public class PCThreading {

public static void main(String[] args) throws InterruptedException {

ProducerConsumer pc = new ProducerConsumer();

// Producer Thread

Thread t1 = new Thread(new Runnable() {

@Override

public void run() {

try {

pc.produce();

} catch (InterruptedException e) {

e.printStackTrace();

}

}

});

// consumer thread

Thread t2 = new Thread(new Runnable() {

@Override

public void run() {

try {

pc.consume();

} catch (InterruptedException e) {

e.printStackTrace();

}

}

});

// Starting the P & C threads:

t1.start();

t2.start();

// t1 should finish before t2.

t1.join();

t2.join();

}

}

class ProducerConsumer {

private int index=0, val=0;

private int[] buffer = new int[2];

void produce() throws InterruptedException {

while (true) {

synchronized (this) {

int size = 2;

while (index == size -1)

wait();

System.out.println(("Produced value: " + val));

buffer[index++] = val++;

notify();

Thread.sleep(1000);

}

}

}

void consume() throws InterruptedException {

while (true) {

synchronized (this) {

while (index==0)

wait();

System.out.println("Consumed value: " + buffer[--index]);

notify();

Thread.sleep(1000);

}

}

}

}

"C:\Program Files\Java\jdk1.8.0\_202\bin\java.exe" "-

Produced value: 0

Consumed value: 0

Produced value: 1

Consumed value: 1

Produced value: 2

Consumed value: 2

Produced value: 3

Consumed value: 3

Produced value: 4

Consumed value: 4

Produced value: 5

Consumed value: 5

Produced value: 6

Consumed value: 6

Produced value: 7

Consumed value: 7

Produced value: 8

Consumed value: 8

Produced value: 9

Consumed value: 9

Produced value: 10

Consumed value: 10

Process finished with exit code -1