

Projects × **Files** **Services**

- JavaApplication1
- JavaApplication2
- JavaApplication4
- ProducerConsumer
 - Source Packages
 - producerconsumer
 - ProducerConsumer.java
 - ProducerConsumerMain.java
 - threadProduce.java
 - threeadConsume.java
 - Test Packages
 - Libraries
 - Test Libraries

main - Navigator ×

Members <empty>

- ProducerConsumerMain
 - main(String[] args)

ProducerConsumerMain.java × ProducerConsumer.java × threadProduce.java × threeadConsume.java ×

Source History

```

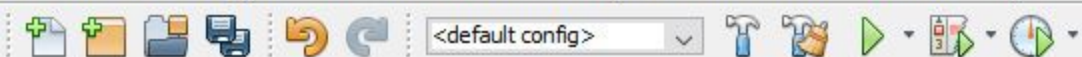
4      * and open the template in the editor.
5      */
6      package producerconsumer;
7      import java.util.Scanner;
8      /**
9       *
10      * @author Towkir
11      */
12      public class ProducerConsumerMain {
13
14          /**
15           * @param args the command line arguments
16           */
17          public static void main(String[] args) throws Exception{
18              // TODO code application logic here
19              int n;
20              System.out.print("Enter number of Items : ");
21              Scanner sc = new Scanner(System.in);
22              n = sc.nextInt();
23              ProducerConsumer pc = new ProducerConsumer(n);
24              threadProduce t1 = new threadProduce(pc);
25              threeadConsume t2 = new threeadConsume(pc);
26              t1.start();
27              t2.start();
28          }
29
30      }
31

```

producerconsumer.ProducerConsumerMain >

Output - ProducerConsumer (run) ×

>> Enter number of Items : 2



Projects | **Files** | **Services**

- JavaApplication1
- JavaApplication2
- JavaApplication4
- ProducerConsumer
 - Source Packages
 - producerconsumer
 - ProducerConsumer.java
 - ProducerConsumerMain.java
 - threadProduce.java
 - threadConsume.java
 - Test Packages
 - Libraries
 - Test Libraries

producerItem - Navigator

Members: <empty>

- ProducerConsumer
 - ProducerConsumer(int capacity)
 - consumeItem()
 - produceItem()
 - capacity : int
 - i : int
 - j : int
 - list : ArrayList<Integer>
 - sc : Scanner

ProducerConsumerMain.java | ProducerConsumer.java | threadProduce.java | threadConsume.java

```

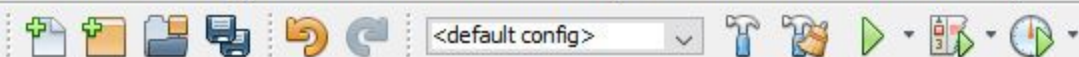
6 package producerconsumer;
7 import java.util.*;
8
9 /**
10  *
11  * @author Towkir
12  */
13 public class ProducerConsumer {
14     ArrayList<Integer> list = new ArrayList<>();
15     int capacity,i=0,j=0;
16     Scanner sc=new Scanner(System.in);
17
18     public ProducerConsumer(int capacity) {
19         this.capacity = capacity;
20     }
21     public void produceItem() throws InterruptedException {
22         int itemNumber = 0;
23         while (true) {
24             synchronized (this) {
25                 while (list.size() == capacity) {
26                     i=0;
27                     wait();
28                 }
29                 System.out.println("Producer-"+i+":Data to insert in the buffer");
30                 int x=sc.nextInt();
31                 System.out.print("Data:"+x);
32                 System.out.println("\nProducer-"+i+":Data " + x+" is inserted");
33                 System.out.println("Empty Space:"+capacity-i);

```

producerconsumer.ProducerConsumer > produceItem > while (true) >

Output - ProducerConsumer (run)

>> Enter number of Items : 2



Projects | **Files** | **Services**

- JavaApplication1
- JavaApplication2
- JavaApplication4
- ProducerConsumer
 - Source Packages
 - producerconsumer
 - ProducerConsumer.java
 - ProducerConsumerMain.java
 - threadProduce.java
 - threadConsume.java
 - Test Packages
 - Libraries
 - Test Libraries

producerItem - Navigator

Members: <empty>

- ProducerConsumer
 - ProducerConsumer(int capacity)
 - consumeItem()
 - produceItem()
 - capacity : int
 - i : int
 - j : int
 - list : ArrayList<Integer>
 - sc : Scanner

ProducerConsumerMain.java | **ProducerConsumer.java** | threadProduce.java | threadConsume.java

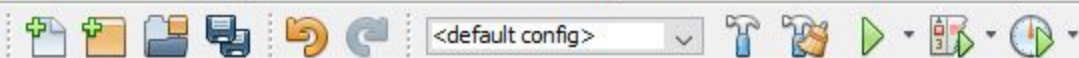
```

23  while (true) {
    33      System.out.println("Empty Space:"+(capacity-1));
    34      System.out.println("Full Space:"+(i+1));
    35      i++;
    36      list.add(x);
    37      notify();
    38      Thread.sleep(2000);
    39  }
    40  }
    41  }
    42
    43  public void consumeItem() throws InterruptedException {
    44      while (true) {
    45          synchronized (this) {
    46              while (list.size() == 0) {
    47                  j=0;
    48                  wait();
    49              }
    50              int item = list.remove(0);
    51              System.out.println("Consumer-" + j+":consumed data "+item);
    52              j++;
    53              notify();
    54              Thread.sleep(2000);
    55          }
    56      }
    57  }
    58  }
    59  }
    
```

producerconsumer.ProducerConsumer > produceItem > while (true) >

Output - ProducerConsumer (run)

