Wait & Notify

```
public class WaitNotifyExample {

    public static void main(String[] args) {
        Buffer buffer = new Buffer(5);
        Thread producer = new Thread(new Producer(buffer));
        Thread consumer = new Thread(new Consumer(buffer));

        producer.start();
        consumer.start();

    }

    static class Buffer {
        private final int capacity;
        private final Queue<Integer> queue = new LinkedList<Integer>();

        public Buffer(int capacity){
            this.capacity = capacity;
        }

        public synchronized void produce(int value) throws InterruptedException{
            while(queue.size() == capacity) {
                System.out.println("buffer is full);
                wait(); // Attende che ci sia spazio nel buffer
            }

            queue.add(value);
            System.out.println("produced: " + value);
            notify(); // Notifica che è disponibile un nuovo elemento
        }

        public synchronized int consume() throws InterruptedException {
            while(queue.isEmpty()) {
                System.out.println("bufffer is empty);
                wait(); // Attende almeno un elemento nel buffer
            }

            int value = queue.poll();
            System.out.println("consumed: " + value);
            notify(); // Notifica ai produttori che c'è spazio nel buffer
            return value;
        }
    }
    static class Producer implements Runnable {
        private final Buffer buffer;

        public Producer(Buffer buffer) {
            this.buffer = buffer;
        }

        @Override
        public void run() {
            try {
                for(int i = 0; i < 10; i++) {
                    buffer.produce(i);
                    Thread.sleep(100); // Simula la produzione
                }
            } catch(InterruptedException e) {
                Thread.currentThread().interrupt();
            }
        }
    }

    static class Consumer implements Runnable {
        private final Buffer buffer;

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

        @Override
        public void run() {
            try {
                for(int i = 0; i < 10; i++) {
                    buffer.consume();
                    Thread.sleep(150); // Simula la produzione
                }
            } catch(InterruptedException e) {
                Thread.currentThread().interrupt();
            }
        }
    }
}
```