

Foundations of Programming (in Java) 2002

p11 q2
fHK

Buffer Question - Solution Notes

The methods start() and run() are inherited from class Thread. The methods wait() and notifyAll() are inherited from class Object. [2 marks]

The most important keyword when answering the second part of the question is 'Monitor'. The description should explain that the synchronized modifier provides a means of managing shared data via the Monitor. If thread A calls a synchronized method when thread B has ownership of the Monitor then thread A will be 'blocked on synchronized'. If thread B calls wait() it gives up ownership and becomes 'blocked on wait' allowing thread A to grab the Monitor. A call of notifyAll() by thread A changes the state of thread B from being 'blocked on wait' to being 'blocked on synchronized'. Thus notifyAll() is advisory.

Some candidates may provide supporting diagrams along the lines of those on the attached sheet.

Here is a complete program which more than adequately attends to the final part of the question (and includes class Buffer which is not required).

```
public class BufferExample
{
    public static void main(String[] args)
    {
        Buffer buf = new Buffer();
        Consumer cons = new Consumer(buf);
        cons.start();
        try
        {
            for (int i = 1; i<=7; i++)
                buf.put(i);
        }
        catch (InterruptedException e)
        {
            System.out.println("Producer Interrupted!");
        }
        cons.halt();
    }
}

class Buffer
{
    private int value;
    private boolean full = false;

    public synchronized void put(int i) throws InterruptedException
    {
        while (this.full)
            this.wait();
        this.value = i;
        this.full = true;
        this.notifyAll();
    }

    public synchronized int get() throws InterruptedException
    {
        while (!this.full)
            this.wait();
        this.full = false;
        this.notifyAll();
        return this.value;
    }
}

class Consumer extends Thread
{
    private Buffer buffer;
    private volatile boolean running = false;

    public Consumer(Buffer b)
    {
        this.buffer = b;
    }

    public void run()
    {
        synchronized(this)
        {
            if (running)
                return;
            running = true;
        }
        while (running)
        {
            try
            {
                System.out.println("Found " + this.buffer.get());
            }
            catch (InterruptedException e)
            {
                System.out.println("Interrupted while consuming!");
            }
        }
    }

    public void halt()
    {
        running = false;
        this.interrupt();
    }
}
```

How the data fields of buf vary

