```
/**
==============================================================================
|   Source code:  Triangle.java and TriangleTester.java
|        Author:  Alp Karavil
|    Student ID:  5827197
|    Assignment:  Program #2 - Triangle
|
|          Course:  COP 3337 (Programming II)
|         Section:  U00
|      Instructor:  William Feild
|      Due Date:   20 September 2018, by the beginning of class
|
| I hereby certify that this collective work is my own
| and none of it is the work of any other person or entity.
| Alp Karavil
|
|  Language:  Java
|  Compile/Run:
|  javac Triangle.java
     javac TriangleTester.java
|  java TriangleTester
|
|  +-----------------------------------------------------------------------
|
|  Description:  This program creates geometrical Triangle objects and prints
its properties.
|
|              This is possible through utilizing the Triangle class,
which utilizes the Point2D class the create 3 points in a 2d space which
represents a triangle.
|
|       Input:  The user is asked to input 3 coordinates, which is then
used to create the 3 corners of the triangle object.
|
|      Output:  This program will print side lengths, angles (in degrees),
perimeter, area, if its equilateral, if its a right triangle, centroid, and
incenter coordinates of the triangle object that is created by user input.
|
|     Process:  The program's steps are as follows:
|
|                   1.  Input is requested
|                   2.  Triangle object is created through user input
|                   3.  Output is printed through the tester class
|
|              No particular algorithms are used.  The Java BigDecimal
class and Java.math.RoundingMode is imported to round values (other than
angles) to 4 decimal place accuracy. Scanner is imported to get user input,
and the Point2D class is implemented to print general information about a
point's location in 2D space.
|
|   Required Features Not Included:  All required features are included,
along with extra credit methods for incenter and centroid of the triangle.
```

```
 |
 |  Known Bugs:  None; the program operates correctly.
 |
 *===================================================================================*/

//Point2D objects that are used as corners of the Triangle
import java.awt.geom.Point2D;
//BigDecimal is imported to do rounding calculations
import java.math.BigDecimal;
//RoundingMode is imported to do rounding along with BigDecimal methods
import java.math.RoundingMode;
//Scanning is imported to get user input for Triangle coordinates
import java.util.Scanner;

/**
 * This is a tester class for Triangle class. This class is able to get user
 * input for the coordinate locations, validate them as double values, then
 * create Triangle objects with the inputs. After creating Triangle objects
 * this class will print vital information about the triangle such as its
 * coordinates, side lengths, angles in degrees, its perimeter, and area. It
 * will also print boolean values on the characteristics of the triangle,
 * such as if it is Equilateral or Right-angled.
 *
 * This tester class also is able to print INCENTER and CENTROID coordinates
 * of the triangle for extra-credit.
 */
public class TriangleTester
{
  //Create Scanner object to be used by main and other static methods
  public static Scanner in = new Scanner(System.in);

  //Main method to ask for user input and print output
  public static void main(String[] args) {
    //Ask the user for input
    System.out.println("Enter the x, y coordinates of three points in this " +
          "order (x1, y1) (x2, y2) (x3, y3). \nSeparate each coordinate " +
          "input with a <return> character.");

    //Create a Triangle object with 6 user inputs for 3 coordinate points
    Triangle myTriangle = new Triangle(getValidatedDouble(),
          getValidatedDouble(), getValidatedDouble(),
          getValidatedDouble(), getValidatedDouble(), getValidatedDouble());

    //Print the location info of the Triangle object's corners
    System.out.println();
    printPointLocation(myTriangle.getCornerOne(), "1");
    printPointLocation(myTriangle.getCornerTwo(), "2");
    printPointLocation(myTriangle.getCornerThree(), "3");

    //Print length of all sides
    System.out.println();
    printTriangleSideLengths(myTriangle);
  //Print angle of each corner
  System.out.println();
  printTriangleAngles(myTriangle);

  //Print the perimeter of triangle
  System.out.println("\nThe perimeter of the triangle is "
        + roundDouble(myTriangle.getPerimeter(), 4) + " units.");

  //Print the area of triangle
  System.out.println("The area of the triangle is "
        + roundDouble(myTriangle.getArea(), 4) + " square units.");

  //Print equilateral triangle check value
  System.out.println("\nThis triangle is Equilateral?: "
        + myTriangle.isEquilateral());

  //Print right triangle check value
  System.out.println("This triangle is Right-angled?:  "
        + myTriangle.isRightTriangle());

  //Print location of incenter
  System.out.println("\nIncenter coordintes: ("
        + roundDouble(myTriangle.getIncenter().getX(), 4) + ", "
        + roundDouble(myTriangle.getIncenter().getY(), 4) + ")");

  //Print location of centroid
  System.out.println("Centroid coordinates: ("
        + roundDouble(myTriangle.getCentroid().getX(), 4) + ", "
        + roundDouble(myTriangle.getCentroid().getY(), 4) + ")");
}

/**
 * This method will round the inputted value to the desired decimal place.
