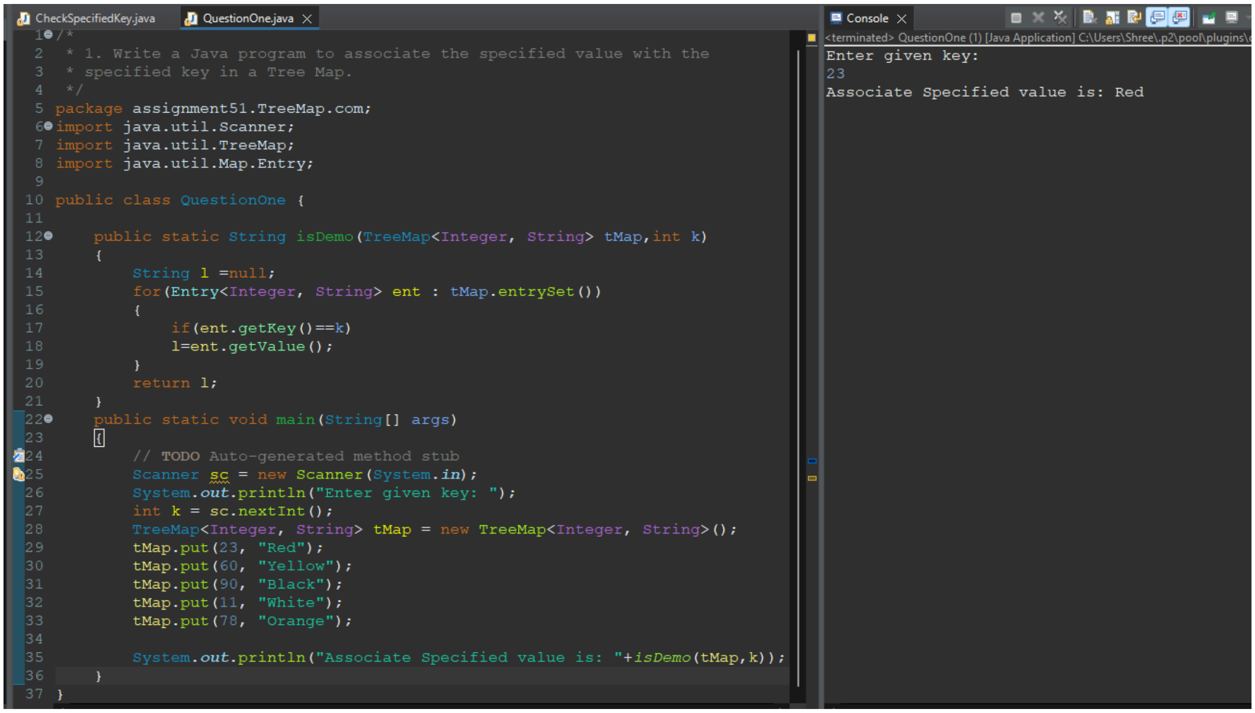
Assignment No:-51

Name:-Suryawanshi Sangramsingh Sambhaji

Batch: - Delta - DCA (Java) 2024 Date:-23/7/2024

1. Write a Java program to associate the specified value with the specified key in a Tree Map.

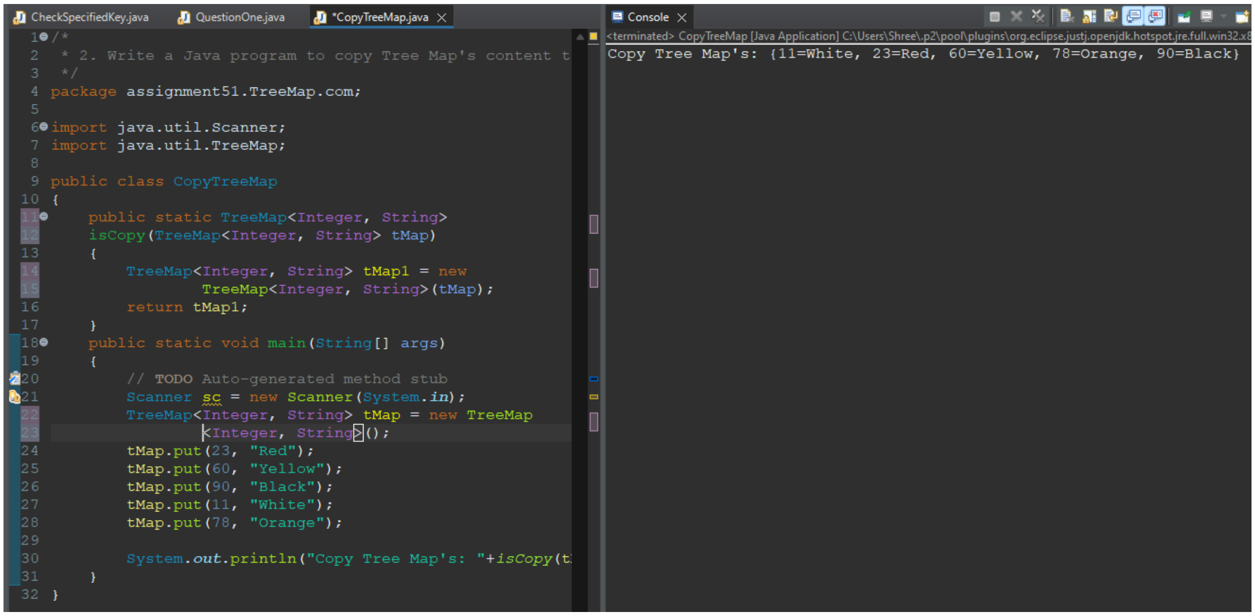


CheckSpecifiedKey.java
QuestionOne.java
Console
<terminated> QuestionOne (1) [Java Application] C:\Users\Shree\.p2\pool\plugins\

```java
/*
 * 1. Write a Java program to associate the specified value with the
 * specified key in a Tree Map.
 */
package assignment51.TreeMap.com;
import java.util.Scanner;
import java.util.TreeMap;
import java.util.Map.Entry;

public class QuestionOne {

    public static String isDemo(TreeMap<Integer, String> tMap,int k)
    {
        String l =null;
        for(Entry<Integer, String> ent : tMap.entrySet())
        {
            if(ent.getKey()==k)
            l=ent.getValue();
        }
        return l;
    }
    public static void main(String[] args)
    {
        // TODO Auto-generated method stub
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter given key: ");
        int k = sc.nextInt();
        TreeMap<Integer, String> tMap = new TreeMap<Integer, String>();
        tMap.put(23, "Red");
        tMap.put(60, "Yellow");
        tMap.put(90, "Black");
        tMap.put(11, "White");
        tMap.put(78, "Orange");

        System.out.println("Associate Specified value is: "+isDemo(tMap,k));
    }
}
```

```
Enter given key:
23
Associate Specified value is: Red
```

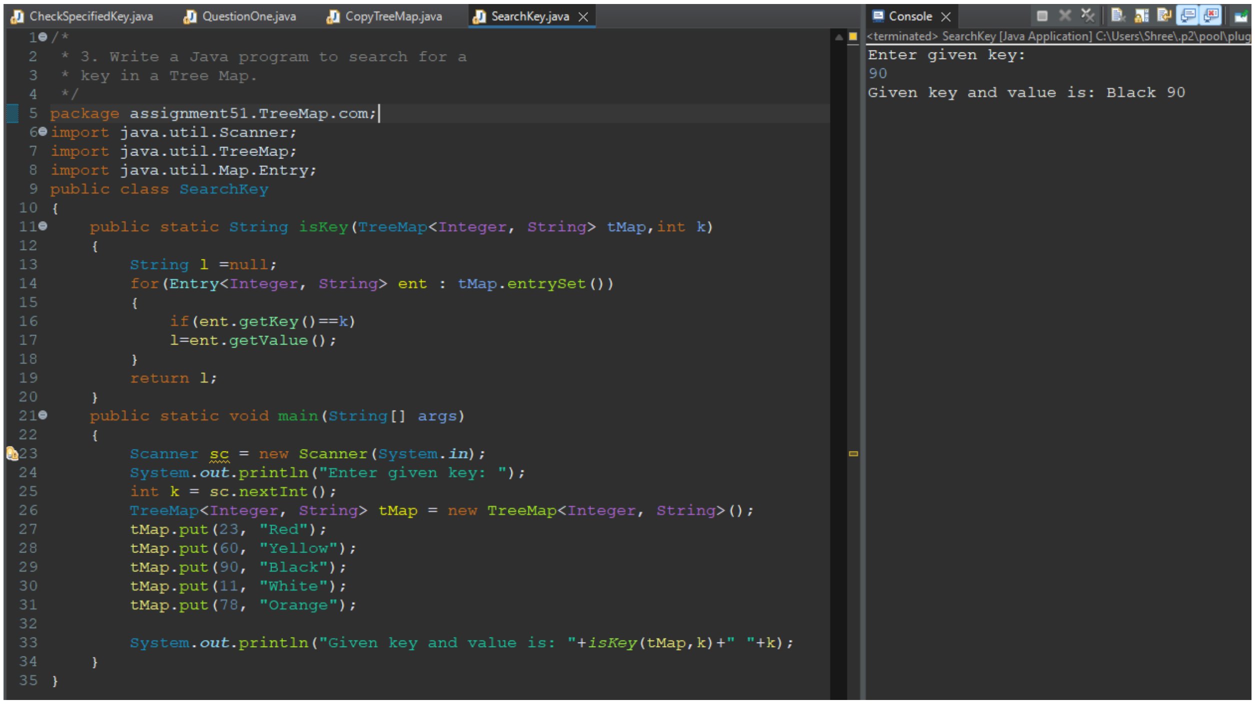
2. Write a Java program to copy Tree Map's content to another Tree Map.



```java
/*
 * 2. Write a Java program to copy Tree Map's content t
 */
package assignment51.TreeMap.com;

import java.util.Scanner;
import java.util.TreeMap;

public class CopyTreeMap
{
    public static TreeMap<Integer, String>
    isCopy(TreeMap<Integer, String> tMap)
    {
        TreeMap<Integer, String> tMap1 = new
                TreeMap<Integer, String>(tMap);
        return tMap1;
    }
    public static void main(String[] args)
    {
        // TODO Auto-generated method stub
        Scanner sc = new Scanner(System.in);
        TreeMap<Integer, String> tMap = new TreeMap
                <Integer, String>();
        tMap.put(23, "Red");
        tMap.put(60, "Yellow");
        tMap.put(90, "Black");
        tMap.put(11, "White");
        tMap.put(78, "Orange");

        System.out.println("Copy Tree Map's: "+isCopy(t
    }
}
```

```
<terminated> CopyTreeMap [Java Application] C:\Users\Shree\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x8
Copy Tree Map's: {11=White, 23=Red, 60=Yellow, 78=Orange, 90=Black}
```

