Assignment 2 Intersection of Two Segments

**package** geometry; //under the package of geometry

**import** java.awt.geom.Point2D; //invoke a geometry package Point2D that can generate points' coordinate

**import** java.lang.Math; //invoke java Math package

**public** **class** Segment{ //declare a class name Segment

**public** Point2D startPoint; //declare a class variable named startPoint in Point2D type

**public** Point2D endPoint; //declare a class variable named endPoint in Point2D type

**public** **double** k; //declare a k variable in type double to represent the slope of a segment

**public** **double** b; //declare a b variable in type double to represent the intercept of a segment with vertical coordinate

**public** **void** setK() { //declare a method with no return value named setK

**if** (endPoint.getX() == startPoint.getX()) { //if the horizontal coordinate of the startPoint equals to that of endPoint/

**this**.k = Double.***MAX\_VALUE***; //k can be the largest positive finite value

}

**else** { //if not

**this**.k = (endPoint.getY() - startPoint.getY())/(endPoint.getX() - startPoint.getX()); //calculate k with slope formula

}

}

**public** **void** setB(){ //declare a method with no return value named setB

**this**.b = endPoint.getY() - **this**.k\*endPoint.getX(); //the formula of b is b=y-kx

}

**public** Segment (Point2D ptStart, Point2D ptEnd){ //declare a constructor with ptStart and ptEnd in Point2D type

**this**.startPoint = ptStart; //assign Point2D function to ptStart

**this**.endPoint = ptEnd; //assign Point2D function to ptEnd

}

**public** **double** calculateLength(){ //declare a method to calculate the length of segments

**double** seg\_length; //declare a local variable in double type named seg\_length

seg\_length = Math.*sqrt*(Math.*pow*(**this**.startPoint.getX() - **this**.endPoint.getX(), 2) + Math.*pow*(**this**.startPoint.getY() - **this**.endPoint.getY(), 2)); //calculate the length of a segment

**return** seg\_length; //return the calculate result

}

**public** **boolean** isLongerThan(Segment a){ //declare a boolean method to compare the length of segments

**double** length\_a = a.calculateLength(); //declare a local variable length\_a in double type, call calculateLength() function

**double** length\_b = **this**.calculateLength(); //declare a local variable length\_b in double type, call calculateLength() function

**if** (length\_a > length\_b) //if length\_a > length\_b

**return** **true**; //return it is true

**else** //if not

**return** **false**; //return it is false

}

**public** **static** Point2D calculateIntersection(Segment a, Segment b){ //public a static method named calculateIntersection, call the methods of two segments

a.setK(); //declare the k value of segment A

a.setB(); //declare the b value of segment A

b.setK(); //declare the k value of segment B

b.setB(); //declare the b value of segment B

**double** x = 0; //initialize variable x in double type, assign its value as 0, x is the horizontal coordinate of this hypothetical intersection

**double** y = 0; //initialize variable y in double type, assign its value as 0, y is the vertical coordinate of this hypothetical intersection

**if** (a.k == Double.***MAX\_VALUE***) { //if the slope k of segment A can be infinite

x = a.startPoint.getX();

y = b.k\*x+b.b;

}

**else** **if** (b.k == Double.***MAX\_VALUE***) { //if the slope k of segment B can be infinite

x = b.startPoint.getX();

y = a.k\*x+a.b;

}

**else** { //if not

x = (b.b-a.b)/(a.k - b.k); //use k and b to calculate intersection's coordinate

y = b.k\*x+b.b; //use k and b to calculate intersection's coordinate

}

Point2D intersection = **new** Point2D.Double(x,y); //declare a local variable intersection in Point2D type, assign (x, y) to this intersection

**return** intersection; //return this intersection variable

}

**public** **static** **boolean** isOn(Segment a, Point2D b) { //declare a static method in boolean called isOn, call functions of Segment a and Point2D

**double** x1,y1,x2,y2; //declare four local variables

x1 = a.startPoint.getX(); //assign values to x1. It is the horizontal coordinate of start point on segment a

y1 = a.startPoint.getY(); //assign values to y1. It is the vertical coordinate of start point on segment a

x2 = a.endPoint.getX(); //assign values to x2. It is the horizontal coordinate of end point on segment a

y2 = a.endPoint.getY(); //assign values to y2. It is the vertical coordinate of end point on segment a

**double** bigAx = x1 > x2 ? x1 : x2; //compare x1 and x2, the bigger one is bigAx

**double** smallAx = x1 > x2 ? x2 : x1; //compare x1 and x2, the smaller one is bigAx

**double** bigAy = y1 > y2 ? y1 : y2; //compare y1 and y2, the bigger one is bigAy

**double** smallAy = y1 > y2 ? y2 : y1; //compare y1 and y2, the smaller one is bigAx

**if** (b.getX() <= bigAx && b.getX() >= smallAx && b.getY() <= bigAy && b.getY() >= smallAy) { //跨立实验

**return** **true**;

}

**else** **return** **false**;

}

**public** **static** **void** main(String[]args){

Point2D pt1 = **new** Point2D.Double(40, 10); //declare a local variable pt1 under main method with type Point2D. Instantiate it with coordinate values

Point2D pt2 = **new** Point2D.Double(40, 90); //declare a local variable pt2 under main method with type Point2D. Instantiate it with coordinate values

Point2D pt3 = **new** Point2D.Double(15, 15); //declare a local variable pt3 under main method with type Point2D. Instantiate it with coordinate values

Point2D pt4 = **new** Point2D.Double(95, 95); //declare a local variable pt4 under main method with type Point2D. Instantiate it with coordinate values

Segment segmentA = **new** Segment (pt1, pt2); //declare a local variable segment A under main method with type Segment. Instantiate it with coordinates

Segment segmentB = **new** Segment (pt3, pt4); //declare a local variable segment B under main method with type Segment. Instantiate it with coordinates

segmentA.setK(); //declare a method to calculate the k for segment A

segmentB.setK(); //declare a method to calculate the k for segment B

**if** (segmentA.k == segmentB.k && (pt1 != pt3 || pt2 != pt4)) { //if the two slopes equal to each other, it means these two segments are parallel

System.***out***.println("There is no intersection"); //print out "There is no intersection"

**return**; //return the if condition

}

**boolean** result = segmentA.isLongerThan(segmentB); //declare a local variable named result with type boolean. Instantiate it.

**if**(result)

System.***out***.println("Segment A is longer than Segment B.");

**else**

System.***out***.println("Segment B is longer than Segment A.");

Point2D resultPoint = Segment.*calculateIntersection*(segmentA, segmentB); //declare a local variable resultPoint with type Point2D. Call function calculateIntersection

**if** (Segment.*isOn*(segmentA, resultPoint) && Segment.*isOn*(segmentB, resultPoint)) { //call "isOn" method in this if condition to test if the intersection is on both segments

System.***out***.println("The intersection of these two segments is: (" + resultPoint.getX() + ", " + resultPoint.getY() + ")");

**return**;

}

**else** {

System.***out***.println("There is no intersection");

**return**;

}

}

}

Segment A is longer than Segment B.

The intersection of these two segments is: (40.0, 40.0)