# 黑盒&白盒测试

小组名: 致富经

组长: 蔡佳泉 1711216

组员: 阙定 1757127

杨阳 1859105

蔡心慧 1859209

王润炜 1859205

# 目录

1.	实验目的	3
2.	实验内容	3
3.	实验步骤	3
	1.1. 对 CALCULATOR 类,使用黑盒测试	3
	1.1.1. 原代码	3
	1.1.2. 等价类划分	4
	1.1.3. 测试用例	£
	1.1.4. Junit 测试	7
	1.2. 对 TRIANGLE 类,先画出程序流程图,再使用白盒测试	
	1.2.1. 原代码	<i>8</i>
	1.2.2. 程序流程图	10
	1.2.3. Junit 测试	11
4.	成员分工	13

## 1.实验目的

- 掌握单元测试软件 JUnit 的使用
- 掌握黑盒测试及白盒测试基本技术
- 实验内容及要求
- 编写测试用例

# 2.实验内容

- 对 Calculator 类,使用黑盒测试【等价划分法】
- 对 Triangle 类,先画出程序流程图,再使用白盒测试【条件覆盖】方法进行测试

## 3.实验步骤

- 1.1.对 Calculator 类,使用黑盒测试
- 1.1.1.原代码

Calculator.java

```
    package com.ybq.calculator;
    public class Calculator {
    private static int result = 0;
    .
```

```
public int add(int n) {
7.
           result = result + n;
           return result;
8.
9.
       }
10.
11.
       public int substract(int n) {
12.
           result = result - n;
13.
           return result;
14.
       }
15.
16.
       public int multiply(int n) {
17.
           result = result * n;
           return result;
18.
19.
       }
20.
       public int divide(int n) {
21.
22.
           result = result / n;
23.
           return result;
24.
       }
25.
       public int square(int n) {
26.
           result = n * n;
27.
           return result;
28.
29.
       }
30.
31.
       public int squareRoot(int n) {
32.
           for (; ; );
33.
       }
34.}
```

#### 1.1.2.等价类划分

输入等价类	有效等价类	无效等价类
一个整数	在-2^31 到 2^31 之间的整数	小于-2^31 的整数
	小数点后为 0 的浮点数	大于 2^31 的整数
		(即小于-2147483648 或大于
		2147483647)
		小数点后不为 0 的浮点数
		含有非数字字符

### 1.1.3.测试用例

输入条件	测试用例说明	测试数据	期望结果	选取理由
一个整数	小于-2 <sup>31</sup> 的整数	-2147483649	显示出错	比有效整数小1
	大于 2^31 的整数	2147483649	显示出错	比有效整数大 1
	小数点后为零的浮点数	1.0	输入有效	小数点后为 0
	小数点后不为零的浮点数	1.1	显示出错	小数点后不为 0
	有一个是非数字字符	1a	显示出错	只有一个非法字符
	全部是非数字字符	+ <sub>VX</sub>	显示出错	三个非法字符
	在-2^31 到 2^31 之间的整数	5	输入有效	类型和范围均有效

测试模块	测试内容	输入数据	预期输出结果	运行输出结果	测试结果
Add 方法	"n"字段数据类	-2147483649	提示输入字段	提示输入字段超过可接	通过
	型不匹配		超过可接受范	受范围	
			围		
Add 方法	"n"字段数据类	2147483649	提示输入字段	提示输入字段超过可接	通过
	型不匹配		超过可接受范	受范围	
			围		
Add 方法	"n"字段数据类	1. 1	提示输入字段	提示输入字段应为纯整	通过
	型不匹配		应为纯整数,类	数,类型不正确	
			型不正确		
Add 方法	"n"字段数据类	1a	提示输入字段	提示输入字段应为纯整	通过
	型不匹配		应为纯整数,类	数,类型不正确	
			型不正确		
Add 方法	"n"字段数据类	+ <sub>VX</sub>	提示输入字段	提示输入字段应为纯整	通过
	型不匹配		应为纯整数,类	数,类型不正确	
			型不正确		
Add 方法	"n"字段数据类	1. 0	输入成功	输入成功	通过
	型匹配				
Add 方法	"n"字段数据类	5	输入成功	输入成功	通过
	型匹配				
Substract 方法	"n"字段数据类	-2147483649	提示输入字段	提示输入字段超过可接	通过
	型不匹配		超过可接受范	受范围	
			围		
Substract 方法	"n"字段数据类	2147483649	提示输入字段	提示输入字段超过可接	通过
	型不匹配		超过可接受范	受范围	
			围		_
Substract 方法	"n"字段数据类	1. 1	提示输入字段	提示输入字段应为纯整	通过
	型不匹配		应为纯整数,类	数,类型不正确	
			型不正确		

