

生产者消费者分为三个部分，第一是仓库，第二是生产者，第三是消费者。

```
public class CustomerAndProducer {

    static class Storehouse {
        private int capacity;
        private int size;

        public Storehouse(int capacity, int size) {
            super();
            this.capacity = capacity;
            this.size = size;
        }

        public synchronized void product(int count) {
            try {
                while (count > 0) {
                    if(size >= capacity) {
                        wait();
                    } else {
                        int inc = (count + size) > capacity ? capacity - size : count;
                        size += inc;
                        count -= inc;
                        System.out.println("本次生产产品: " + inc + "个, 当前产品: " + size + "个");
                    }
                    notifyAll();
                }
            } catch (Exception e) {
                // TODO: handle exception
            }
        }

        public synchronized void consume(int count) {
            try {
                while (count > 0) {
                    if(size <= 0) {
                        wait();
                    } else {
                        int inc = count > size ? size : count;
                        size -= inc;
                        count -= inc;
                        System.out.println("本次消费产品: " + inc + "个, 当前总产品: " + size + "个");
                    }
                }
            }
        }
    }
}
```

```

        System.out.println("本次消费产品: " + item.getName() + " 当前总产品: " + size + " 个");
    }
    notifyAll();
}
} catch (Exception e) {
    // TODO: handle exception
}
}

}

static class Consumer{

    private Storehouse storehouse;

    public Consumer(Storehouse storehouse) {
        super();
        this.storehouse = storehouse;
    }

    public void consume(final int count) {
        new Thread(new Runnable() {

            @Override
            public void run() {
                storehouse.consume(count);
            }
        }, "消费者").start();
    }

}

static class Producer{

    private Storehouse storehouse;

    public Producer(Storehouse storehouse) {
        super();
        this.storehouse = storehouse;
    }

    public void produce(final int count) {
        new Thread(new Runnable() {

```

```

static class Producer{
    private Storehouse storehouse;

    public Producer(Storehouse storehouse) {
        super();
        this.storehouse = storehouse;
    }

    public void produce(final int count) {
        new Thread(new Runnable() {

            @Override
            public void run() {
                storehouse.product(count);
            }
        }, "消费者").start();
    }
}

public static void main(String [] args) {
    Storehouse storehouse = new Storehouse(100, 0);

    Customer customer = new Customer(storehouse);
    Producer producer = new Producer(storehouse);

    customer.consume(10);
    producer.produce(110);
    customer.consume(50);
    customer.consume(70);
    customer.consume(20);
    customer.consume(100);
    producer.produce(60);
    producer.produce(10);
    producer.produce(510);

}
}

```