

LAB EXERCISE

RECTANGLE

Background:

1. Professional programmers carefully design the classes they need before any coding is done. With well-designed classes, programming is much easier and the program has fewer bugs. Object-oriented design consists of deciding what classes are needed, what data they will hold, and how they will behave. All these decisions are documented (written up) and then examined. If something doesn't look right, it is fixed before any programming is done.
2. The specifications of a class that models a rectangular shape would be:

Variables

```
private double myX;           // the x coordinate of the rectangle
private double myY;           // the y coordinate of the rectangle
private double myWidth;       // the width of the rectangle
private double myHeight;      // the height of the rectangle

// Creates a 500 x 500 SketchPad with a DrawingTool, pen, that is used
// to display Rectangle objects. The Drawingtool is declared static
// so that multiple Rectangle objects can be drawn on the Sketchpad
// at the same time.
private static DrawingTool pen =
    new DrawingTool(new SketchPad(500, 500));
```

Constructors

```
// Creates a default instance of a Rectangle object with all dimensions
// set to zero.
Rectangle()

// Creates a new instance of a Rectangle object with the left and right
// edges of the rectangle at x and x + width. The top and bottom edges
// are at y and y + height.
Rectangle(double x, double y, double width, double height)
```

Methods

```
// calculates and returns the perimeter of the rectangle
public double getPerimeter()

// Calculates and returns the are of the rectangle.
public double getArea()

// Draws a new instance of a Rectangle object with the left and right
// edges of the rectangle at x and x + width. The top and bottom edges
// are at y and y + height.
public void draw()
```

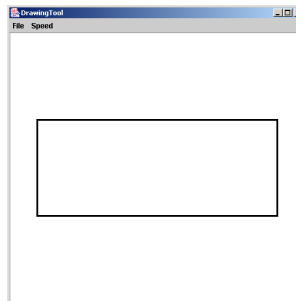
Assignment:

1. Implement a `Rectangle` class with the following properties.
 - a. A `Rectangle` object is specified in the constructor with the left and right edges of the rectangle at `x` and `x + width`. The top and bottom edges are at `y` and `y + height`.
 - b. A method `getPerimeter` calculates and returns the perimeter of the `Rectangle`.
 - c. A method `getArea` calculates and returns the area of the `Rectangle`.
 - d. A method `draw` displays a new instance of a `Rectangle` object. Refer to handout, *H.A.I.I – DrawingTool*, for details on `DrawingTools` methods.
2. Write a testing class with a `main` method that constructs a `Rectangle` and calls `getPerimeter` and `getArea` for each `Rectangle` created. **Sample** usage would be:

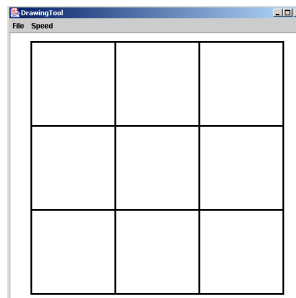
```
// Construct a 400 x 160 rectangle at location -200, -80.
Rectangle rectA = new Rectangle(-200, -80, 400, 160);
rectA.draw(); // draw the rectangle

System.out.println("Perimeter = " + rectA.getPerimeter());
System.out.println("Area = " + rectA.getArea());
```

The resulting images would be similar to the one shown below:



3. Use the concepts in 2. above to construct a 3x3 grid of `Rectangle` objects as show below. You should be able to produce the grid with only 3 rectangles. In addition, calculate and display the perimeter and area of the rectangles.



4. It is recommended that the `Rectangle` class and the testing class be in two source files. Call your instructor to your workstation for scoring.