SPOS Lab Exp No.6 (Mutex And Semaphore)

import java.util.Scanner;

import java.util.concurrent.Semaphore;

public class ReaderWriter {

    static Semaphore mutex = new Semaphore (1);

    static Semaphore wrt = new Semaphore (1);

    static int readCount = 0;

    static String message = "Hello";

    static Scanner sc = new Scanner (System.in);

    static class Reader implements Runnable

    {

        public void run()

        {

            try

            {

                mutex.acquire();

                readCount++;

                if (readCount == 1)

                {

                    wrt.acquire();

                }

                mutex.release();

                System.out.println("Thread "+Thread.currentThread().getName()+" is reading : " + message);

                Thread.sleep(1500);

                System.out.println("Thread "+Thread.currentThread().getName()+" has finished reading");

                mutex.acquire();

                readCount--;

                if (readCount == 0)

                {

                    wrt.release();

                }

                mutex.release();

            }

            catch(InterruptedException e)

            {

                System.out.println(e.getMessage());

            }

        }

    }

    static class Writer implements Runnable

    {

        public void run ()

        {

            try

            {

                wrt.acquire();

                message = "Good Morning";

                System.out.println("Thread "+Thread.currentThread().getName()+" is writing : " + message);

                Thread.sleep(1500);

                System.out.println("Thread "+Thread.currentThread().getName()+" has finished writing");

                wrt.release();

            }

            catch(InterruptedException e)

            {

                System.out.println(e.getMessage());

            }

        }

    }

    public static void main(String[] args) throws Exception {

        Reader read = new Reader();

        Writer write = new Writer();

        Thread r1 = new Thread(read);

        r1.setName("Reader-1");

        Thread w1 = new Thread(write);

        w1.setName("Writer-1");

        Thread r2 = new Thread(read);

        r2.setName("Reader-2");

        Thread w2 = new Thread(write);

        w2.setName("Writer-2");

        Thread r3 = new Thread(read);

        r3.setName("Reader-3");

        Thread w3 = new Thread(write);

        w3.setName("Writer-3");

        r1.start();

        r2.start();

        r3.start();

        w1.start();

        w2.start();

        w3.start();

    }

}

OUTPUT:-