Substract 方法       "n"字段数据类型不匹配       担示输入字段应为纯整数,类型不正确         Substract 方法       "n"字段数据类型不匹配       提示输入字段 提示输入字段应为纯整数,类型不正确         Substract 方法       "n"字段数据类型不正确       提示输入字段应为纯整数,类型不正确         Substract 方法       "n"字段数据类型匹配       1.0       输入成功       输入成功       通过         Substract 方法       "n"字段数据类型匹配       5       输入成功       输入成功       通过         Multiply 方法       "n"字段数据类型不匹配       是示输入字段型流输入字段型过可接型范围       超过可接受范围       超过可接受范围       通过         Multiply 方法       "n"字段数据类型不匹配       2147483649       提示输入字段型流输入字段超过可接型范围       通过         Multiply 方法       "n"字段数据类型不匹配       提示输入字段型流输入字段超过可接型范围       通过         Multiply 方法       "n"字段数据类型不匹配       2147483649       提示输入字段型流输入字段应为纯整型流输入字段应为纯整通过       通过	
Substract 方法       "n"字段数据类型不匹配       #x 输入字段 虚示输入字段应为纯整 通过应为纯整数,类型不正确         Substract 方法       "n"字段数据类型工正确       1.0       输入成功 输入成功 通过型工配         Substract 方法       "n"字段数据类型匹配       5       输入成功 输入成功 通过型工配         Multiply 方法       "n"字段数据类型工工程       2147483649 提示输入字段超过可接型范围       提示输入字段超过可接型范围         Multiply 方法       "n"字段数据类型工工程       2147483649 提示输入字段超过可接型范围       提示输入字段超过可接型范围         Multiply 方法       "n"字段数据类型工工程       2147483649 提示输入字段型式可接型范围       提示输入字段超过可接型范围         Multiply 方法       "n"字段数据类型工工程       2147483649 提示输入字段型型工程       提示输入字段型工程         Multiply 方法       "n"字段数据类型工程       2147483649 提示输入字段型       提示输入字段型型可接型       通过可接受范围         Multiply 方法       "n"字段数据类型工程       2147483649 提示输入字段型       提示输入字段应为纯整       通过	
Substract 方法       "n"字段数据类型不匹配       提示输入字段 提示输入字段应为纯整数,类型不正确         Substract 方法       "n"字段数据类型工正确       量不正确         Substract 方法       "n"字段数据类型工配       输入成功 输入成功 输入成功 通过         Multiply 方法       "n"字段数据类型不匹配       是示输入字段型压管         Multiply 方法       "n"字段数据类型不匹配       是示输入字段型压管型         Multiply 方法       "n"字段数据类型不匹配       是示输入字段型示输入字段型过可接型范型         超过可接受范型       是示输入字段超过可接通过可接通过可接受范型         超过可接受范型       是流输入字段超过可接通过         Multiply 方法       "n"字段数据类型不匹配       是示输入字段型         Multiply 方法       "n"字段数据类型、2147483649       提示输入字段型       提示输入字段超过可接通过         Multiply 方法       "n"字段数据类型不匹配       是示输入字段型       通过	
型不匹配       应为纯整数,类型不正确       数,类型不正确         Substract 方法       "n"字段数据类型匹配       1.0       输入成功 输入成功 输入成功 通过 输入成功 通过 输入成功 通过 通过 输入成功 编入成功 通过 通过 通过 通过可接受范型不匹配         Multiply 方法       "n"字段数据类型不匹配 超过可接受范围 图	