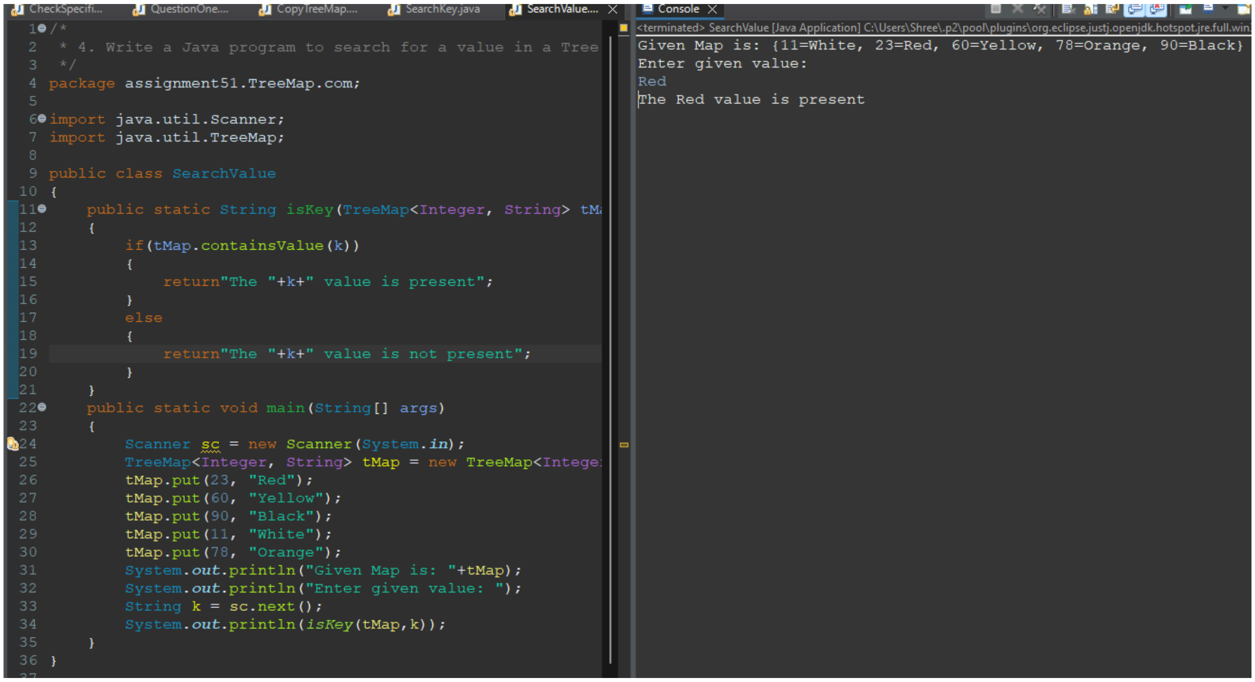
3. Write a Java program to search for a key in a Tree Map.



```java
/*
 * 3. Write a Java program to search for a
 * key in a Tree Map.
 */
package assignment51.TreeMap.com;
import java.util.Scanner;
import java.util.TreeMap;
import java.util.Map.Entry;
public class SearchKey
{
    public static String isKey(TreeMap<Integer, String> tMap,int k)
    {
        String l =null;
        for(Entry<Integer, String> ent : tMap.entrySet())
        {
            if(ent.getKey()==k)
            l=ent.getValue();
        }
        return l;
    }
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter given key: ");
        int k = sc.nextInt();
        TreeMap<Integer, String> tMap = new TreeMap<Integer, String>();
        tMap.put(23, "Red");
        tMap.put(60, "Yellow");
        tMap.put(90, "Black");
        tMap.put(11, "White");
        tMap.put(78, "Orange");

        System.out.println("Given key and value is: "+isKey(tMap,k)+" "+k);
    }
}
```

```
<terminated> SearchKey [Java Application] C:\Users\Shree\.p2\pool\plug
Enter given key:
90
Given key and value is: Black 90
```

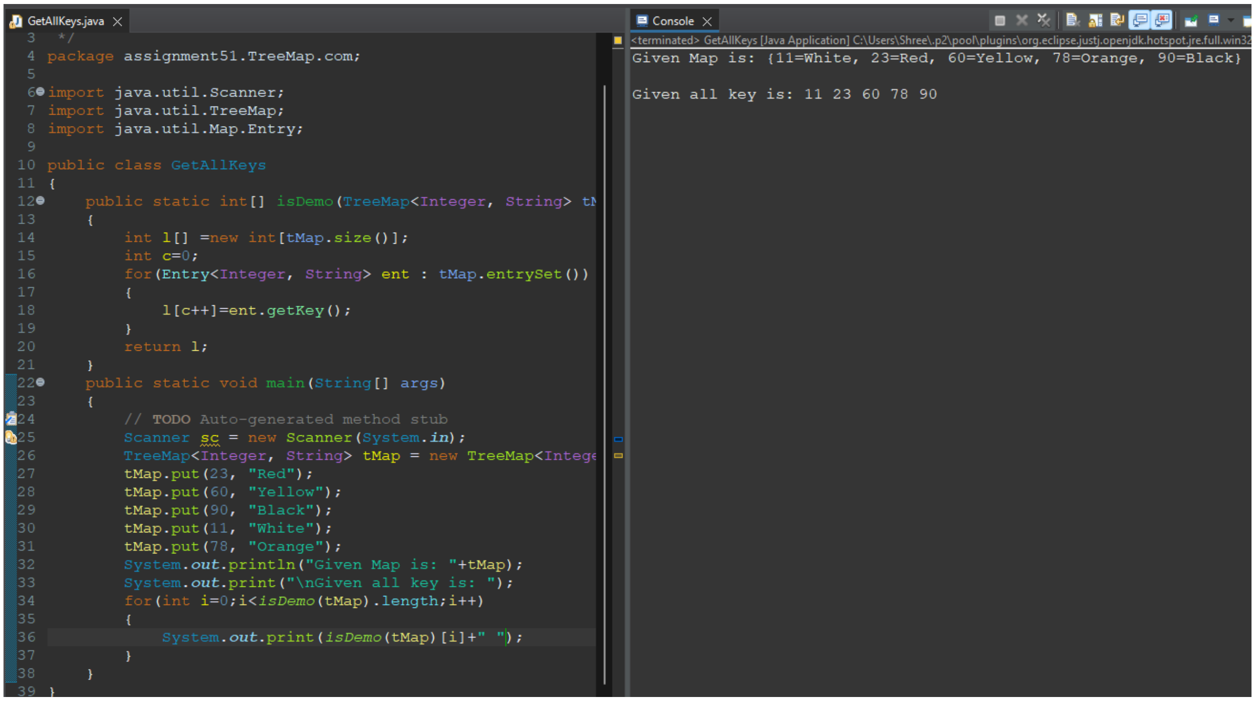
4. Write a Java program to search for a value in a Tree Map.



```java
/*
 * 4. Write a Java program to search for a value in a Tree
 */
package assignment51.TreeMap.com;

import java.util.Scanner;
import java.util.TreeMap;

public class SearchValue
{
    public static String isKey(TreeMap<Integer, String> tMa
    {
        if(tMap.containsValue(k))
        {
            return"The "+k+" value is present";
        }
        else
        {
            return"The "+k+" value is not present";
        }
    }
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        TreeMap<Integer, String> tMap = new TreeMap<Intege
        tMap.put(23, "Red");
        tMap.put(60, "Yellow");
        tMap.put(90, "Black");
        tMap.put(11, "White");
        tMap.put(78, "Orange");
        System.out.println("Given Map is: "+tMap);
        System.out.println("Enter given value: ");
        String k = sc.next();
        System.out.println(isKey(tMap,k));
    }
}
```

```
<terminated> SearchValue [Java Application] C:\Users\Shree\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win
Given Map is: {11=White, 23=Red, 60=Yellow, 78=Orange, 90=Black}
Enter given value:
Red
The Red value is present
```

