public void mostrarResultado() {

1

double lado1, lado2, lado3;

4

3

2

if (lado1ok && lado2ok && lado3ok) {

lado1 = Double.valueOf(textLado1.getText().replace(',', '.'));

5

lado2 = Double.valueOf(textLado2.getText().replace(',', '.'));

lado3 = Double.valueOf(textLado3.getText().replace(',', '.'));

// Es triangulo

8

7

6

if (lado1 + lado2 > lado3 && lado1 + lado3 > lado2 && lado2 + lado3 > lado1) {

9

labelResultado.setForeground(Color.GREEN);

10

if (lado1 == lado2)

11

if (lado1 == lado3)

labelResultado.setText("Es un triángulo equilátero");

12

else

labelResultado.setText("Es un triángulo isósceles");

15

14

13

else if (lado1 == lado3 || lado2 == lado3) {

13

labelResultado.setText("Es un triángulo isósceles");

else

labelResultado.setText("Es un triángulo escaleno");

16

else

labelResultado.setForeground(Color.RED);

17

labelResultado.setText("No es un triángulo, la suma de dos lados debe ser mayor al tercero");

} else {

labelResultado.setForeground(Color.RED);

labelResultado.setText("Entrada no válida, debe ingresarse un número mayor a 0");

18

}

19

} // Fin

Caminos:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Caminos | | | | | | | | | | | | | | |
| C1 | 1 | 2 | 17 | 19 |  |  |  |  |  |  |  |  |  |  |
| C2 | 1 | 2 | 3 | 17 | 19 |  |  |  |  |  |  |  |  |  |
| C3 | 1 | 2 | 3 | 4 | 17 | 19 |  |  |  |  |  |  |  |  |
| C4 | 1 | 2 | 3 | 4 | 5 | 6 | 18 | 19 |  |  |  |  |  |  |
| C5 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 18 | 19 |  |  |  |  |  |
| C6 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 18 | 19 |  |  |  |  |
| C7 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 19 |  |
| C8 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 13 | 19 |  |
| C9 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 14 | 13 | 19 |  |
| C10 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 14 | 15 | 13 | 19 |
| C11 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 14 | 15 | 16 | 19 |

La complejidad ciclomática según el algoritmo de McCabe es de 11.

(28 aristas – 19 nodos + 2 = 11)

Por lo tanto, deberán realizarse un mínimo de 11 pruebas para poder probar el programa.

17

18

19/FIN

12

15

16

13

11

14

10

9

8

7

6

5

4

3

2

1