EXERCISE VI

Go as far as you can!

6.1 (The **Rectangle** class) Following the example of the Circle class, design a class named Rectangle to represent a rectangle. The class contains:

- Two double data fields named **width** and **height** that specify the **width** and **height** of the rectangle. The default values are 1 for both width and height.
- A no-arg constructor that creates a default rectangle.
- A constructor that creates a rectangle with the specified **width** and **height**.
- A method named **getArea()** that returns the area of this rectangle.
- A method named **getPerimeter()** that returns the perimeter.

Implement the class. Write a test program that creates two Rectangle objects—one with width **4** and height **40** and the other with width **3.5** and height **35.9**. Display the width, height, area, and perimeter of each rectangle in this order.

6.2 (Geometry: n-sided regular polygon) In an n-sided regular polygon all sides have the same length and all angles have the same degree (i.e., the polygon is both equilateral and equiangular). Design a class named RegularPolygon that contains:

- A private **int** data field named **n** that defines the number of sides in the polygon with default value **3**.
- A private **double** data field named **side** that stores the length of the side with default value **1**.
- A private **double** data field named **x** that defines the x-coordinate of the center of the polygon with default value **0**.
- A private **double** data field named **y** that defines the y-coordinate of the center of the polygon with default value **0**.
- A no-arg constructor that creates a regular polygon with default values.
- A constructor that creates a regular polygon with the specified number of sides and length of side, centered at (**0**, **0**).
- A constructor that creates a regular polygon with the specified number of sides, length of side, and x-and y-coordinates.
- The accessor and mutator methods for all data fields.
- The method **getPerimeter()** that returns the perimeter of the polygon.
- The method **getArea()** that returns the area of the polygon. The formula for computing the area of a regular polygon is $$Area = \frac{n \times s^2}{4 \times \tan\left(\frac{\mathrm{p}}{n}\right)}.$$

Implement the class. Write a test program that creates three RegularPolygon objects, created using the no-arg constructor, using RegularPolygon(6, 4), and using RegularPolygon(10, 4, 5.6, 7.8). For each object, display its perimeter and area.

6.3 (*Using the* `GregorianCalendar` *class*) Java API has the `GregorianCalendar` class in the `java.util` package that can be used to obtain the year, month, and day of a date. The no-arg constructor constructs an instance for the current date, and the methods `get(GregorianCalendar.YEAR)`, `get(GregorianCalendar.MONTH)`, and `get(GregorianCalendar.DAY_OF_MONTH)` return the year, month, and day. Write a program to perform two tasks:

- Display the current year, month, and day.
- The `GregorianCalendar` class has the `setTimeInMillis(long)`, which can be used to set a specified elapsed time since January 1, 1970. Set the value to `1234567898765L` and display the year, month, and day.

6.4 (*The* `Location` *class*) Design a class named `Location` for locating a maximal value and its location in a two-dimensional array. The class contains public data fields `row`, `column`, and `maxValue` that store the maximal value and its indices in a two dimensional array with `row` and `column` as `int` type and `maxValue` as `double` type.

Write the following method that returns the location of the largest element in a two-dimensional array.

```
public static Location locateLargest(double[][] a)
```

The return value is an instance of `Location`. Write a test program that prompts the user to enter a two-dimensional array and displays the location of the largest element in the array. Here is a sample run:

```
Enter the number of rows and columns of the array: 3 4 ↵Enter
Enter the array:
23.5 35 2 10 ↵Enter
4.5 3 45 3.5 ↵Enter
35 44 5.5 9.6 ↵Enter
The location of the largest element is 45 at (1, 2)
```