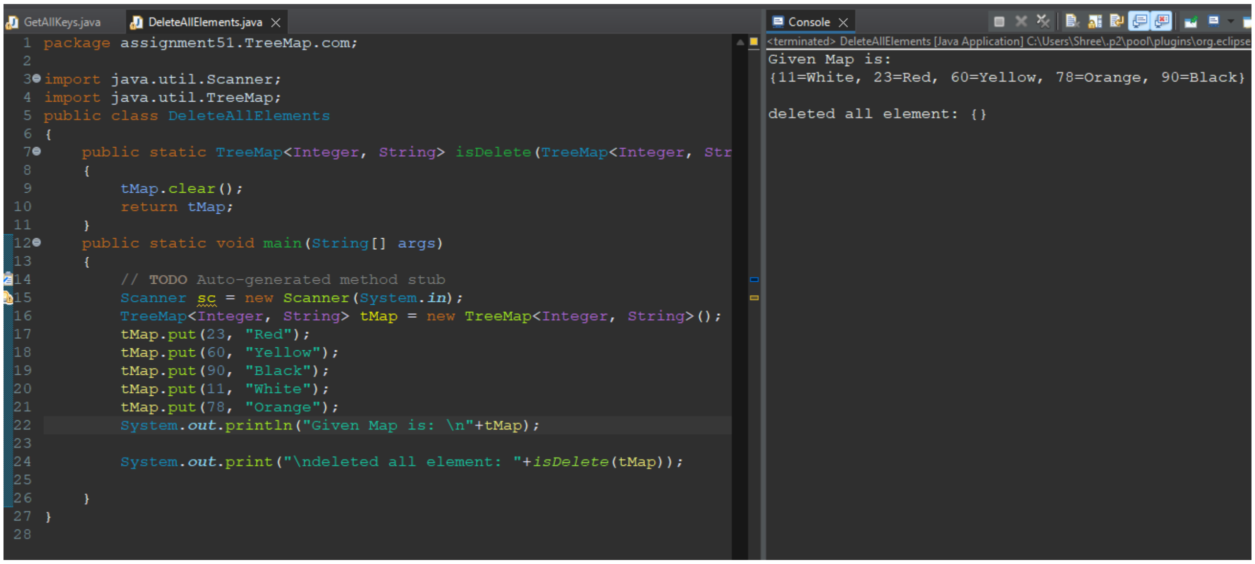
5. Write a Java program to get all keys from a Tree Map.



```java
 */
package assignment51.TreeMap.com;

import java.util.Scanner;
import java.util.TreeMap;
import java.util.Map.Entry;

public class GetAllKeys
{
    public static int[] isDemo(TreeMap<Integer, String> tM
    {
        int l[] =new int[tMap.size()];
        int c=0;
        for(Entry<Integer, String> ent : tMap.entrySet())
        {
            l[c++]=ent.getKey();
        }
        return l;
    }
    public static void main(String[] args)
    {
        // TODO Auto-generated method stub
        Scanner sc = new Scanner(System.in);
        TreeMap<Integer, String> tMap = new TreeMap<Intege
        tMap.put(23, "Red");
        tMap.put(60, "Yellow");
        tMap.put(90, "Black");
        tMap.put(11, "White");
        tMap.put(78, "Orange");
        System.out.println("Given Map is: "+tMap);
        System.out.print("\nGiven all key is: ");
        for(int i=0;i<isDemo(tMap).length;i++)
        {
            System.out.print(isDemo(tMap)[i]+" ");
        }
    }
}
```

```
<terminated> GetAllKeys [Java Application] C:\Users\Shree\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32
Given Map is: {11=White, 23=Red, 60=Yellow, 78=Orange, 90=Black}

Given all key is: 11 23 60 78 90
```

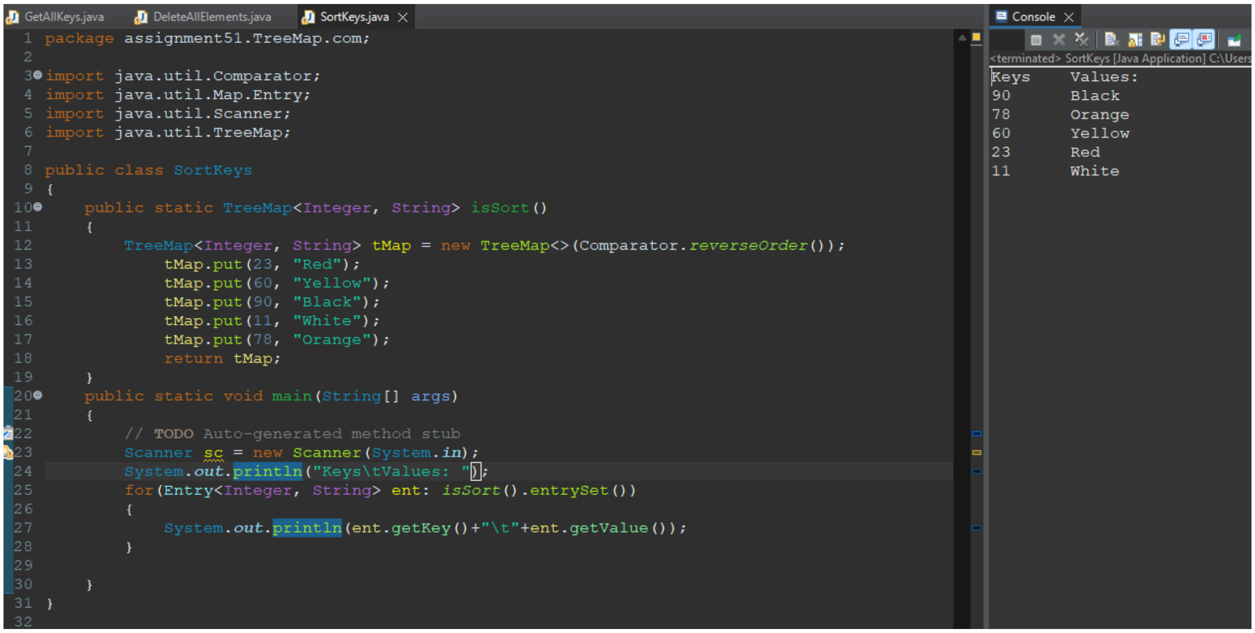
6. Write a Java program to delete all elements from a Tree Map.



```java
package assignment51.TreeMap.com;

import java.util.Scanner;
import java.util.TreeMap;
public class DeleteAllElements
{
    public static TreeMap<Integer, String> isDelete(TreeMap<Integer, Str
    {
        tMap.clear();
        return tMap;
    }
    public static void main(String[] args)
    {
        // TODO Auto-generated method stub
        Scanner sc = new Scanner(System.in);
        TreeMap<Integer, String> tMap = new TreeMap<Integer, String>();
        tMap.put(23, "Red");
        tMap.put(60, "Yellow");
        tMap.put(90, "Black");
        tMap.put(11, "White");
        tMap.put(78, "Orange");
        System.out.println("Given Map is: \n"+tMap);

        System.out.print("\ndeleted all element: "+isDelete(tMap));

    }
}
```

```
<terminated> DeleteAllElements [Java Application] C:\Users\Shree\.p2\pool\plugins\org.eclipse
Given Map is:
{11=White, 23=Red, 60=Yellow, 78=Orange, 90=Black}

deleted all element: {}
```

