Name:- Bhavesh Kewalramanis

Roll No.:- 25

Batch:- A1

Semester:-4th

Shift:- 1st

Section:- A

PRACTICAL-8

AIM: Demonstrate the multithreading capabilities of Java.

Problem Statement: Create a reader thread that reads from a file word by word and stores it in shared memory (use arraylist). Also create a writer thread which writes from the shared memory to an output file. Write a main() to show the simulation of three readers, each reading from a different file and a single writer.

Code:

Reader part :

```
package eigth;

import java.io.FileReader;
import java.util.ArrayList;

public class Reader implements Runnable {
     Thread t;
     FileReader read;
     ArrayList<String> sm;
     int readerId;
```

```
    boolean eof=false;

    public Reader(int readerId,String
threadName,FileReader read, ArrayList<String> sm) {
        this.readerId=readerId;
        t=new Thread(this,threadName);
        this.read=read;
        this.sm=sm;
    }

    public String wordRead() {
        int ch,k=0;
        char w[] = new char[30];
        try {
            while((ch=read.read())!=-1) {
                if(ch==' '|| ch=='.' || ch==',' ||
ch=='\n') {
                    String w1=new String(w,0,k);
                    return w1;
                }
                w[k++]=(char)ch;
            }
        }catch(Exception e) {
            e.printStackTrace();
        }
        eof=true;
        return new String(w,0,k);
    }

    @Override
    public void run() {
        while(eof==false) {
            String w= wordRead();
            synchronized(sm) {
                System.out.println("Reader "+readerId+"
read  : "+w);
                sm.add(w);
                sm.notify();
            }
        }
    }
```

```
}
```

Writer part:

```
package eigth;

import java.io.FileReader;
import java.io.FileWriter;
import java.util.ArrayList;

public class Writer implements Runnable{
	Thread t;
	FileWriter write;
	ArrayList<String> sm;
	int writerId;

	public Writer(int writerId,String
threadName,FileWriter write, ArrayList<String> sm) {
		this.writerId=writerId;
		t=new Thread(this,threadName);
		this.write=write;
		this.sm=sm;
	}

	public void wordWrite(String w) {
		try {
			write.write(w + " ");
		}catch(Exception e) {
			e.printStackTrace();
		}

	}

	@Override
	public void run() {
		for(int i=0;i<5;i++) {
			synchronized(sm) {
				try {
					if(sm.isEmpty()) {
```

```
                              System.out.println("Shared
Memory is Empty. Waiting... ");
                              sm.wait();
                         }
                         else {
                              String w=sm.remove(0);
                              System.out.println("Writer
"+this.writerId+" wrote : "+w);
                              wordWrite(w);
                         }
                    }catch(Exception e) {
                         e.printStackTrace();
                    }
               }
          }

     }

}
```

Main part:

```
package eigth;

import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;
import java.util.concurrent.locks.Lock;
import java.util.concurrent.locks.ReadWriteLock;


public class ReaderWriter{
     public static void main(String[] args) {
          ArrayList<String> sm = new ArrayList<>();
          FileReader r1 = null,r2 = null,r3 = null;
          FileWriter w = null;
```

