

实验六 软件测试

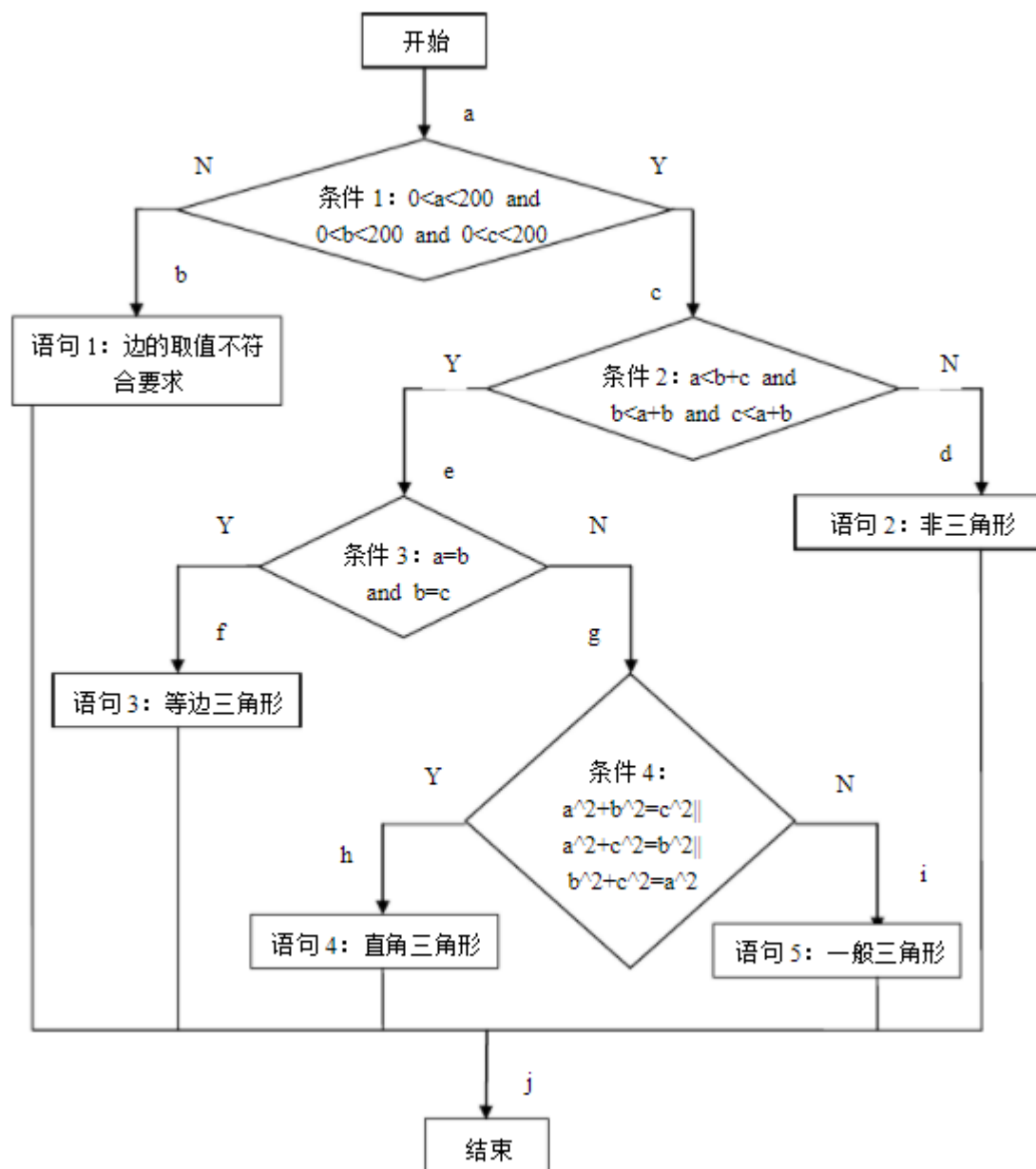
一、实验目的

1. 掌握软件测试的基本方法和技术。
2. 掌握测试用例设计。

二、实验内容

1、三角形判定程序

1. 【说明】输入三个整数 a 、 b 、 c ，分别作为三角形的三条边，通过程序判断这三条边是否构成一般三角形、等腰三角形、等边三角形。
 - 1) 用逻辑覆盖设计测试用例。
 - (1) 画出程序的流程图。



2) 按表 1 形式，用语句覆盖、判定覆盖、条件覆盖、条件组合覆盖设计测试用例。

(1) 语句覆盖



(2) 判定覆盖

(1) 语句覆盖

| 用例编号 | 输入数据 a | 输入数据 b | 输出数据 c | 输出 | 备注 |
|------|--------|--------|--------|-----------|------|
| 1 | -2 | 3 | 4 | 边的取值不符合要求 | 语句 1 |
| 2 | 3 | 4 | 8 | 非三角形 | 语句 2 |
| 3 | 3 | 3 | 3 | 等边三角形 | 语句 3 |
| 4 | 3 | 4 | 5 | 直角三角形 | 语句 4 |
| 5 | 2 | 3 | 4 | 一般三角形 | 语句 5 |

