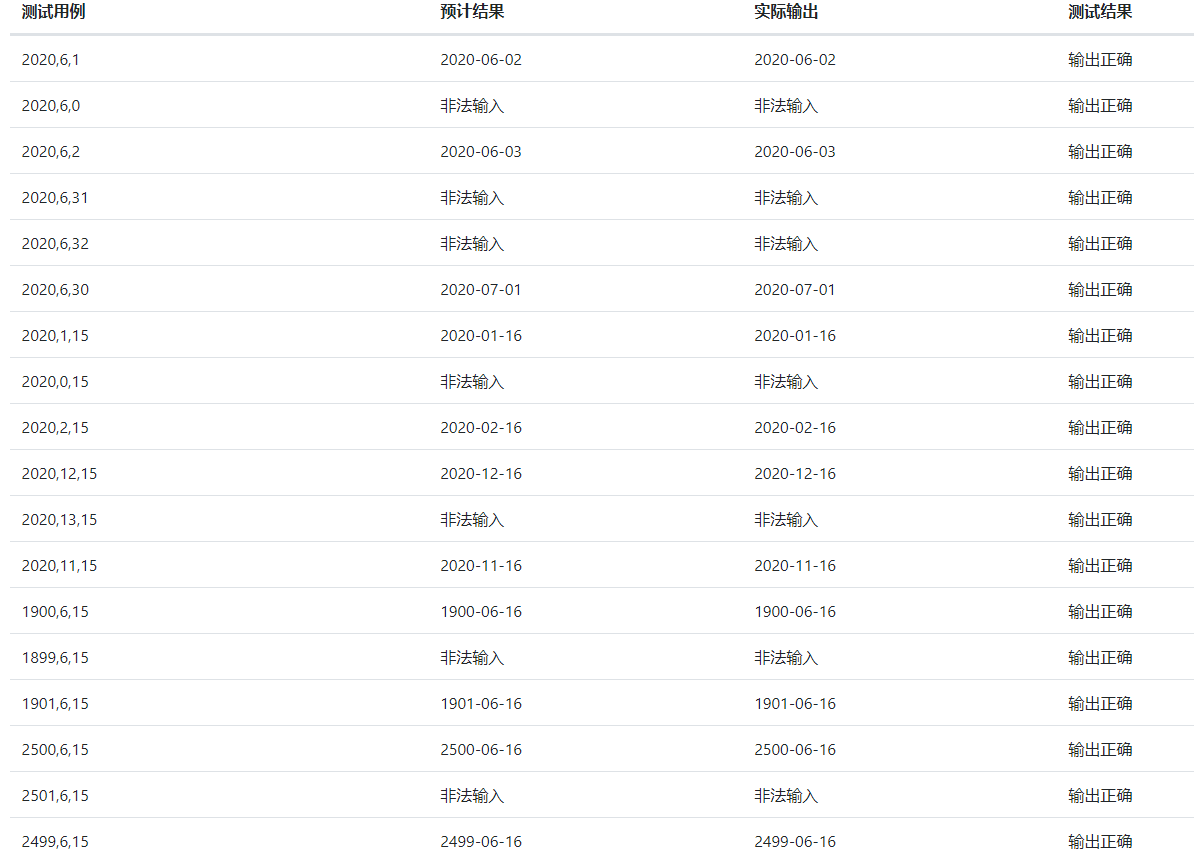
## 测试代码设计

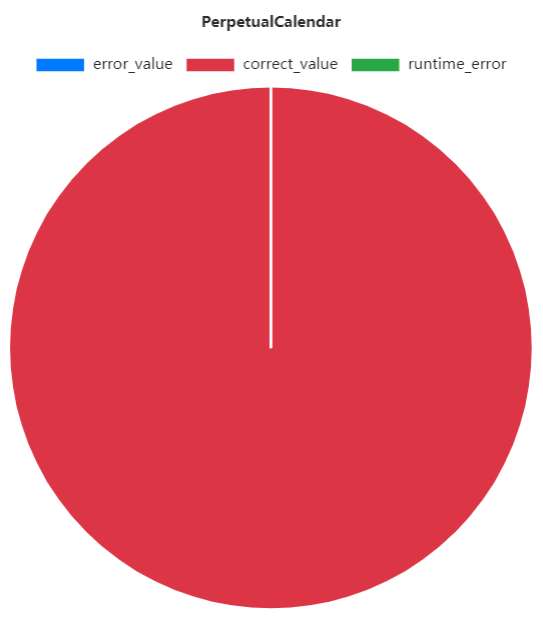
1. **import** java.text.SimpleDateFormat;
2. **import** java.util.Calendar;
4. **public** **class** PerpetualCalendar{
5. **public** **static** String getNextDay(**int** y,**int** m,**int** d){
6. **if**(y<1900||y>2500||m<1||m>12){
7. **return** "非法输入";
8. }
9. SimpleDateFormat dateFormat=**new** SimpleDateFormat("yyyy-MM-dd");
10. Calendar date=Calendar.getInstance();
11. date.set(Calendar.YEAR,y);
12. date.set(Calendar.MONTH,m-1);
13. var i=date.getActualMaximum(Calendar.DATE);
14. **if**(d<1||d>i){
15. **return** "非法输入";
16. }
17. date.set(Calendar.DATE,d);
18. date.add(Calendar.DATE,1);
19. **return** dateFormat.format(date.getTime());
20. }
21. }

## 测试用例执行情况

使用自制代码测试工具对编写代码加以测试，得到如下结果：

### 边界值法





代码测试全部通过

### 等价类法

由于测试用例过多，在此仅展示统计数据



代码测试全部通过