JAVA EXAMPLES - PRODUCER CONSUMER PROBLEM

Problem Description:

How to solve the producer consumer problem using thread?

Solution:

Following example demonstrates how to solve the producer consumer problem using thread.

```
public class ProducerConsumerTest {
   public static void main(String[] args) {
      CubbyHole c = new CubbyHole();
      Producer p1 = new Producer(c, 1);
      Consumer c1 = new Consumer(c, 1);
      p1.start();
      c1.start();
   }
}
class CubbyHole {
   private int contents;
   private boolean available = false;
   public synchronized int get() {
      while (available == false) {
         try {
            wait();
         }
         catch (InterruptedException e) {
         }
      }
      available = false;
      notifyAll();
      return contents;
   }
   public synchronized void put(int value) {
      while (available == true) {
         try {
            wait();
         }
         catch (InterruptedException e) {
         }
      }
      contents = value;
      available = true;
      notifyAll();
   }
}

class Consumer extends Thread {
   private CubbyHole cubbyhole;
   private int number;
   public Consumer(CubbyHole c, int number) {
      cubbyhole = c;
      this.number = number;
   }
   public void run() {
      int value = 0;
         for (int i = 0; i < 10; i++) {
            value = cubbyhole.get();
            System.out.println("Consumer #"
   + this.number
+ " got: " + value);
}
}
}
```

```
class Producer extends Thread {
private CubbyHole cubbyhole;
private int number;

public Producer(CubbyHole c, int number) {
cubbyhole = c;
this.number = number;
}

public void run() {
for (int i = 0; i < 10; i++) {
cubbyhole.put(i);
System.out.println("Producer #" + this.number
+ " put: " + i);
try {
sleep((int)(Math.random() * 100));
} catch (InterruptedException e) { }
}
}
}
```

Result:

The above code sample will produce the following result.

```
Producer #1 put: 0
Consumer #1 got: 0
Producer #1 put: 1
Consumer #1 got: 1
Producer #1 put: 2
Consumer #1 got: 2
Producer #1 put: 3
Consumer #1 got: 3
Producer #1 put: 4
Consumer #1 got: 4
Producer #1 put: 5
Consumer #1 got: 5
Producer #1 put: 6
Consumer #1 got: 6
Producer #1 put: 7
Consumer #1 got: 7
Producer #1 put: 8
Consumer #1 got: 8
Producer #1 put: 9
Consumer #1 got: 9
```