// ------------------------------------------

// Author: Lauren Escobedo

// Assignment: Chapter 9 Problem 9.8

// Date: 02/22/2023

// Language: Java

// File Name: Quadrilateral.java

// Description: Exercise 9.8

// - Quadrilaterals

// ------------------------------------------

// Super class to the shapes

public class Quadrilateral {

// Shapes points

protected double x1, y1, x2, y2, x3, y3, x4, y4;

// Constructor - sets points for the shapes

public Quadrilateral(double x1, double y1,

double x2, double y2,

double x3, double y3,

double x4, double y4) {

this.x1 = x1;

this.y1 = y1;

this.x2 = x2;

this.y2 = y2;

this.x3 = x3;

this.y3 = y3;

this.x4 = x4;

this.y4 = y4;

}

// Returns string representation of object

@Override

public String toString() {

return "\nPoints:\n" + "x1 = " + x1 +

"\t y1 = " + y1 +

"\nx2 = " + x2 +

"\t y2 = " + y2 +

"\nx3 = " + x3 +

"\t y3 = " + y3 +

"\nx4 = " + x4 +

"\t y4 = " + y4 + "\n\n";

}

}

// ------------------------------------------

// Author: Lauren Escobedo

// Assignment: Chapter 9 Problem 9.8

// Date: 02/22/2023

// Language: Java

// File Name: Trapezoid.java

// Description: Exercise 9.8

// - Quadrilaterals

// ------------------------------------------

// Subclass trapezoid, derived from quadrilateral

public class Trapezoid extends Quadrilateral {

// Instance variables

protected double height, base1, base2;

// Constructor

public Trapezoid(double x1, double y1, double x2, double y2, double x3, double y3, double x4, double y4) {

// Utilize superclasses constructor for points

super(x1, y1, x2, y2, x3, y3, x4, y4);

height = Math.abs(x1 - x2);

base1 = y2 - y3;

base2 = y1 - y4;

}

// Returns area

public double area() {

return Math.abs(0.5 \* (base1 + base2) \* height);

}

// Display the shape

@Override

public String toString() {

String output = String.format("\nTrapezoid:\n %sHeight: %2.2f \nBase1: %2.2f \nBase2: %2.2f \nArea: %2.2f\n",

super.toString(), height, base1, base2, area());

return output;

}

}

// ------------------------------------------

// Author: Lauren Escobedo

// Assignment: Chapter 9 Problem 9.8

// Date: 02/22/2023

// Language: Java

// File Name: Parallelogram.java

// Description: Exercise 9.8

// - Quadrilaterals

// ------------------------------------------

// Subclass parallelogram, derived from trapezoid

public class Parallelogram extends Quadrilateral {

// Instance variables

private double height, width;

// Constructor

public Parallelogram(double x1, double y1, double x2, double y2, double x3, double y3, double x4, double y4) {

// Utilize superclasses constructor for points

super(x1, y1, x2, y2, x3, y3, x4, y4);

height = Math.abs(x1 - x2);

width = y2 - y3;

}

// Returns area

public double area() {

return height \* width;

}

// Displays height and width

public String displayHW() {

return "Height: " + height + "\nWidth: " + width;

}

// Display the shape

@Override

public String toString() {

return "\nParallelogram: \n" + super.toString() + displayHW() + "\nArea:" + area() + "\n";

}

}

// ------------------------------------------

// Author: Lauren Escobedo

// Assignment: Chapter 9 Problem 9.8

// Date: 02/22/2023

// Language: Java

// File Name: Rectangle.java

// Description: Exercise 9.8

// - Quadrilaterals

// ------------------------------------------

// Subclass rectange, derived from quadrilateral

public class Rectangle extends Quadrilateral {

// Instance variables

private double height;

protected double width;

// Constructor

public Rectangle(double x1, double y1, double x2, double y2, double x3, double y3, double x4, double y4) {

// Utilize superclasses constructor for points

super(x1, y1, x2, y2, x3, y3, x4, y4);

width = Math.sqrt(Math.pow((x2 - x1), 2) + Math.pow((y2 - y1), 2));

height = Math.sqrt(Math.pow((x4 - x1), 2) + Math.pow((y4 - y1), 2));

}

// Returns area

private double area() {

return height \* width;

}

// Display the shape

@Override

public String toString() {

String output = String.format("\nRectangle: \n%sHeight: %2.2f\nLength: %2.2f\nArea:%2.2f\n", super.toString(),

height, width, area());

return output;

}

}

// ------------------------------------------

// Author: Lauren Escobedo

// Assignment: Chapter 9 Problem 9.8

// Date: 02/22/2023

// Language: Java

// File Name: Square.java

// Description: Exercise 9.8

// - Quadrilaterals

// ------------------------------------------

// Subclass square, derived from rectangle

public class Square extends Quadrilateral {

private double width;

public Square(double x1, double y1, double x2, double y2, double x3, double y3, double x4, double y4) {

// Utilize superclasses constructor for points

super(x1, y1, x2, y2, x3, y3, x4, y4);

this.width = Math.sqrt(Math.pow((x2 - x1), 2) + Math.pow((y2 - y1), 2));

}

private double area() {

return this.width \* 4;

}

@Override

public String toString() {

String output = String.format("\nSquare: \n%sWidth: %2.2f\nArea:%2.2f\n", super.toString(), width, area());

return output;

}

}

// ------------------------------------------

// Author: Lauren Escobedo

// Assignment: Chapter 9 Problem 9.8

// Date: 02/22/2023

// Language: Java

// File Name: Execise9\_8.java

// Description: Exercise 9.8

// - Quadrilaterals

// ------------------------------------------

// Driver class

public class Exercise9\_8 {

public static void main(String[] args) throws Exception {

Trapezoid trapezoid = new Trapezoid(1.2, 4.5, 5.6, 3.8, 4.2, 2.4, 1.2, 4.1);

System.out.println(trapezoid.toString());

Parallelogram parallelogram = new Parallelogram(4.0, 2.0, 7.0, 10.0, 2.0, 7.0, 7.0, 1.0);

System.out.println(parallelogram.toString());

Rectangle rectangle = new Rectangle(7.0, 4.0, 3.0, 4.0, 3.0, 2.0, 1.0, 2.0);

System.out.println(rectangle.toString());

Square square = new Square(4.0, 0.0, 8.0, 0.0, 8.0, 4.0, 4.0, 4.0);

System.out.println(square.toString());

System.exit(0);

}

}

A picture containing text

Description automatically generated