Substract 方法       "n"字段数据类型匹配       1.0       输入成功       输入成功       通过         Substract 方法       "n"字段数据类型匹配       5       输入成功       输入成功       通过         Multiply 方法       "n"字段数据类型不匹配       —2147483649       提示输入字段提示输入字段超过可接超过可接超过可接受范围       通过         Multiply 方法       "n"字段数据类型不匹配       是示输入字段型式可接受范围       超过可接受范型       是示输入字段超过可接超过可接超过可接超过可接超过可接受范围       通过         Multiply 方法       "n"字段数据类型不匹配       提示输入字段型设可接超过可接超过可接超过可接超过可接超过可接受范围       超过可接受范围       通过         Multiply 方法       "n"字段数据类型、1.1       提示输入字段型表,是示输入字段应为纯整通过       通过	
Substract 方法       "n"字段数据类型匹配       1.0       输入成功       输入成功       通过         Substract 方法       "n"字段数据类型匹配       5       输入成功       输入成功       通过         Multiply 方法       "n"字段数据类型不匹配       是示输入字段是超过可接超过可接超过可接超过可接超过可接超过可接超过可接超过可接超过可接超过可接	
型匹配       输入成功       输入成功       通过         Multiply 方法       "n"字段数据类型下匹配       —2147483649       提示输入字段 提示输入字段超过可接通过可接通过可接通过可接受范围       通过         Multiply 方法       "n"字段数据类型下匹配       是示输入字段型过可接受范围       通过可接受范围         Multiply 方法       "n"字段数据类型下匹配       是示输入字段型过可接受范围       通过可接受范围         Multiply 方法       "n"字段数据类型下匹配       是示输入字段型过可接型范型可接型范型可接型范围       通过可接受范围         Multiply 方法       "n"字段数据类型       1.1       提示输入字段       提示输入字段应为纯整       通过	
Substract 方法       "n"字段数据类型匹配       5       输入成功       输入成功       通过         Multiply 方法       "n"字段数据类型不匹配       是示输入字段是过可接是范型、受范围       超过可接受范型、受范围         Multiply 方法       "n"字段数据类型不匹配       是示输入字段是过可接是范型的工程。       超过可接受范型、受范围         Multiply 方法       "n"字段数据类型不匹配       是示输入字段是过可接是范型的工程。       通过可接受范型的工程。         Multiply 方法       "n"字段数据类型、1.1       提示输入字段       提示输入字段应为纯整       通过	
型匹配       型匹配       提示输入字段 提示输入字段超过可接 通过 超过可接受范 图	
型不匹配     超过可接受范	
型不匹配     超过可接受范	
Multiply 方法     "n"字段数据类型不匹配     是示输入字段型进可接型范型可接型范型       Multiply 方法     "n"字段数据类型、1.1     提示输入字段型进可接型范型       提示输入字段型进可接型范围     型流围       Multiply 方法     "n"字段数据类型、1.1     提示输入字段型	
型不匹配     超过可接受范     受范围       围     围       Multiply 方法     "n"字段数据类     1.1     提示输入字段     提示输入字段应为纯整     通过	
Multiply 方法     "n"字段数据类     1.1     提示输入字段     提示输入字段应为纯整     通过	
Multiply 方法 "n"字段数据类 1.1 提示输入字段 提示输入字段应为纯整 通过	
型不匹配 应为纯整数,类 数,类型不正确	
型不正确	
Multiply 方法 "n"字段数据类 la 提示输入字段 提示输入字段应为纯整 通过	
型不匹配 应为纯整数,类 数,类型不正确	
型不正确	
Multiply 方法 "n"字段数据类 +vx 提示输入字段 提示输入字段应为纯整 通过	
型不匹配	
型不正确	
Multiply 方法 "n"字段数据类 1.0 输入成功 输入成功 通过	
型匹配	
Multiply 方法 "n"字段数据类 5 输入成功 输入成功 通过	
型匹配	
Divide 方法 "n"字段数据类 -2147483649 提示输入字段 提示输入字段超过可接 通过	
型不匹配 超过可接受范 受范围	
围	
Divide 方法 "n"字段数据类 2147483649 提示输入字段 提示输入字段超过可接 通过	
型不匹配 超过可接受范 受范围	
围	
Divide 方法 "n"字段数据类 1.1 提示输入字段 提示输入字段应为纯整 通过	
型不匹配 应为纯整数,类 数,类型不正确	
型不正确	
Divide 方法 "n"字段数据类 la 提示输入字段 提示输入字段应为纯整 通过	
型不匹配 应为纯整数,类 数,类型不正确	
型不正确	
Divide 方法 "n"字段数据类 +vx 提示输入字段 提示输入字段应为纯整 通过	
型不匹配 应为纯整数,类 数,类型不正确	
型不正确	

