

```
class Buffer {  
    private int n;           // Pufferlaenge  
    private int[] contents; // Pufferinhalt  
    private int num, ipos, opos = 0;  
  
    public Buffer(int size) {  
        n = size; contents = new int[size];  
    }  
  
    public synchronized void append(int item)  
        throws InterruptedException {  
        while (num==n) wait();  
        contents[ipos] = item;  
        ipos = (ipos+1)%n; num++;  
        notify();  
    }  
}
```

...

```
public synchronized int remove()
    throws InterruptedException {
    while (num==0) wait();
    int item = contents[opos];
    opos = (opos+1)%n; num--;
    notify();
    return item;
}
}
```

```
class Producer extends Thread {
    private Buffer b;
    public Producer(Buffer b) { this.b = b; }
    public void run() {
        try {
            for (int i=1; i<=10; i++) {
                System.out.println("into buffer: "+i);
                b.append(i);
                sleep(5);
            }
        } catch (InterruptedException e) {
            return; // end this thread
        }
    }
}
```

```
class Consumer extends Thread {
    private Buffer b;
    public Consumer(Buffer b) { this.b = b; }
    public void run() {
        try {
            for (int i=1; i<=10; i++) {
                System.out.println("from buffer: "
                    + b.remove());
                sleep(20);
            }
        } catch (InterruptedException e) {
            return; // end this thread
        }
    }
}
```

```
...
```

```
Buffer b = new Buffer(4);
```

```
new Consumer(b).start();
```

```
new Producer(b).start();
```

```
...
```