public class PerimeterAssignmentRunner {

public int getNumPoints (Shape s){

int totalPt = 0;

for (Point currPt: s.getPoints()){

totalPt = totalPt + 1;

}

return totalPt;

}

public double getPerimeter (Shape s) {

// Start with totalPerim = 0

double totalPerim = 0.0;

// Start wth prevPt = the last point

Point prevPt = s.getLastPoint();

// For each point currPt in the shape,

for (Point currPt : s.getPoints()) {

// Find distance from prevPt point to currPt

double currDist = prevPt.distance(currPt);

// Update totalPerim by currDist

totalPerim = totalPerim + currDist;

// Update prevPt to be currPt

prevPt = currPt;

}

// totalPerim is the answer

return totalPerim;

}

public void testPerimeter () {

FileResource fr = new FileResource();

Shape s = new Shape(fr);

double length = getPerimeter(s);

System.out.println("perimeter = " + length);

int numPoints = getNumPoints(s);

System.out.println("points = " + numPoints);

double average = getAverageLength(s);

System.out.println("average = " + average);

double largeX = getLargestX(s);

System.out.println("largest x = " + largeX);

double longestSide = getLargestSide(s);

System.out.println("largest side = " + longestSide);

}

public double getAverageLength(Shape s) {

double perimeter = getPerimeter(s);

int points = getNumPoints(s);

double average = perimeter / points;

return average;

//1. get the perimeter (add up all side lengths) -- save into variable

//2. divide by the number of points -- save into variable

//3. divide the two variables to get the average -- save in variable

//4. return the average variable

}

public double getLargestSide(Shape s) {

Point prevPt = s.getLastPoint();

double prevLgth = 0;

double currLgth = 0;

for(Point currPt : s.getPoints()){

currLgth = prevPt.distance(currPt);

if(currLgth > prevLgth){

prevLgth = currLgth;

}

}

return prevLgth;

//1. create a variable to hold the value that will be returned

//2. loop from point to point

//3. get distance from previous point to current point -- set to variable currentSide

//4. if currentSide is greater that largestSide, largestSide = currentSide

//5. return currentsSide

}

public double getLargestX(Shape s) {

Point prevPoint = s.getLastPoint();

double prevPointX = prevPoint.getX();

double currPointX = 0;

for(Point currPoint: s.getPoints()){

currPointX= currPoint.getX();

if(currPointX > prevPointX){

prevPointX = currPointX;

}

}

return prevPointX;

}

//return larger x that is stored in this variable

// 1. create variable to hold value that will be returned - largestX

// 3. loop points / getPoints?

// 4. get coordinates (specifically x)?

// 5. if currentX > largestX, then largestX = currentX

// 6. return largest

//Assignment 2

public double getLargestPerimeterMultipleFiles() {

DirectoryResource dr = new DirectoryResource();

double lgPeri = 0;

for(File f: dr.selectedFiles()){

FileResource fr = new FileResource(f);

Shape s = new Shape(fr);

double currPerimeter = getPerimeter(s);

if(currPerimeter > lgPeri){

lgPeri = currPerimeter;

}

}

return lgPeri;

}

public String getFileWithLargestPerimeter() {

DirectoryResource dr = new DirectoryResource();

File lgPeri = null;

double largePeri = 0;

for(File f : dr.selectedFiles()){

FileResource fr = new FileResource(f);

Shape s = new Shape(fr);

double currPerimeter = getPerimeter(s);

if(currPerimeter > largePeri){

largePeri = currPerimeter;

lgPeri = f;

}

}

return lgPeri.getName();

}

public void testPerimeterMultipleFiles() {

FileResource fr = new FileResource();

System.out.println("Largest perimeter is = " + getLargestPerimeterMultipleFiles());

System.out.println("File with largest perimeter is " + getFileWithLargestPerimeter());

}

public void testFileWithLargestPerimeter() {

FileResource fr = new FileResource();

System.out.println(getFileWithLargestPerimeter());

}

// This method creates a triangle that you can use to test your other methods

public void triangle(){

Shape triangle = new Shape();

triangle.addPoint(new Point(0,0));

triangle.addPoint(new Point(6,0));

triangle.addPoint(new Point(3,6));

for (Point p : triangle.getPoints()){

System.out.println(p);

}

double peri = getPerimeter(triangle);

System.out.println("perimeter = " + peri);

}

// This method prints names of all files in a chosen folder that you can use to test your other methods

public void printFileNames() {

DirectoryResource dr = new DirectoryResource();

for (File f : dr.selectedFiles()) {

System.out.println(f);

}

}

public static void main (String[] args) {

PerimeterAssignmentRunner pr = new PerimeterAssignmentRunner();

pr.testPerimeter();

pr.testPerimeterMultipleFiles();

}

}