Divide 方法	"n"字段数据类 型匹配	1.0	输入成功	输入成功	通过
Divide 方法	"n"字段数据类 型匹配	5	输入成功	输入成功	通过
Square 方法	"n"字段数据类型不匹配	-2147483649	提示输入字段 超过可接受范 围	提示输入字段超过可接 受范围	通过
Square 方法	"n"字段数据类型不匹配	2147483649	提示输入字段 超过可接受范围	提示输入字段超过可接 受范围	通过
Square 方法	"n"字段数据类型不匹配	1.1	提示输入字段 应为纯整数,类 型不正确	提示输入字段应为纯整 数,类型不正确	通过
Square 方法	"n"字段数据类型不匹配	1a	提示输入字段 应为纯整数,类 型不正确	提示输入字段应为纯整 数,类型不正确	通过
Square 方法	"n"字段数据类型不匹配	+ <sub>VX</sub>	提示输入字段 应为纯整数,类 型不正确	提示输入字段应为纯整 数,类型不正确	通过
Square 方法	"n"字段数据类 型匹配	1.0	输入成功	输入成功	通过
Square 方法	"n"字段数据类 型匹配	5	输入成功	输入成功	通过

### 1.1.4. Junit 测试

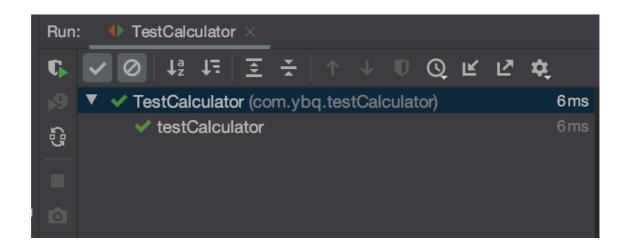
#### TestCalculator.java

```
    package com.ybq.testCalculator;

2.
3. import com.ybq.calculator.Calculator;
4. import org.junit.Test;
5.
6. import static org.junit.Assert.assertEquals;
7.
8. public class TestCalculator {
9.
10.
       @Test
       public void testCalculator() {
11.
12.
           Calculator calculator = new Calculator();
13.
           int sum = calculator.add(100);
```

```
14.
           assertEquals(sum, 100);
15.
           sum = calculator.substract(50);
16.
17.
           assertEquals(sum, 50);
18.
19.
           sum = calculator.multiply(2);
20.
           assertEquals(sum, 100);
21.
22.
           sum = calculator.divide(2);
23.
           assertEquals(sum,50);
24.
25.
           sum = calculator.square(2);
26.
           assertEquals(sum,4);
27.
       }
28.}
```