(2) 判定覆盖

| 用例编号 | 输入数据 a | 输入数据 b | 输出数据 c | 输出 | 备注 |
|------|--------|--------|--------|-----------|-------------------|
| 1 | 201 | 2 | 3 | 边的取值不符合要求 | 条件 1 “假” |
| 2 | 2 | 3 | 6 | 非三角形 | 条件 1 “真” 条件 2 “假” |
| 3 | 4 | 4 | 4 | 等边三角形 | 条件 2 “真” 条件 3 “真” |
| 4 | 3 | 4 | 5 | 直角三角形 | 条件 3 “假” 条件 4 “真” |
| 5 | 2 | 4 | 5 | 一般三角形 | 条件 4 “假” |

```
Microsoft Visual Studio 调试控制台
请依次输入三角形的三条边，各边以空格分开：
201 2 3
边的取值不符合要求，请检查后重新输入！
继续运行请输入'Y'，否则输入'N'
Y
请依次输入三角形的三条边，各边以空格分开：
2 3 6
该三条边不构成三角形！
继续运行请输入'Y'，否则输入'N'
Y
请依次输入三角形的三条边，各边以空格分开：
4 4 4
该三条边构成等边三角形！
继续运行请输入'Y'，否则输入'N'
Y
请依次输入三角形的三条边，各边以空格分开：
3 4 5
该三条边构成直角三角形！
继续运行请输入'Y'，否则输入'N'
Y
请依次输入三角形的三条边，各边以空格分开：
2 4 5
该三条边构成普通三角形！
继续运行请输入'Y'，否则输入'N'
N
程序结束！

C:\Users\清风\source\repos\judge_triangle\Debug\judge_triangle.exe (进程 10668)已退出，返回代码为: 0。
若要在调试停止时自动关闭控制台，请启用“工具”->“选项”->“调试”->“调试停止时自动关闭控制台”。
按任意键关闭此窗口...
```

(3) 条件覆盖

对于第一个判断：条件 $0 \leq a \leq 200$ 取真为 T1, 取假为 F1；
条件 $0 \leq b \leq 200$ 取真为 T2, 取假为 F2； 条件取真为 T3, 取假为 F3；
对于第二个判断：条件 $a \leq b \leq c$ 取真为 T4, 取假为 F4；
条件 $b \leq a \leq c$ 取真为 T5, 取假为 F5； 条件 $c \leq a \leq b$ 取真为 T6, 取假为 F6；
对于第三个判断：条件 $a=b$ 取真为 T7, 取假为 F7；
条件 $b=c$ 取真为 T8, 取假为 F8；
对于第四个判断：条件 $a^2 + b^2 = c^2$ 取真为 T9, 取假为 F9； 条件 $a^2 + c^2 = b^2$
取真为 T10, 取假为 F10；
条件 $b^2 + c^2 = a^2$ 取真为 T11, 取假为 F11；

(3) 条件覆盖

对于第一个判断：条件 $0 < a < 200$ 取真为 T1,取假为 F1；

条件 $0 < b < 200$ 取真为 T2,取假为 F2；

条件 $0 < c < 200$ 取真为 T3,取假为 F3；

对于第二个判断：条件 $a < b + c$ 取真为 T4,取假为 F4；

条件 $b < a + c$ 取真为 T5,取假为 F5；

条件 $c < a + b$ 取真为 T6,取假为 F6；

对于第三个判断：条件 $a = b$ 取真为 T7,取假为 F7；

条件 $b = c$ 取真为 T8,取假为 F8；

对于第四个判断：条件 $a^2 + b^2 = c^2$ 取真为 T9,取假为 F9；

条件 $a^2 + c^2 = b^2$ 取真为 T10,取假为 F10；

条件 $b^2 + c^2 = a^2$ 取真为 T11,取假为 F11；

| 用例编号 | 输入数据 a | 输入数据 b | 输出数据 c | 输出 | 备注 |
|------|--------|--------|--------|---------------|------------------|
| 1 | -2 | 2 | 4 | 边的取值不 符合要求 | F1,T2,T3 |
| 2 | 3 | -4 | 220 | 边的取值不 符合要求 | T1,F2,F3 |
| 3 | 3 | 5 | 9 | 非三角形 | T4,T5,F6 |
| 4 | 8 | 1 | 3 | 非三角形 | F4,T5,T6 |
| 5 | 2 | 8 | 4 | 非三角形 | T4,F5,T6 |
| 6 | 3 | 3 | 3 | 等边三角形 | T7,T8 |
| 7 | 2 | 3 | 4 | 一般三角形 | F7,F8,F9,F10,F11 |
| 8 | 3 | 4 | 5 | 直角三角形 | T9 |
| 9 | 4 | 5 | 3 | 直角三角形 | T11 |
| 10 | 5 | 3 | 4 | 直角三角形 | T10 |

```
Microsoft Visual Studio 调试控制台
请依次输入三角形的三条边，各边以空格分开：
-2 2 4
边的取值不符合要求，请检查后重新输入！
继续运行请输入'Y'，否则输入'N'
Y
请依次输入三角形的三条边，各边以空格分开：
3 -4 220
边的取值不符合要求，请检查后重新输入！
继续运行请输入'Y'，否则输入'N'
Y
请依次输入三角形的三条边，各边以空格分开：
3 5 9
该三条边不构成三角形！
继续运行请输入'Y'，否则输入'N'
Y
请依次输入三角形的三条边，各边以空格分开：
8 1 3
该三条边不构成三角形！
继续运行请输入'Y'，否则输入'N'
Y
请依次输入三角形的三条边，各边以空格分开：
2 8 4
该三条边不构成三角形！
继续运行请输入'Y'，否则输入'N'
Y
请依次输入三角形的三条边，各边以空格分开：
3 3 3
该三条边构成等边三角形！
继续运行请输入'Y'，否则输入'N'
Y
请依次输入三角形的三条边，各边以空格分开：
2 3 4
该三条边构成普通三角形！
继续运行请输入'Y'，否则输入'N'
Y
请依次输入三角形的三条边，各边以空格分开：
3 4 5
该三条边构成直角三角形！
继续运行请输入'Y'，否则输入'N'
Y
请依次输入三角形的三条边，各边以空格分开：
4 5 3
该三条边构成直角三角形！
继续运行请输入'Y'，否则输入'N'
Y
请依次输入三角形的三条边，各边以空格分开：
5 3 4
该三条边构成直角三角形！
继续运行请输入'Y'，否则输入'N'
N
程序结束！

C:\Users\清风\source\repos\judge_triangle\Debug\judge_triangle.exe (进程 2448) 已退出，返回代码为: 0。
若要在调试停止时自动关闭控制台，请启用“工具”->“选项”->“调试”->“调试停止时自动关闭控制台”。
```

