|  |
| --- |
| PRODUCER CONSUMER USING THREADS |
|  |  |
|  | import java.util.\*; |
|  | class producer implements Runnable |
|  | { |
|  | ArrayList<Integer>b=null; |
|  | final int limit=10;int i=0; |
|  | producer(ArrayList<Integer>buffer) |
|  | { |
|  | b=buffer; |
|  | } |
|  | public void run() |
|  | { |
|  | while(true) |
|  | { |
|  | try |
|  | { |
|  | i++; |
|  | produce(i); |
|  | } |
|  | catch(Exception e) |
|  | { |
|  | System.out.println(e); |
|  | } |
|  | } |
|  | } |
|  | public void produce(int i) throws InterruptedException |
|  | { |
|  | synchronized(b) |
|  | { |
|  | while(b.size()==limit) |
|  | { |
|  | System.out.println("Producer is waiting for consumer"); |
|  | b.wait(); |
|  | } |
|  | } |
|  | synchronized(b) |
|  | { |
|  | b.add(i); |
|  |  |
|  | Thread.sleep(10); |
|  | b.notify(); |
|  | } |
|  | } |
|  | } |
|  | class consumer implements Runnable |
|  | { |
|  | ArrayList<Integer>b=null; |
|  | final int limit=10;int i=0; |
|  | consumer(ArrayList<Integer>buffer) |
|  | { |
|  | b=buffer; |
|  | } |
|  | public void run() |
|  | { |
|  | while(true) |
|  | { |
|  | try |
|  | { |
|  |  |
|  | consume(0); |
|  | } |
|  | catch(Exception e) |
|  | { |
|  | System.out.println(e); |
|  | } |
|  | } |
|  | } |
|  | public void consume(int i) throws InterruptedException |
|  | { |
|  | synchronized(b) |
|  | { |
|  | while(b.isEmpty()) |
|  | { |
|  | System.out.println("Consumer is waiting for producer"); |
|  | b.wait(); |
|  | } |
|  | } |
|  | synchronized(b) |
|  | { |
|  | b.remove(i); |
|  | Thread.sleep(10); |
|  | b.notify(); |
|  | } |
|  | } |
|  | } |
|  | public class Producerconsumer{ |
|  | public static void main(String args[]) |
|  | { |
|  | ArrayList<Integer>buffer=new ArrayList<Integer>(); |
|  | Thread t1=new Thread(new producer(buffer)); |
|  | Thread t2=new Thread(new consumer(buffer)); |
|  | t1.start(); |
|  | t2.start(); |
|  | } |
|  | } |