Activity 7.1 - BubbleSorter

Overview

In this activity you will create a class named BubbleSorter. It will have an `int` array as an instance variable that will be initialized with random integer values when a new BubbleSorter object is instantiated. The BubbleSorter will provide a `sort()` method that will sort the values within the int array with the smallest value being stored at index 0 and the largest value stored at index `n-1`, where `n` is the length of the array.

To perform the sort operation you will implement the Bubble Sort algorithm. See the Wikipedia page for Bubble Sort for a nice animation. As sorting algorithms go, bubble sort is one of the worst in terms of performance, but it does have the benefit of being one of the easiest to conceptualize and implement. :)

Instructions

Getting Started

1. Create an "`Activity7.1`" directory in VScode.
2. Create a new Java class file named BubbleSorter.java
3. Create a new Java class file named Driver.java with a main() method

Part 1: Initialize BubbleSorter and display array contents

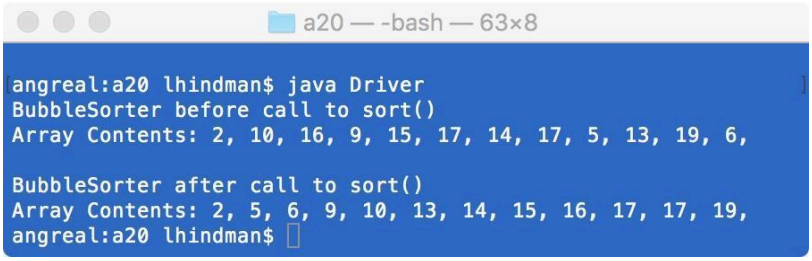
- Add an int array as an instance variable to BubbleSorter and in the constructor use a Random number generator to initialize the array with random values. The number of elements (size) in the array should be passed as a parameter to the BubbleSorter constructor
- Add a toString() method to the BubbleSorter class that uses a foreach loop to build a nicely-formatted String containing the comma-separated values of the int array.
- In the Driver class, create a new instance of the BubbleSorter with 12 elements and print the contents of the int array using the toString() method.

Part 2: Implement the Bubble Sort algorithm

- Add a private swap() method that will swap the values at two positions (indexes) within the int array. This method will take two int values as parameters that represent the indexes to swap in the array and will not return any value.
- Add a public sort() method to the BubbleSorter. This method will take no parameters and will not return any values. It will simply sort the values within the int array by swapping their positions. The table below provides a **pseudocode** implementation of the Bubble sort algorithm.

```
function sort():
     done = false
     while !done
          done = true
          for( i = 1; i < dataArray.length; i++)
               if (dataArray[i-1] > dataArray[i])
                    swap(i-1,i)
                    done = false
```

- Update the Driver class to call the sort() method and then call the toString() method a second time to display the sorted array.



Terminology Identification

In your code add comments identifying examples of the following: array declaration, array initialization, bounds checking, array element. These should be identified with an @keyterm tag within the comment.

Code Review

When you are finished with this activity, pair up with a classmate and review each other's code to make sure it meets all the requirements.

Submission

After completing the assignment, use the assignment link in Canvas and follow the submission instructions there. You will upload your `BubbleSorter.java` and `Driver.java` files and put your reflection in the "Comments" box.

Reflection Requirements

Write a one paragraph reflection describing your experience with this activity. The reflection should also include the name of your code review partner AND something interesting you found in their code. Please review the activity rubric for more details.