package com.tma;  
  
 /\*  
This program helps in finding the properties of the quadrilateral such as area, perimeter, angles, etc just by  
selecting 4 points of the quadrilateral and then entering their variables from X and Y, which means  
that we need 8 variables, all of them should be integers in order for the program to work efficiently.  
The program calculates many quadrilaterals, and to calculate each shape, the required data must be entered,  
 After each of them has been successfully calculated, a summary of the average properties of all the calculated  
shapes will be printed and the highest values will appear in the output panel in addition to indicating the  
coordinates of the chosen shape .  
\*/  
import java.util.Scanner;  
  
  
 public class Main {  
  
 public static void main(String[] args) {  
  
 // Definition of the variables that will be processed  
 double The\_Aggregate\_of\_Perimeters = 0.0, The\_Largest\_Perimeter = 0.0;  
 double The\_Entirety\_of\_Areas = 0.0, The\_Largest\_Area = 0.0;  
 double Biggest\_Angle\_of\_Quadrilateral = 0.0, Biggest\_Angle\_of\_Quadrilaterals = 0.0;  
  
 String Biggest\_Angle\_of\_Quadrilaterals\_coordinates = "";  
 String The\_Largest\_Area\_coordinates = "";  
 String The\_Largest\_Perimeter\_coordinates = "";  
  
  
 Scanner Record = new Scanner(System.*in*);  
 System.*out*.print("Write down The Quadrilaterals' number: ");  
 int The\_Amount\_of\_Quadrilaterals = Record.nextInt();  
  
  
 /\* it's required that we need to find the varierles of 15 quadrilaterals or larger so i used while loope to  
 make the program not process any input less than 15, but if it's 15 or larger the program will start to apply the operations  
 on all of them, also the while loope is important because the summarizaition won't be printed unless all of  
 the quadrilaterals are calculated so we used it to make the program calculate all the entered values then after  
 it ends, the summarizaition will be printed  
 \*/  
 while (The\_Amount\_of\_Quadrilaterals < 15) {  
 System.*out*.print("please, use a number equals or larger than 15 :");  
 The\_Amount\_of\_Quadrilaterals = Record.nextInt();  
 }  
  
 /\*  
 to start the operations the user must enter the x,y Axises of 4 points, which means the  
 program won't work unless 8 valuse are in the inputs:  
 \*/  
  
 for (int i = 1; i <= The\_Amount\_of\_Quadrilaterals; i++) {  
 System.*out*.print("Pick out 4 points of your Quadrilateral. put a space between each Axis ");  
 int x\_in\_1st\_point = Record.nextInt(), y\_in\_1st\_point = Record.nextInt();  
 int x\_in\_2nd\_point = Record.nextInt(), y\_in\_2nd\_point = Record.nextInt();  
 int x\_in\_3rd\_point = Record.nextInt(), y\_in\_3rd\_point = Record.nextInt();  
 int x\_in\_4th\_point = Record.nextInt(), y\_in\_4th\_point = Record.nextInt();  
  
 /\*  
 we need to find the length sides of the quadrilateral ( the space between two points).  
 (they are in the while loope because they will be applied on every quadrilateral )  
 \*/  
 double ER = Math.*sqrt*(Math.*pow*((x\_in\_1st\_point - x\_in\_2nd\_point), 2) + Math.*pow*((y\_in\_1st\_point - y\_in\_2nd\_point), 2));  
 double RT = Math.*sqrt*(Math.*pow*((x\_in\_2nd\_point - x\_in\_3rd\_point), 2) + Math.*pow*((y\_in\_2nd\_point - y\_in\_3rd\_point), 2));  
 double TY = Math.*sqrt*(Math.*pow*((x\_in\_3rd\_point - x\_in\_4th\_point), 2) + Math.*pow*((y\_in\_3rd\_point - y\_in\_4th\_point), 2));  
 double EY = Math.*sqrt*(Math.*pow*((x\_in\_1st\_point - x\_in\_4th\_point), 2) + Math.*pow*((y\_in\_1st\_point - y\_in\_4th\_point), 2));  
  
 double RY = Math.*sqrt*(Math.*pow*((x\_in\_2nd\_point - x\_in\_4th\_point), 2) + Math.*pow*((y\_in\_2nd\_point - y\_in\_4th\_point), 2));  
 double ET = Math.*sqrt*(Math.*pow*((x\_in\_1st\_point - x\_in\_3rd\_point), 2) + Math.*pow*((y\_in\_1st\_point - y\_in\_3rd\_point), 2));  
  
