Starting with Program 3 …

Add a superclass named Polygon. Make Triangle a subclass of Polygon. Add another subclass named Quad.

Quad will deal with simple, convex quadrilaterals. The input file will list the vertices consecutively, so you do not have to do any complex calculations to decide which of the six segments are sides and which are diagonals. (How nice is that?)

For each quadrilateral, you will want to be print the name, the vertices, the side lengths, and the lengths of the diagonal:

* sideLength(p1,p2) will return the distance between two points. This will be called from the constructor.
* printName() will return the name of the quadrilateral as a nicely formatted string.
* listVertices() will return the vertices’ coordinates as a nicely formatted string.
* listSides() will return the quadrilateral’s sides’ lengths as a nicely formatted 3-line string.
* listDiagonals() will return the quadrilaterals diagonal’s lengths.
* Constructor accepts the name (String) and 4 points as vertices.
* Quad myQuad[0] = new Quad(name,p1, p2, p3,p4);
* The attributes of a triangle will be:
  + name (a string)
  + vert1, vert2, vert3, vert4 (each of type Point)
  + side1, side2, side3, side4 (each of type double)
  + diag1, diag2 (each of type double)
* Note that some methods should now be in the superclass. For example,
  + sideLength(p1,p2) will return the distance between two points. This will be called from the constructor.
* For each polygon, also indicate if any of the sides are of equal length. (Note that for these vertices (-2.161,-3.366),(2.161,3.366),(-5.83, 3.743), the lengths of the sides are actually not identical. However, we will state that anything within an epsilon of 0.0001 is “close enough”.)

***main***

* read in a line of data from **polyInput.txt**
* this data will be used to create three or four points
* throw an exception if any three points are collinear or there are duplicates
* each triangle will be placed in an arraylist of triangles (named myTri)
* each quadrilateral will be placed in an arraylist of quadrilaterals (named myQuad)
* each polygon’s data will be determined and printed
* this will be repeated for multiple polygons, until the end of the data file is reached

***sample output for each triangle***

The name of the triangle is Sally.

Vertices: (1.4, 2.3), (3.2, 4.1), (5.0, 6.9)

Side lengths: 5.84 3.33 2.55

No sides are of equal length.

***sample output for each quadrilateral***

The name of the triangle is Billy.

Vertices: (0,0),(0,1),(1,1),(1,0)

Side lengths: 1.00 1.00 1.00 1.00

Diagonals: 1.41 1.41

At least two sides are of equal length.

***sample input file (Note: The data file will have one space to separate items.)***

Sally 1.4 2.3 3.2 4.1 5.0 6.9

Billy 0 0 0 1 1 1 1 0

Program 4: This program will be due on Wednesday, March 21, by 11:59 PM.

The input file will be named **polyInput.txt** (Hard-code the name into your program.)

The output should be displayed on the screen and written to the file called **XXmyPoly.txt**

You will submit five files … **Prog5.java, Point.java, Triangle.java, Quad.java** *and* **Poly.java**