Midterm 2 recover

Due Jul 15 by 11:59pm **Points** 10

Goal

This assignment consists of programming a solution for the readers-writers problem using semaphores in java.

The Sources

Four classes are given as a template to start programming:

- **Main**: initializes everything and start the readers and writers. You can change this class to add anything you need, but keep the initialization routines untouched.
- **SharedResource**: this class represents the resource readers and writers are going to share. You should not change this class.
- **Writer**: this class represents a writer. You may add anything you need in this class, but must leave the sleep calls as is.
- **Reader**: this class represents a reader. You may add anything you need in this class, but must leave the sleep calls as is.

The Problem

Even though the sources compile and run, some RuntimeExceptions are thrown because the shared resource is being accessed without any mutual exclusion.

You should add the use of Semaphore objects to prevent this from happening.

Tips

Remember that many readers can read at the same time, but only one writer can write at the same time. In addition, when a writer is writing, no other readers should be reading.

You can assume the semaphores are FIFO.

Your solution should prevent writers from starving. One way of testing so is decreasing the readers sleep time to something low and test that writers are still being able to write.

Grading

You can do this assignments in groups of as much members as you want. However, the grading will involve a 5 minute interview with me (face to face) in which you will have to explain your solution, how it works and what where the problems you found.

Be prepared to explain any details of the given template code as well.

Your solution should show both reader and writer activity and no thrown exceptions.

This assignment will count as 10 points in your midterm 2 grade.