条件覆盖测试

(4) 判定/条件覆盖

(4)判定/条件覆盖

| 用例编号 | 输入数据 a | 输入数据 b | 输出数据 c | 输出 | 备注 |
|------|--------|--------|--------|---------------|-------------------------------------|
| 1 | -1 | 201 | 206 | 边的取值不 符合要求 | F1,F2,F3 路径: a-b-j |
| 2 | 3 | 1 | 5 | 非三角形 | T1,T2,T3,T4,T5,F6 路径: a-c-d-j |
| 3 | 3 | 5 | 1 | 非三角形 | T4,F5,T6 路径: a-c-d-j |
| 4 | 8 | 1 | 3 | 非三角形 | F4,T5,T6 路径: a-c-d-j |
| 5 | 4 | 4 | 4 | 等边三角形 | T7,T8 路径: a-c-e-f-j |
| 6 | 2 | 3 | 4 | 一般三角形 | F7,F8 F9,F10,F11 路径: a-c-e-g-i-j |
| 7 | 3 | 4 | 5 | 直角三角形 | T9 路径: a-c-e-g-h-j |
| 8 | 4 | 5 | 3 | 直角三角形 | T11 路径: a-c-e-g-h-j |
| 9 | 5 | 3 | 4 | 直角三角形 | T10 路径: a-c-e-g-h-j |

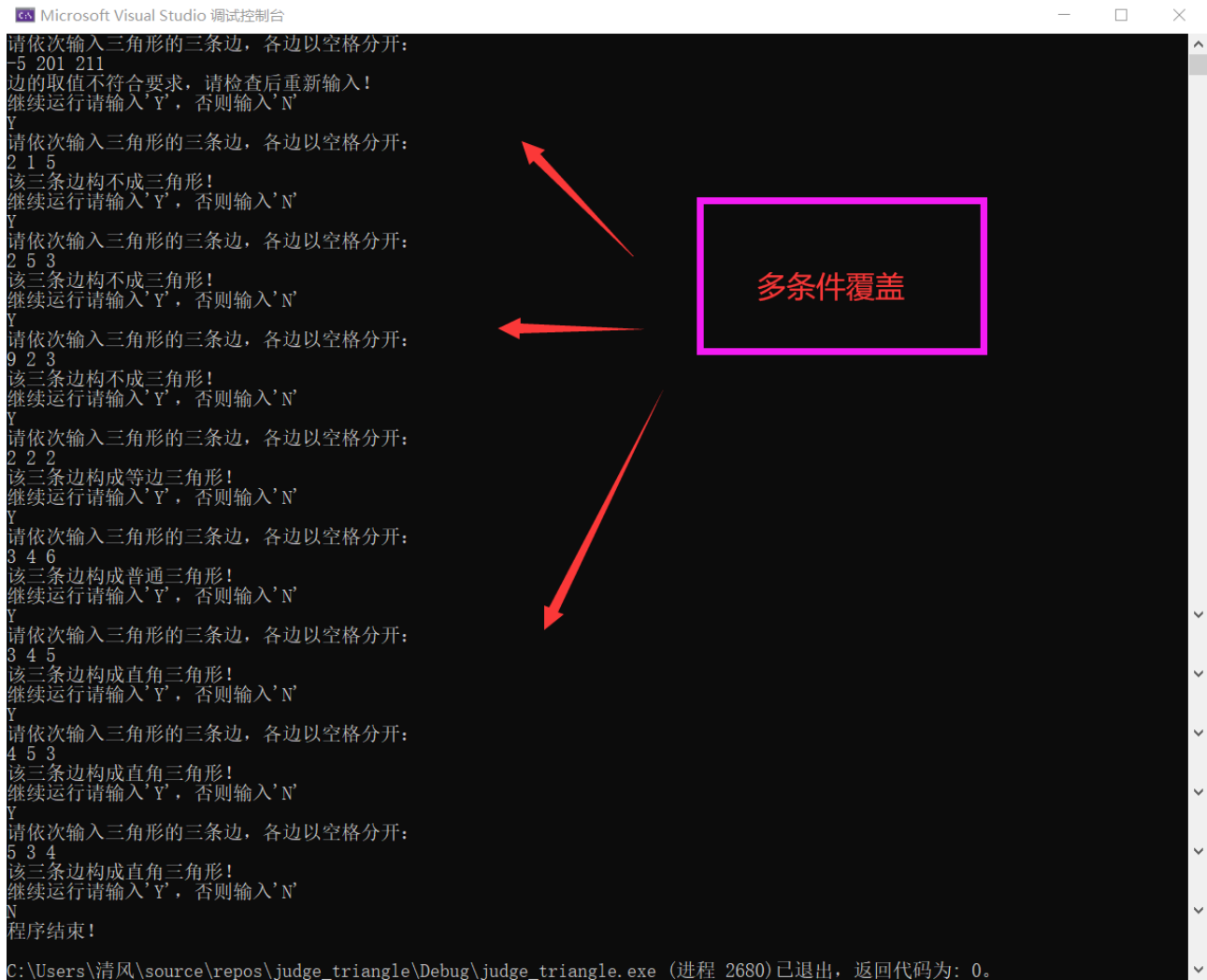
Microsoft Visual Studio 调试控制台

```
请依次输入三角形的三条边，各边以空格分开：
-1 201 206
边的取值不符合要求，请检查后重新输入！
继续运行请输入'Y'，否则输入'N'
Y
请依次输入三角形的三条边，各边以空格分开：
3 1 5
该三条边不构成三角形！
继续运行请输入'Y'，否则输入'N'
Y
请依次输入三角形的三条边，各边以空格分开：
3 5 1
该三条边不构成三角形！
继续运行请输入'Y'，否则输入'N'
Y
请依次输入三角形的三条边，各边以空格分开：
8 1 3
该三条边不构成三角形！
继续运行请输入'Y'，否则输入'N'
Y
请依次输入三角形的三条边，各边以空格分开：
4 4 4
该三条边构成等边三角形！
继续运行请输入'Y'，否则输入'N'
Y
请依次输入三角形的三条边，各边以空格分开：
2 3 4
该三条边构成普通三角形！
继续运行请输入'Y'，否则输入'N'
Y
请依次输入三角形的三条边，各边以空格分开：
3 4 5
该三条边构成直角三角形！
继续运行请输入'Y'，否则输入'N'
Y
请依次输入三角形的三条边，各边以空格分开：
4 5 3
该三条边构成直角三角形！
继续运行请输入'Y'，否则输入'N'
Y
请依次输入三角形的三条边，各边以空格分开：
5 3 4
该三条边构成直角三角形！
继续运行请输入'Y'，否则输入'N'
N
程序结束！
```

