

6/2/24

classmate

Date

Page

## Program 10

### IPC

class Q {

int n;

boolean valueSet = false;

synchronized int get() {

while (!valueSet)

try {

System.out.println("Interrupted \n consumer waiting \n");  
wait();

} catch (InterruptedException e) {

System.out.println("InterruptedException caught");  
}

System.out.println("out:" + n);

valueSet = false;

System.out.println("in intimate producer \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("Exception caught");  
}

}

this.n = 1;

01 number?

valueSet = line;

591

System.out.println("Put: " + n);

3 0 (ans)

System.out.println("In infinite consumer will?");

? () top line, bottom line?

notify();

(2) 2 enter! () enter

{  
}

3 null

class producer implements Runnable {

{  
// producer produces n/ buffer size

@ q;

() show

? () will pass buffer size? data

producer(@ q) {  
// produce & put producer

this.q = q;

(n + " = full") notify; would up?

new Thread(this, "producer").start();

? () will wait? consumer? notify; no. sleep?

public void run() {

() get in

consumer

int i = 0;

while (i < 5) {

q.put(i++);

? (n bit) top line, bottom line?

(2) 2 enter? enter

}

3 null

}

? () will pass buffer size? "full" no. sleep?

() show

class consumer implements Runnable {

? () will pass buffer size? data

@ q;

? () will pass buffer size? notify; no. sleep?



```
consumer (Q q) {
```

```
    this.q = q;
```

```
    new Thread (this, "consumer").start();
```

```
}
```

```
public void run() {
```

```
    int i = 0;
```

```
    while (i < 15) {
```

```
        int x = q.get();
```

```
        System.out.println ("consumed:" + x);
```

```
        i++;
```

```
    }
```

```
}
```

```
}
```

```
class Main {
```

```
    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

3 (10 0) consumers

Press control - c last step

10 p. all

Put : 0

10 (10 0) consumers

Initial consumer

Product writing

21 - 10 = 11 items

Got : 0

10 = 1 item

Initial producer

Put : 1

2 (21 1) items

10 - 1 = 9 items

Initial consumer

1 (21 1) items

Product writing

10 - 1 = 9

consumed : 0

got : 1

Initial producer

3 items

1 (10 1) items

consumed : 1

put : 2

10 - 1 = 9

Initial consumer

1 (10 2) items

1 (10 2) items

Product writing

1 (10 2) items

Got : 2

Initial producer

consumed : 2



consumer writing

put = 3

Infinite consumer

product avail

total = 3

consumed : 3

Run  
6/2/2024