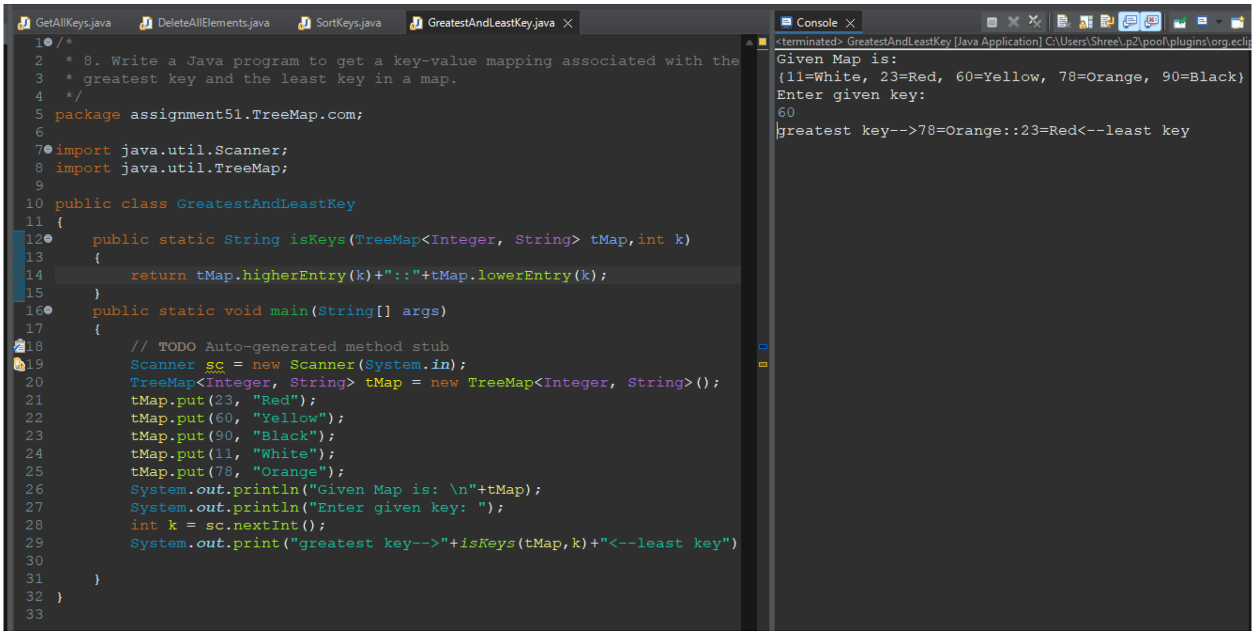
7. Write a Java program to sort keys in a Tree Map by using a comparator.



```java
package assignment51.TreeMap.com;

import java.util.Comparator;
import java.util.Map.Entry;
import java.util.Scanner;
import java.util.TreeMap;

public class SortKeys
{
    public static TreeMap<Integer, String> isSort()
    {
        TreeMap<Integer, String> tMap = new TreeMap<>(Comparator.reverseOrder());
            tMap.put(23, "Red");
            tMap.put(60, "Yellow");
            tMap.put(90, "Black");
            tMap.put(11, "White");
            tMap.put(78, "Orange");
            return tMap;
    }
    public static void main(String[] args)
    {
        // TODO Auto-generated method stub
        Scanner sc = new Scanner(System.in);
        System.out.println("Keys\tValues: ");
        for(Entry<Integer, String> ent: isSort().entrySet())
        {
            System.out.println(ent.getKey()+"\t"+ent.getValue());
        }

    }
}
```

```
<terminated> SortKeys [Java Application] C:\Users
Keys    Values:
90      Black
78      Orange
60      Yellow
23      Red
11      White
```

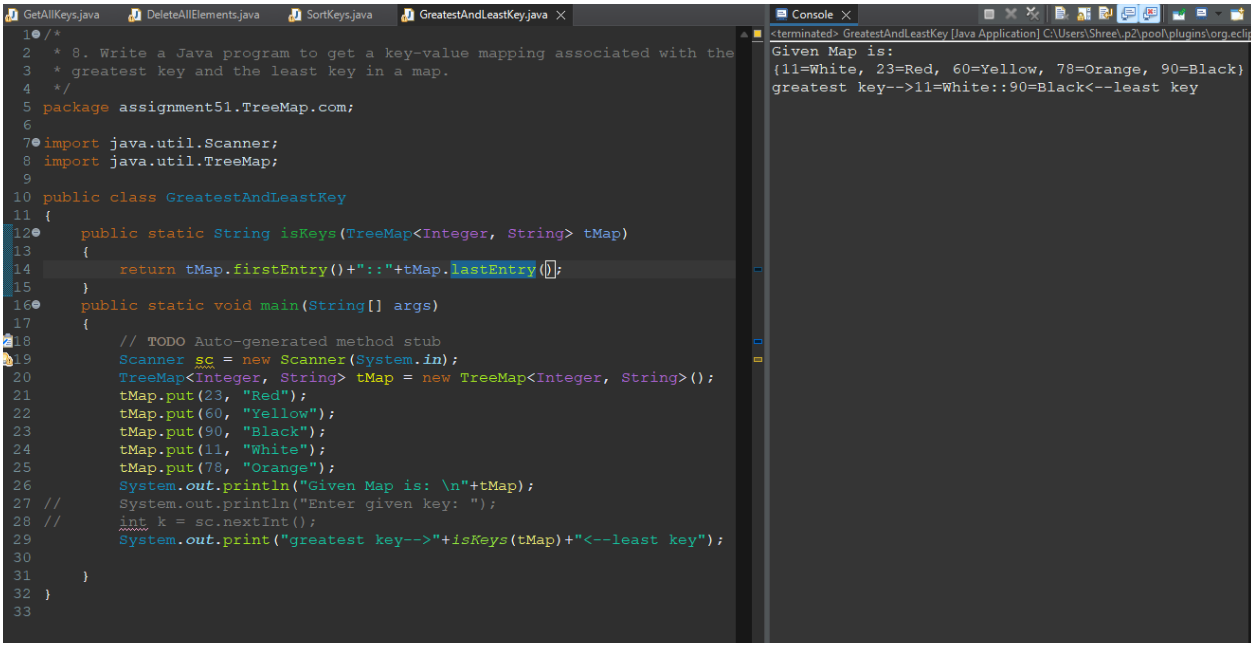
8. Write a Java program to get a key-value mapping associated with the greatest key and the least key in a map.


GetAllKeys.java
DeleteAllElements.java
SortKeys.java
GreatestAndLeastKey.java
Console
<terminated> GreatestAndLeastKey [Java Application] C:\Users\Shree\.p2\pool\plugins\org.eclip

```java
/*
 * 8. Write a Java program to get a key-value mapping associated with the
 * greatest key and the least key in a map.
 */
package assignment51.TreeMap.com;

import java.util.Scanner;
import java.util.TreeMap;

public class GreatestAndLeastKey
{
    public static String isKeys(TreeMap<Integer, String> tMap,int k)
    {
        return tMap.higherEntry(k)+"::"+tMap.lowerEntry(k);
    }
    public static void main(String[] args)
    {
        // TODO Auto-generated method stub
        Scanner sc = new Scanner(System.in);
        TreeMap<Integer, String> tMap = new TreeMap<Integer, String>();
        tMap.put(23, "Red");
        tMap.put(60, "Yellow");
        tMap.put(90, "Black");
        tMap.put(11, "White");
        tMap.put(78, "Orange");
        System.out.println("Given Map is: \n"+tMap);
        System.out.println("Enter given key: ");
        int k = sc.nextInt();
        System.out.print("greatest key-->"+isKeys(tMap,k)+"<--least key")

    }
}
```

```
Given Map is:
{11=White, 23=Red, 60=Yellow, 78=Orange, 90=Black}
Enter given key:
60
greatest key-->78=Orange::23=Red<--least key
```

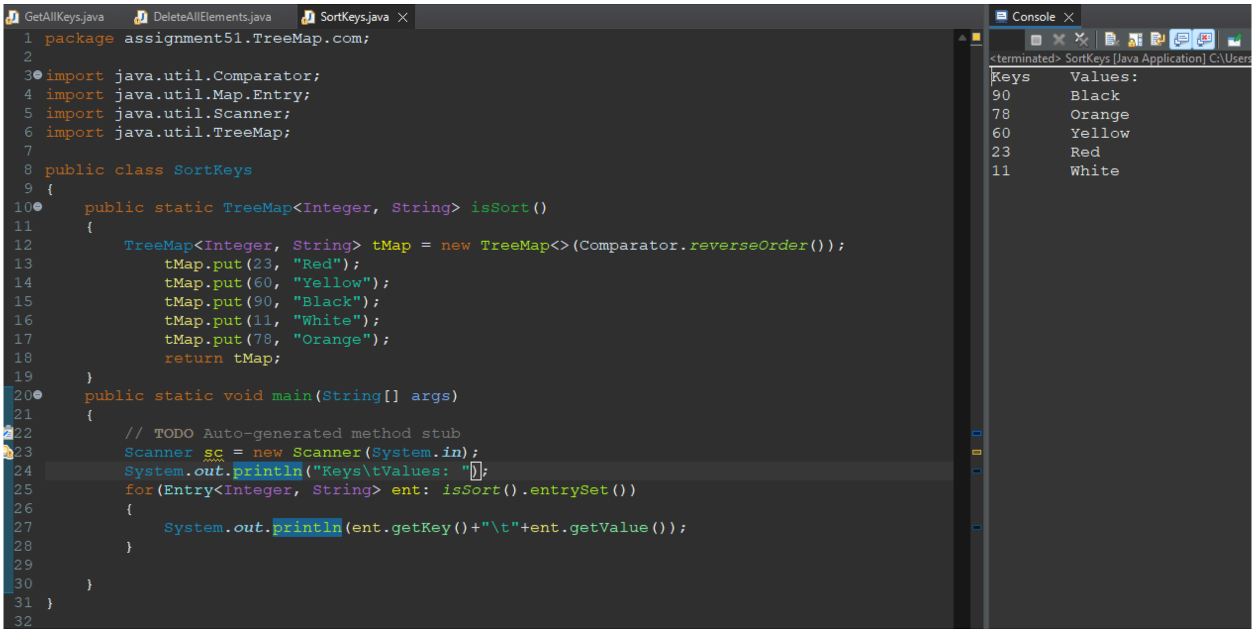
9. Write a Java program to get the first (lowest) key and the last (highest) key currently in a map.


