

```
J TokenRing.java 1 X J Tring.java
J TokenRing.java > TokenRing > token
1 import java.util.concurrent.TimeUnit;
2 import java.util.Scanner;
3 public class TokenRing {
4     static int numProcesses; // Total number of processes in the ring
5     static int token = 0; // Initially, process 0 has the token
6     public static void main(String[] args) {
7         Scanner sc = new Scanner(System.in);
8         System.out.print(s:"Enter the number of node : ");
9         numProcesses = sc.nextInt();
10        while (true) {
11            // The process that holds the token can enter the Critical Section
12            System.out.println("Process " + token + " has the token.");
13
14            // ● CRITICAL SECTION (CS) STARTS
15            System.out.println("Process " + token + " is entering the Critical Section...");
16            try {
17                TimeUnit.SECONDS.sleep(timeout:5); // Simulating Critical Section execution
18            } catch (InterruptedException e) {
19                e.printStackTrace();
20            }
21            System.out.println("Process " + token + " is exiting the Critical Section.");
22            // ● CRITICAL SECTION (CS) ENDS
23
24            // Pass the token to the next process in the ring
25            token = (token + 1) % numProcesses;
26
27            try {
28                TimeUnit.SECONDS.sleep(timeout:1); // Simulating token passing delay
29            } catch (InterruptedException e) {
30                e.printStackTrace();
31            }
32        }
33    }
34 }
```

```
J TokenRing.java 1 J Tring.java X
J Tring.java > Tring
1 import java.util.Scanner;
2
3 public class Tring {
4     public static void main(String[] args) {
5         Scanner sc = new Scanner(System.in);
6         System.out.print(s:"Enter the number of processes: ");
7         int n = sc.nextInt();
8         int token = 0;
9
10        while (true) {
11            System.out.println("Token is with process " + token);
12            System.out.print(s:"Enter process to enter critical section (-1 to exit): ");
13            int process = sc.nextInt();
14
15            if (process == -1) break;
16            if (process == token) {
17                System.out.println("Process " + process + " is in critical section");
18                System.out.println("Process " + process + " is exiting critical section");
19            } else {
20                System.out.println("Process " + process + " does not have the token, cannot enter CS");
21            }
22            token = (token + 1) % n;
23        }
24        sc.close();
25    }
26 }
```