**2012 Final Thread**

**Buffer.java**

public interface Buffer {

public void set( int value );

public int get();

}

**SynchronizedBuffer.java**

public class SynchronizedBuffer implements Buffer {

private Lock accessLock = new ReentrantLock();

private int buffer = 0; // shared by producer and consumer threads

// place int value into buffer

@Override

public void set( int value )

{

accessLock.lock(); // lock this object

try

{

buffer = value; // set new buffer value

System.out.println((buffer-1)+" . eleman kontrol edildi " );

} // end try

finally

{

accessLock.unlock(); // unlock this object

} // end finally

} // end method set

// return value from buffer

@Override

public int get()

{

int readValue = 0; // initialize value read from buffer

accessLock.lock(); // lock this object

try

{

readValue = buffer; // retrieve value from buffer

} // end try

finally

{

accessLock.unlock(); // unlock this object

} // end finally

return readValue;

} // end method get

}

**Thread1.java**

public class Thread1 extends Thread {

int[] gelen = new int[40];

String isim;

private Buffer sharedLocation;

public Thread1(int[] gelen, String isim, Buffer sharedLocation)

{

this.gelen = gelen;

this.isim = isim;

this.sharedLocation = sharedLocation;

}

@Override

public void run()

{

//synchronized(this)

for(int i=0 ; i<10 ; i++)

{

try {

Thread.sleep(500);

} catch (InterruptedException ex) {

Logger.getLogger(Thread1.class.getName()).log(Level.SEVERE, null, ex);

}

System.out.println("şuanki thread "+isim+" işleme başladı");

if(gelen[sharedLocation.get()]==8)

{

System.out.println("aranan sayı "+sharedLocation.get()+". sırada bulunmuştur");

sharedLocation.set(sharedLocation.get()+1);

}else{

sharedLocation.set(sharedLocation.get()+1);}

}

}

}

**ThreadArama.java**

public class ThreadArama {

public static void main(String[] args) {

Buffer sharedLocation = new SynchronizedBuffer();

int [] dizi={1,1,1,1,1,1,1,1,1,1,

1,1,1,1,8,1,1,1,1,1,

1,1,1,1,1,8,1,1,1,1,

1,1,1,1,1,1,1,1,1,1};

Thread1 t1=new Thread1(dizi,"t1",sharedLocation);

Thread1 t2=new Thread1(dizi,"t2",sharedLocation);

Thread1 t3=new Thread1(dizi,"t3",sharedLocation);

Thread1 t4=new Thread1(dizi,"t4",sharedLocation);

t1.start();

t2.start();

t3.start();

t4.start();

}

}