GetAllKeys.java
DeleteAllElements.java
SortKeys.java
GreatestAndLeastKey.java
Console
<terminated> GreatestAndLeastKey [Java Application] C:\Users\Shree\.p2\pool\plugins\org.eclip

```java
/*
 * 8. Write a Java program to get a key-value mapping associated with the
 * greatest key and the least key in a map.
 */
package assignment51.TreeMap.com;

import java.util.Scanner;
import java.util.TreeMap;

public class GreatestAndLeastKey
{
    public static String isKeys(TreeMap<Integer, String> tMap)
    {
        return tMap.firstEntry()+"::"+tMap.lastEntry();
    }
    public static void main(String[] args)
    {
        // TODO Auto-generated method stub
        Scanner sc = new Scanner(System.in);
        TreeMap<Integer, String> tMap = new TreeMap<Integer, String>();
        tMap.put(23, "Red");
        tMap.put(60, "Yellow");
        tMap.put(90, "Black");
        tMap.put(11, "White");
        tMap.put(78, "Orange");
        System.out.println("Given Map is: \n"+tMap);
//      System.out.println("Enter given key: ");
//      int k = sc.nextInt();
        System.out.print("greatest key-->"+isKeys(tMap)+"<--least key");

    }
}
```

```
Given Map is:
{11=White, 23=Red, 60=Yellow, 78=Orange, 90=Black}
greatest key-->11=White::90=Black<--least key
```

10. Write a Java program to get a reverse order view of the keys contained in a given map.

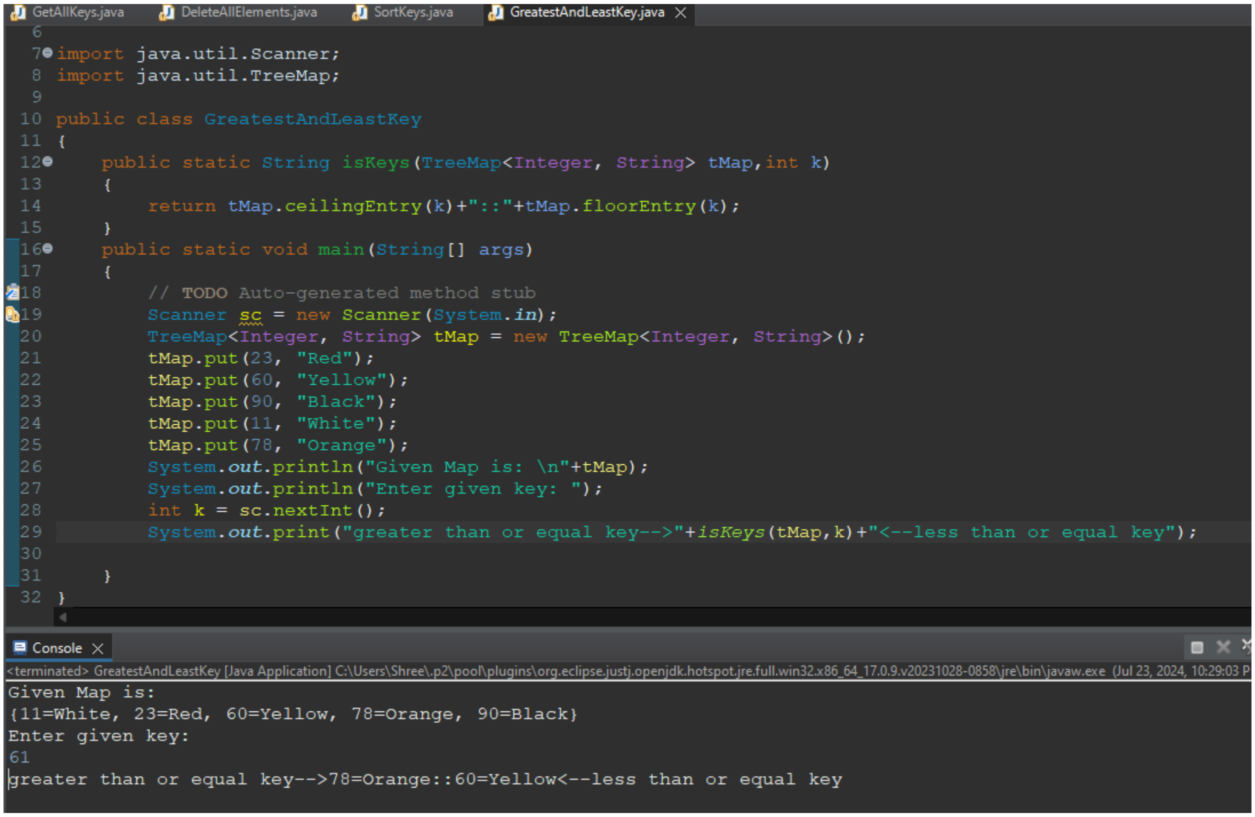


```java
package assignment51.TreeMap.com;

import java.util.Comparator;
import java.util.Map.Entry;
import java.util.Scanner;
import java.util.TreeMap;

public class SortKeys
{
    public static TreeMap<Integer, String> isSort()
    {
        TreeMap<Integer, String> tMap = new TreeMap<>(Comparator.reverseOrder());
            tMap.put(23, "Red");
            tMap.put(60, "Yellow");
            tMap.put(90, "Black");
            tMap.put(11, "White");
            tMap.put(78, "Orange");
            return tMap;
    }
    public static void main(String[] args)
    {
        // TODO Auto-generated method stub
        Scanner sc = new Scanner(System.in);
        System.out.println("Keys\tValues: ");
        for(Entry<Integer, String> ent: isSort().entrySet())
        {
            System.out.println(ent.getKey()+"\t"+ent.getValue());
        }

    }
}
```

```
<terminated> SortKeys [Java Application] C:\Users
Keys     Values:
90       Black
78       Orange
60       Yellow
23       Red
11       White
```

11. Write a Java program to get a key-value mapping associated with the greatest key less than or equal to the given key.

