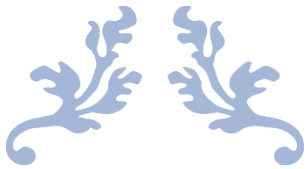


SECOND THEORETICAL WORK

Exercise 2

GROUP: ISO2-2022-A01

MEMBERS: GONZALO GÓMEZ VIDAL GONZALO ARÉVALO GÓMEZ

Index:

Statement one..... **¡Error! Marcador no definido.**
Statement two..... 2
Statement three..... 3
Statement four..... 3
Statement five..... 3
Statement six..... 4
Statement seven..... 5
Statement eight..... 8
Statement nine..... 10

1. Statement one

Write, at least the pseudocode of the identified method:

```
public String GetTypeTriangle(Triangle tri) throws TriangleException {
    String Type="";
    try {
        if(tri.getangle1()<=0 || tri.getangle2()<=0 || tri.getangle3()<=0 || !tri.isValid() ) {
            throw new TriangleException("Numbers cannot be negative or 0 or the sum of the angles must be 180 ");}
        }
            catch (TriangleException e) {
        System.out.println("Invalid input: " + e.getMessage());}
    if(tri.getangle1()==60 && tri.getangle2()==60 && tri.getangle3()==60) {
        Type+="equilateral,acute-angled";}
    else if(tri.getangle1()==tri.getangle2() || tri.getangle2()==tri.getangle3() || tri.getangle1()==tri.getangle3())   {
                if(tri.getangle1()>90 || tri.getangle2()>90 || tri.getangle3()>90) {
                    Type="isosceles,obtuse-angled";
                    }
                else if(tri.getangle1()==90 || tri.getangle2()==90 || tri.getangle3()==90) {
                    Type="isosceles,rectangle"
                }
                else {
                    Type="isosceles,acute-angled"
                }

                }
    else {
        if(tri.getangle1()>90 || tri.getangle2()>90 || tri.getangle3()>90) {
            Type="scalene,obtuse-angled";
            }
        else if(tri.getangle1()==90 || tri.getangle2()==90 || tri.getangle3()==90) {
            Type="scalene,rectangle"
        }
        else {
            Type="scalene,acute-angled"
        }

        }
```

2. Statement two

Identify the variables that must be considered to test the method

To test the method we must check the variables:

angle1,angle2,angle3

3. Statement three

Identify the test values for each one of the variables previously identified, specifying the technique used to obtain each of those values).

Test Values

Parameter	Equivalence Class	Values *Error guessing	Boundary values (Light variant)
Tri.getangle1()	[-∞,0] (0,60] (60,90) [90,180) [180, ∞)	-4 13 79 93 *120 200	0 60 90 180
Tri.getangle2()	[-∞,0] (0,60] (60,90) [90,180) [180, ∞)	-4 13 81 111 *123 210	0 60 90 180
Tri.getangle3()	[-∞,0] (0, ∞) [180, ∞)	-4 20 65 147 *122 210	0 60 90 180

4. Statement four

Calculate the maximum possible number of test cases that could be generated from the test values.

The maximum number of test cases is 10*10*10=1000

5 Statement five

Define some test suites using each use

Each Use
CP1:{-4,-4,-4}
CP2:{13,13,20}
CP3:{79,85,65}
CP4:{93,111,147}
CP5:{120,123,122}
CP6:{0,0,0}
CP7:{60,60,60}
CP8:{90,90,90,180}
CP9:{180,180,180}
CP10:{200,210,210}

6. Statement six

Define test suits to achieve pairwaise coverage by using the proposed algorithm in Lectures. You can check the results by means of the software PICT

Tri.getangle1()	Tri.getangle2()	Tri.getangle3()
90	111	180
120	-4	122
13	60	0
90	13	122
90	60	210
120	210	147
93	210	-4
120	60	65
180	13	65
60	0	122
180	111	0
0	123	-4
90	210	65
-4	123	180
79	180	65
0	90	147
79	111	90
90	180	60
180	180	122
90	90	20
79	13	20
-4	0	20
79	123	210
13	123	60
-4	-4	147
79	0	0
180	60	90
13	-4	90
200	180	180
0	81	0
-4	81	90
93	-4	0
79	90	60
93	13	90
0	210	90
120	13	-4
200	-4	65
13	210	20
13	81	210
60	111	60
-4	210	210
13	13	180
120	123	90
120	0	180
180	90	180
60	123	20
60	180	90
-4	180	-4
200	210	60
120	90	0
-4	90	122
120	111	210
200	60	122
60	81	65
93	180	20
93	81	147
93	90	65
60	60	147
79	81	122
200	90	90
93	123	122

