READ ME Sort Manager Project

First Java Project after learning Java core at Sparta Global. The Sort Manager utilises the Factory design pattern to produce its Sorter objects. All sorters implement a sorters interface that allows all sorters to be called as sorters as opposed to their derived classes.

Upon start up the program produces an instance of Logger using the log4j library from the maven central repository. This instance of logger is created so the program can log any exceptions that are thrown within a log file.

Once the logger has been instantiated the program creates an instance of Sorter factory that takes a user input using Scanner in the form of an int that is used by the sorter factory to determine the type of object it needs to produce. If a number is inputted that is not contained within the list of possible options, the sorter factory will throw an Illegal argument exception.

The program then uses scanner to get an input for the array. It starts by asking for the length of the array. The array length can be any positive integer. If a value of zero is entered as the array length the program will generate an array of random length (between 3 and 20000) and populate the array with random integer values (between -50000 and 50000) using the random method in the Math class. If a positive integer is entered the program will then preassign an empty array of the desired length and ask for the values to be entered into the array one at a time, again using a scanner. If a negative number is entered as a desired length of the array, then an Illegal argument exception will be thrown.

This array is then passed through a Sorted check before being passed to the sorter object that is created by the sorter factory. This sorted check consists of considering an element and the element after it until all elements in the array have been considered. If the program finds all elements are entered into the array in ascending order the program will output the Array, with the message that the array has already been sorted (in the case of Bubble Sort and Merge Sort). It will also check at this stage if an array consisting of a single number has been entered and again if it is found to be the case the program will return the single number with the message that “A single element array cannot be sorted”.

If the sorted check returns false (the array is not sorted) then the array is then passed to the sorter object created by the Sorter factory currently there are 3 different sorter implementations, and they are as follows:

1. Bubble Sort
2. Merge Sort
3. Binary Sort

For each of these implementations there is a null check to ensure the value of the array entered is not null. If null is passed into any of the objects, then the program will throw a null pointer exception.

In the section below the details to how the program processes the different types of sorts it can do are outlined.

Bubble Sort

Merge Sort

Binary Sort