1. Создать интерфейс программы с использованием элементов Delphi, таких как меню, кнопки и текстовые поля для ввода и вывода данных.

2. Реализовать алгоритм для определения вида треугольника по его длинам и внутренним углам, используя конструкции языка Delphi.

3. Реализовать функционал для работы с текстовыми и и типизированными файлами, используя компоненты Delphi, такие как TMemo и TOpenDialog(?).

4. Реализовать графическое представление результатов работы программы с использованием компонентов Delphi, таких как TShape и TImage(?).

5. Реализовать тестирование и отладку программы, используя инструменты Delphi.

6. Разместить код программы в соответствующих секциях интерфейса и добавить обработчики событий для реализации функционала программы.

7. Наконец, собрать и скомпилировать программу в исполняемый файл, чтобы она могла быть запущена на компьютере пользователя.

Для определения вида треугольника по его длинам и внутренним углам можно использовать следующий алгоритм:

1. Запросить у пользователя длины трех сторон треугольника и внутренние углы.
2. Использовать теорему Пифагора, чтобы проверить, является ли треугольник прямоугольным. Если так, то треугольник является прямоугольным.
3. Использовать теорему косинусов для определения типа треугольника по внутренним углам.
4. Использовать теорему о равнобедренных треугольниках для определения, является ли треугольник равнобедренным.
5. Использовать теорему о равносторонних треугольниках для определения, является ли треугольник равносторонним.
6. В зависимости от полученных результатов, определить и вывести на экран вид треугольника (прямоугольный, равнобедренный, равносторонний или простой треугольник).
7. Вывести результат на экран или сохранить в файл, в зависимости от настроек программы.

Важно отметить, что данный алгоритм является общим и может быть модифицирован или расширен в зависимости от конкретной задачи и потребностей программы.

Пример кода алгоритма определения вида треугольника на современном delphi, у меня нет опыта со старой версией, не факт, что это будет работать на ней, нужно проверить

var

a, b, c: Double; // lengths of triangle sides

alpha, beta, gamma: Double; // angles of the triangle

begin

// get input from user for side lengths and angles

a := InputBox('Triangle sides', 'Enter length of side a:', '');

b := InputBox('Triangle sides', 'Enter length of side b:', '');

c := InputBox('Triangle sides', 'Enter length of side c:', '');

alpha := InputBox('Triangle angles', 'Enter angle alpha:', '');

beta := InputBox('Triangle angles', 'Enter angle beta:', '');

gamma := InputBox('Triangle angles', 'Enter angle gamma:', '');

// use Pythagorean theorem to check if triangle is right

if (a^2 + b^2 = c^2) or (a^2 + c^2 = b^2) or (b^2 + c^2 = a^2) then

ShowMessage('This is a right triangle.')

else

begin

// use cosine theorem to check if triangle is acute or obtuse

if (cos(alpha) > 0) and (cos(beta) > 0) and (cos(gamma) > 0) then

ShowMessage('This is an acute triangle.')

else if (cos(alpha) < 0) or (cos(beta) < 0) or (cos(gamma) < 0) then

ShowMessage('This is an obtuse triangle.')

else

begin

// check if triangle is isosceles or equilateral

if (a = b) and (b = c) then

ShowMessage('This is an equilateral triangle.')

else if (a = b) or (b = c) or (a = c) then

ShowMessage('This is an isosceles triangle.')

else

ShowMessage('This is a scalene triangle.');

end;

end;

end;