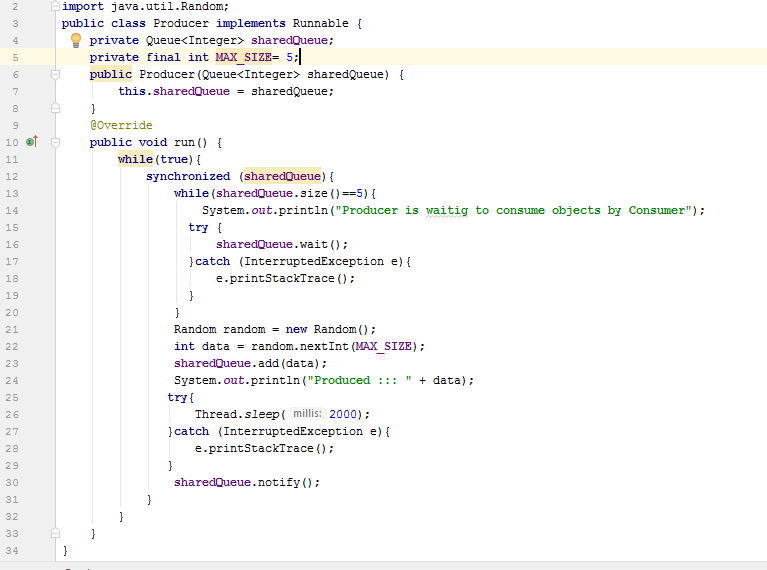
**Producer Consumer problem in Java using wait() and notify() method:**

Producer and Consumer are two separate threads which share a same BoundedQueue. The role of Producer is to produce the elements and moves into the Queue. Producer holds producing if the queue is full and resumes producing if the size of the queue is not full. The consumer consumes the element from the queue .Consumer holds consuming if the size of the queue is empty or zero and resumes consuming once the queue has an element.

This problem can be solved by many ways :

1. **Solving** **Producer Consumer problem problem using wait() and notify() method:**

