

Name : Ramzy Izza Wardhana
NIM : 21/472698/PA/20322
Class : IUP CS 1

Assignment 11 – Geometry Problem (Convex Hull)

Complete Source Code Zipped File:

https://drive.google.com/file/d/1U9FTTrfgC_dfgXFcQ1IjkPNX3eh9PpDd8/view?usp=sharing

1. Add getConvexHullArea method in the Geometry class that takes input of the convex hull of a set of points S, CH(S), and return the area of the area enclosed by the convex hull.

Output:

```
Convex Hull:
(0.0, 4.0)
(0.5, 1.5)
(4.75, 2.1)
(5.6, 5.8)
(1.75, 6.75)
Area of Given Convex Hull is : 21.182499999999997
```

```
//Get the area of convex hull
Geometry.getConvexHullArea(X, Y);
```

```
public static void getConvexHullArea(double X[], double Y[]){
    double area = 0.0; int n = X.length; int i = 0; int j = n - 1;

    while(i != n){
        area += (X[j] + X[i]) * (Y[j] - Y[i]);
        j = i;
        i++;
    }

    double result = Math.abs(area / 2.0);
    System.out.println("Area of Given Convex Hull is : " + result + "\n");
}
```

2. Add getConvexHullLength method in the Geometry class that takes input of the convex hull of a set of points S, CH(S), and return the perimeter of the convex hull

Output:

```
Convex Hull:
(0.0, 4.0)
(0.5, 1.5)
(4.75, 2.1)
(5.6, 5.8)
(1.75, 6.75)
Perimeter / Length of Given Convex hull is: 17.863110812124233
```

```
//Get the perimeter of convex hull
Geometry.getConvexHullLength(hull);
```

```

public static void getConvexHullLength(Point[] P){

    double perimeter = 0; int i = 0;

    while(i != P.length - 1){
        perimeter += distance(P[i], P[i + 1]);
        i++;
    }

    perimeter += distance(P[0], P[P.length - 1]);

    System.out.println("Perimeter / Length of Given Convex hull is: " + perimeter + "\n");
}

```

Point.java

```

package GeometryProblem;

public class Point implements Comparable<Point>{

    double x, y;

    public Point(){
        y = 0.0;
        x = 0.0;
    }

    public Point(double _x, double _y){
        x = _x;
        y = _y;
    }

    public int compareTo(Point other) {
        double EPS = 1e-9;
        double tmp;

        if(Math.abs(x - other.x) > EPS){
            tmp = x - other.x;
            if(tmp > EPS)
                return 1;
            else
                return -1;
        }
    }
}

```

```

        else if(Math.abs(y - other.y) > EPS){
            tmp = y - other.y;
            if(tmp > EPS)
                return 1;
            else
                return -1;
        }
        else{
            return 0;
        }
    }

    public String toString(){
        return "(" + x + ", " + y + ")";
    }
}

```

Geometry.java

```

package GeometryProblem;

import java.util.Arrays;

public class Geometry {

    public static double cross(Point O, Point A, Point B){
        return (A.x - O.x) * (B.y - O.y) - (A.y - O.y) * (B.x - O.x);
    }

    public static boolean ccw(Point p, Point q, Point r){
        return cross(p, q, r) > 0;
    }

    public static Point[] convexHull(Point[] P){
        if(P.length > 2){
            int n = P.length, upperLength = 0, lowerLength = 0;
            Point[] lowerhull = new Point[n];
            Point[] upperHull = new Point[n];

            Arrays.sort(P);

            lowerhull[0] = P[0];
            lowerhull[1] = P[1];

```

```

        lowerLength = 2;
        for(int i = 2; i < n; i++){
            while(lowerLength >= 2 && !ccw(lowerhull[lowerLength - 2],
lowerhull[lowerLength - 1], P[i])){
                lowerLength--;
            }
            lowerhull[lowerLength] = P[i];
            lowerLength++;
        }
        upperHull[0] = P[n-1];
        upperHull[1] = P[n-2];
        upperLength = 2;
        for(int i = n - 3; i >= 0; i--){
            while(upperLength >= 2 && !ccw(upperHull[upperLength - 2],
upperHull[upperLength - 1], P[i])){
                upperLength--;
            }
            upperHull[upperLength] = P[i];
            upperLength++;
        }
        Point[] result = new Point[2 * n];
        int t = 0;
        for(int i = 0; i < lowerLength - 1; i++){
            result[t] = lowerhull[i];
            t++;
        }
        for(int i = 0; i < upperLength - 1; i++){
            result[t] = upperHull[i];
            t++;
        }

        if(t > 1)
            result = Arrays.copyOfRange(result, 0, t);
        return result;
    }

    else if(P.length <= 2)
        return P.clone();
    else
        return null;
}

public static double distance(Point A, Point B){
    return Math.sqrt((A.x - B.x) * (A.x - B.x) + (A.y - B.y) * (A.y - B.y));
}

```

```

public static void getConvexHullArea(double X[], double Y[]){
    double area = 0.0; int n = X.length; int i = 0; int j = n - 1;

    while(i != n){
        area += (X[j] + X[i]) * (Y[j] - Y[i]);
        j = i;
        i++;
    }

    double result = Math.abs(area / 2.0);
    System.out.println("Area of Given Convex Hull is : " + result + "\n");
}

public static void getConvexHullLength(Point[] P){

    double perimeter = 0; int i = 0;

    while(i != P.length - 1){
        perimeter += distance(P[i], P[i + 1]);
        i++;
    }

    perimeter += distance(P[0], P[P.length - 1]);

    System.out.println("Perimeter / Length of Given Convex hull is: " +
perimeter + "\n");
}
}

```

Main.java

```

package GeometryProblem;

public class Main{
    public static void main(String[] args){

        Point[] points = new Point[7];
        points[0] = new Point(3.6, 4.5);
        points[1] = new Point(0, 4);
        points[2] = new Point(1.75, 6.75);
        points[3] = new Point(2.4, 3);
        points[4] = new Point(5.6, 5.8);
        points[5] = new Point(0.5, 1.5);
        points[6] = new Point(4.75, 2.1);
    }
}

```

```
Point[] hull = Geometry.convexHull(points);
double X[] = new double[hull.length];
double Y[] = new double[hull.length];
for(int i = 0 ; i < hull.length; i++){
    X[i] = hull[i].x;
    Y[i] = hull[i].y;
}

//Print the convex hull
System.out.println("Convex Hull: ");
for(int i = 0; i < hull.length; i++){
    if(hull[i] != null)
        System.out.println(hull[i]);
}
//Get the area of convex hull
Geometry.getConvexHullArea(X, Y);
//Get the perimeter of convex hull
Geometry.getConvexHullLength(hull);
}
}
```