

06/02/24

classmate

Date \_\_\_\_\_  
Page \_\_\_\_\_

class BMS extends Thread

{  
    public void run()  
    {

        for (int i=1; i<=5; i++)  
            System.out.println ("BMS college of engineering "+i);

    try  
    {

        Thread.sleep(10000);  
    }

    catch (InterruptedException e) {

        e.printStackTrace();  
    }

    }

}

class CS extends Thread

{  
    public void run()  
    {

        for (int i=1; i<=5; i++)  
            System.out.println ("CSE "+i);

    try  
    {

        Thread.sleep(2000);  
    }

    catch (InterruptedException e) {

        e.printStackTrace();  
    }

    }

}

    }

}

class overmain:

{  
    public static void main (String args [ ] )

    BMS b1 = new BMS ();

    CS c1 = new CSC();

    b1.start ();

    c1.start ();

}

Output :

BMS college of engineering,

ESG 1

RCE 1

CSE 3

CSE 4

CSE 5

BMS college of engineering 2

BMS college of engineering 3

BMS college of engineering 4

BMS college of engineering 5

class Q {  
 int n;  
 boolean valueSet = false;  
 synchronized int get() {  
 while (!valueSet) {  
 try {  
 System.out.println("In Consumer waiting (" + n + ");");  
 wait();  
 } catch (InterruptedException e) {  
 System.out.println("InterruptedException caught");  
 }  
 System.out.println("Error: " + n);  
 valueSet = false;  
 System.out.println("In Producer waiting (" + n + ");");  
 notify();  
 return n;  
 }  
 synchronized void put(int n) {  
 while (valueSet) {  
 try {  
 System.out.println("In Producer waiting (" + n + ");");  
 wait();  
 } catch (InterruptedException e) {  
 System.out.println("InterruptedException caught");  
 }  
 }  
 System.out.println("Put: " + n);  
 System.out.println("In Intimate Consumer (" + n + ");");  
 notify();  
 }  
 }  
}

## class Producer implements Runnable

```
    @Q q  
    producer (@Q q) {  
        this.q = q;  
        new Thread(this, "Producer").start();  
    }  
  
    public void run() {  
        int i = 0;  
        while (i < 10) {  
            q.put (i++);  
        }  
    }  
}
```

## class Consumer implements Runnable

```
    @Q q  
    consumer (@Q q) {  
        this.q = q;  
        new Thread(this, "Consumer").start();  
    }  
  
    public void run() {  
        int i = 0;  
        while (i < 10) {  
            int x = q.get();  
            System.out.println("Consumed: " + x);  
            i++;  
        }  
    }  
}
```

class P  
public  
@Q =  
new R  
new C  
System  
3

Output

Put : 1  
026 F : 1  
Put : 2  
Get : 1  
Put : 2  
Get : 2  
Put : 3  
Get : 3  
Put : 4  
Get : 4  
Put : 5  
Get : 5  
Put : 6  
Get : 6  
Put : 7  
Get : 7  
Put : 8  
Get : 8  
Put : 9  
Get : 9  
Put : 10  
Get : 10

```
class PCFixed {  
public static void main (String args[]) {  
    Q q = new Q();
```

```
    new Producer(q);
```

```
    new Consumer(q);
```

```
    System.out.println ("Press control-C to stop");  
}}
```

```
}
```

Output:

Put : 1

Get : 1

Put : 2

Get : 2

Put : 3

Get : 3

Put : 4

Get : 4

Put : 5

Get : 5

See Screenshot

Put : 0 ~ m