 /\*  
 we are doing here operations to find the angles of the quadrilateral,and convert them from the Radian form to Degree form .  
 (we will need both forms)  
 \*/  
 double First\_Angle = Math.*acos*((Math.*pow*(ER, 2) + Math.*pow*(EY, 2) - Math.*pow*(RY, 2)) / (2 \* ER \* EY));  
 double First\_Angle\_degree = First\_Angle \* 180 / Math.*PI*;  
 double Second\_Angle = Math.*acos*((Math.*pow*(ER, 2) + Math.*pow*(RT, 2) - Math.*pow*(ET, 2)) / (2 \* ER \* RT));  
 double Second\_Angle\_degree = Second\_Angle \* 180 / Math.*PI*;  
 double Third\_Angle = Math.*acos*((Math.*pow*(TY, 2) + Math.*pow*(RT, 2) - Math.*pow*(RY, 2)) / (2 \* TY \* RT));  
 double Third\_Angle\_degree = Third\_Angle \* 180 / Math.*PI*;  
 double Fourth\_Angle = Math.*acos*((Math.*pow*(EY, 2) + Math.*pow*(TY, 2) - Math.*pow*(ET, 2)) / (2 \* EY \* TY));  
 double Fourth\_Angle\_degree = Fourth\_Angle \* 180 / Math.*PI*;  
  
 /\*  
 the lengths we found before are now being added to find a perimeter  
 \*/  
  
 double Single\_Perimeter = ER + RT + TY + EY;  
 The\_Aggregate\_of\_Perimeters = The\_Aggregate\_of\_Perimeters + Single\_Perimeter;  
 if (Single\_Perimeter > The\_Largest\_Perimeter) {  
 The\_Largest\_Perimeter = Single\_Perimeter;  
 The\_Largest\_Perimeter\_coordinates = " The Coordinates of The Largest Perimeter are: E:(" + x\_in\_1st\_point + "," + y\_in\_1st\_point + ") R:(" + x\_in\_2nd\_point + "," + y\_in\_2nd\_point + ") T:(" + x\_in\_3rd\_point + "," + y\_in\_3rd\_point + ") Y:(" + x\_in\_4th\_point + "," + y\_in\_4th\_point + ")";  
 }  
 // to find an area we need a radian degree  
 double One\_Area = 0.5 \* ER \* EY \* Math.*sin*(First\_Angle)  
 + 0.5 \* RT \* TY \* Math.*sin*(Third\_Angle);  
 The\_Entirety\_of\_Areas = The\_Entirety\_of\_Areas + One\_Area;  
 if (One\_Area > The\_Largest\_Area) ;  
  
 // we collect the coordinates of the area so we can print them later  
 {  
 The\_Largest\_Area = One\_Area;  
 The\_Largest\_Area\_coordinates = "The Coordinates of The Largest Area: E:(" + x\_in\_1st\_point + "," + y\_in\_1st\_point + ") R:(" + x\_in\_2nd\_point + "," + y\_in\_2nd\_point + ") T:(" + x\_in\_3rd\_point + "," + y\_in\_3rd\_point + ") Y:(" + x\_in\_4th\_point + "," + y\_in\_4th\_point + ")";  
 }  
  
 /\*  
 each quadrilateral has 4 angles, if we want to know the largest angle we use the following operation  
 \*/  
 Biggest\_Angle\_of\_Quadrilateral = Math.*max*(First\_Angle, Math.*max*(Second\_Angle\_degree, Math.*max*(Third\_Angle\_degree, Fourth\_Angle\_degree)));  
  
 if (Biggest\_Angle\_of\_Quadrilateral > Biggest\_Angle\_of\_Quadrilaterals)  
 /\*  
 the 8 values that the user entered are now being collecting in point form (we will print them  
 with the outputs)  
 \*/ {  
 Biggest\_Angle\_of\_Quadrilaterals = Biggest\_Angle\_of\_Quadrilateral;  
 Biggest\_Angle\_of\_Quadrilaterals\_coordinates = "its Coordinates: E:(" + x\_in\_1st\_point + "," + y\_in\_1st\_point + ") R:(" + x\_in\_2nd\_point + "," + y\_in\_2nd\_point + ") T:(" + x\_in\_3rd\_point + "," + y\_in\_3rd\_point + ") Y:(" + x\_in\_4th\_point + "," + y\_in\_4th\_point + ")";  
 }  
  