>> Enter number of Items : 2



Projects × **Files** **Services**

- JavaApplication1
- JavaApplication2
- JavaApplication4
- ProducerConsumer
 - Source Packages
 - producerconsumer
 - ProducerConsumer.java
 - ProducerConsumerMain.java
 - threadProduce.java
 - threadConsume.java
 - Test Packages
 - Libraries
 - Test Libraries

Navigator ×

Members <empty>

- threadProduce :: Thread
 - threadProduce(ProducerConsumer producerC
 - run() ↑ Thread
 - producerConsumer : ProducerConsumer

ProducerConsumerMain.java × ProducerConsumer.java × threadProduce.java × threadConsume.java ×

Source History

```

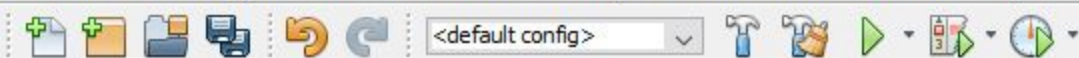
1  /*
2   * To change this license header, choose License Headers in Project Properties.
3   * To change this template file, choose Tools | Templates
4   * and open the template in the editor.
5   */
6   package producerconsumer;
7
8   /**
9    *
10   * @author Towkir
11   */
12   public class threadProduce extends Thread{
13       private ProducerConsumer producerConsumer;
14
15       public threadProduce(ProducerConsumer producerConsumer) {
16           this.producerConsumer = producerConsumer;
17       }
18
19       public void run() {
20           try {
21               producerConsumer.produceItem();
22           } catch (InterruptedException e) {
23               e.printStackTrace();
24           }
25       }
26   }
27

```

producerconsumer.threadProduce >>

Output - ProducerConsumer (run) ×

>> Enter number of Items : 2



Projects **Files** **Services**

- JavaApplication1
- JavaApplication2
- JavaApplication4
- ProducerConsumer
 - Source Packages
 - producerconsumer
 - ProducerConsumer.java
 - ProducerConsumerMain.java
 - threadProduce.java
 - threadConsume.java
 - Test Packages
 - Libraries
 - Test Libraries

threadConsume - Navigator

Members <empty>

- threadConsume :: Thread
 - threadConsume(ProducerConsumer producerConsumer)
 - run() ↑ Thread
 - producerConsumer : ProducerConsumer

ProducerConsumerMain.java x ProducerConsumer.java x threadProduce.java x threadConsume.java x

Source History

```

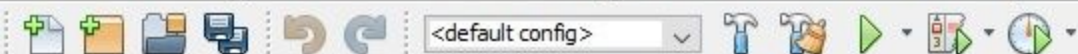
1  /*
2   * To change this license header, choose License Headers in Project Properties.
3   * To change this template file, choose Tools | Templates
4   * and open the template in the editor.
5   */
6   package producerconsumer;
7
8   /**
9    *
10   * @author Towkir
11   */
12   public class threadConsume extends Thread{
13       private ProducerConsumer producerConsumer;
14
15       public threadConsume(ProducerConsumer producerConsumer) {
16           this.producerConsumer = producerConsumer;
17       }
18
19       public void run() {
20           try {
21               producerConsumer.consumeItem();
22           } catch (InterruptedException e) {
23               e.printStackTrace();
24           }
25       }
26   }
27

```

producerconsumer.threadConsume >

Output - ProducerConsumer (run) x

>> Enter number of Items : 2



Projects | **Files** | **Services**

- JavaApplication1
- JavaApplication2
- JavaApplication4
- ProducerConsumer
 - Source Packages
 - producerconsumer
 - ProducerConsumer.java
 - ProducerConsumerMain.java
 - threadProduce.java
 - threadConsume.java
 - Test Packages
 - Libraries
 - Test Libraries

main - Navigator

Members: <empty>

- ProducerConsumerMain
 - main(String[] args)

ProducerConsumerMain.java | **ProducerConsumer.java** | threadProduce.java | threadConsume.java

Source | History | [Icons]

producerconsumer.ProducerConsumerMain | main

Output - ProducerConsumer (run)

```

Enter number of Items : 2
Producer-0:Data to insert in the buffer
10
Data:10
Producer-0:Data 10 is inserted
Empty Space:1
Full Space:1
Producer-1:Data to insert in the buffer
20
Data:20
Producer-1:Data 20 is inserted
Empty Space:1
Full Space:2
Consumer-0:consumed data 10
Consumer-1:consumed data 20
Producer-2:Data to insert in the buffer
10
Data:10
Producer-2:Data 10 is inserted
Empty Space:1
Full Space:3
Producer-3:Data to insert in the buffer
30
Data:30
Producer-3:Data 30 is inserted
Empty Space:1
Full Space:4
Consumer-0:consumed data 10
Consumer-1:consumed data 30
Producer-0:Data to insert in the buffer
20
Data:20
Producer-0:Data 20 is inserted
    
```