Lab Assignment 3

Plagiarism policies shall be strictly enforced

You are given the following requirements :

A shape can be defined as a series of 2-D points called vertices. It is a collection of edges where an edge links adjacent vertices. The shape will have at least 3 points and these points do not crossover themselves anywhere (i.e. 3 or more points would not lie in the same line). Each shape is closed (i.e. last point is linked to the first point). The center of a shape is the arithmetic average of all the shape points. The shape can be a triangle, rectangle, pentagon, hexagon or any polygon.

Given 2 shapes A and B, compute if A "crosses" B. A "crosses" B, if A has an edge such that one end point of that edge is within B and another end point is outside B (if any end point of this edge of A is on an edge of B, then assume that this end point of A's edge is within B). If both the end points of an edge of A are outside B, but this edge is passing through B, then ignore this case (on the basis of this condition you cannot say that A crosses B, as one end point must be within B and another must be outside B).·

Given two shapes A and B, compute how A "encircles" B in one of three ways, encoded as an integer
1 : The center of B is within A
0: otherwise, the two shapes have no intersection

A shape can be accepted as a String from the user. This string should list the points or vertices that makeup the shape (each point has x and y coordinates). For Example :
"0 0 0 1 1 1 1.0 0.0"
Parse this string to get the points that make up the shape:
point0 =(0,0)
point1 = (0,1)
point2 = (1,1)
point3 = (1.0,0.0)

Perform necessary checks on input string to ensure that it is a valid shape.

Keeping above requirements in mind, construct an OO design and code it in java such that the program accepts "n" number of shapes and compares first given shape with the other remaining shapes and print the results, one per line, if two shapes crosses or encircles each other.
Example :
Input shapes string :

a. "0 0 0 1 1 1 1 0"
b. "10 10 10 11 11 11 11 10"
c. "0.5 0.5 0.5 -10 1.5 0"
d. "0.5 0.5 0.75 0.75 0.75 0.2"

Output :

```
a crosses b :false
a crosses c :true
a crosses d:false
a encircles b:0
a encircles c:0
a encircles d:1
```