

```

class Stock {
    int item;
    boolean available = false;

    synchronized void addStock(int item) {
        while (available) {
            try {
                wait();
            } catch (InterruptedException e) {}
        }
        this.item = item;
        available = true;
        System.out.println("Produced item: " + item);
        notify();
    }

    synchronized int getStock() {
        while (!available) {
            try {
                wait();
            } catch (InterruptedException e) {}
        }
        System.out.println("Consumed item: " + item);
        available = false;
        notify();
        return item;
    }
}

class Producer extends Thread {
    Stock stock;

    Producer(Stock stock) {
        this.stock = stock;
    }

    public void run() {
        for (int i = 1; i <= 5; i++) {
            stock.addStock(i);
            try { Thread.sleep(500); } catch (InterruptedException e) {}
        }
    }
}

class Consumer extends Thread {
    Stock stock;

    Consumer(Stock stock) {
        this.stock = stock;
    }

    public void run() {
        for (int i = 1; i <= 5; i++) {
            stock.getStock();
            try { Thread.sleep(1000); } catch (InterruptedException e) {}
        }
    }
}

public class ProducerConsumerDemo {
    public static void main(String[] args) {
        Stock s = new Stock();
        new Producer(s).start();
        new Consumer(s).start();
    }
}

```

} }