C:\Users\清风\source\repos\judge_triangle\Debug\judge_triangle.exe (进程 10680) 已退出，返回代码为: 0。

判定/条件覆盖

(5) 多条件覆盖



```
Microsoft Visual Studio 调试控制台
请依次输入三角形的三条边，各边以空格分开：
-5 201 211
边的取值不符合要求，请检查后重新输入！
继续运行请输入'Y'，否则输入'N'
Y
请依次输入三角形的三条边，各边以空格分开：
2 1 5
该三条边不构成三角形！
继续运行请输入'Y'，否则输入'N'
Y
请依次输入三角形的三条边，各边以空格分开：
2 5 3
该三条边不构成三角形！
继续运行请输入'Y'，否则输入'N'
Y
请依次输入三角形的三条边，各边以空格分开：
9 2 3
该三条边不构成三角形！
继续运行请输入'Y'，否则输入'N'
Y
请依次输入三角形的三条边，各边以空格分开：
2 2 2
该三条边构成等边三角形！
继续运行请输入'Y'，否则输入'N'
Y
请依次输入三角形的三条边，各边以空格分开：
3 4 6
该三条边构成普通三角形！
继续运行请输入'Y'，否则输入'N'
Y
请依次输入三角形的三条边，各边以空格分开：
3 4 5
该三条边构成直角三角形！
继续运行请输入'Y'，否则输入'N'
Y
请依次输入三角形的三条边，各边以空格分开：
4 5 3
该三条边构成直角三角形！
继续运行请输入'Y'，否则输入'N'
Y
请依次输入三角形的三条边，各边以空格分开：
5 3 4
该三条边构成直角三角形！
继续运行请输入'Y'，否则输入'N'
N
程序结束！
C:\Users\清风\source\repos\judge_triangle\Debug\judge_triangle.exe (进程 2680) 已退出，返回代码为: 0。
```

(6) 路径覆盖

(5) 多条件覆盖

| 用例编号 | 输入数据 a | 输入数据 b | 输出数据 c | 输出 | 备注 |
|------|--------|--------|--------|---------------|-------------------|
| 1 | -5 | 201 | 211 | 边的取值不 符合要求 | F1,F2,F3 |
| 2 | 2 | 1 | 5 | 非三角形 | T1,T2,T3,T4,T5,F6 |
| 3 | 2 | 5 | 3 | 非三角形 | T4,F5,T6 |
| 4 | 9 | 2 | 3 | 非三角形 | F4,T5,T6 |
| 5 | 2 | 2 | 2 | 等边三角形 | T7,T8 |
| 6 | 3 | 4 | 6 | 一般三角形 | F7,F8 F9,F10,F11 |
| 7 | 3 | 4 | 5 | 直角三角形 | T9 |
| 8 | 4 | 5 | 3 | 直角三角形 | T1 |
| 9 | 5 | 3 | 4 | 直角三角形 | T10 |

(6) 路径覆盖

| 用例编号 | 输入数据 a | 输入数据 b | 输出数据 c | 输出 | 备注 |
|------|--------|--------|--------|---------------|-------------|
| 1 | -3 | 201 | 4 | 边的取值不 符合要求 | a-b-j |
| 2 | 2 | 5 | 9 | 非三角形 | a-c-d-j |
| 3 | 3 | 3 | 3 | 等边三角形 | a-c-e-f-j |
| 4 | 4 | 5 | 3 | 直角三角形 | a-c-e-g-h-j |
| 5 | 6 | 3 | 4 | 一般三角形 | a-c-e-g-i-j |

```
Microsoft Visual Studio 调试控制台
请依次输入三角形的三条边，各边以空格分开：
-3 201 4
边的取值不符合要求，请检查后重新输入！
继续运行请输入'Y'，否则输入'N'
Y
请依次输入三角形的三条边，各边以空格分开：
2 5 9
该三条边不构成三角形！
继续运行请输入'Y'，否则输入'N'
Y
请依次输入三角形的三条边，各边以空格分开：
3 3 3
该三条边构成等边三角形！
继续运行请输入'Y'，否则输入'N'
Y
请依次输入三角形的三条边，各边以空格分开：
4 5 3
该三条边构成直角三角形！
继续运行请输入'Y'，否则输入'N'
Y
请依次输入三角形的三条边，各边以空格分开：
6 3 4
该三条边构成普通三角形！
继续运行请输入'Y'，否则输入'N'
N
程序结束！

C:\Users\清风\source\repos\judge_triangle\Debug\judge_triangle.exe (进程 18552) 已退出，返回代码为：0。
若要在调试停止时自动关闭控制台，请启用“工具”->“选项”->“调试”->“调试停止时自动关闭控制台”。
```

