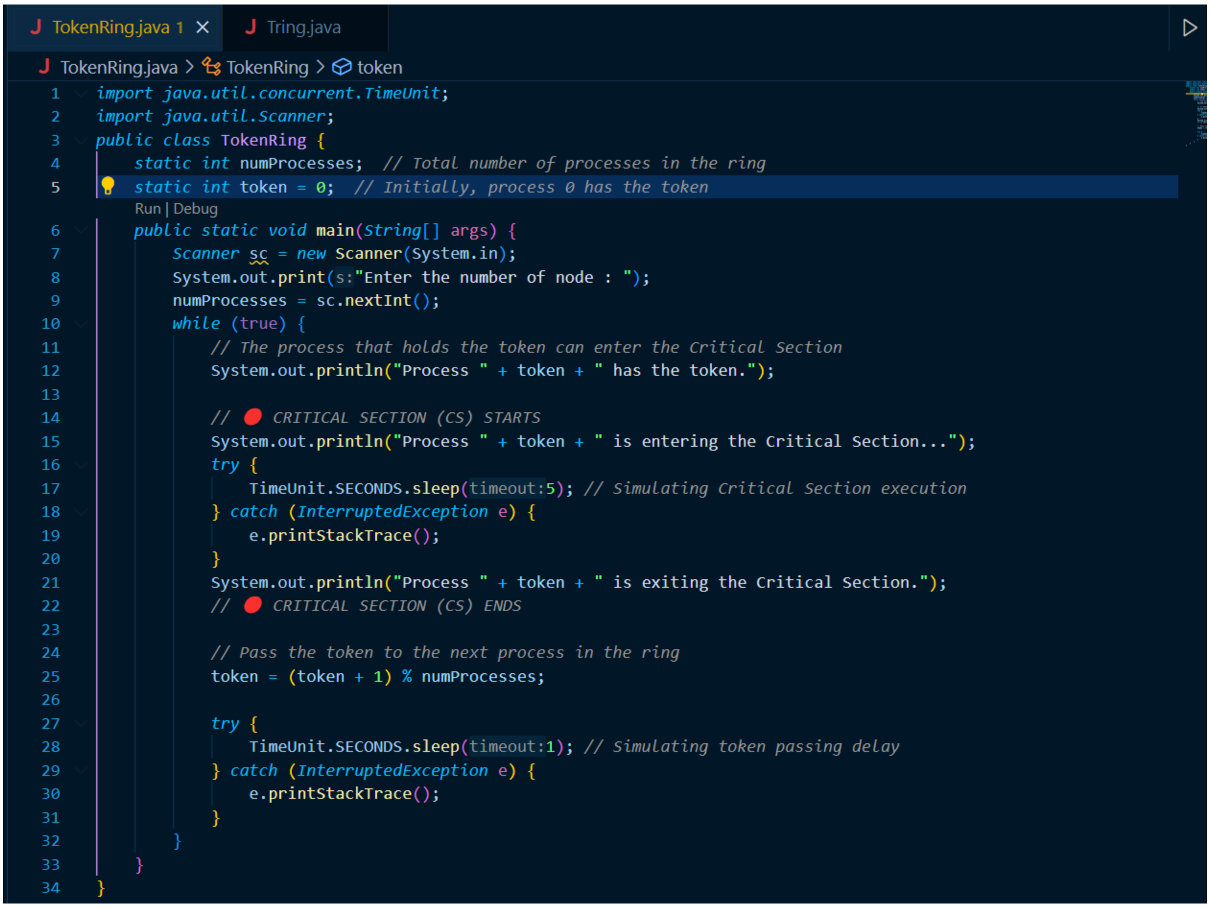
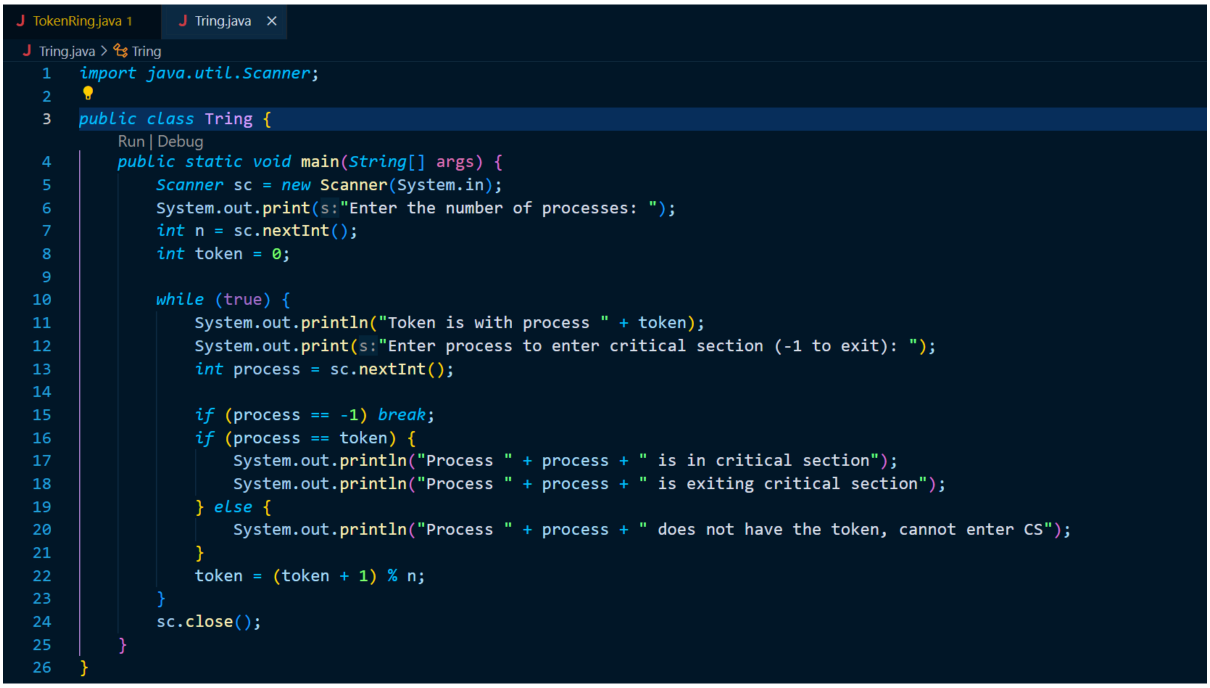


```java
import java.util.concurrent.TimeUnit;
import java.util.Scanner;
public class TokenRing {
    static int numProcesses;  // Total number of processes in the ring
    static int token = 0;  // Initially, process 0 has the token
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print(s:"Enter the number of node : ");
        numProcesses = sc.nextInt();
        while (true) {
            // The process that holds the token can enter the Critical Section
            System.out.println("Process " + token + " has the token.");

            // 🔴 CRITICAL SECTION (CS) STARTS
            System.out.println("Process " + token + " is entering the Critical Section...");
            try {
                TimeUnit.SECONDS.sleep(timeout:5); // Simulating Critical Section execution
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
            System.out.println("Process " + token + " is exiting the Critical Section.");
            // 🔴 CRITICAL SECTION (CS) ENDS

            // Pass the token to the next process in the ring
            token = (token + 1) % numProcesses;

            try {
                TimeUnit.SECONDS.sleep(timeout:1); // Simulating token passing delay
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }
}
```



```java
import java.util.Scanner;

public class Tring {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print(s:"Enter the number of processes: ");
        int n = sc.nextInt();
        int token = 0;

        while (true) {
            System.out.println("Token is with process " + token);
            System.out.print(s:"Enter process to enter critical section (-1 to exit): ");
            int process = sc.nextInt();

            if (process == -1) break;
            if (process == token) {
                System.out.println("Process " + process + " is in critical section");
                System.out.println("Process " + process + " is exiting critical section");
            } else {
                System.out.println("Process " + process + " does not have the token, cannot enter CS");
            }
            token = (token + 1) % n;
        }
        sc.close();
    }
}
```