



Current scope: [all classes](#) | [com.sselab.tritypejunittest](#)

Coverage Summary for Package: com.sselab.tritypejunittest

| Package | Class, % | Method, % | Line, % |
|-----------------------------|------------|-----------|---------------|
| com.sselab.tritypejunittest | 100% (1/1) | 60% (3/5) | 25.5% (12/47) |

| Class | Class, % | Method, % | Line, % |
|-------------------------|------------|-----------|---------------|
| Tritype | 100% (1/1) | 60% (3/5) | 25.5% (12/47) |

generated on 2023-12-05 13:52

Coverage Summary for Class: Tritype (com.selab.tritypejunittest)

| Class | Class, % | Method, % | Line, % |
|---------|------------|-----------|---------------|
| Tritype | 100% (1/1) | 60% (3/5) | 25.5% (12/47) |

```
1 package com.selab.tritypejunittest;
2
3 import java.io.*;
4
5 class Tritype
6 {
7     private static String[] triTypes = { "", // ignore 0,
8     "scalene", "isosceles", "equilateral",
9     "not a valid triangle" };
10    private static String instructions = "This is the ancient " +
11    "Trityp program.\nEnter three integers that represent the " +
12    "lengths of the sides of a triangle.\nThe triangle will be " +
13    "categorized as either scalene, isosceles, equilateral\n" +
14    "or invalid.\n";
15
16    public static void main (String[] argv)
17    { // Driver program for trityp
18        int A, B, C;
19        int T;
20
21        // If (argv.length == 3) {
22        //     A = Integer.parseInt(argv[0]);
23        //     B = Integer.parseInt(argv[1]);
24        //     C = Integer.parseInt(argv[2]);
25        // } else {
26        System.out.println (instructions);
27        System.out.println ("Enter side 1: ");
28        A = getW();
29        System.out.println ("Enter side 2: ");
30        B = getW();
31        System.out.println ("Enter side 3: ");
32        C = getW();
33        T = Triang (A, B, C);
34        // }
35        System.out.println ("Result is: " + triTypes[T]);
36    }
37
38    // =====
39    // The main triangle classification method
40    static int Triang (int Side1, int Side2, int Side3)
41    {
42        int triOut;
43
44        // triOut is output from the routine:
45        //   Triang = 1 if triangle is scalene
46        //   Triang = 2 if triangle is isosceles
47        //   Triang = 3 if triangle is equilateral
48        //   Triang = 4 if not a triangle
49
50        // After a quick confirmation that it's a legal
51        // triangle, detect any sides of equal length
52        if (Side1 <= 0 || Side2 <= 0 || Side3 <= 0)
53        {
54            triOut = 4;
55            return (triOut);
56        }
57
58        triOut = 0;
59        if (Side1 == Side2)
60            triOut = triOut + 1;
61        if (Side1 == Side3)
62            triOut = triOut + 2;
63        if (Side2 == Side3)
64            triOut = triOut + 3;
65        if (triOut == 0)
66        { // Confirm it's a legal triangle before declaring
67            // it to be scalene
68            if (Side1+Side2 <= Side3 || Side2+Side3 <= Side1 ||
69                Side1+Side3 <= Side2)
70                triOut = 4;
71            else
72                triOut = 1;
73            return (triOut);
74        }
75
76        // Confirm it's a legal triangle before declaring */
77        // it to be isosceles or equilateral */
78
79        if (triOut > 3)
80            triOut = 3;
81        else if (triOut == 1 && Side1+Side2 > Side3)
82            triOut = 2;
83        else if (triOut == 2 && Side1+Side3 > Side2)
84            triOut = 2;
85        else if (triOut == 3 && Side2+Side3 > Side1)
86            triOut = 3;
87        else
88            triOut = 4;
89        return (triOut);
90    } // end Triang
91
92    // =====
93    // Read (or choose) an integer
94    private static int getW ()
95    {
96        int inputInt = 1;
97        BufferedReader in = new BufferedReader (new InputStreamReader (System.in));
98        String instr;
99
100        try
101        {
102            instr = in.readLine ();
103            inputInt = Integer.parseInt(instr);
104        }
105        catch (IOException e)
106        { // JDE requires the IOException to be caught.
107            System.out.println ("Could not read input, choosing 1.");
108        }
109        catch (NumberFormatException e)
110        {
111            System.out.println ("Entry must be a number, choosing 1.");
112        }
113        return (inputInt);
114    } // end getW
115
116    } // end Tritype class
```