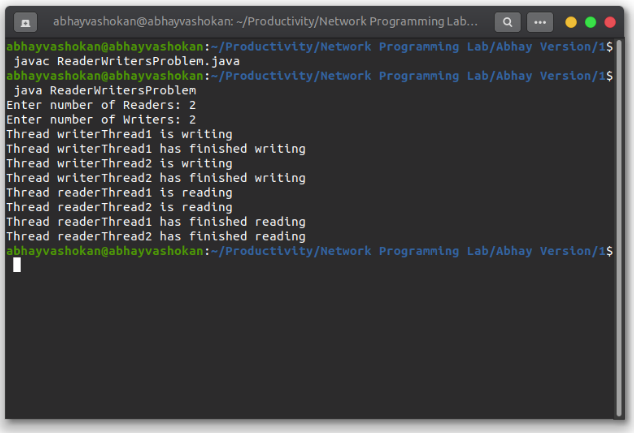
Program

```java
// Implement the First Readers-Writers Problem (Using Threads and Shared
Memory)

import java.util.Scanner;
import java.util.concurrent.Semaphore;

class ReaderWritersProblem {

    static Semaphore readLock = new Semaphore(1, true);
    static Semaphore writeLock = new Semaphore(1, true);
    static int readCount = 0;

    // Reader class
    static class Read implements Runnable {
        @Override
        public void run() {
            try {
                // Acquire Section
                readLock.acquire();
                readCount++;
                if (readCount == 1) {
                    writeLock.acquire();
                }
                readLock.release();

                // Reading section
                System.out.println("Thread " +
Thread.currentThread().getName() + " is reading");
                Thread.sleep(1500);
                System.out.println("Thread " +
Thread.currentThread().getName() + " has finished reading");

                // Releasing section
                readLock.acquire();
                readCount--;
                if (readCount == 0) {
                    writeLock.release();
                }
                readLock.release();
            } catch (InterruptedException e) {
                System.out.println(e.getMessage());
            }
        }
    }

    // Writer class
    static class Write implements Runnable {
        @Override
        public void run() {
            try {
                writeLock.acquire();
                System.out.println("Thread " +
Thread.currentThread().getName() + " is writing");
                Thread.sleep(2500);
                System.out.println("Thread " +
Thread.currentThread().getName() + " has finished writing");
                writeLock.release();
            } catch (InterruptedException e) {
                System.out.println(e.getMessage());
            }
        }
    }

    public static void main(String[] args) throws Exception {
        int readerNo, writerNo;
        Thread readerThread[] = new Thread[10];
        Thread writerThread[] = new Thread[10];
        Scanner scan = new Scanner(System.in);

        System.out.print("Enter number of Readers: ");
        readerNo = scan.nextInt();
        System.out.print("Enter number of Writers: ");
        writerNo = scan.nextInt();
        scan.close();

        if (readerNo < 0) {
            System.out.println("Error Message");
            System.exit(0);
        }
        if (writerNo < 0) {
            System.out.println("Error Message");
            System.exit(0);
        }

        Read read = new Read();
        Write write = new Write();

        for (int i = 1; i <= readerNo; i++) {
            readerThread[i] = new Thread(read);
            readerThread[i].setName("readerThread" + i);
        }
        for (int i = 1; i <= writerNo; i++) {
            writerThread[i] = new Thread(write);
            writerThread[i].setName("writerThread" + i);
        }

        int j = 1, k = 1;
        if (readerNo == 0) {
            while (k <= writerNo) {
                writerThread[k].start();
                k++;
            }
        } else if (writerNo == 0) {
                while (j <= readerNo) {
                    readerThread[j].start();
                    j++;
                }
            } else {
                while (j <= readerNo && k <= writerNo) {
                    readerThread[j].start();
                    writerThread[k].start();
                    j++;
                    k++;
                }
                while (j <= readerNo) {
                    readerThread[j].start();
                    j++;
                }
                while (k <= writerNo) {
                    writerThread[k].start();
                    k++;
                }
            }
        }
    }
```

Screenshot


abhayvashokan@abhayvashokan: ~/Productivity/Network Programming Lab...

```
abhayvashokan@abhayvashokan:~/Productivity/Network Programming Lab/Abhay Version/1$
 javac ReaderWritersProblem.java
abhayvashokan@abhayvashokan:~/Productivity/Network Programming Lab/Abhay Version/1$
 java ReaderWritersProblem
Enter number of Readers: 2
Enter number of Writers: 2
Thread writerThread1 is writing
Thread writerThread1 has finished writing
Thread writerThread2 is writing
Thread writerThread2 has finished writing
Thread readerThread1 is reading
Thread readerThread2 is reading
Thread readerThread1 has finished reading
Thread readerThread2 has finished reading
abhayvashokan@abhayvashokan:~/Productivity/Network Programming Lab/Abhay Version/1$
```

Output

```
Enter number of Readers: 2
Enter number of Writers: 2
Thread writerThread1 is writing
Thread writerThread1 has finished writing
Thread writerThread2 is writing
Thread writerThread2 has finished writing
Thread readerThread1 is reading
Thread readerThread2 is reading
Thread readerThread1 has finished reading
Thread readerThread2 has finished reading
```

ReadMe

1. `javac ReaderWritersProblem.java`
2. `java ReaderWritersProblem`