AssignmentDFSD702

MultiThreading

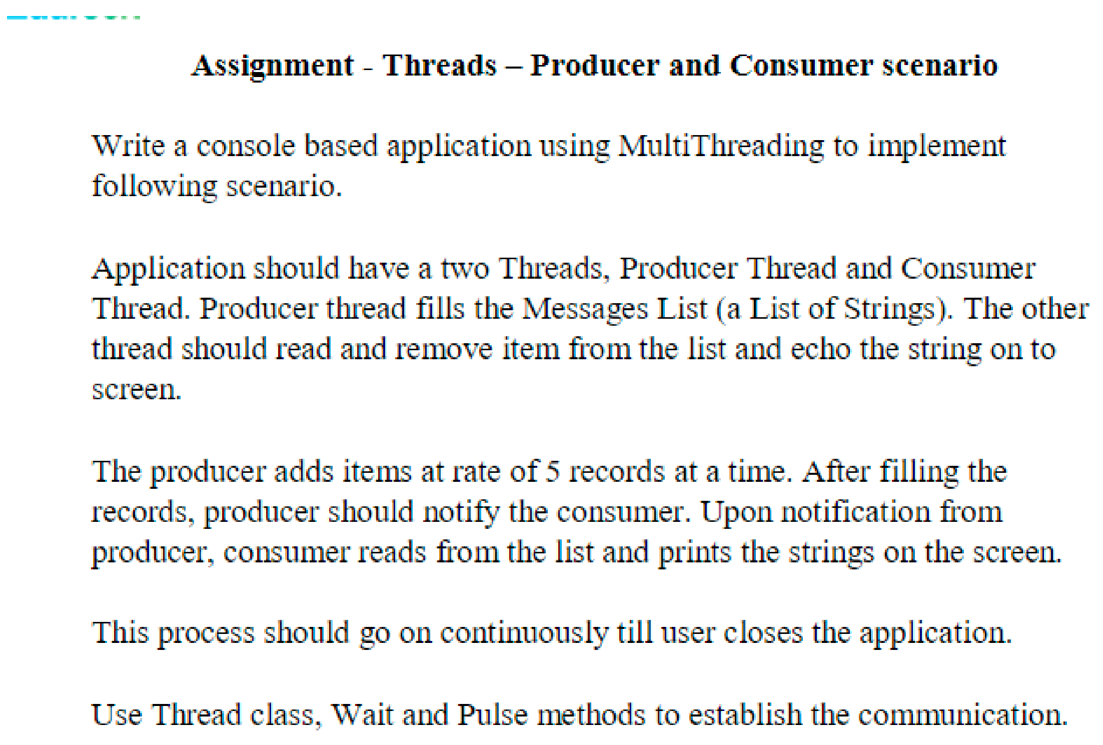
Name:B Karthik

MID:M1082972

Source Code:

https://github.com/BKarthik1/BKarthik1.git

Question



Assignment - Threads – Producer and Consumer scenario

Write a console based application using MultiThreading to implement following scenario.

Application should have a two Threads, Producer Thread and Consumer Thread. Producer thread fills the Messages List (a List of Strings). The other thread should read and remove item from the list and echo the string on to screen.

The producer adds items at rate of 5 records at a time. After filling the records, producer should notify the consumer. Upon notification from producer, consumer reads from the list and prints the strings on the screen.

This process should go on continuously till user closes the application.

Use Thread class, Wait and Pulse methods to establish the communication.

Solution

```
using System;
using System.Collections.Generic;
using System.Threading.Channels;
using System.Threading.Tasks;
namespace Console App ProducerConsumerPattern {
class Program {
static void Main(string[] args) {
var pc = new ProducerConsumer();
pc.StartChannel();
Console.ReadKey();
}
}
public class ProducerConsumer {
static int messageLimit = 5;
Channel < string > channel = Channel.CreateBounded < string > (messageLimit);
public void StartChannel() {
List < string > names = new List < string > ();
names.Add("John Smith");
names.Add("Jane Smith");
names.Add("John Doe");
names.Add("Jane Doe");
Task producer = Task.Factory.StartNew(() => {
foreach(var name in names) {
channel.Writer.TryWrite(name);
}
channel.Writer.Complete();
});
Task[] consumer = new Task[2];
for (inti = 0; i < consumer.Length; i++) {
consumer[i] = Task.Factory.StartNew(async () => {
while (awaitchannel.Reader.WaitToReadAsync()) {
if (channel.Reader.TryRead(out
var data)) {
Console.WriteLine($ " Data read from Consumer
No.{Task.CurrentId} is {data}");
}
}
});
}
producer.Wait();
Task.WaitAll(consumer);
}
}
```

```
}
```