**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**

**FACULTY OF ELECTRICAL AND COMPUTER ENGINEERING**

**SOFTWARE ENGINEERING II**

**eMPIRICAL WORKSHOP - I TERM 2021**

**Objectives:**

* Test the program freely using intuitive test cases.

**Requirements**

* Eclipse IDE for Java Developers >= 2019-09 R (4.13.0)
* Git

**Introduction**

The triangle problem is the most widely used example in software testing literature. The logic used for the problem is clear but complex, meaning that behind some intuitive conditions are other hidden ones more difficult to get.

The traditional problem states the following: the triangle program accepts three integers, a, b, and c, as input. These are taken to be sides of a triangle. The output of the program is the type of triangle determined by the three sides: Equilateral, Isosceles, Scalene, or Not Triangle. [1]

To start, it is important to define what a triangle is. Let us see some definitions and compare them with yours:

* A closed plane figure having three sides and three angles [2].
* A polygon having three sides [3].
* A triangle is a polygon with three edges and three vertices [4].

These definitions describe what it is explicitly seen by a person, not what mathematical/ geometrical implications and theorems really define a triangle. Here is where this simple problem turns into a difficult and complex task. Not because of triangle’s properties and rules, but the assumptions the problem does. One assumption is that developers know some details about triangles, particularly the triangle inequality theorem: the sum of any pair of sides must be strictly greater than the third side.

Here is an improved version of the problem and the one we will use during the lab [1]:

The triangle program accepts three integers, a, b, and c, as input. These are taken to be sides of a triangle. The integers a, b, and c must satisfy the following conditions:

c1. 1 ≤ a ≤ 200.

c3. 1 ≤ c ≤ 200

c2. 1 ≤ b ≤ 200

c5. b < a + c

c4. a < b + c

c6. c < a + b

The output of the program is the type of triangle determined by the three sides: Equilateral, Isosceles, Scalene, or Not Triangle. If an input value fails any of conditions c1, c2, or c3, the program notes this with an output message, for example, “Value of b is not in the range of permitted values.” If values of a, b, and c satisfy conditions c4, c5, and c6, one of four mutually exclusive outputs is given:

1. If all three sides are equal, the program output is Equilateral.
2. If exactly one pair of sides is equal, the program output is Isosceles.
3. If no pair of sides is equal, the program output is Scalene.
4. If any of conditions c4, c5, and c6 is not met, the program output is Not Triangle.

**Development**

Create a Java program that solves the triangle problem, fulfilling all the conditions involved. Below, a pseudocode that you can use to create your program:

Program triangle’

Dim a, b, c As Integer

Dim c1, c2, c3, Is Triangle as Boolean

‘Step 1: Get Input

Do

Output (“Enter 3 integers which are sides of a triangle”)

Input (a, b, c)

c1 = (1 ≤ a) AND (a ≤ 200)

c2 = (1 ≤ b) AND (b ≤ 200)

c3 = (1 ≤ c) AND (c ≤ 200)

If NOT(c1)

Then Output (“Value of a is not in the range of permitted values”)

End If

If NOT(c2)

Then Output (“Value of b is not in the range of permitted values”)

End If

If NOT(c3)

Then Output (“Value of c is not in the range of permitted values”)

End If

Until c1 AND c2 AND c3

Output (“Side A is”, a)

Output (“Side B is”, b)

Output (“Side C is”, c)

‘Step 2: Is A Triangle?

If (a < b + c) AND (b < a + c) AND (c < a + b)

Then Is Triangle = True

Else Is Triangle = False

End If

‘Step 3: Determine Triangle Type

If Is Triangle

Then If (a = b) AND (b = c)

Then Output (“Equilateral”)

Else If (a ≠ b) AND (a ≠ c) AND (b ≠ c)

Then Output (“Scalene”)

Else Output (“Isosceles”)

End If

End If

Else Output (“Not a Triangle”)

End If

End triangle

After writing your code, it is time to test whether it behaves correctly. To do this, you should write certain combinations of inputs and expected outputs. This must be done **without using your code**; you must produce a table like the one presented below. To do it, you must use the requirements of the problem, your *assumptions* as well as your experience and knowledge.

Test Cases

|  |  |  |
| --- | --- | --- |
| Test | Input Values (a, b, c) | Expected Output |
| 1 | 3, 3, 3 | Equilateral |
| 2 | 4, 5, “?” | Invalid Input |

Assumptions are critical to test planning, since the set of test cases needed to test the behavior of the program will vary based on the assumptions you made. Assumptions need to be documented. The most dangerous assumptions are those which are not documented but which are simply assumed to be true. [5]

For example:

Assumption:

* Non numerical values are invalid, i.e. “A” or “?”

For this lab, you must:

1) Complete the table with tests and assumptions you consider necessary.

2) Implement the tests into your code. For example: Suppose you created a method called “triangleType” that receives 3 integers and returns a string with the type of the triangle. To test if the output is correct based on your test cases, do the following:

System.out.println(“Test 1: ”+ “Equilateral”.equals(triangleType(3,3,3)));

System.out.println(“Test 2: ”+ “Invalid Input”.equals(triangleType(4,5,’?’)));

If any of yours tests fails, create a new java class, copy your code, and modify it so that it behaves according to your test cases.

**Deliverables**

1. Document your test cases and assumptions.
2. **URL** of the repository where you performed the lab.

**Rubric**

|  |  |
| --- | --- |
| **Description** | **Value** |
| Project code (in the repository) | 50 |
| Test cases and assumptions | 50 |
| Penalty per hour or fraction of delay. | -30 |
| Penalty for not uploading required deliverables as specified | -30 |

# **References**

|  |  |
| --- | --- |
| [1] | P. Jorgensen, Software Testing, CRC Press, 2014. |
| [2] | Dictionary.com, "Triangle," [Online]. Available: https://www.dictionary.com/browse/triangle. |
| [3] | Merriam-Webster, "Triangle," [Online]. Available: https://www.merriam-webster.com/dictionary/triangle. |
| [4] | Wikipedia, "Triangle," [Online]. Available: https://en.wikipedia.org/wiki/Triangle. |
| [5] | Collard & Company, "ANALYZING THE TRIANGLE PROBLEM," 2004. |