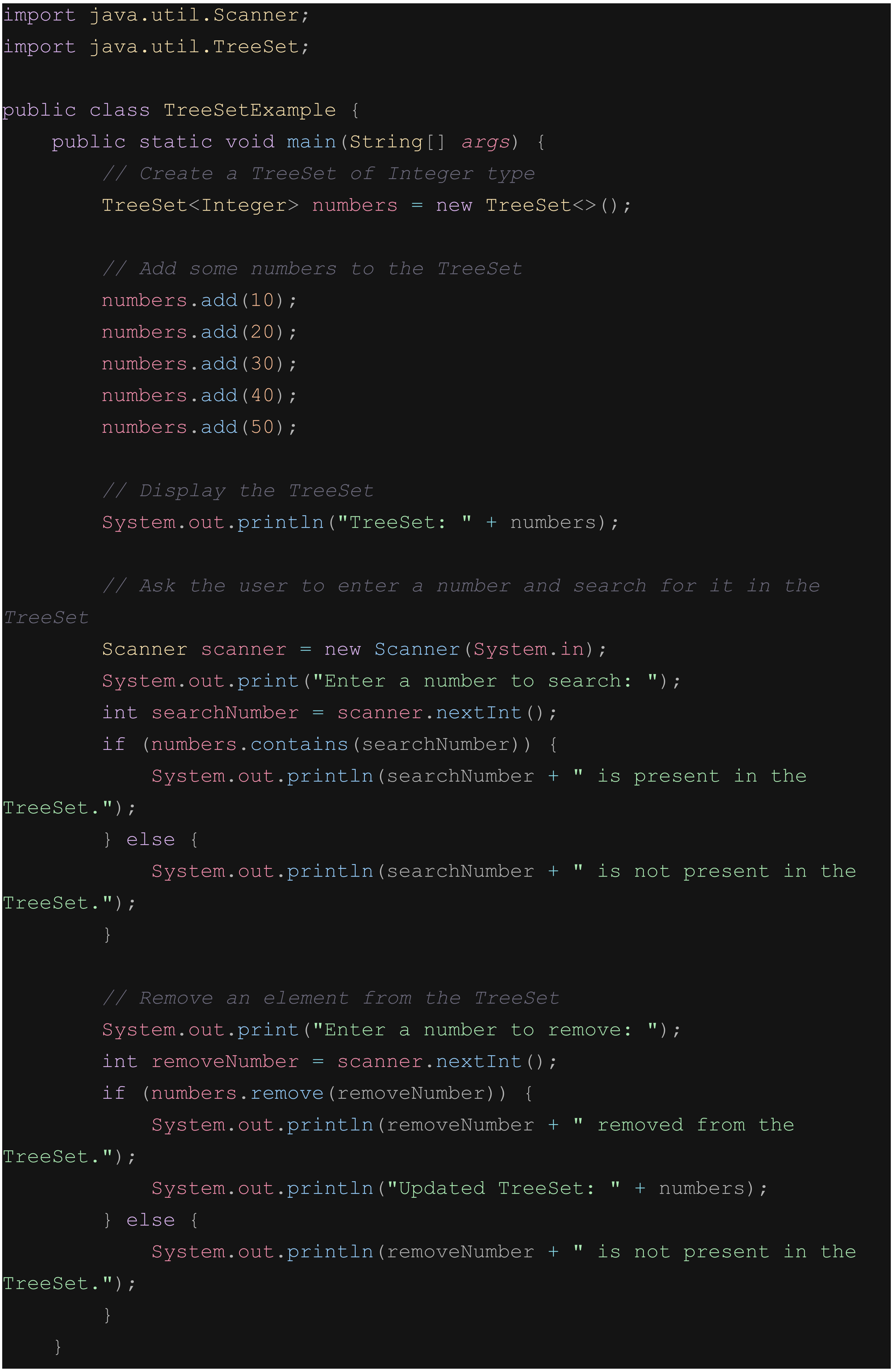
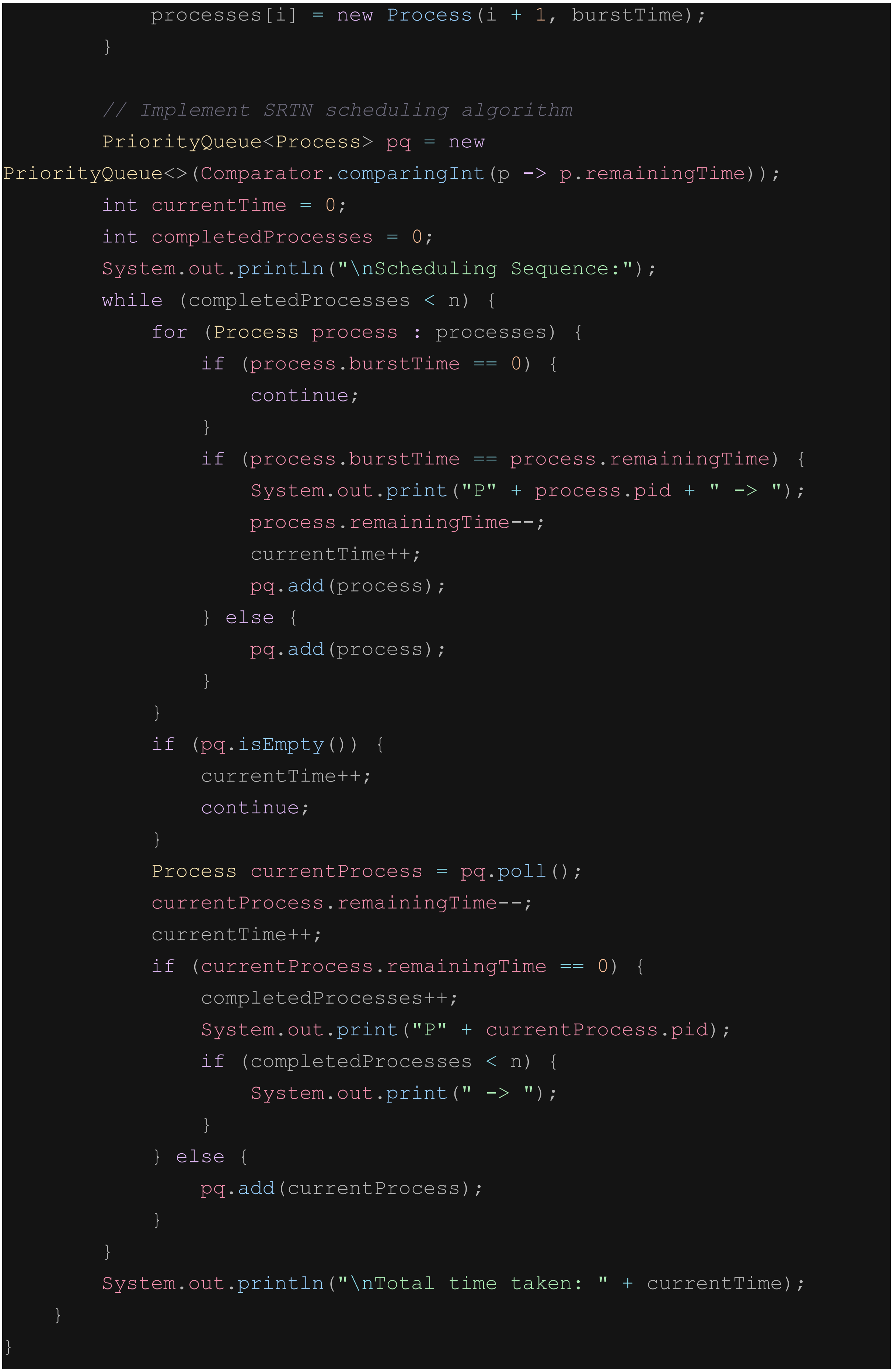


```
import java.util.Scanner;
import java.util.TreeSet;

public class TreeSetExample {
    public static void main(String[] args) {
        // Create a TreeSet of Integer type
        TreeSet<Integer> numbers = new TreeSet<>();

        // Add some numbers to the TreeSet
        numbers.add(10);
        numbers.add(20);
        numbers.add(30);
        numbers.add(40);
        numbers.add(50);

        // Display the TreeSet
        System.out.println("TreeSet: " + numbers);

        // Ask the user to enter a number and search for it in the
TreeSet
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a number to search: ");
        int searchNumber = scanner.nextInt();
        if (numbers.contains(searchNumber)) {
            System.out.println(searchNumber + " is present in the
TreeSet.");
        } else {
            System.out.println(searchNumber + " is not present in the
TreeSet.");
        }

        // Remove an element from the TreeSet
        System.out.print("Enter a number to remove: ");
        int removeNumber = scanner.nextInt();
        if (numbers.remove(removeNumber)) {