路径覆盖

测试用例

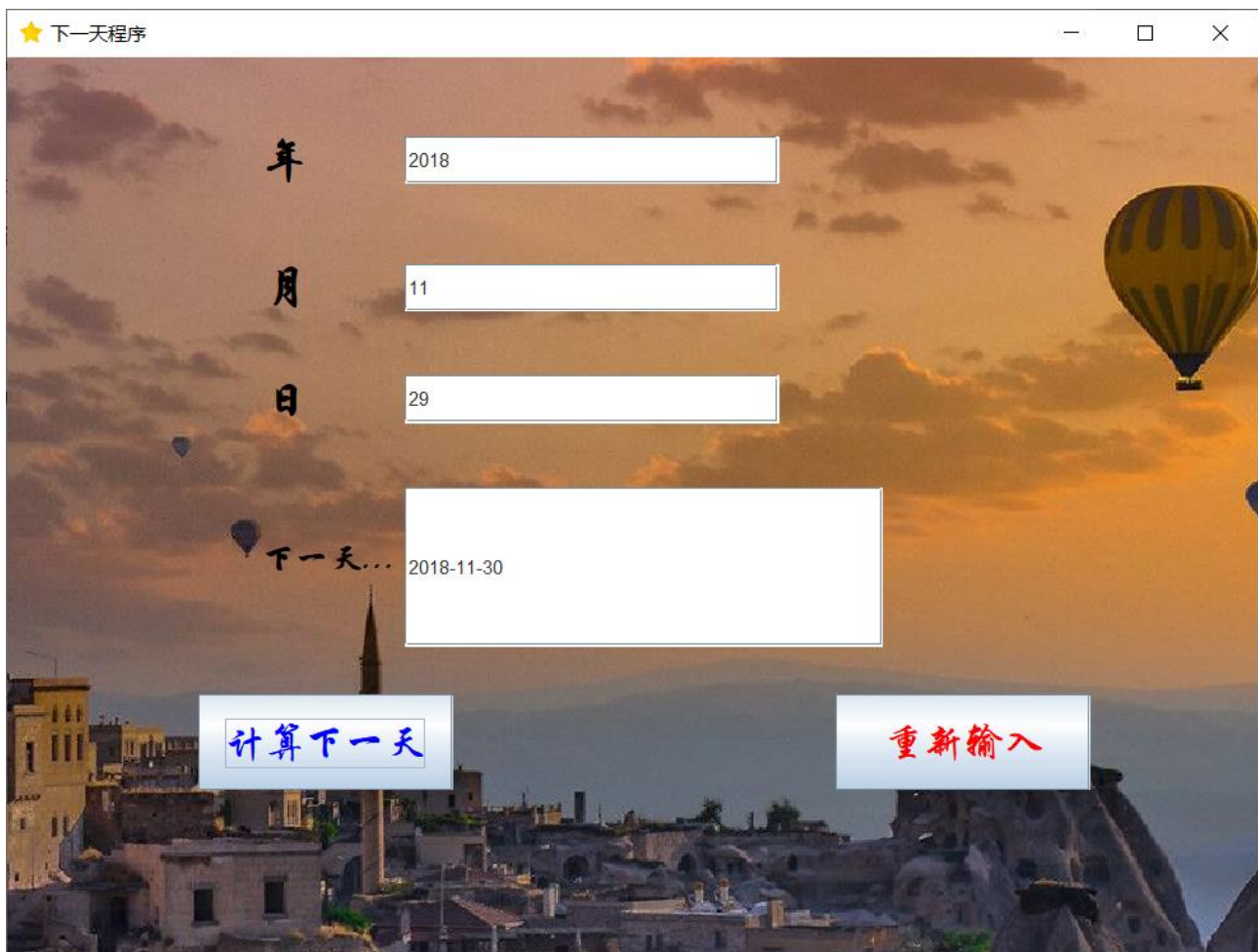
2、下一天程序

问题描述：

函数有三个输入变量 month、day、year（month、day 和 year 均为整数值，并且满足： $1 \leq \text{month} \leq 12$ 、 $1 \leq \text{day} \leq 31$ 、 $1900 \leq \text{year} \leq 2050$ 。），分别作为输入日期的月份、日、年份，通过程序可以输出下一天的日期。例如，输入为 2018 年 11 月 29 日，则该程序的输出为 2018 年 11 月 30 日。

请用边界值测试方法设计测试用例。并对你所设计的测试用例的不足之处加以说明。

程序界面



一、年：

(1) 使输入刚好等于最小的整数。 输入：1990

预期的输入：1900

(2) 使输入刚好等于最大的整数。 输入：2050

预期的输入：2050

(3) 使输入刚好小于最小的整数。 输入：1989 预期的输入：“错误——无效输入”

(4) 使输入刚好大于最大的整数。 输入：2051 预期的输入：“错误——无效输入”

二、月：

(5) 使输入刚好等于最小的整数。 输入：1

预期的输入：1

(6) 使输入刚好等于最大的整数。 输入：12

预期的输入：12

(7) 使输入刚好小于最小的整数。 输入：0

预期的输入：“错误——无效输入”

(8) 使输入刚好大于最大的整数。 输入：13 预期的输入：“错误——无效输入”

三、日：

(9) 使输入刚好等于最小的整数。 输入：1

预期的输入：1

(10) 使输入刚好等于最大的整数。 输入：31

预期的输入：31

(11) 使输入刚好小于最小的整数。 输入：0

预期的输入：“错误——无效输入”

(12) 使输入刚好大于最大的整数。 输入：32 预期的输入：“错误——无效输入”

四、测试用例：结果截图

使用（年，月，日）形式：

正确输入：

1、(1900, 1, 1)、(1900, 1, 31)、(1900, 12, 1)、(1900, 12, 31)、(2050, 1, 1)、(2050, 1, 31)、(2050, 12, 1)、(2050, 12, 31)。

★ 下一天程序

年 1900

月 1

日 1

下一天... 1900-01-02

计算下一天

重新输入

★ 下一天程序

年

月

日

下一天...

★ 下一天程序

年

月

日

下一天...

★ 下一天程序

年

月

日

下一天...

计算下一天

重新输入

★ 下一天程序

年

月

日

下一天...

