Homework 03
Sorting Rectangles!
Due 10/06/2023 by 11:55PM

Objective:

Write a program where the user creates and sorts a collection of rectangles in a variety of ways based on their area, and gives the average area, minimum area, and maximum area.

Requirements:

- Functionality. (80pts)
  - No Syntax Errors. (80pts*)
    - *Code that cannot be compiled due to syntax errors is nonfunctional code and will receive no points for this entire section.
  - Only use Arrays to store the collection. (80pts*)
    - *Any other data structure such as LinkedLists, ArrayLists, etc. will result in no points awarded for this entire section.
  - Do not use built-in sorters. (80pts*)
    - *You must implement all sorting yourself.
    - *You MAY NOT use built-in Java sorters such as anything in “import java.util.*” or “import java.util.Arrays”.
    - *Any violation will result in no points awarded for this entire section.
  - Clear and Easy-To-Use Interface. (10pts)
    - Users should easily understand what the program does and how to use it.
    - Users should be prompted for input and should be able to enter data easily.
    - Users should be presented with output after major functions, operations, or calculations.
    - The user must be presented with the options:
      - Enter Rectangle Data – Both the size of the collection and each individual rectangle’s length and width.
      - Sort and display the Rectangles’ areas from Smallest to Largest.
      - Sort and display the Rectangles’ areas form Largest to Smallest.
      - Display the Average Area of the rectangles
      - Display the Minimum area in the collection.
      - Display the Maximum area in the collection.
      - Quit the program.
    - If the user picks an invalid option, then the program must inform the user and continue.

        - All the above must apply for full credit.
    - Specify the Size of the Collection and Entering Data. (10pts)
        - The user must be able to specify the size of the collection (the array). If the size is invalid, then the program must keep asking the user for a valid size. The program should not terminate if an invalid size is entered.
        - After creating the collection, the user must be able to enter the data – the length and width of the rectangle. The length and width can be any decimal value: positive, negative, or zero.
        - All the above must apply for full credit.
    - Sorting and Displaying the Data. (30pts)
        - The user must be able to sort the rectangles by their area either from smallest to largest (ascending order) or from smallest to largest (descending order). Both versions of sort must be implemented for full credit.
        - Area of a rectangle is the length x width.
        - If the user tries to sort a collection that has not been created (in other words an empty Array), then the program must inform them it cannot sort an empty collection.
        - Once the collection has been sorted the areas must be printed to the console.
        - All the above must apply for full credit.
    - Displaying the Average Area (10pts)
        - The user must be able to request the average area of the collection and the program must display the average area.
        - The average is the sum of all the areas divided by the number of items in the collection.
        - If the user tries to sort a collection that has not been created (in other words an empty Array), then the program must inform them it cannot sort an empty collection.
        - All the above must apply for full credit.
    - Displaying the Minimum Area (10pts)
        - The user must be able to request the minimum area in the collection and the program must display it.
        - If the user tries to find the minimum in a collection that has not been created (in other words an empty Array), then the program must inform them it cannot find the minimum in an empty collection.
        - All the above must apply for full credit.
    - Displaying the Maximum Area (10pts)
        - The user must be able to request the maximum area in the collection and the program must display it.
        - If the user tries to find the maximum in a collection that has not been created (in other words an empty Array), then the program must inform them it cannot find the maximum in an empty collection.
        - All the above must apply for full credit.
- Coding Style. (10pts)
    - Readable Code

- Meaningful identifiers for data and methods.
- Proper indentation that clearly identifies statements within the body of a class, a method, a branching statement, a loop statement, etc.
- All the above must apply for full credit.

- Comments. (10pts)
    - Your name in the file. (5pts)
    - At least 5 meaningful comments in addition to your name. These must describe the function of the code it is near. (5pts)

Finally:
Upload the .java file to the CSCE Dropbox