Nama : Andyan Yogawardhana

NIM : 21/482180/PA/21030

Kelas : KOMB1

Tugas 11 – Convex Hull

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.
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.

Source Code

```java
import java.util.Arrays;

public class ConvexHull {
    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);

        System.out.println("Convex Hull");
        for (int i = 0; i < hull.length; i++) {
            if (hull[i] != null) {
                System.out.println(hull[i]);
            }
        }

        Geometry.getConvexHullArea(hull);
        Geometry.getConvexHullLength(hull);
    }
}

class Point implements Comparable<Point> {
    double x, y;

    public Point() {
```

```java
        x = 0.0;
        y = 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 + ")";
    }
}

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);
    }

    // return true if pqr turns left (counter-clockwise)
    public static boolean ccw(Point p, Point q, Point r) {
        return cross(p, q, r) > 0;
    }

```

```
79      public static Point[] convexHull(Point[] P) {
80          if (P.length > 2) {
81              int n = P.length, upperLength = 0, lowerLength = 0;
82              Point[] lowerHull = new Point[n];
83              Point[] upperHull = new Point[n];
84
85              Arrays.sort(P);
86
87              // build lower hull first
88              lowerHull[0] = P[0];
89              lowerHull[1] = P[1];
90              lowerLength = 2;
91              for (int i = 2; i < n; i++) {
92                  while (lowerLength >= 2 &&
    !ccw(lowerHull[lowerLength - 2], lowerHull[lowerLength - 1], P[i]))
    {
93                      lowerLength--;
94                  }
95                  lowerHull[lowerLength] = P[i];
96                  lowerLength++;
97              }
98
99              // build upper hull
100                 upperHull[0] = P[n - 1];
101                 upperHull[1] = P[n - 2];
102                 upperLength = 2;
103                 for (int i = n - 3; i >= 0; i--) {
104                     while (upperLength >= 2 &&
    !ccw(upperHull[upperLength - 2], upperHull[upperLength - 1], P[i]))
    {
105                         upperLength--;
106                     }
107                     upperHull[upperLength] = P[i];
108                     upperLength++;
109                 }
110
111                 // combine lower hull and upper hull
112                 Point[] result = new Point[2 * n];
113                 int t = 0;
114                 for (int i = 0; i < lowerLength - 1; i++) {
115                     result[t] = lowerHull[i];
116                     t++;
117                 }
118                 for (int i = 0; i < upperLength - 1; i++) {
119                     result[t] = upperHull[i];
120                     t++;
121                 }
122
```

```
123                 result = Arrays.copyOfRange(result, 0, t);
124
125                 return result;
126             } else if (P.length <= 2) {
127                 return P.clone();
128             } else {
129                 return null;
130             }
131         }
132
133         // 1
134         public static void getConvexHullArea(Point[] P) {
135             double result = 0;
136
137             // deklarasi point koordinat pusat (0, 0)
138             Point O = new Point();
139
140             // menghitung luas dengan menjumlahkan cross product
    titik-titik yang terhubung
141             // (metode segitiga)
142             for (int i = 0; i < P.length - 1; i++) {
143                 result += cross(O, P[i], P[i + 1]);
144             }
145             // untuk sisi yang menghubungkan convex point pertama
    dan terakhir
146             result += cross(O, P[P.length - 1], P[0]);
147
148             // hasil akhir dibagi dua agar sesuai dengan
149             // rumus metode luas segitiga (|a x b|) / 2
150             System.out.println("Convex Hull Area   = " + result /
    2);
151         }
152
153         // 2
154         public static void getConvexHullLength(Point[] P) {
155             double result = 0;
156
157             // menghitung keliling dengan menambahkan setiap sisi
    convex hull
158             for (int i = 0; i < P.length - 1; i++) {
159                 result += Math.sqrt(Math.pow(P[i].x - P[i + 1].x,
    2) + Math.pow(P[i].y - P[i + 1].y, 2));
160             }
161             // untuk sisi yang menghubungkan convex point pertama
    dan terakhir
162             result += Math.sqrt(Math.pow(P[0].x - P[P.length -
    1].x, 2) + Math.pow(P[0].y - P[P.length - 1].y, 2));
163
```

```
164             System.out.println("Convex Hull Length = " + result);
165         }
166     }
```

Output Terminal

```
(0.0, 4.0)
(0.5, 1.5)
(4.75, 2.1)
(5.6, 5.8)
(1.75, 6.75)
Convex Hull Area   = 21.1825
Convex Hull Length = 17.863110812124233
```