7.Statement seven

For code snippets that include decisions, propose a set of test cases to achieve coverage of decisions.

angle1<=0 or angle2<=0 or angle3<=0 or not valid

A :angle1 <=0
B: angle2<=0
C: angle3<=0
D: not valid

A	B	C	D	A or B or C or D	Dominant
T	T	T	T	T	A,B,C,D
T	T	T	F	T	A,B,C
T	T	F	T	T	A,B,D
T	T	F	F	T	A,B
T	F	T	T	T	A,C,D
T	F	T	F	T	A,C
T	F	F	T	T	A,D
T	F	F	F	T	A
F	T	T	T	T	B,C,D
F	T	T	F	T	B,C
F	T	F	T	T	B,D
F	T	F	F	T	B
F	F	T	T	T	C,D
F	F	T	F	T	C
F	F	F	T	T	D
F	F	F	F	F	A,B,C,D

TEST CASES

Tri.getMaxAngle()	Tri.getSide1()	Tri.getSide2()	Tri.isValid	Result
70	12	11	False	T
30	30	120	True	F

angle1==60 and angle2==60 and angle3==60
A: angle1==60
B: angle1==60
C: angle1==60

A	B	C	A and B and C	Dominant
T	**T**	**T**	**T**	**A,B,C**
T	**T**	**F**	**F**	**C**
T	**F**	**T**	**F**	**B**
T	**F**	**F**	**F**	**B,C**
F	**T**	**T**	**F**	**A**
F	**T**	**F**	**F**	**A,C**
F	**F**	**T**	**F**	**A,B**
F	**F**	**F**	**F**	**A,B,C**

TEST CASES

Tri.getangle1()	Tri.getangle2()	Tri.getangle3()	Result
60	**60**	**60**	**T**
80	**60**	**60**	**F**

(Angle1=angle2 or angle2=angle3 or angle1=angle3) and not ((angle1=angle2 and angle1=angle3) or (angle1=angle2 and angle2=angle3) or (angle1=angle3 and angle2=angle3)
A: Angle1=Angle2
B: angle2=angle3
C: angle1=angle3

A	B	C	(A or B or C) and not ((A and C) or (B and A) or (C and B)	Dominant
T	**T**	**T**	**F**	**A,C,B**
T	**T**	**F**	**F**	**A,B**
T	**F**	**T**	**F**	**A,C**
T	**F**	**F**	**T**	**A**
F	**T**	**T**	**F**	**B,C**
F	**T**	**F**	**T**	**B**
F	**F**	**T**	**T**	**C**
F	**F**	**F**	**F**	**A,C,B**

.
TEST CASES

Tri.getangle1()	Tri.getangle2()	Tri.getangle3()	Result
45	**90**	**45**	**T**
36	**90**	**54**	**F**

angle1>90 or angle2>90 or angle3>90:

A:angle1>90
B:angle2>90
C:angle3>90

A	B	C	A or B or C	Dominant
T	T	T	T	A,B,C
T	T	F	T	A,B
T	F	T	T	A,C
T	F	F	T	A
F	T	T	T	B,C
F	T	F	T	B
F	F	T	T	C
F	F	F	F	A,B,C

Tri.getangle1()	Tri.getangle2()	Tri.getangle3()	Result
45	90	45	F
40	40	100	T

Angle1=90 or angle2=90 or angle3=90
A:angle1=90
B: angle2=90
C: angle3=90

A	B	C	A or B or C	Dominant
T	T	T	T	A,B,C
T	T	F	T	A,B
T	F	T	T	A,C
T	F	F	T	A
F	T	T	T	B,C
F	T	F	T	B
F	F	T	T	C
F	F	F	F	A,B,C

TEST CASES

Tri.getangle1()	Tri.getangle2()	Tri.getangle3()	Result
45	45	90	T
40	40	100	F

8.Statement eight

For code snippets that include decisions, propose test case sets to achieve MC/DC coverage.

angle1<=0 or angle2<=0 or angle3<=0 or not valid

A :angle1 <=0
B: angle2<=0
C: angle3<=0
D: not valid

A	B	C	D	A or b or C or D	Dominant
T	T	T	T	T	A,B,C,D
T	T	T	F	T	A,B,C
T	T	F	T	T	A,B,D
T	T	F	F	T	A,B
T	F	T	T	T	A,C,D
T	F	T	F	T	A,C
T	F	F	T	T	A,D
T	F	F	F	T	A
F	T	T	T	T	B,C,D
F	T	T	F	T	B,C
F	T	F	T	T	B,D
F	T	F	F	T	B
F	F	T	T	T	C,D
F	F	T	F	T	C
F	F	F	T	T	D
F	F	F	F	F	A,B,C,D