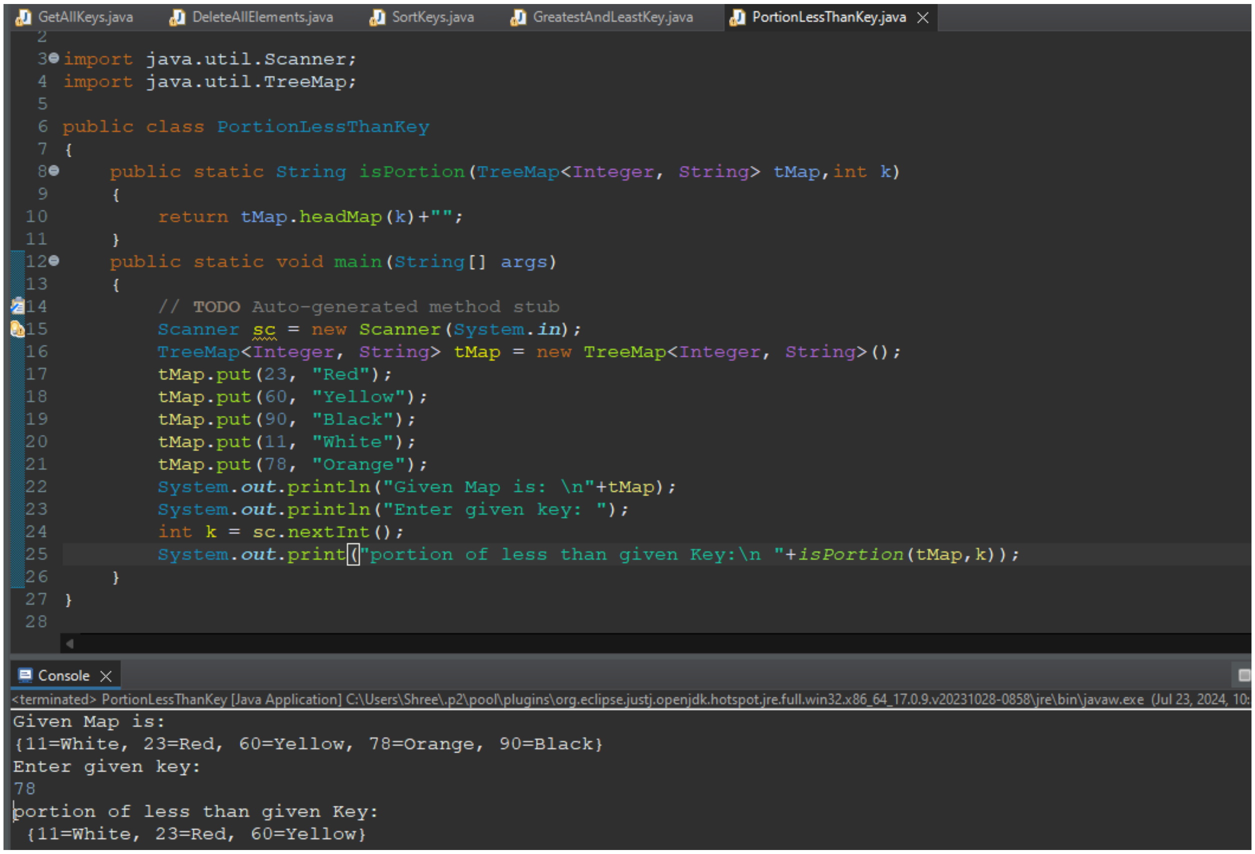
12. Write a Java program to get the greatest key less than or equal to the given key.



```java

import java.util.Scanner;
import java.util.TreeMap;

public class GreatestAndLeastKey
{
    public static String isKeys(TreeMap<Integer, String> tMap,int k)
    {
        return tMap.ceilingEntry(k)+"::"+tMap.floorEntry(k);
    }
    public static void main(String[] args)
    {
        // TODO Auto-generated method stub
        Scanner sc = new Scanner(System.in);
        TreeMap<Integer, String> tMap = new TreeMap<Integer, String>();
        tMap.put(23, "Red");
        tMap.put(60, "Yellow");
        tMap.put(90, "Black");
        tMap.put(11, "White");
        tMap.put(78, "Orange");
        System.out.println("Given Map is: \n"+tMap);
        System.out.println("Enter given key: ");
        int k = sc.nextInt();
        System.out.print("greater than or equal key-->"+isKeys(tMap,k)+"<--less than or equal key");

    }
}
```

```
<terminated> GreatestAndLeastKey [Java Application] C:\Users\Shree\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.9.v20231028-0858\jre\bin\javaw.exe  (Jul 23, 2024, 10:29:03 P
Given Map is:
{11=White, 23=Red, 60=Yellow, 78=Orange, 90=Black}
Enter given key:
61
greater than or equal key-->78=Orange::60=Yellow<--less than or equal key
```

13. Write a Java program to get the portion of a map whose keys are strictly less than a given key.

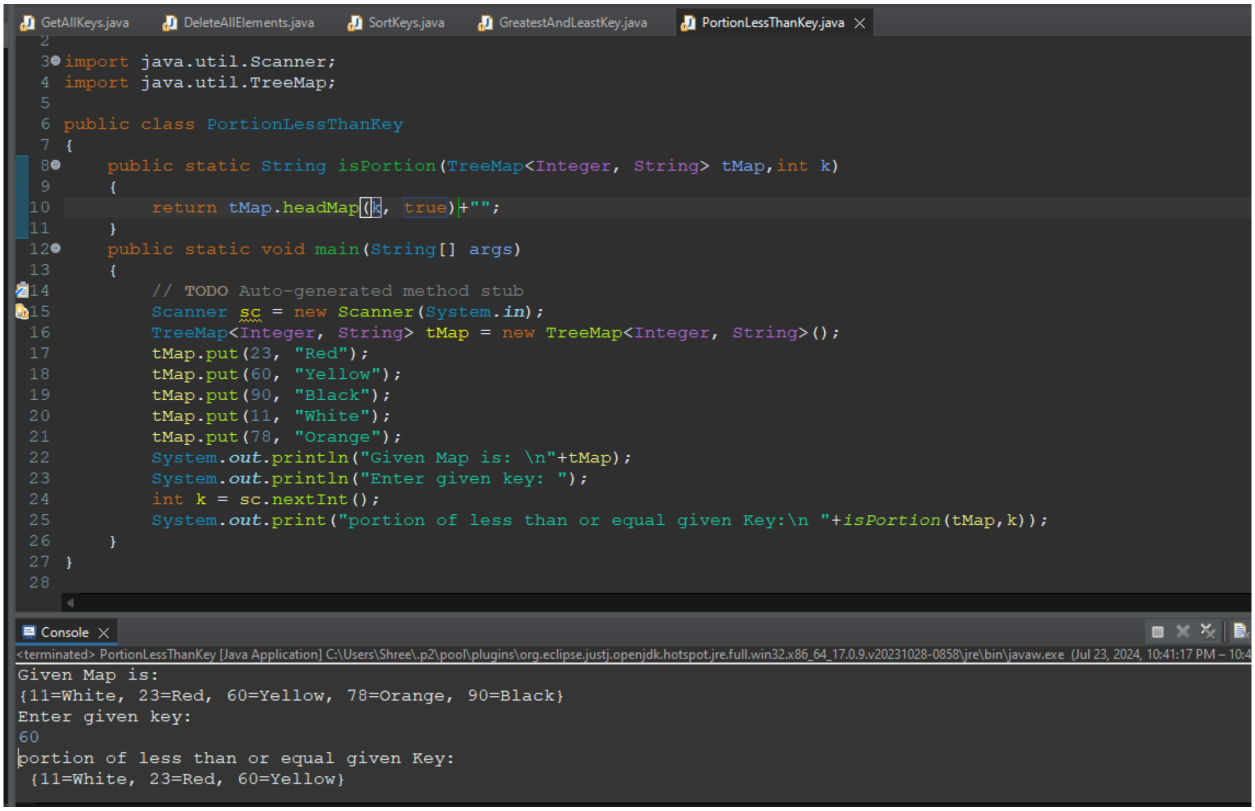


```java

import java.util.Scanner;
import java.util.TreeMap;

public class PortionLessThanKey
{
    public static String isPortion(TreeMap<Integer, String> tMap,int k)
    {
        return tMap.headMap(k)+"";
    }
    public static void main(String[] args)
    {
        // TODO Auto-generated method stub
        Scanner sc = new Scanner(System.in);
        TreeMap<Integer, String> tMap = new TreeMap<Integer, String>();
        tMap.put(23, "Red");
        tMap.put(60, "Yellow");
        tMap.put(90, "Black");
        tMap.put(11, "White");
        tMap.put(78, "Orange");
        System.out.println("Given Map is: \n"+tMap);
        System.out.println("Enter given key: ");
        int k = sc.nextInt();
        System.out.print("portion of less than given Key:\n "+isPortion(tMap,k));
    }
}

```

<terminated> PortionLessThanKey [Java Application] C:\Users\Shree\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.9.v20231028-0858\jre\bin\javaw.exe (Jul 23, 2024, 10:

```
Given Map is:
{11=White, 23=Red, 60=Yellow, 78=Orange, 90=Black}
Enter given key:
78
portion of less than given Key:
 {11=White, 23=Red, 60=Yellow}
```