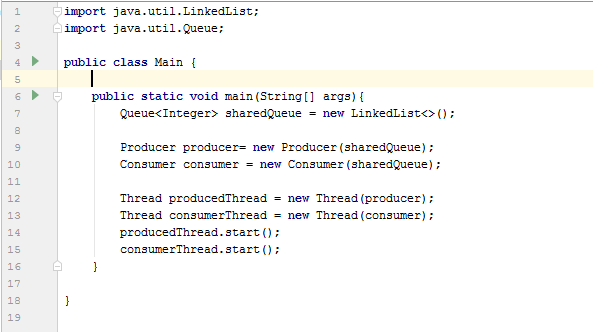
**Producer.java:**

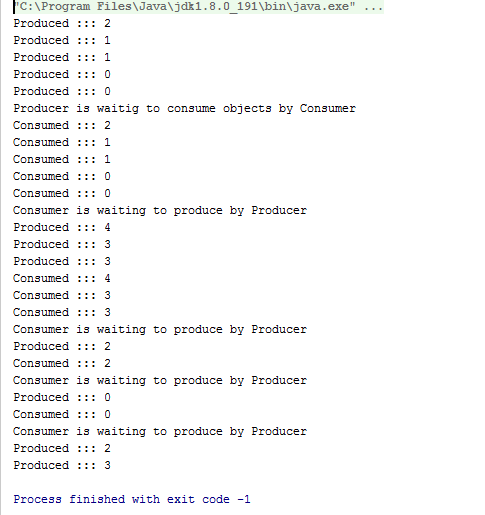


**Consumer.java :**



**Main.java**





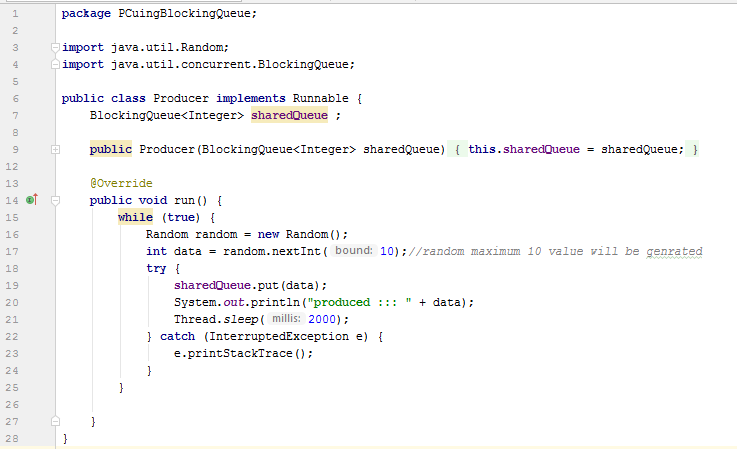
1. **Solving Producer Consumer problem using BlockingQueue :**

BlockingQueue solves much of the problems of synchronization mechanism handle by wait() and notify() in Producer Consumer problem. BlockingQueue has a method called take() and put() .The producer starts to produce objects and pushing into the Queue. Once the Queue is full Producer is in wait until consumer will consume it and it will start producing again. Similar behavior is followed by Consumer where Consumer waits until there is a single element in the queue.

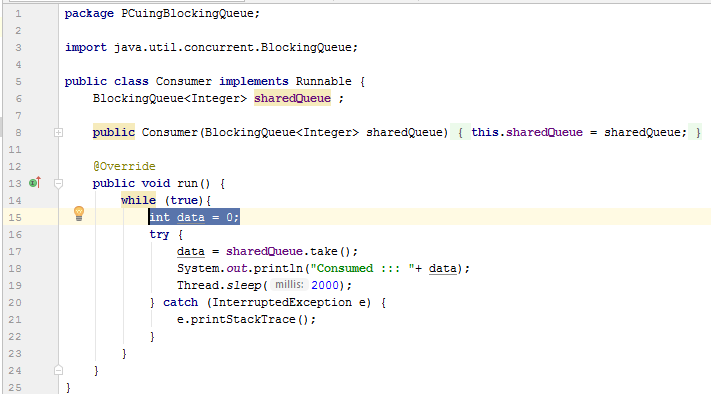
**Difference between put() and add() method of BlockingQueue :**

If there is not any space available in the BlockingQueueon and our requirement is insert more element in that , by using add() method it shows exception but on using put() method it waits until the blocking becomes vacant to add elements.

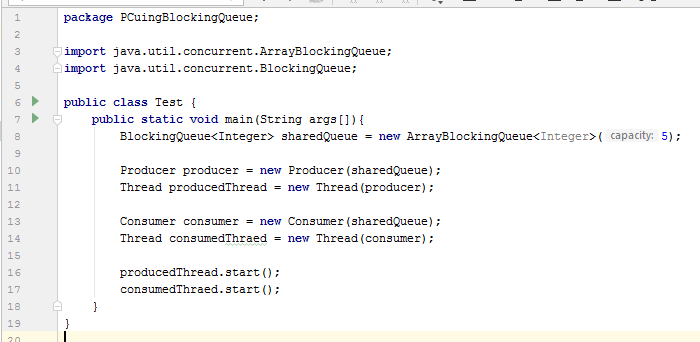
**Producer.java:**



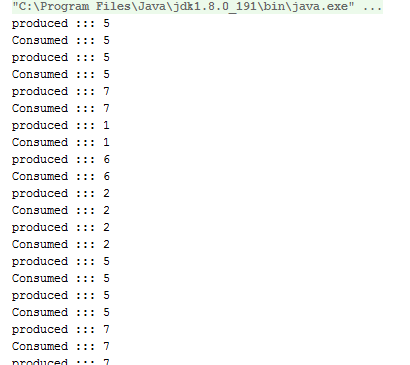
**Consumer.java:**



**Test.java:**



**Output :**



**From above output we can observe that producer is producing and Consumer is Consuming in a synchronized fashion without using wait() and notify() method.so instead of wait and notify java provide the implementation of BlockingQueue(came from jdk 1.5)**