            System.out.println(removeNumber + " removed from the
TreeSet.");
            System.out.println("Updated TreeSet: " + numbers);
        } else {
            System.out.println(removeNumber + " is not present in the
TreeSet.");
        }
    }
}
```

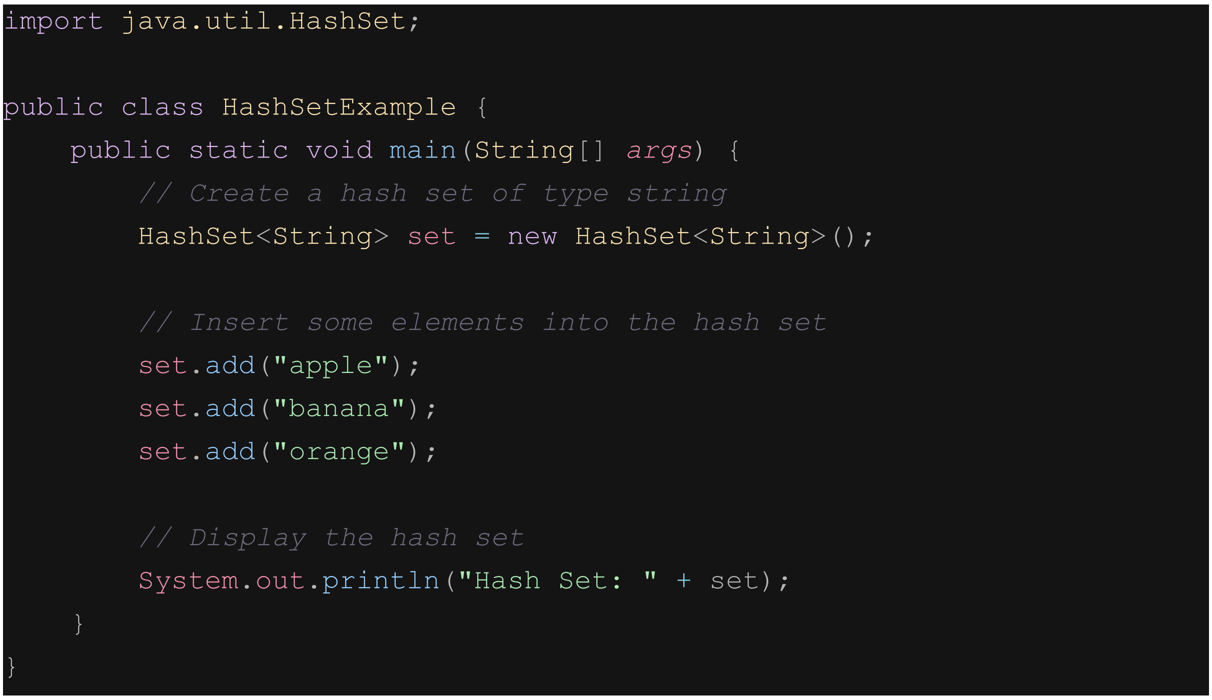
```
import java.util.TreeMap;

class Address {
    private String plotNo;
    private String at;
    private String post;

    public Address(String plotNo, String at, String post) {
        this.plotNo = plotNo;
        this.at = at;
        this.post = post;
    }

    public String getPlotNo() {
        return plotNo;
    }

    public String getAt() {
        return at;
    }

    public String getPost() {
        return post;
    }
}

public class TreeMapExample {
    public static void main(String[] args) {
        // Create a TreeMap with keys as names and values as addresses
        TreeMap<String, Address> addressBook = new TreeMap<>();

        // Insert some data into the TreeMap
        addressBook.put("John Doe", new Address("123", "Main Street",
"New York"));
        addressBook.put("Jane Smith", new Address("456", "High Street",