TEST CASES

Tri.getangle1()	Tri.getangle2()	Tri.getangle3()	Tri.isValid()	Result
70	12	11	False	T
30	30	120	True	F
90	95	-5	True	T
90	-5	95	True	T
-5	90	95	True	T

angle1==60 and angle2==60 and angle3==60
A: angle1==60
B: angle1==60
C: angle1==60

A	B	C	A and B and C	Dominant
T	T	T	T	A,B,C
T	T	F	F	C
T	F	T	F	B
T	F	F	F	B,C
F	T	T	F	A
F	T	F	F	A,C
F	F	T	F	A,B
F	F	F	F	A,B,C

TEST CASES

Tri.getangle1()	Tri.getangle2()	Tri.getangle3()	Result
60	60	60	T
80	60	60	F
60	50	60	F
70	60	60	F

(Angle1=angle2 or angle2=angle3 or angle1=angle3) and not ((angle1=angle2 and angle1=angle3) or (angle1=angle2 and angle2=angle3) or (angle1=angle3 and angle2=angle3)
A: Angle1=Angle2
B: angle2=angle3
C: angle1=angle3

A	B	C	(A or B or C) and not ((A and C) or (B and A) or (C and B)	Dominant
T	T	T	F	A,C,B
T	T	F	F	A,B
T	F	T	F	A,C
T	F	F	T	A
F	T	T	F	B,C
F	T	F	T	B
F	F	T	T	C
F	F	F	F	A,C,B

TEST CASES

Tri.getangle1()	Tri.getangle2()	Tri.getangle3()	Result
45	90	45	T
36	90	54	F
90	45	45	T
45	45	90	T

angle1>90 or angle2>90 or angle3>90
A:angle1>90
B:angle2>90
C:angle3>90

A	B	C	A or B or C	Dominant
T	T	T	T	A,B,C
T	T	F	T	A,B
T	F	T	T	A,C
T	F	F	T	A
F	T	T	T	B,C
F	T	F	T	B
F	F	T	T	C
F	F	F	F	A,B,C

TEST CASES

Tri.getangle1()	Tri.getangle2()	Tri.getangle3()	Result
45	90	45	F
40	40	100	T
40	100	40	T
100	40	40	T

Angle1=90 or angle2=90 or angle3=90
A:angle1=90
B: angle2=90
C: angle3=90

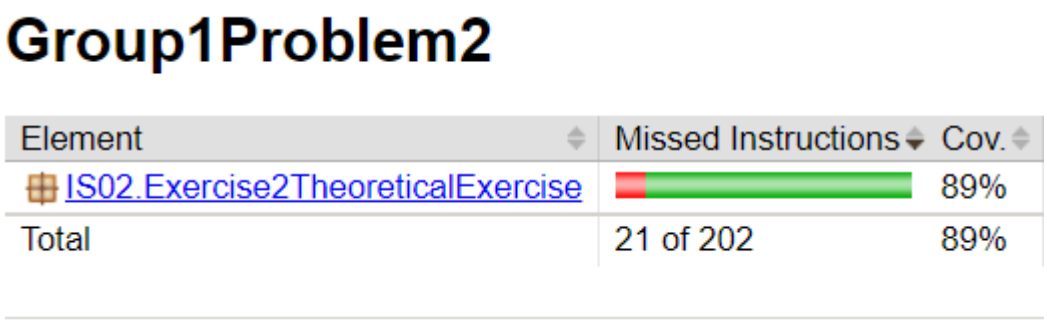
A	B	C	A or B or C	Dominant
T	T	T	T	A,B,C
T	T	F	T	A,B
T	F	T	T	A,C
T	F	F	T	A
F	T	T	T	B,C
F	T	F	T	B
F	F	T	T	C
F	F	F	F	A,B,C

TEST CASES

Tri.getangle1()	Tri.getangle2()	Tri.getangle3()	Result
45	45	90	T
40	40	100	F
45	90	45	T
90	45	45	T

9.Statement nine

Comment on the results of the number of test cases obtained in section 4, 5, and 6, as well as the execution of the oracles: what could be said about the coverage achieved?



Doing this type of test we achieve a coverage of 89% which is a high coverage that assure as the program is well tested