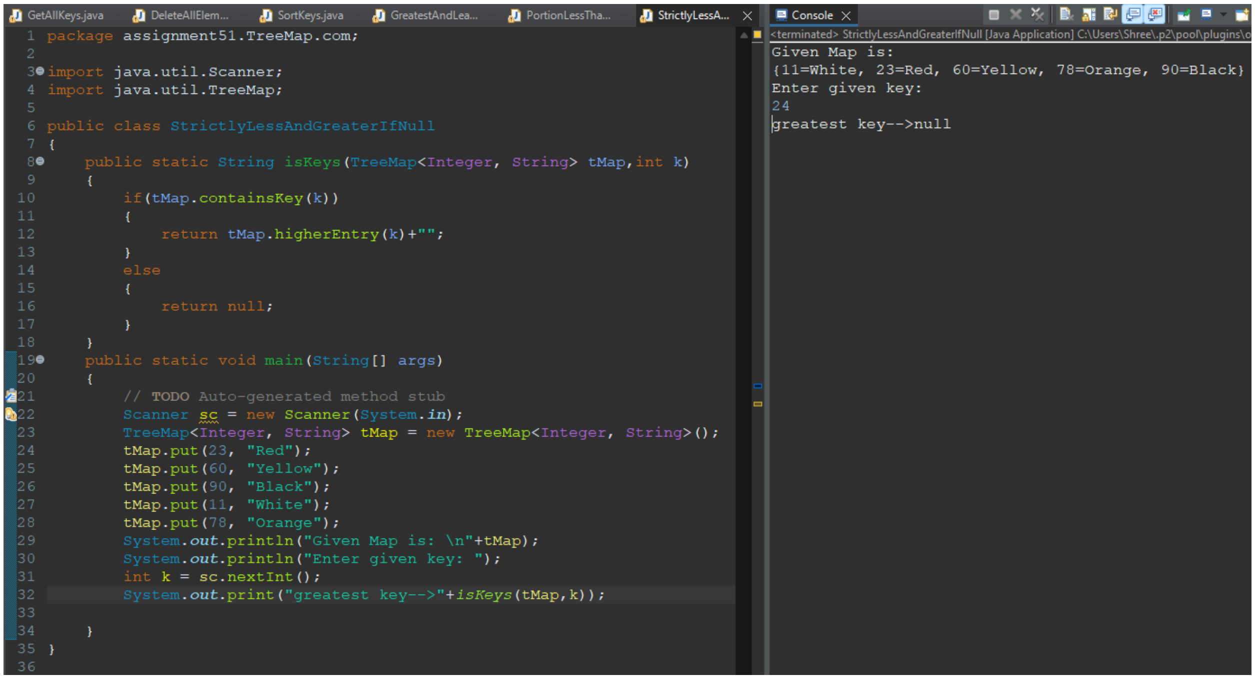
14. Write a Java program to get the portion of this map whose keys are less than (or equal to, if inclusive is true) a given key.



```java

import java.util.Scanner;
import java.util.TreeMap;

public class PortionLessThanKey
{
    public static String isPortion(TreeMap<Integer, String> tMap,int k)
    {
        return tMap.headMap(k, true)+"";
    }
    public static void main(String[] args)
    {
        // TODO Auto-generated method stub
        Scanner sc = new Scanner(System.in);
        TreeMap<Integer, String> tMap = new TreeMap<Integer, String>();
        tMap.put(23, "Red");
        tMap.put(60, "Yellow");
        tMap.put(90, "Black");
        tMap.put(11, "White");
        tMap.put(78, "Orange");
        System.out.println("Given Map is: \n"+tMap);
        System.out.println("Enter given key: ");
        int k = sc.nextInt();
        System.out.print("portion of less than or equal given Key:\n "+isPortion(tMap,k));
    }
}

```

```
<terminated> PortionLessThanKey [Java Application] C:\Users\Shree\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.9.v20231028-0858\jre\bin\javaw.exe  (Jul 23, 2024, 10:41:17 PM – 10:4
Given Map is:
{11=White, 23=Red, 60=Yellow, 78=Orange, 90=Black}
Enter given key:
60
portion of less than or equal given Key:
 {11=White, 23=Red, 60=Yellow}
```

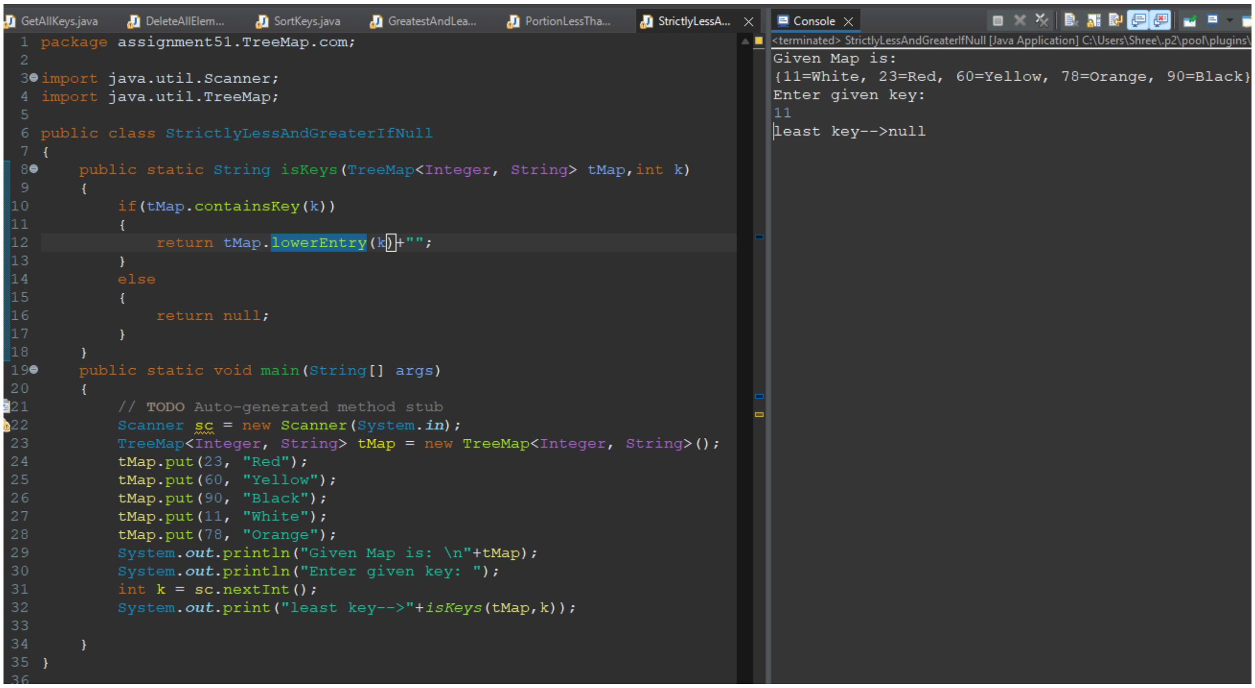
15. Write a Java program to get the least key strictly greater than the given key. Return null if there is no such key.



```java
package assignment51.TreeMap.com;

import java.util.Scanner;
import java.util.TreeMap;

public class StrictlyLessAndGreaterIfNull
{
    public static String isKeys(TreeMap<Integer, String> tMap,int k)
    {
        if(tMap.containsKey(k))
        {
            return tMap.higherEntry(k)+"";
        }
        else
        {
            return null;
        }
    }
    public static void main(String[] args)
    {
        // TODO Auto-generated method stub
        Scanner sc = new Scanner(System.in);
        TreeMap<Integer, String> tMap = new TreeMap<Integer, String>();
        tMap.put(23, "Red");
        tMap.put(60, "Yellow");
        tMap.put(90, "Black");
        tMap.put(11, "White");
        tMap.put(78, "Orange");
        System.out.println("Given Map is: \n"+tMap);
        System.out.println("Enter given key: ");
        int k = sc.nextInt();
        System.out.print("greatest key-->"+isKeys(tMap,k));

    }
}

```

```
<terminated> StrictlyLessAndGreaterIfNull [Java Application] C:\Users\Shree\.p2\pool\plugins\o
Given Map is:
{11=White, 23=Red, 60=Yellow, 78=Orange, 90=Black}
Enter given key:
24
greatest key-->null
```

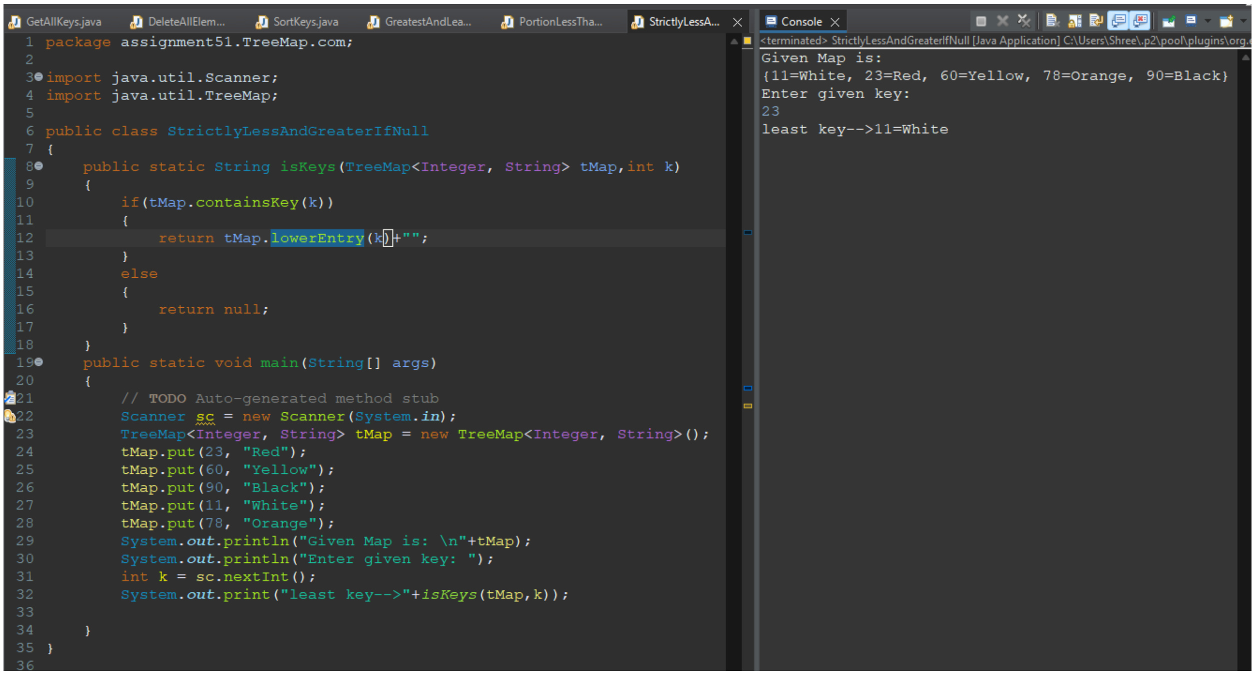
16. Write a Java program to get a key-value mapping associated with the greatest key strictly less than the given key. Return null if there is no such key.



```java
package assignment51.TreeMap.com;

import java.util.Scanner;
import java.util.TreeMap;

public class StrictlyLessAndGreaterIfNull
{
    public static String isKeys(TreeMap<Integer, String> tMap,int k)
    {
        if(tMap.containsKey(k))
        {
            return tMap.lowerEntry(k)+"";
        }
        else
        {
            return null;
        }
    }
    public static void main(String[] args)
    {
        // TODO Auto-generated method stub
        Scanner sc = new Scanner(System.in);
        TreeMap<Integer, String> tMap = new TreeMap<Integer, String>();
        tMap.put(23, "Red");
        tMap.put(60, "Yellow");
        tMap.put(90, "Black");
        tMap.put(11, "White");
        tMap.put(78, "Orange");
        System.out.println("Given Map is: \n"+tMap);
        System.out.println("Enter given key: ");
        int k = sc.nextInt();
        System.out.print("least key-->"+isKeys(tMap,k));

    }
}
```

```
Given Map is:
{11=White, 23=Red, 60=Yellow, 78=Orange, 90=Black}
Enter given key:
11
least key-->null
```

