

Ejemplo de Productores y Consumidores

1. Cargue y ejecute las siguientes clases y observe sus resultados

```
class Consumer implements Runnable {  
    Q q;  
    Consumer(Q q) {  
        this.q = q;  
        new Thread(this, "Consumer").start();  
    }  
    public void run() {  
        while(true) {  
            q.get();  
            try{ Thread.sleep(20); }catch (InterruptedException ie){}  
        }  
    }  
}
```

```
class Producer implements Runnable {  
    Q q;  
    Producer(Q q) {  
        this.q = q;  
        new Thread(this, "Producer").start();  
    }  
    public void run() {  
        int i = 0;  
        while(true) {  
            q.put(i++);  
            try{ Thread.sleep(20); }catch (InterruptedException ie){}  
        }  
    }  
}
```

```
public class ProduceConsume {  
    public static void main(String args[]) {  
        Q q = new Q();  
        new Producer(q);  
        new Consumer(q);  
        System.out.println("Press Control-C to stop.");  
    }  
}
```

```
class Q {  
    //este es un objeto compartido entre productores y consumidores  
    //esta versión es insegura  
    int n;  
    boolean valueSet = false;  
    synchronized int get() {  
        if(!valueSet)  
            System.out.println("Got: " + n);  
        valueSet = false;  
        return n;  
    }  
    synchronized void put(int n) {  
        if(valueSet)  
            this.n = n;  
        valueSet = true;  
        System.out.println("Put: " + n);  
    }  
}
```

2. Ahora cambie la clase Q por la siguiente, ejecute varias veces y compare sus resultados con los anteriores

```
class Q {
    int n;
    boolean valueSet = false;
    synchronized int get() {
        if(!valueSet)
            try {
                wait();
            } catch (InterruptedException e) {
                System.out.println("InterruptedException caught");
            }
        System.out.println("Got: " + n);
        valueSet = false;
        notify();
        return n;
    }
    synchronized void put(int n) {
        if(valueSet)
            try {
                wait();
            } catch (InterruptedException e) {
                System.out.println("InterruptedException caught");
            }
        this.n = n;
        valueSet = true;
        System.out.println("Put: " + n);
        notify();
    }
}
```

3. Modifique el programa para que existan varios productores y varios consumidores

4. Ponga un freno por tiempo