"London"));
```

```
        addressBook.put("Bob Johnson", new Address("789", "Elm Street",
"Los Angeles"));

        // Display the TreeMap
        for (String name : addressBook.keySet()) {
            Address address = addressBook.get(name);
            System.out.println(name + ": Plot No. " +
address.getPlotNo() + ", At " + address.getAt() + ", Post " +
address.getPost());
        }
    }
}
```

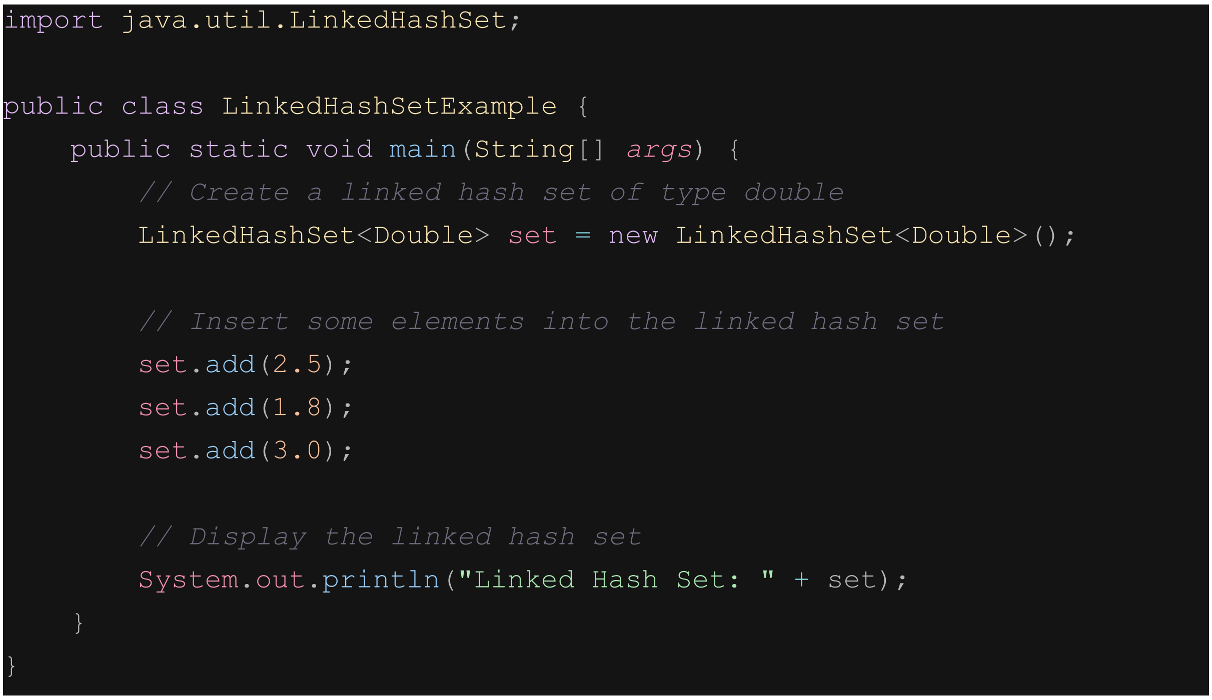
```
import java.util.*;

class Process {
    int pid;
    int burstTime;
    int remainingTime;

    public Process(int pid, int burstTime) {
        this.pid = pid;
        this.burstTime = burstTime;
        this.remainingTime = burstTime;
    }
}

public class SRTN {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        // Take input as burst time of N processes
        System.out.print("Enter the number of processes: ");
        int n = scanner.nextInt();
        Process[] processes = new Process[n];
        for (int i = 0; i < n; i++) {
            System.out.print("Enter burst time for process " + (i + 1)
+ ": ");
            int burstTime = scanner.nextInt();
```



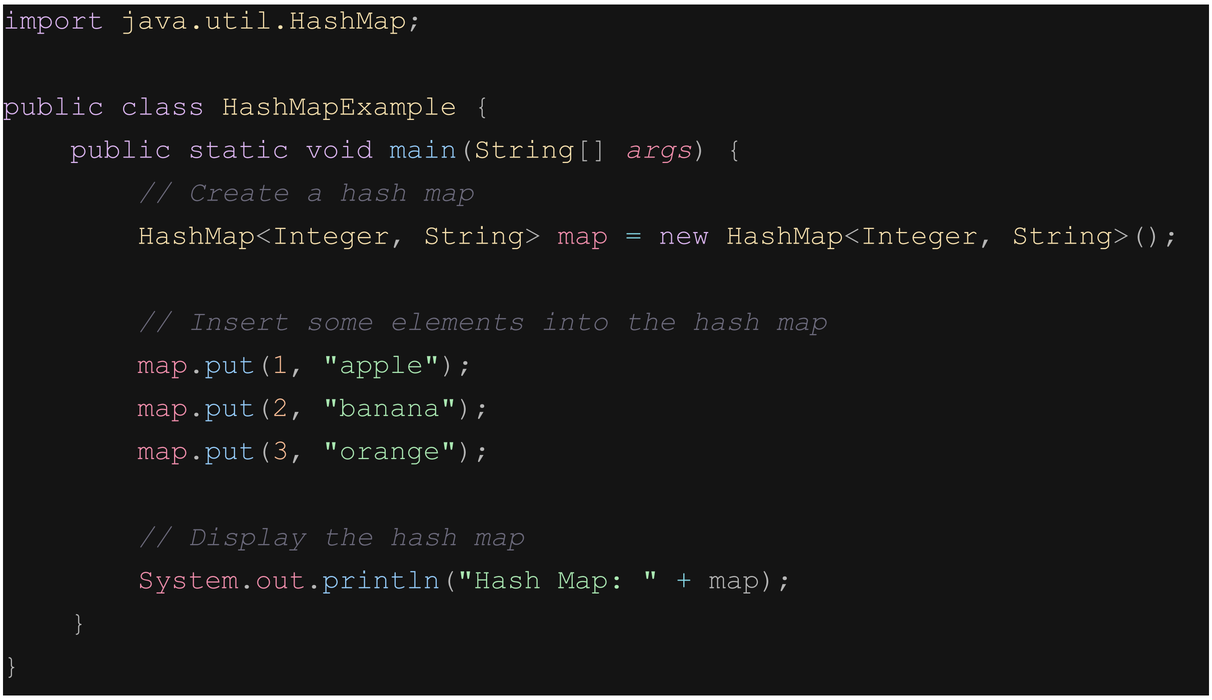
