CSE-2008 Assignment - 10

Academic year: 2020-2021 **Semester:** FAL

Faculty Name: Mr. Sanket Mishra sir **Date:** 18 /11/2021

Student name: M.Taran **Reg. no.:** 19BCE7346

QUESTION:

Write a Java program to simulate the Producer and Consumer problem and use notify() method and wait method with synchronized block.

CODE :

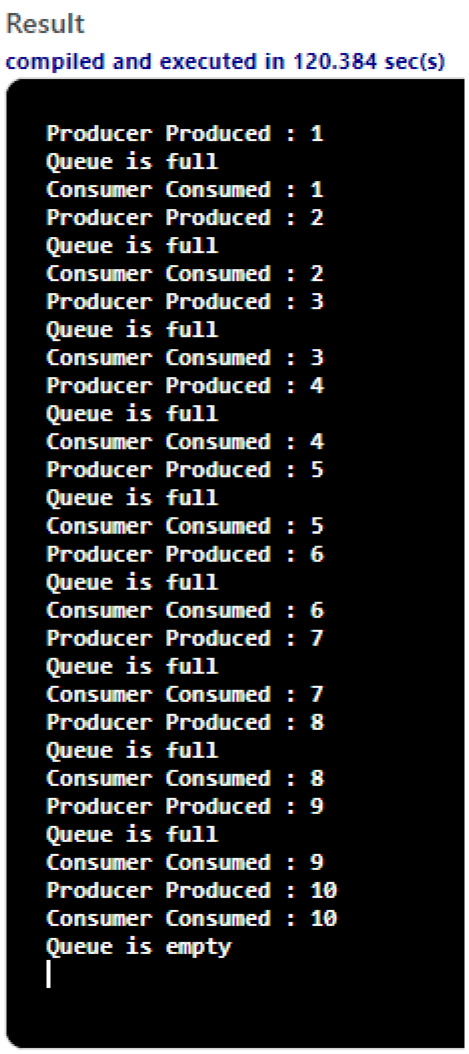
```
import java.util.ArrayList;
import java.util.List;
public class Main {
  public static void main(String[] args) {
    List < Integer > sharedQueue = new ArrayList < Integer > ();
    Producer producer = new Producer(sharedQueue);
    Consumer consumer = new Consumer(sharedQueue);
    Thread p = new Thread(producer, "Producer Thread");
    Thread c = new Thread(consumer, "Consumer Thread");
    p.start();
    c.start();
  }
}

class Producer implements Runnable {
  List < Integer > sharedQueue = new ArrayList < Integer > ();
  public Producer(List < Integer > sharedQueue) {
    this.sharedQueue = sharedQueue;
  }
  public void run() {
    for (int i = 1; i <= 10; i++) {
      try {
        produce(i);
```

```java
      } catch (InterruptedException e) {
        e.printStackTrace();
      }
    }
  }
  private void produce(int i) throws InterruptedException {
    synchronized(sharedQueue) {
      if (sharedQueue.size() == 1) {
        System.out.println("Queue is full");
        sharedQueue.wait();
      }
    }
    synchronized(sharedQueue) {
      System.out.println("Producer Produced : " + i);
      sharedQueue.add(i);
      Thread.sleep(1000);
      sharedQueue.notify();
    }
  }
}

class Consumer implements Runnable {
  List < Integer > sharedQueue = new ArrayList < Integer > ();
  public Consumer(List < Integer > sharedQueue) {
    this.sharedQueue = sharedQueue;
  }
  @Override public void run() {
    while (true) {
      try {
        consume();
        Thread.sleep(1000);
      } catch (InterruptedException e) {
        e.printStackTrace();
      }
    }
  }
  private void consume() throws InterruptedException {
    synchronized(sharedQueue) {
      while (sharedQueue.size() == 0) {
        System.out.println("Queue is empty");
        sharedQueue.wait();
      }
    }
    synchronized(sharedQueue) {
      Thread.sleep(1000);
      System.out.println("Consumer Consumed : " + sharedQueue.remove(0));
      sharedQueue.notify();
    }
  }
}
```

OUTPUT:



Result

compiled and executed in 120.384 sec(s)

```
Producer Produced : 1
Queue is full
Consumer Consumed : 1
Producer Produced : 2
Queue is full
Consumer Consumed : 2
Producer Produced : 3
Queue is full
Consumer Consumed : 3
Producer Produced : 4
Queue is full
Consumer Consumed : 4
Producer Produced : 5
Queue is full
Consumer Consumed : 5
Producer Produced : 6
Queue is full
Consumer Consumed : 6
Producer Produced : 7
Queue is full
Consumer Consumed : 7
Producer Produced : 8
Queue is full
Consumer Consumed : 8
Producer Produced : 9
Queue is full
Consumer Consumed : 9
Producer Produced : 10
Consumer Consumed : 10
Queue is empty
```