

Estaciones Espaciales Triangulares

Análisis del problema: la agencia espacial necesita clasificar los 3 tipos de triángulos ya que con esta forma triangular ayudan a la optimización y captación de energía solar, se necesita que cada estructura se clasifique en alguno de los tipos de triángulo como lo son equilátero, isósceles y escaleno. Al igual se necesita descartar las estructuras que no formen un triángulo con la medida de sus 3 lados.

Requisitos

Funcionales: Se deben ingresar 3 valores numéricos que sean los lados de el triángulo a analizar.

- se debe verificar el cumplimiento de la formación del triángulo luego de haber ingresado los 3 valores numéricos sabiendo que la suma de 2 lados sean los que sean tiene que ser mayor que el 3er lado sobrante

- luego de demostrar si es valido el triángulo se necesita clasificar en equilátero, isósceles y escaleno.

- al realizar todos estos procesos y verificar si es o no es triángulo, al usuario se le tiene que dar la respectiva información de que tipo de triángulo es, o si no es un triángulo valido.

No funcionales: El programa tiene que ser simple a la hora de manejarlo pidiendo solo lo necesario y no añadiendo cosas que puedan perjudicar el desarrollo del código

- se tiene que ejecutar de manera rápida y correcta la clasificación con los datos ingresados, sin errores.

- tiene que validar solo la entrada de valores numéricos para que no se generen errores en los cálculos

- se tiene que ejecutar en cualquier programa que soporte Python o c++.

- si es necesaria una revisión del código se necesita una buena organización de este para que no halla problema con el mantenimiento de este o visualización de alguna falla.

Casos de uso

- El usuario ingresa los valores para ser analizados por el programa y que este clasifique un triángulo respecto a estos, si el usuario ingresa un valor que no sea numérico entero este generaría un error y necesitaría un reinicio a el análisis

- Se ingresan valores al programa para determinar si es un triángulo o no lo es, el sistema analiza la clasificación y genera equilátero cuando los 3 lados son iguales, isósceles cuando

2 lados son iguales, escaleno cuando los 3 lados son diferentes y no es un triángulo cuando la suma de 2 lados es menor o igual al 3er lado, para que al final en la pantalla se podrá visualizar como se clasificó.

ENTRADAS:

- 3 valores numéricos enteros positivos ingresados por el usuario
- posibles errores si se ingresa otro valor diferente a un entero positivo

PROCESOS:

- Se verifica si los valores ingresados son números positivos
- Comprobar si la suma de dos lados es mayor que el tercero
- Determinar si es equilátero, isósceles, escaleno o no es triángulo según sus lados.

SALIDAS:

Mensaje indicando si es equilátero, isósceles, escaleno o no es un triángulo

JUSTIFICACIÓN DE SOLUCIÓN

Busque con un diagrama de flujo (anexado al final del informe) los diferentes casos que se podrían presentar en el programa para así realizar los mismos pasos en el código, primeramente se tiene que validar si los valores ingresados son válidos para la realización del problema, primeramente se tienen que exigir 3 lados para poder hacer el cálculo de que tipo de triángulo eso, así que a la hora de ingresar los valores se empieza el análisis, si los valores no son enteros positivos esto generaría un error ya que no se podría hacer el cálculo de manera clara y tendría que volver a ingresar datos válidos para que el programa no genere algún error, ya con los valores correctos se pasa a una prueba que determina si los valores podrían generar un triángulo dado que la suma de 2 lados tiene que ser mayor a el 3 lado, si con los valores proporcionados cumple esta regla se sigue clasificando entre los 3 tipos de triángulo, si no sirven los valores se tendrá que evidenciar un “no es un triángulo” y se le mostrará este resultado al usuario. Siguiendo con la clasificación dependerá de las reglas para clasificarlo, si los 3 lados son iguales será equilátero, si 2 lados son iguales será isósceles, y si ninguno de los lados es igual, pero cumple con la regla para formar un triángulo este será escaleno, y al final de este análisis se le generaría una respuesta al usuario con el tipo de triángulo que se forma con los valores ingresados al programa.

Esta organización de datos es la más óptima ya que permite determinar si primero los datos son válidos, segundo que el triángulo sea válido, también haciendo el recorrido por los diferentes triángulos haciendo la comparación de los valores ingresados siendo más fácil de reconocer el equilátero al ser los 3 valores iguales y así siguiendo con el isósceles con dos valores iguales y por último con el escaleno que ninguno de sus lados son iguales, pero sigue la regla de la formación de triángulos

Se podría pensar que obviar la regla de la creación de triángulos y dejarla como ultimo paso es correcto, pero yo analice ese factor y al probarlo seria imposible poder dar un resultado con veracidad, ya que al no dar esta regla como primera medida para clasificar podría dar resultados erróneos como por ejemplo la suma de 2 lados den igual al tercer lado y clasificar ese resultado como escaleno, ya que la regla estando al final no se podría ejecutar primero que las demás determinaciones de tipos de triangulo.

DIAGRAMA UML

