

```

public class ProducerConsumer {
    static final int N = 100; // 定义缓冲区大小的常量
    static producer p = new producer(); // 初始化一个新的生产者线程
    static consumer c = new consumer(); // 初始化一个新的消费者线程
    static our_monitor mon = new our_monitor(); // 初始化一个新的管程

    public static void main(String args[]) {
        p.start(); // 开始生产者线程
        c.start(); // 开始消费者线程
    }

    static class producer extends Thread {
        public void run() { // run方法包含了线程代码
            int item;
            while (true) { // 生产者循环
                item = produce_item();
                mon.insert(item);
            }
        }
        private int produce_item() { ... } // 实际生产
    }

    static class consumer extends Thread {
        public void run() { // run方法包含了线程代码
            int item;
            while (true) { // 消费者循环
                item = mon.remove();
                consume_item(item);
            }
        }
        private void consume_item(int item) { ... } // 实际消费
    }

    static class our_monitor { // 这是一个管程
        private int buffer[] = new int[N];
        private int count = 0, lo = 0, hi = 0; // 计数器和索引

        public synchronized void insert(int val) {
            if (count == N) go_to_sleep(); // 如果缓冲区满, 则进入休眠
            buffer[hi] = val; // 向缓冲区中插入一个新的数据项
            hi = (hi + 1) % N; // 设置下一个数据项的槽
            count = count + 1; // 缓冲区中的数据项又多了一项
            if (count == 1) notify(); // 如果消费者在休眠, 则将其唤醒
        }

        public synchronized int remove() {
            int val;
            if (count == 0) go_to_sleep(); // 如果缓冲区空, 进入休眠
            val = buffer[lo]; // 从缓冲区中取出一个数据项
            lo = (lo + 1) % N; // 设置待取数据项的槽
            count = count - 1; // 缓冲区中的数据项数目减少1
            if (count == N - 1) notify(); // 如果生产者在休眠, 则将其唤醒
            return val;
        }

        private void go_to_sleep() { try{wait();} catch(InterruptedException exc) {} }
    }
}

```

图2-35 用Java语言实现的生产者-消费者问题的解法

与管程和信号量有关的另一个问题是, 这些机制都是设计用来解决访问公共内存的一个或多个CPU上的互斥问题的。通过将信号量放在共享内存中并用TSL或XCHG指令来保护它们, 可以避免竞争。如果一个分布式系统具有多个CPU, 并且每个CPU拥有自己的私有内存, 它们通过一个局域网相连, 那么这些原语将失效。这里的结论是: 信号量太低级了, 而管程在少数几种编程语言之外又无法使用, 并且, 这些原语均未提供机器间的信息交换方法。所以还需要其他的方法。