Lab 10

# IPC

Date
Page

```
Class Q{
int n;
boolean valueSet = false;
Synchronized int get(){
while (!valueSet)
try {
wait();
} catch(InterruptedException e){}
System.out.println("InterruptedException caught");
}
System.out.println("Got: " + n);
valueSet = false;
notify();
return n;
}
Synchronized void put(int n){
while(valueSet)
try{
wait();
} Catch(InterruptedException e) {
System.out.println("InterruptedException caught");
}
this.n = n;
valueSet = true;
System.out.println("Put: " + n);
notify();
}
}.

Class Producer implements Runnable{
Q q;
Producer(Q q, a){
this.q = q;
new Thread(this, "Producer").start();
}
```

```
public void run(){
int i=0;
while(i<15){
q.put(i++);
}
}
}
class PCFixed{
public static void main(String args[]){
Q q=new Q();
new Producer(q);
new Consumer(q);
System.out.println("Press Control-C to stop");
}
}
```

Press Control-C to stop.

| | |
|---|---|
| Put 0 | Put 8 |
| Got 0 | Got 8 |
| Put 1 | Put 9 |
| Got 1 | Got 9 |
| Put 2 | Put 10 |
| Got 2 | Got 10 |
| Put 3 | Put 11 |
| Got 3 | Got 11 |
| Put 4 | Put 12 |
| Got 4 | Got 12 |
| Put 5 | Put 13 |
| Got 5 | Got 13 |
| Put 6 | Put 14 |
| Got 6 | Got 14 |
| Put 7 | |
| Got 7 | |

# Deadlock.

```
Class A{
Synchronized void foo(B b){
String name = Thread.currentThread().getName();
System.out.println(name + " entered A.foo");
try{
Thread.sleep(1000);
}catch(Exception e){
System.out.println("A Interrupted");
}
System.out.println(name + " trying to call B.last()");
b.last();
}
void last(){
System.out.println("Inside A.last");
}
}
Class B{
Synchronized void bax(A a){
String name = Thread.currentThread().getName();
System.out.println(name + " entered B.bax");
try{
Thread.sleep(1000);
}catch(Exception e){
System.out.println("B Interrupted");
}
System.out.println(name + " trying to Call A.last()");
a.last();
}
void last(){
System.out.println("Inside A.last");
}
}
Class Deadlock implements Runnable
```

```
{
A a = new A()
B b = new B();
Deadlock(){
Thread.currentThread().setName("MainThread");
Thread t = new Thread(this, "RacingThread");
t.start();
a.foo(b);
System.out.println("Back in main thread");
}
Public void run(){
b.bar(a);
System.out.println("Back in other thread");
}
Public static void main(String args[]){
new Deadlock();
}
}
```

Op

```
MainThread entered A.foo
RacingThread entered B.bar
RacingThread trying to call A.last()
Inside A.last
Back in other thread
MainThread trying to call B.last()
Inside A.last
Back in main thread.
```

06.02.24