 * @param roundValue value being rounded
 * @param decimalPlaces decimal places requested after rounding
 * @return rounded value
 */
private static double roundDouble(double roundValue, int decimalPlaces)
{
  //Crete a BigDecimal object so that value is precise
  BigDecimal convertedDouble = new BigDecimal(Double.toString(roundValue));

  //Round to the input specified by the user in the decimalPlaces input.
  convertedDouble = convertedDouble.setScale(decimalPlaces,
        RoundingMode.HALF_UP);

  //Return converted value as a double
  return convertedDouble.doubleValue();
}

/**
```

```
 * This method utilizes the scanner class to validate double inputs into
 * Scanner objects. If the input is not a double, the method will send an
 * error message and ask the user to input a double value.
 * @return double value by user input
 */
private static double getValidatedDouble()
{
  //While the current input is not a double value
  while (in.hasNextDouble() == false)
  {
    //Send error message
    System.out.println("Please enter a double value.");
    //Clear Scanner cache for next input
    in.next();
  }
  //If input is a double, store it.
  double input = in.nextDouble();
  //Return double value
  return input;
}

/**
 * Prints the location of an inputted Point2D (x, y) object along with its
 * name. Rounds to 4 decimals.
 * @param point Point2D object whose location is being requested
 * @param pointName Name of the point in String form
 */
private static void printPointLocation(Point2D point, String pointName)
{
  //Print the x and y values to 4 decimal place accuracy
  System.out.println("Point " + pointName + " coordinates: ("
        + roundDouble(point.getX(), 4) + ", "
        + roundDouble(point.getY(), 4) + ")");
}

/**
 * Prints the side length of the 3 sides of inputted Triangle object.
 * Rounds to 4 decimals.
 * @param triangleInput Triangle object
 */
private static void printTriangleSideLengths(Triangle triangleInput)
{
  //Print all side lengths to 4 decimal place accuracy
  System.out.println("Side 1 length: " +
        roundDouble(triangleInput.getSideLength1(), 4));

  System.out.println("Side 2 length: " +
        roundDouble(triangleInput.getSideLength2(), 4));

  System.out.println("Side 3 length: " +
        roundDouble(triangleInput.getSideLength3(), 4));
}
```

```
  /**
   * Prints all 3 angles (in degrees) of the triangle object inputted by the
   * user. Rounds to an integer value when being printed.
   * @param triangleInput Triangle object
   */
  private static void printTriangleAngles(Triangle triangleInput)
  {
    //Print all triangle angles to 0 decimal place accuracy (as an integer)
    System.out.println("Angle 1 " +
          roundDouble(triangleInput.getAngle(1), 0)
          + " degrees");

    System.out.println("Angle 2 " +
          roundDouble(triangleInput.getAngle(2), 0)
          + " degrees");

    System.out.println("Angle 3 " +
          roundDouble(triangleInput.getAngle(3), 0)
          + " degrees");
  }
}
```