计算下一天

重新输入

★ 下一天程序

年 2050

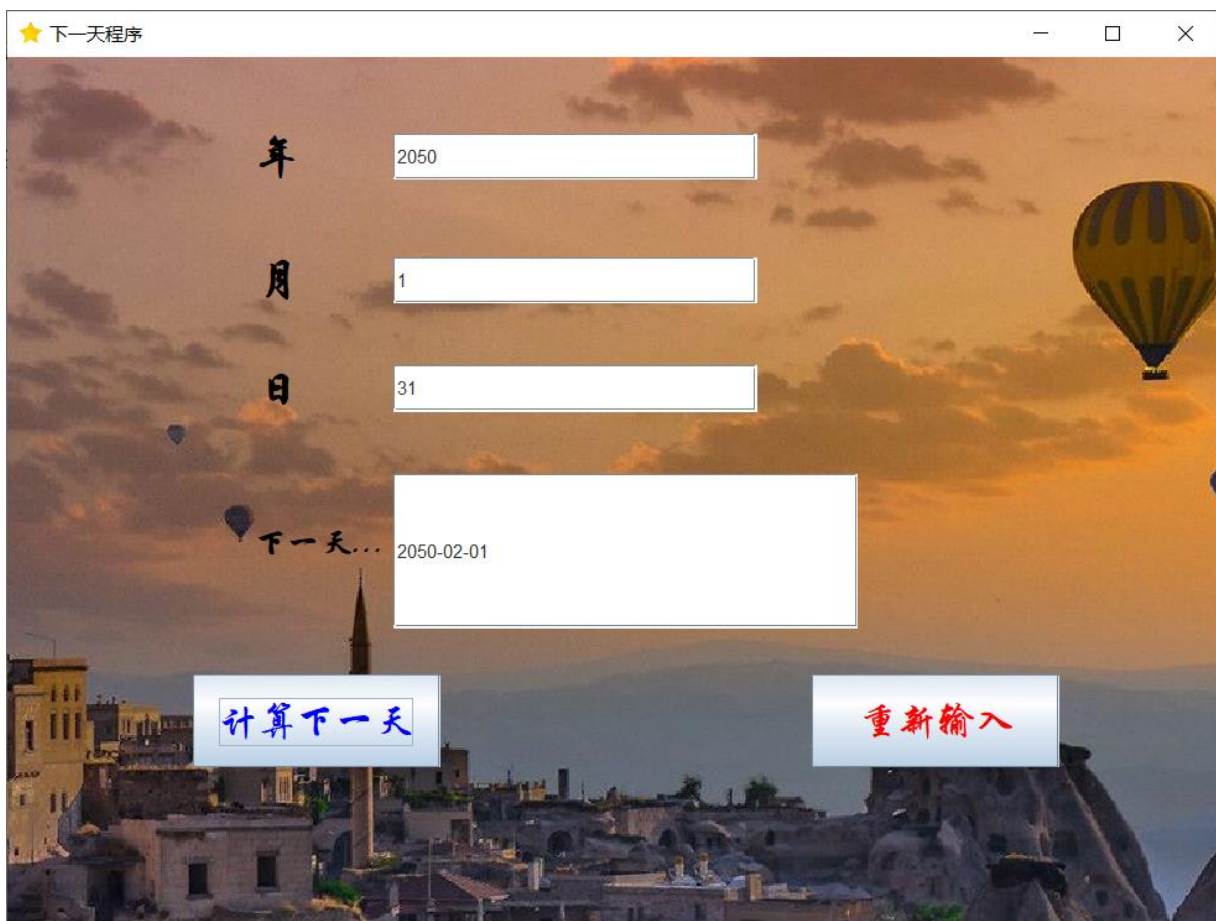
月 1

日 31

下一天... 2050-02-01

计算下一天

重新输入



★ 下一天程序

年 2050

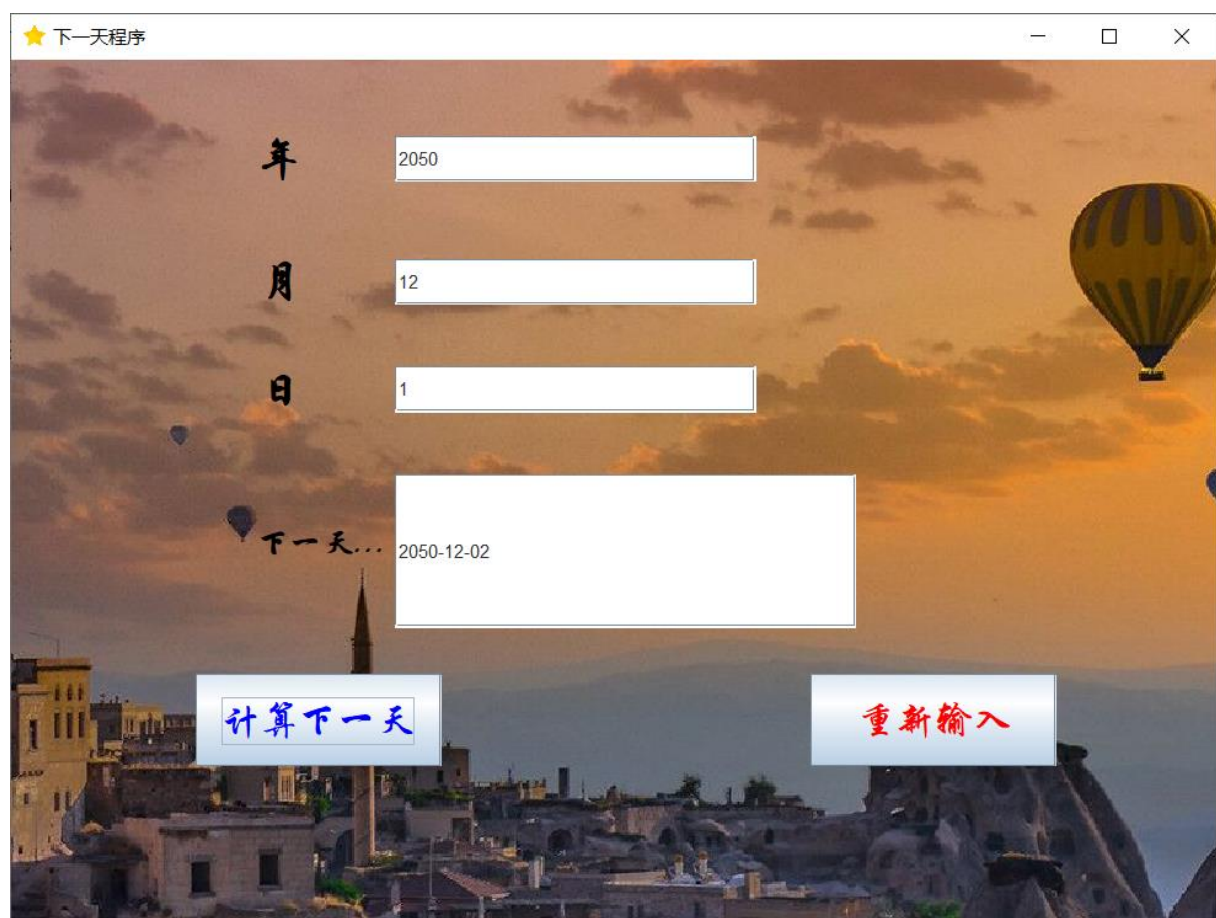
月 12

日 1

下一天... 2050-12-02

计算下一天

重新输入



★ 下一天程序

年

月

日

下一天...