 /\*  
 now we are printing the outputs that the user need to know.  
 \*/  
 System.*out*.println("The 4 sides area of the Quadrilateral are: ");  
 System.*out*.printf("(ER) Side: %.2f\n", ER);  
 System.*out*.printf("(RT) Side: %.2f\n", RT);  
 System.*out*.printf("(TY) Side: %.2f\n", TY);  
 System.*out*.printf("(EY) Side: %.2f\n", EY);  
 // the value of angles are required so we used the degree form to find them  
 System.*out*.println("The Inside Angles are: ");  
 System.*out*.printf("Angle E: %.2f\n", First\_Angle\_degree);  
 System.*out*.printf("Angle R: %.2f\n", Second\_Angle\_degree);  
 System.*out*.printf("Angle T: %.2f\n", Third\_Angle\_degree);  
 System.*out*.printf("Angle Y: %.2f\n", Fourth\_Angle\_degree);  
  
 System.*out*.printf("The perimeter is: %.2f\n", Single\_Perimeter);  
 System.*out*.printf("The Area is: %.2f\n", One\_Area);  
  
 // now all the above operations will be applied as many time as the user entered in the beggining  
 }  
 /\*  
 the next operations will give us a summarization of the outputs  
 \*/  
  
 double Intermediate\_Perimeter = The\_Aggregate\_of\_Perimeters / The\_Amount\_of\_Quadrilaterals;  
 System.*out*.printf("the Intermediate Perimeter of EVERY entered Quadrilateral: %.2f %n", Intermediate\_Perimeter);  
 double Intermediate\_Area = The\_Entirety\_of\_Areas / The\_Amount\_of\_Quadrilaterals;  
 System.*out*.printf("the Intermediate Area of ALL entered quadrilaterals: %.2f %n", Intermediate\_Area);  
 System.*out*.printf("the quadrilateral that has the largest perimeter is : %.2f %s %n", The\_Largest\_Perimeter, The\_Largest\_Perimeter\_coordinates);  
 System.*out*.printf("the quadrilateral that has the largest area is : %.2f %s %n", The\_Largest\_Area, The\_Largest\_Area\_coordinates);  
 System.*out*.printf(" the largest area: %.2f %s %n", The\_Largest\_Area, The\_Largest\_Area\_coordinates);  
 System.*out*.printf("the largest Inside angle of All the quadrilaterals is : %.2f %s %n", Biggest\_Angle\_of\_Quadrilaterals, Biggest\_Angle\_of\_Quadrilaterals\_coordinates);  
  
  
 }  
 }

صورة تحتوي على نص, لقطة شاشة, شاشة عرض, إلكترونيات

تم إنشاء الوصف تلقائياً

صورة تحتوي على نص, لقطة شاشة, شاشة عرض, إلكترونيات

تم إنشاء الوصف تلقائياً

صورة تحتوي على نص, شاشة عرض, لقطة شاشة, داخلي

تم إنشاء الوصف تلقائياً

صورة تحتوي على نص, لقطة شاشة, شاشة عرض, إلكترونيات

تم إنشاء الوصف تلقائياً

صورة تحتوي على نص, لقطة شاشة, شاشة عرض, إلكترونيات

تم إنشاء الوصف تلقائياً

صورة تحتوي على نص, لقطة شاشة, شاشة عرض, إلكترونيات

تم إنشاء الوصف تلقائياً

صورة تحتوي على نص, لقطة شاشة, إلكترونيات, شاشة عرض

تم إنشاء الوصف تلقائياً

صورة تحتوي على نص, لقطة شاشة, إلكترونيات, شاشة عرض

تم إنشاء الوصف تلقائياً

صورة تحتوي على نص, لقطة شاشة, شاشة عرض, إلكترونيات

تم إنشاء الوصف تلقائياً

صورة تحتوي على نص, لقطة شاشة, شاشة عرض, إلكترونيات

تم إنشاء الوصف تلقائياً

Asma Aljasem – 20414512

TM105 - 1305