```java
            processes[i] = new Process(i + 1, burstTime);
        }

        // Implement SRTN scheduling algorithm
        PriorityQueue<Process> pq = new
PriorityQueue<>(Comparator.comparingInt(p -> p.remainingTime));
        int currentTime = 0;
        int completedProcesses = 0;
        System.out.println("\nScheduling Sequence:");
        while (completedProcesses < n) {
            for (Process process : processes) {
                if (process.burstTime == 0) {
                    continue;
                }
                if (process.burstTime == process.remainingTime) {
                    System.out.print("P" + process.pid + " -> ");
                    process.remainingTime--;
                    currentTime++;
                    pq.add(process);
                } else {
                    pq.add(process);
                }
            }
            if (pq.isEmpty()) {
                currentTime++;
                continue;
            }
            Process currentProcess = pq.poll();
            currentProcess.remainingTime--;
            currentTime++;
            if (currentProcess.remainingTime == 0) {
                completedProcesses++;
                System.out.print("P" + currentProcess.pid);
                if (completedProcesses < n) {
                    System.out.print(" -> ");
                }
            } else {
                pq.add(currentProcess);
            }
        }
        System.out.println("\nTotal time taken: " + currentTime);
    }
}
```



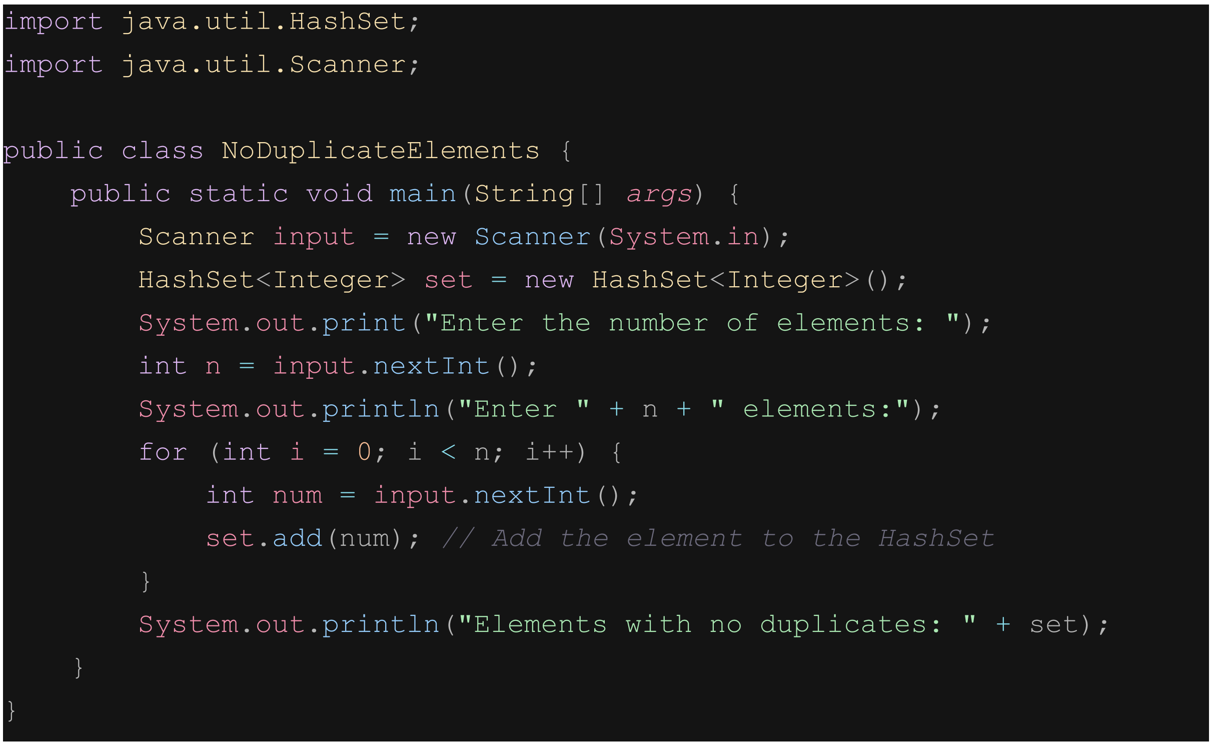
```java
import java.util.HashSet;

public class HashSetExample {
    public static void main(String[] args) {
        // Create a hash set of type string
        HashSet<String> set = new HashSet<String>();

        // Insert some elements into the hash set
        set.add("apple");
        set.add("banana");
        set.add("orange");

        // Display the hash set
        System.out.println("Hash Set: " + set);
    }
}
```



```java
import java.util.LinkedHashSet;

public class LinkedHashSetExample {
    public static void main(String[] args) {
        // Create a linked hash set of type double
        LinkedHashSet<Double> set = new LinkedHashSet<Double>();

        // Insert some elements into the linked hash set
        set.add(2.5);
        set.add(1.8);
        set.add(3.0);

        // Display the linked hash set
        System.out.println("Linked Hash Set: " + set);
    }
}
```



```java
import java.util.HashMap;

public class HashMapExample {
    public static void main(String[] args) {
        // Create a hash map
        HashMap<Integer, String> map = new HashMap<Integer, String>();

        // Insert some elements into the hash map
        map.put(1, "apple");
        map.put(2, "banana");
        map.put(3, "orange");

        // Display the hash map
        System.out.println("Hash Map: " + map);
    }
}
```



```java
import java.util.HashSet;
import java.util.Scanner;

public class NoDuplicateElements {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        HashSet<Integer> set = new HashSet<Integer>();
        System.out.print("Enter the number of elements: ");
        int n = input.nextInt();
        System.out.println("Enter " + n + " elements:");
        for (int i = 0; i < n; i++) {
            int num = input.nextInt();
            set.add(num); // Add the element to the HashSet
        }
        System.out.println("Elements with no duplicates: " + set);
    }
}
```