## Lab 10

In lab 10, you will finish a multithreaded application where the threads cooperate to print numbers in sorted order. Two of the three classes are given to you in [lab10.zip](http://cs.utsa.edu/~cs3443/laboratories/lab10.zip).

The name of your Eclipse project should be abc123-lab10, where you replace abc123 with your abc123 id. The name of the file that contains the main method should remain SortingThreadTest.java. The other file should also keep its original name. To submit the project, export the project and upload the zip file to Blackboard.

### Task

Your task is to write the SortingThread class so it works in combination with the code in [lab10.zip](http://cs.utsa.edu/~cs3443/laboratories/lab10.zip). SortingThread should implement the Runnable interface so that when SortingThreadTest is run, it will print something similar to the following:

0 unsynchronized 1

0 unsynchronized 4

0 unsynchronized 4

0 unsynchronized 5

0 unsynchronized 6

0 unsynchronized 7

0 unsynchronized 8

0 unsynchronized 9

0 unsynchronized 9

0 unsynchronized 9

1 unsynchronized 0

1 unsynchronized 1

1 unsynchronized 2

1 unsynchronized 3

1 unsynchronized 3

1 unsynchronized 6

1 unsynchronized 6

1 unsynchronized 7

1 unsynchronized 8

1 unsynchronized 9

1 synchronized 0

0 synchronized 1

1 synchronized 1

1 synchronized 2

1 synchronized 3

1 synchronized 3

0 synchronized 4

0 synchronized 4

0 synchronized 5

0 synchronized 6

1 synchronized 6

1 synchronized 6

0 synchronized 7

1 synchronized 7

0 synchronized 8

1 synchronized 8

1 synchronized 9

0 synchronized 9

0 synchronized 9

0 synchronized 9

Each thread will get 10 numbers (generated randomly) to sort. Each thread first prints out the numbers without any synchronization. In the above printout, thread 0 printed its numbers before thread 1. Next, each thread prints out the numbers synchronizing with the other thread using a SortingBuffer object. One example in the above printout is where thread 0 had to wait for thread 1 to print a few numbers before thread 0 could print a 4.

### SortingThread

The SortingThread class must implement the Runnable interface. The constructor needs to remember the parameters in instance variables (the first parameter is the thread number).

The run method of SortingThread is where all the work happens. This is how to synchronize with the SortingBuffer object. For each value in the array, you must first call SortingBuffer's waitUntilMinimum method before printing out the value in the array (printing a line like those shown above). After printing all the values in the array, you must call SortingBuffer's finished method.

### SortingBuffer

The waitUntilMinimum method in SortingBuffer stores a value for each thread. It makes a thread wait until its value is the minimum. Sentinel values are used for initialization and finalization. Note that whenever the currentValue array is changed, notifyAll is called.

SortingBuffer could be improved in a couple of ways (not required for this lab, nor do you get extra credit). One would be to compute the minimum more efficiently. Another would be to add a generic type parameter so that SortingBuffer (and SortingThread) could be used for any Comparable type; this would make the initialization and finalization a little bit more difficult.

### Comments

Add javadoc comments to your SortingThread.java.

### UML Class Diagram

Create a UML class diagram using Violet.

### Rubric

* An incorrect submission will possibly get zero points. A project that does not compile will receive at most 50 points total.
* (40 pts.) If the threads successfully print out the unsynchronized lines similar to the above print out. Each thread should print its numbers in sorted order.
* (40 pts.) If the threads successfully print out the synchronized lines similar to the above print out. All the numbers from all the threads should be printed in sorted order.
* (10 pts.) If there are javadoc comments for SortingThread.java: at least one for the class and comments for each variable, method and constructor.
* (10 pts.) A UML class diagram is included that has all the classes (including the SortingThreadTest.java class) and all the constructors, methods, variables and constants of each class.