错误输入:

1、(1899, 1, 1)、(1899, 1, 31)、(1989, 12, 1)、(1899, 12, 31)、(2051, 1, 1)、(2051, 1, 31)、(2051, 12, 1)、(2051, 12, 31)。

★ 下一天程序

年

月

日

下一天... 错误——无效输入！请检查年份输入！

计算下一天

重新输入

★ 下一天程序

年

月

日

下一天... 错误——无效输入！请检查年份输入！

计算下一天

重新输入

2、(1900, 1, 0)、(1900, 1, 32)、(1900, 12, 0)、(1900, 12, 32)、(2050, 1, 0)、(2050,

1, 32)、(2050, 12, 0)、(2050, 12, 32)。

★ 下一天程序

年1900

月1

日0

下一天... 错误——无效输入！请检查日期输入！

计算下一天

重新输入

★ 下一天程序

年1900

月12

日32

下一天... 错误——无效输入！请检查日期输入！

计算下一天

重新输入

3、(1900, 0, 1)、(1900, 0, 31)、(1900, 13, 1)、(1900, 13, 31)、(2050, 0, 1)、(2050, 0, 31)、(2050, 13, 1)、(2050, 13, 31)。

★ 下一天程序

年

1900

月

0

日

1

下一天...

错误——无效输入！请检查月份输入！

计算下一天

重新输入

★ 下一天程序

年

1900

月

13

日

1

下一天...

错误——无效输入！请检查月份输入！

计算下一天

重新输入

五、测试不足：

每年有十二月，1 月和 12 月都是 31 天的大月，但是在闰年和平年的区分中，闰年的二月为 28 天，平年的二月为 29 天。同时还有 30 天的小月，如 4 月、6 月、9 月、11 月。如果应用这样的边界值测试方法，无法测试出二月和拥有 30 天数的小月的情况。

这里针对这部分不足，做了程序改进改进，使之能够完成对大月、小月以及闰年的判定和运行。

★ 下一天程序

年 2008

月 2

日 28

下一天... 2008-02-29

计算下一天

重新输入

★ 下一天程序

年 2008

月 2

日 29

下一天... 2008-03-01

计算下一天

重新输入

三、实验心得总结

此次实验实现了两个程序，目的都是以实践的方式体会测试用例如何进行设计和评判。第一个程序测试用例重要侧重在流程测试；第二个程序主要关心逻辑的评判。因此，依据不同程序的特征，设计不同的测试方案和内容。

软件实现过程比较简单。第一个程序是判断三角形的程序，我采用了 C++ 语言进行编写，过程化的设计简洁而又明了。第二个程序是日期的辨别。这里偏重于工程开发（实现了界面），故我使用 Java 进行对象化开发。开发的过程比较顺利，但是需要注意到一些小细节的实现，比如第一个实验数的范围，这里是规定了整数，而若是没有规定，还需要在实数域上进行考虑。又如，第二个实验，要注意考虑公元 0 年的处理。

总的来说，学习计算机还是需要理论和实践相结合才能深入地了解其内涵和本质。在之后的学习中，我会继续学习理论知识，但同时也不忘上机实践。此外，还可以多在网上查阅相关资料，与同专业老师和同学交流，共同进步！

附页：源代码

1 三角形程序：C++源码

```
// judge_triangle.cpp : 此文件包含 "main" 函数。程序执行将在此处开始并结束。

//Author: Yu Shuaiwen

//Department: 2018 级计算机科学与技术

//Class Num: 1

//Date: 2021/5/1

#include <iostream>

#include <stdio.h>

using namespace std;

int main()
```

```
{

char ch='Y';

while (ch=='Y') {

    //输入设置

    int a, b, c;

    cout << "请依次输入三角形的三条边，各边以空格分开：\n";

    cin >> a >> b >> c;

    if (a <= 0 || a >= 200 || b <= 0 || b >= 200 || c <= 0 || c >= 200) {

        cout << "边的取值不符合要求，请检查后重新输入！\n";

    }

    else {

        if (a + b > c && a + c > b && b + c > a) {

            //正确的三角形

            if (a == b && b == c) {

                //等边三角形

                cout << "该三条边构成等边三角形！\n";

            }

            else if (a == b || b == c || c == a) {

                //等腰三角形

                cout << "该三条边构成等腰三角形！\n";

            }

            else if (a * a + b * b == c * c || a * a + c * c == b * b |
| b * b + c * c == a * a) {

                //直角三角形

                cout << "该三条边构成直角三角形！\n";

            }

            else {
```

```

//普通三角形

cout << "该三条边构成普通三角形! \n";

    }

}

else {

//构不成三角形

cout << "该三条边构不成三角形! \n";

}

}

cout << "继续运行请输入'Y', 否则输入'N'\n";

cin >> ch;

}

if (ch == 'N') cout << "程序结束! \n";

return 0;

}

```

2 下一天程序：Java 源码

```

import java.awt.Color;

import java.awt.Container;

import java.awt.Dimension;

import java.awt.FlowLayout;

import java.awt.Font;

import java.awt.GridLayout;

import java.awt.Image;

import java.awt.TextField;

