**READER WRITER PROBLEM CODE**

class ReadWriteBuffer<T> {

private static final Object writeLock = new Object();

private static final Object readWriteLock = new Object();

private int numberReaders = 0;

private int numberWriters = 0;

private int numberWriteRequests = 0;

public void write(T object) throws InterruptedException {

numberWriteRequests++;

synchronized (readWriteLock) {

while (numberReaders > 0) {

readWriteLock.wait();

}

}

synchronized (writeLock) {

numberWriteRequests--;

numberWriters++;

System.out.println("Writer #" + Thread.currentThread().getId() + " started writing.");

Thread.sleep(3000);

System.out.println("Writer #" + Thread.currentThread().getId() + " finished writing.");

numberWriters--;

synchronized (readWriteLock) {

readWriteLock.notifyAll();

}

}

}

public void read() throws InterruptedException {

synchronized (readWriteLock) {

while (numberWriters > 0 || numberWriteRequests > 0) {

readWriteLock.wait();

}

}

numberReaders++;

System.out.println("Reader #" + Thread.currentThread().getId() + " started reading.");

Thread.sleep(1000);

System.out.println("Reader #" + Thread.currentThread().getId() + " finished reading.");

numberReaders--;

synchronized (readWriteLock) {

readWriteLock.notifyAll();

}

}

}

class Reader implements Runnable {

private final ReadWriteBuffer<Object> buffer;

private int priority;

public Reader(int priority, ReadWriteBuffer<Object> buffer) {

this.priority = priority;

this.buffer = buffer;

}

@Override

public void run() {

while (true) {

try {

Thread.sleep(priority);

buffer.read();

} catch (InterruptedException e) {

e.printStackTrace();

}

}

}

}

class Writter implements Runnable {

private final ReadWriteBuffer<Object> buffer;

private int priority;

public Writter(int priority, ReadWriteBuffer<Object> buffer) {

this.priority = priority;

this.buffer = buffer;

}

@Override

public void run() {

while (true) {

try {

Thread.sleep(priority);

buffer.write("Object");

} catch (InterruptedException e) {

e.printStackTrace();

}

}

}

}

public class ReadersWriters {

public static void main(String[] args) {

ReadWriteBuffer<Object> sharedBuffer = new ReadWriteBuffer<Object>();

new Thread(new Reader(1, sharedBuffer)).start();

new Thread(new Reader(700, sharedBuffer)).start();

new Thread(new Reader(100, sharedBuffer)).start();

new Thread(new Writter(2000, sharedBuffer)).start();

new Thread(new Writter(4000, sharedBuffer)).start();

}

}