

```

1 // Implementation of Token Ring Algorithm
2- import java.io.*;
3 import java.lang.*;
4
5- class Token Ring {
6- public static void main(String args[]) throws Throwable {
7     DataInputStream dis = new DataInputStream(System.in);
8     System.out.println("Enter the num of nodes:");
9     int n = Integer.parseInt(dis.readLine());
10    // Decides the number of nodes forming the ring
11    int token = 0;
12    int ch = 1;
13    for (int i = 0; i < n; i++)
14        System.out.print(" " + i);
15    System.out.println(" " + 0);
16-    try {
17-        while (ch == 1) {
18            System.out.println("Enter sender:");
19            int s = Integer.parseInt(dis.readLine());
20            System.out.println("Enter receiver:");
21            int r = Integer.parseInt(dis.readLine());
22            System.out.println("Enter Data:");
23            String d = dis.readLine();
24            System.out.print("Token passing:");
25            for (int i = token; i != s; i++)
26
27                System.out.print(" " + i + "->");
28            System.out.println(" " + s);
29            System.out.println("Sender " + s + "sending data: " + d);
30            for (int i = s + 1; i != r; i = (i + 1) % n)
31                System.out.println("data " + d + " forwarded by " + i);
32            System.out.println("Receiver " + r + "received data: " + d);
33            token = s;
34        } catch (Exception e) {
35        }
36    }
37 }
38

```

eclipse-workspace - algo/src/algo/TokenRing.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Console X

TokenRing [Java Application] C:\Users\Satish\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64_17.0.2.v20220201-1208\jre\bin\

```

Enter the num of nodes:
5
0 1 2 3 4 0
Enter sender:
1
Enter receiver:
4
Enter Data:
ASKAAAA
Token passing: 0-> 1
Sender 1sending data: ASKAAAA
data ASKAAAA forwarded by 2
data ASKAAAA forwarded by 3
Receiver 4received data: ASKAAAA
Enter sender:
4
Enter receiver:
1
Enter Data:
BBBBB
Token passing: 1-> 2-> 3-> 4
Sender 4sending data: BBBBB
data BBBBB forwarded by 5
Receiver 1received data: BBBBB

```