#### 结果



### 1.2.对 Triangle 类,先画出程序流程图,再使用白盒测试

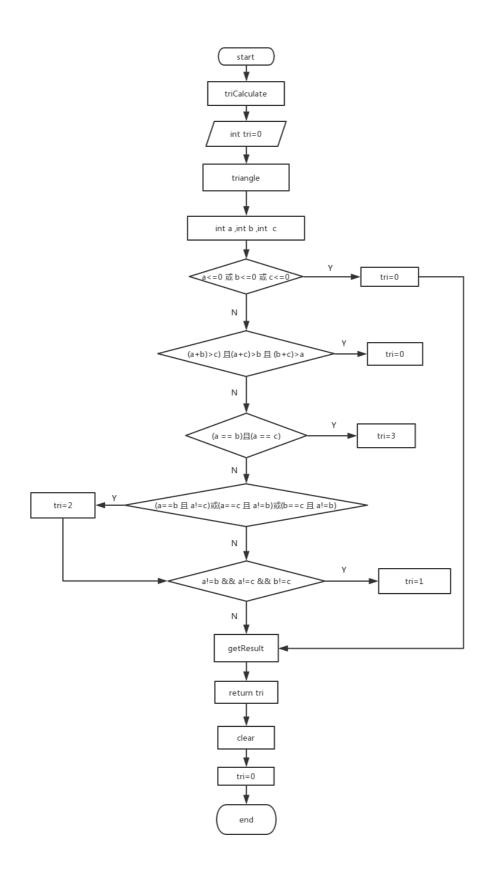
#### 1.2.1.原代码

TriCalculate.java

```
    package com.ybq.triangle;
    public class TriCalculate {
    private static int tri = 0;
```

```
5.
6.
       public void triangle(int a, int b, int c) {
7.
          //有一条边为非正数,不是三角形
          if (a <= 0 || b <= 0 || c <= 0) {
8.
9.
              tri = 0;
10.
          } else {
              //三边任意两边大于第三边,是三角形
11.
12.
              if (((a + b) > c) && ((a + c) > b && (b + c) > a)) {
13.
                  if ((a == b) && (a == c)) {
14.
                     tri = 3;//等边三角形
15.
16.
                  if ((a == b && a != c) || (a == c && a != b) || (
   b == c \&\& a != b)) {
17.
                      tri = 2;//等腰三角形
18.
                  if (a != b && a != c && b != c) {
19.
20.
                     tri = 1;//一般三角形
21.
                  }
22.
              } else {
23.
                  tri = ∅;//三边长度无法构成三角形
24.
25.
          }
26.
27.
       public int getReuslt() {
28.
29.
          return tri;
30.
31.
32.
       public void clear() {
33.
          tri = 0;
34.
35.}
```

### 1.2.2.程序流程图



#### 1.2.3.Junit 测试

#### TestTriCalculate.java

```
    package com.ybq.testTriangle;

2.
3. import com.ybq.triangle.TriCalculate;
4. import org.junit.Test;
5.
import static org.junit.Assert.assertEquals;
7.
8. public class TestTriCalculate {
       /**
9.
10.
        * 测试 clear()
11.
12.
       @Test
       public void testClear() {
13.
14.
           TriCalculate triCalculate = new TriCalculate();
15.
           triCalculate.triangle(1,1,1);
16.
           assertEquals(triCalculate.getReuslt(), 3);
17.
           triCalculate.clear();
           assertEquals(triCalculate.getReuslt(), 0);
18.
19.
       }
20.
       /**
21.
22.
        * 测试等边三角形
23.
        */
24.
       @Test
25.
       public void testEqualTir() {
26.
           TriCalculate triCalculate = new TriCalculate();
27.
           triCalculate.triangle(1,1,1);
28.
           assertEquals(triCalculate.getReuslt(), 3);
29.
       }
30.
31.
       /**
        * 测试等腰三角形
32.
        */
33.
34.
       @Test
       public void testIsoTri() {
35.
36.
           TriCalculate triCalculate = new TriCalculate();
37.
           triCalculate.triangle(1,2,2);
38.
           assertEquals(triCalculate.getReuslt(), 2);
39.
       }
```

```
40.
       /**
41.
        * 测试等腰三角形
42.
43.
        */
44.
       @Test
45.
       public void testComTri() {
46.
           TriCalculate triCalculate = new TriCalculate();
47.
           triCalculate.triangle(3,4,5);
           assertEquals(triCalculate.getReuslt(), 1);
48.
49.
       }
50.
51.
       /**
52.
        * 测试边是非正数
        */
53.
54.
       @Test
       public void testNotPosNum() {
55.
56.
           TriCalculate triCalculate = new TriCalculate();
57.
           triCalculate.triangle(-1,-2,-3);
           assertEquals(triCalculate.getReuslt(), ∅);
58.
59.
       }
60.
       /**
61.
        * 测试三边任意两边不大于第三边
62.
63.
       @Test
64.
65.
       public void testNotTir() {
66.
           TriCalculate triCalculate = new TriCalculate();
67.
           triCalculate.triangle(1,2,3);
68.
           assertEquals(triCalculate.getReuslt(), 0);
69.
       }
70.}
```

#### 结果



# 4.成员分工

学号	姓名	任务分工
1711216	蔡佳泉	Junit 测试
1859105	杨阳	等价类划分&测试用例
1859205	王润炜	Junit 测试
1859209	蔡心慧	程序流程图
1757127	阙定	文档排版