```

```
import java.awt.Toolkit;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.text.ParseException;

import java.text.SimpleDateFormat;

import java.util.Calendar;

import java.util.Date;


import javax.swing.Icon;

import javax.swing.ImageIcon;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JPasswordField;

import javax.swing.JTextField;

public class theFunc extends JFrame{

    static JTextField result=new JTextField();

    public static void main(String[] args) {

        //先创建一个窗口，不过此时创建的是隐藏的。

        JFrame jf=new JFrame();

        jf.setIconImage(Toolkit.getDefaultToolkit().getImage("E://myJar/images/service.png"));

        //再设置窗口大小，单位是像素。

        jf.setSize(800,600);

        //再窗口的标题

        jf.setTitle("下一天程序");

        //设置图标

        Toolkit t=Toolkit.getDefaultToolkit();
```



```
//Image image=t.getImage("image\\a.jpg");

//jf.setIconImage(image);

//设置窗口居中

jf.setLocationRelativeTo(null);

//设置窗口的关闭

jf.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

//存放组件, 标签组件

JLabel jl=new JLabel();

jl.setText("年");

jl.setForeground(Color.black);

jl.setFont(new Font("华文行楷",Font.BOLD,30));

jl.setSize(100, 30);

jl.setLocation(160, 50); //标签的存在位置

//把标签 jl 设置在窗口上面

//一个窗口的面板分为三层 根面板和中间面板 以及内容面板, 组件全部放在内容面板上面

// 获取到内容面板

Container con=jf.getContentPane();

//内容面板的默认布局是麻将布局 东西南北 中 在添加组件的时候如果没有指定放在那里则会
放在中间并且把窗口沾满,

// 重新设置布局

con.setLayout(null);

con.add(jl);

con.setBackground(Color.WHITE); //设置窗口颜色

//设置背景图片

Toolkit t1=Toolkit.getDefaultToolkit();

Image image1=t1.getImage("E://myJar/images/beauty.jpg");

Icon ic=new ImageIcon(image1);

JLabel jl3=new JLabel(ic);
```

```
j13.setSize(800, 600);

j13.setLocation(0, 0);

con.add(j13);

//再创建一个密码标签

JLabel j11=new JLabel();

j11.setText("月");

j11.setForeground(Color.black);

j11.setFont(new Font("华文行楷", Font.BOLD, 30));

j11.setSize(100, 30);

j11.setLocation(160, 130);

j13.add(j11);

//创建文本框

JLabel j14=new JLabel();

j14.setText("下一天日期");

j14.setForeground(Color.black);

j14.setFont(new Font("华文行楷", Font.BOLD, 20));

j14.setSize(100, 30);

j14.setLocation(160, 300);

j13.add(j14);


result.setSize(300, 100);

result.setLocation(250, 270);

j13.add(result);

JTextField year=new JTextField();

year.setSize(235, 30);
```

```
year.setLocation(250, 50);

j13.add(year);

//JPasswordField password=new JPasswordField();

JTextField month = new JTextField();

month.setSize(235, 30);

month.setLocation(250, 130);

j13.add(month);

//创建验证码标签

JLabel j12=new JLabel();

j12.setText("日");

j12.setForeground(Color.black);

j12.setFont(new Font("华文行楷", Font.BOLD, 30));

j12.setSize(100, 30);

j12.setLocation(160, 200);

j13.add(j12);

JTextField day=new JTextField();

day.setSize(235, 30);

day.setLocation(250, 200);

j13.add(day);

/*

JTextField yzm2=new JTextField();

yzm2.setSize(100, 30);

yzm2.setLocation(380, 200);

j13.add(yzm2);

JButton huoqu=new JButton();

huoqu.setText("获取验证码");

huoqu.setSize(100, 30);
```

```
huoqu.setLocation(490, 200);

huoqu.setForeground(Color.GRAY);

j13.add(huoqu);

*/

//设置按获取验证码按钮可获得验证码

JButton ok=new JButton();

ok.setSize(160, 60);

ok.setLocation(120, 400);

ok.setText("计算下一天");

ok.setFont(new Font("华文行楷",Font.PLAIN,25));

ok.setForeground(Color.BLUE);

j13.add(ok);

JButton cancel=new JButton();

cancel.setSize(160, 60);

cancel.setLocation(520, 400);

cancel.setText("重新输入");

cancel.setFont(new Font("华文行楷",Font.PLAIN,25));

cancel.setForeground(Color.red);

j13.add(cancel);

//设置窗口显示出来

jf.setVisible(true);

cancel.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent e) {
```

```
// TODO Auto-generated method stub

//System.exit(0);

year.setText("");

month.setText("");

day.setText("");

result.setText("");

    }

}

);

//注意：此处是匿名内部类

ok.addActionListener(new ActionListener() {

    public void actionPerformed(ActionEvent e) {

        // TODO Auto-generated method stub

        String syear = year.getText();

        String smonth = month.getText();

        String sday = day.getText();

        int dyear = Integer.valueOf(syear).intValue();

        int dmonth = Integer.valueOf(smonth).intValue();

        int dday = Integer.valueOf(sday).intValue();

        if(syear.length()>4||smonth.length()>2||sday.length()>2) {

            result.setText("错误—无效输入！请检查输入的字符长度！");

            return ;

        }else if(dyear<1900||dyear>2050) {

            result.setText("错误—无效输入！请检查年份输入！");

            return ;

        }else if(dmonth<1||dmonth>12) {

            result.setText("错误—无效输入！请检查月份输入！");
```

```

        return ;

    }else if(dday<1||dday>31) {

        result.setText("错误—无效输入！ 请检查日期输入！");

        return ;

    }

    String sdate = syear+"-"+smonth+"-"+sday;

    SimpleDateFormat sformat = new SimpleDateFormat("yyyy-MM-dd");

    //解析日期

    Date date = null;

    try {

        date = sformat.parse(sdate);

    } catch (ParseException e1) {

        // TODO Auto-generated catch block

        e1.printStackTrace();

    }

    Calendar c = Calendar.getInstance();

    c.setTime(date);

    //用 Calendar 工具类给日期加一天

    c.add(Calendar.DATE, 1);

    result.setText(sformat.format(c.getTime()));

}

}

);

}

}

```