Problem 2: Raiders of the lost figures

Submit:

- Your source code to mooshak http://www.deei.fct.ualg.pt/~mooshak and
- Your report (within a zip file) to http://www.deei.fct.ualg.pt/POO/Entregas/

Up to Monday, November 09, 2015, 10H00 AM

Problem G in Mooshak POO 2015/2016

Raiders of the lost figures

A long time ago in a galaxy far, far away… an ancient and wise civilisation has found that certain triangles and rectangles give fortune and power to his owner. Fortune and power can be a quite dangerous combination in wrong hands, so the ancient and wise civilisation has hid these mysterious figures in the 2D Uniform Density Flat World. Obviously, in the 2D Uniform Density Flat World, too many triangles and rectangles exist, what makes any attempt to identify these mysterious figures a really hard job.

You belong to a team of clever raiders who had already discovered a lot on these mysterious figures. From the writings left by the ancient and wise civilisation, you know that both figures are defined by there vertices, three for triangles, four points for rectangles. Moreover, both figures have geometric properties that facilitate their identification. These properties are related with their perimeter and centroid (the same as the center of mass for uniform density). The centroid of a triangle is the intersection point of its medians (the lines joining each vertex with the midpoint of the opposite side), as shown in Fig. 1. More generally, the centroid of a finite set of points can be computed as the arithmetic mean of each coordinate of the points.

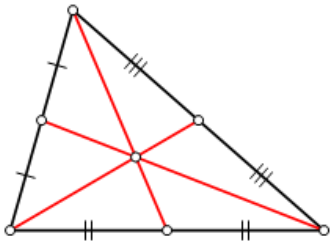

Fig. 1 – Computing the centroid

TASK

For the moment, your task is to use Object-Oriented Principles including unit tests, and test driven development to write a program that checks whether a ordered collection of points given as input define either a triangle or a rectangle and, when it does, computes both its perimeter and the centroid.

In particular, your program should adopt the following class Main (make sure this class is also submitted to mooshak together with your other classes):

```
import java.lang.reflect.*;
import java.util.*;

public class Main {
	public static  String capital(String s) {
		String res = s.toLowerCase();
		Character initial = Character.toUpperCase(res.charAt(0));
		StringBuilder sb = new StringBuilder(res);
		sb.setCharAt(0, initial);
		final String answer = sb.toString();
		return answer;
	}
	public static void main(String[] args) throws Exception {
		Scanner sc = new Scanner (System.in);
		Constructor<?> constructor;
		Class<?> cl;
		Polygon p;
		String s;
		String [] aos;
		while (sc.hasNextLine()) {
		    s = sc.nextLine();
		    aos = s.split(" ");
		    try {
				cl = Class.forName(capital(aos[0]));
				constructor = cl.getConstructor (new Class[]
                                                    {String.class} );
				p = (Polygon) constructor.newInstance(s);

				System.out.print(p.getClass().getName()+": ");
				System.out.print((int) p.perimeter()+" ");
				System.out.println(p.centroid());
		    }
		    catch (Exception e) {
				System.out.println("nwwalf");
		    }
		  }
		  sc.close();
	 }
}
```

Input

The input of your program will be a set of lines. The number of lines will never exceed 99999999999999999999999999999999999999999999999999999999.

Each line has the name of a geometric figure, i..e., either Triangle or Rectangle followed by an ordered sequence of space separated integers, representing the coordinates of points, possibly the vertices of a figure. Each point is represented by its x-coordinate and y-coordinate, both integers. Therefore, a triangle would be represented by the sequence: x1 y1 x2 y2 x3 y3 while a rectangle would be represented by x1 y1 x2 y2 x3 y3 x4 y4.

Output
The output will have as much lines as the number of figures in the input. If an input line represents a triangle or a rectangle, the corresponding output line should start by “Triangle: “or “Rectangle: “ respectively and without “ “, showed by its perimeter and centroid. Only the integer part of these data is outputted. Otherwise “nwwalf” (without “ “) should be outputted.

Therefore, each output line is either “nwwalf” or the name of the class representing the figure followed by a sequence of three separated integers. In the later case, the first integer represents the integer part of the perimeter, and the last two integers represent the integer part of x and y coordinates of the centroid.

Sample Input
Triangle 3 4 9 4 3 8
Triangle 0 0 1 1 2 2
Rectangle 2 13 6 17 8 15 4 11

Sample Output
Triangle: 17 5 5
nwwalf
Rectangle: 16 5 14