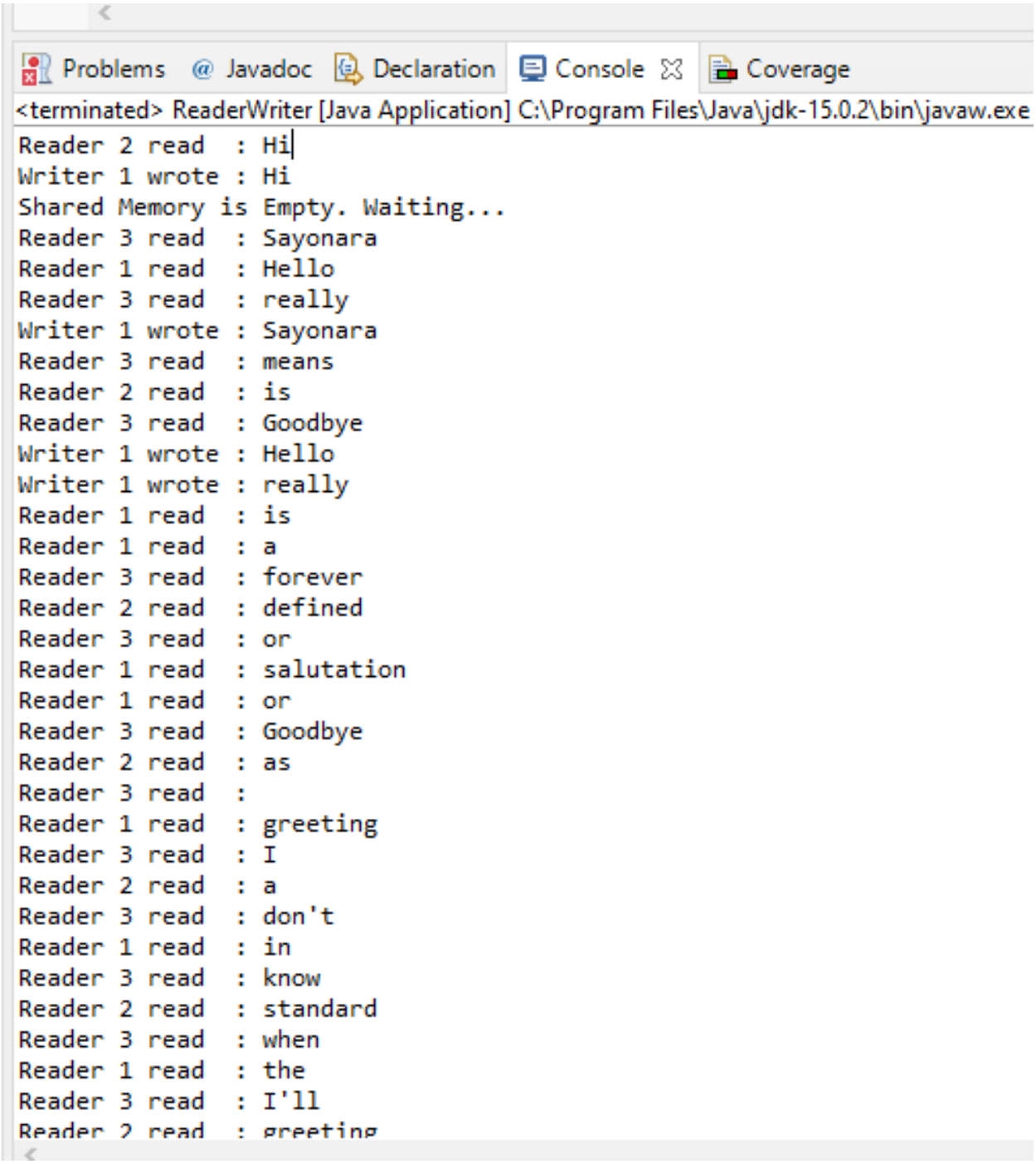
```
            try {
                r1 = new
FileReader("C:\\Users\\HP\\Desktop\\a.txt");
                r2=new
FileReader("C:\\Users\\HP\\Desktop\\b.txt");
                r3=new
FileReader("C:\\Users\\HP\\Desktop\\c.txt");
                w = new
FileWriter("C:\\Users\\HP\\Desktop\\d.txt");
            } catch (Exception e1) {
                e1.printStackTrace();
            }
            Reader reader1=new Reader(1,"reader1",r1,sm);
            Reader reader2=new Reader(2,"reader2",r2,sm);
            Reader reader3=new Reader(3,"reader3",r3,sm);
            Writer writer = new Writer(1,"writer",w,sm);
            reader1.t.start();
            reader2.t.start();
            reader3.t.start();
            writer.t.start();
            try {
            reader1.t.join(5000);
           reader2.t.join(5000);
            reader3.t.join(5000);
            writer.t.join(5000);
            r1.close();
            r2.close();
            r3.close();
            w.close();
            }catch (Exception e) {
                e.printStackTrace();
            }
        }

}
```

Output:

```
Reader 2 read  : Hi
Writer 1 wrote : Hi
Shared Memory is Empty. Waiting...
Reader 3 read  : Sayonara
Reader 1 read  : Hello
Reader 3 read  : really
Writer 1 wrote : Sayonara
Reader 3 read  : means
Reader 2 read  : is
Reader 3 read  : Goodbye
Writer 1 wrote : Hello
Writer 1 wrote : really
Reader 1 read  : is
Reader 1 read  : a
Reader 3 read  : forever
Reader 2 read  : defined
Reader 3 read  : or
Reader 1 read  : salutation
Reader 1 read  : or
Reader 3 read  : Goodbye
Reader 2 read  : as
Reader 3 read  :
Reader 1 read  : greeting
Reader 3 read  : I
Reader 2 read  : a
Reader 3 read  : don't
Reader 1 read  : in
Reader 3 read  : know
Reader 2 read  : standard
Reader 3 read  : when
Reader 1 read  : the
Reader 3 read  : I'll
Reader 2 read  : greeting
Reader 3 read  : see
Reader 1 read  : English
Reader 3 read  : you
Reader 2 read  : and
Reader 3 read  : again
Reader 2 read  : is
```

```
Reader 3 read  :
Reader 1 read  : language
Reader 2 read  : short
Reader 1 read  :
Reader 2 read  : for
Reader 2 read  : hello
Reader 2 read  :
```



```
<terminated> ReaderWriter [Java Application] C:\Program Files\Java\jdk-15.0.2\bin\javaw.exe (Mar 30, 2021, 8:33:09 P
Reader 3 read  : forever
Reader 2 read  : defined
Reader 3 read  : or
Reader 1 read  : salutation
Reader 1 read  : or
Reader 3 read  : Goodbye
Reader 2 read  : as
Reader 3 read  :
Reader 1 read  : greeting
Reader 3 read  : I
Reader 2 read  : a
Reader 3 read  : don't
Reader 1 read  : in
Reader 3 read  : know
Reader 2 read  : standard
Reader 3 read  : when
Reader 1 read  : the
Reader 3 read  : I'll
Reader 2 read  : greeting
Reader 3 read  : see
Reader 1 read  : English
Reader 3 read  : you
Reader 2 read  : and
Reader 3 read  : again
Reader 2 read  : is
Reader 3 read  :
Reader 1 read  : language
Reader 2 read  : short
Reader 1 read  :
Reader 2 read  : for
Reader 2 read  : hello
Reader 2 read  :
```