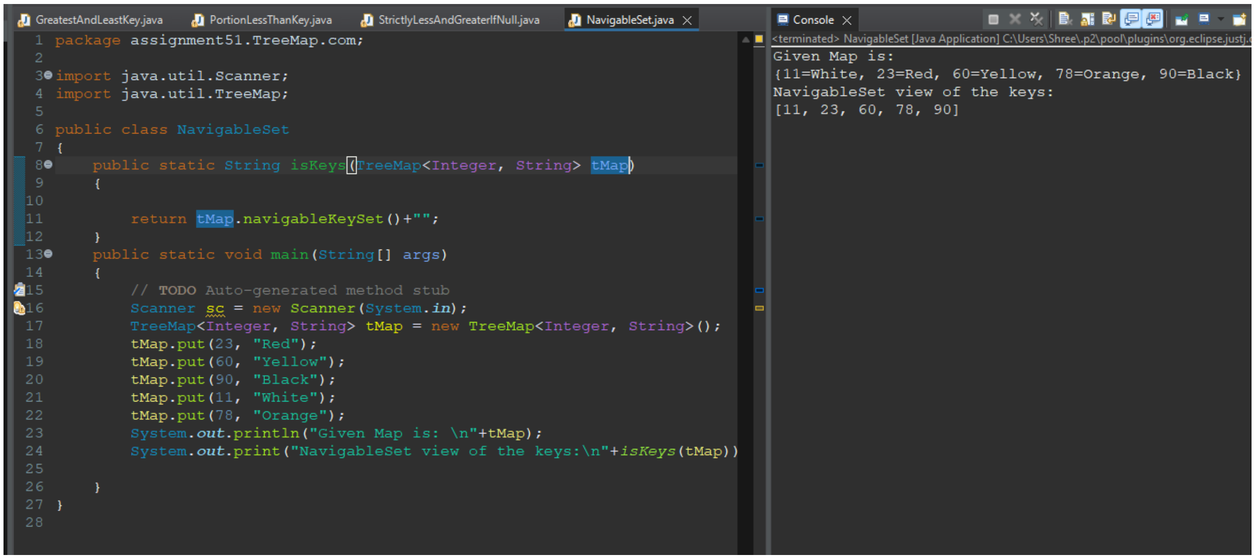
17. Write a Java program to get the greatest key strictly less than the given key. Return null if there is no such key.



```java
package assignment51.TreeMap.com;

import java.util.Scanner;
import java.util.TreeMap;

public class StrictlyLessAndGreaterIfNull
{
    public static String isKeys(TreeMap<Integer, String> tMap,int k)
    {
        if(tMap.containsKey(k))
        {
            return tMap.lowerEntry(k)+"";
        }
        else
        {
            return null;
        }
    }
    public static void main(String[] args)
    {
        // TODO Auto-generated method stub
        Scanner sc = new Scanner(System.in);
        TreeMap<Integer, String> tMap = new TreeMap<Integer, String>();
        tMap.put(23, "Red");
        tMap.put(60, "Yellow");
        tMap.put(90, "Black");
        tMap.put(11, "White");
        tMap.put(78, "Orange");
        System.out.println("Given Map is: \n"+tMap);
        System.out.println("Enter given key: ");
        int k = sc.nextInt();
        System.out.print("least key-->"+isKeys(tMap,k));

    }
}
```

```
Given Map is:
{11=White, 23=Red, 60=Yellow, 78=Orange, 90=Black}
Enter given key:
23
least key-->11=White
```

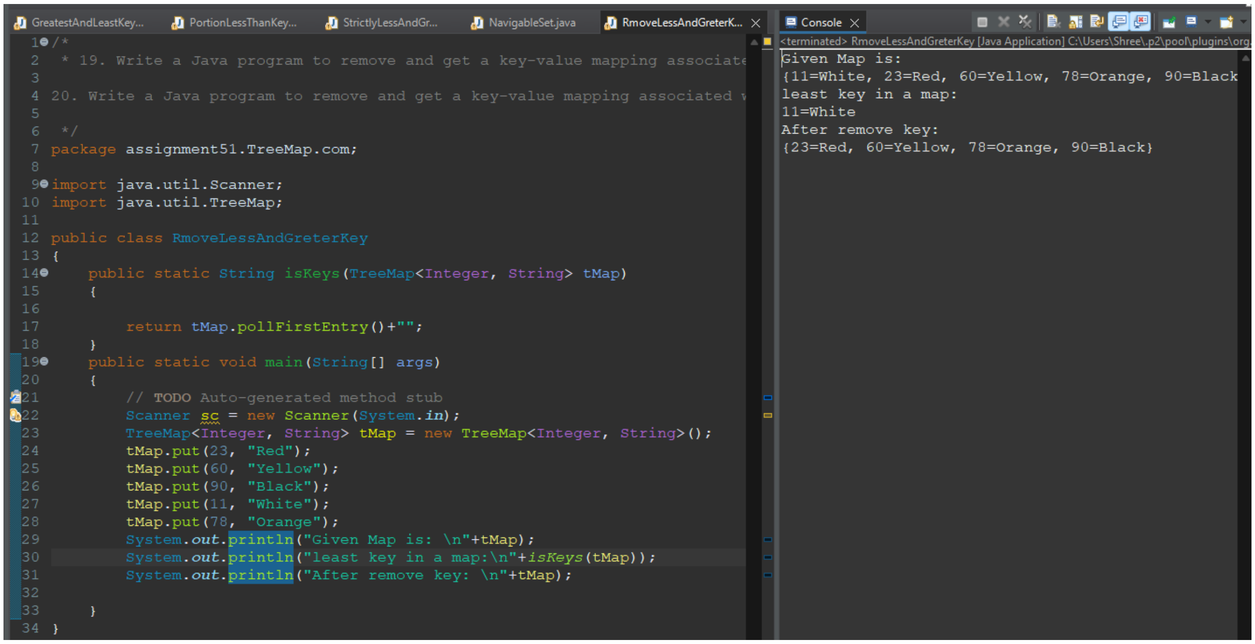
18. Write a Java program to get a NavigableSet view of keys in a map.



```java
package assignment51.TreeMap.com;

import java.util.Scanner;
import java.util.TreeMap;

public class NavigableSet
{
    public static String isKeys(TreeMap<Integer, String> tMap)
    {

        return tMap.navigableKeySet()+"";
    }
    public static void main(String[] args)
    {
        // TODO Auto-generated method stub
        Scanner sc = new Scanner(System.in);
        TreeMap<Integer, String> tMap = new TreeMap<Integer, String>();
        tMap.put(23, "Red");
        tMap.put(60, "Yellow");
        tMap.put(90, "Black");
        tMap.put(11, "White");
        tMap.put(78, "Orange");
        System.out.println("Given Map is: \n"+tMap);
        System.out.print("NavigableSet view of the keys:\n"+isKeys(tMap))

    }
}
```

```
<terminated> NavigableSet [Java Application] C:\Users\Shree\.p2\pool\plugins\org.eclipse.justj.
Given Map is:
{11=White, 23=Red, 60=Yellow, 78=Orange, 90=Black}
NavigableSet view of the keys:
[11, 23, 60, 78, 90]
```

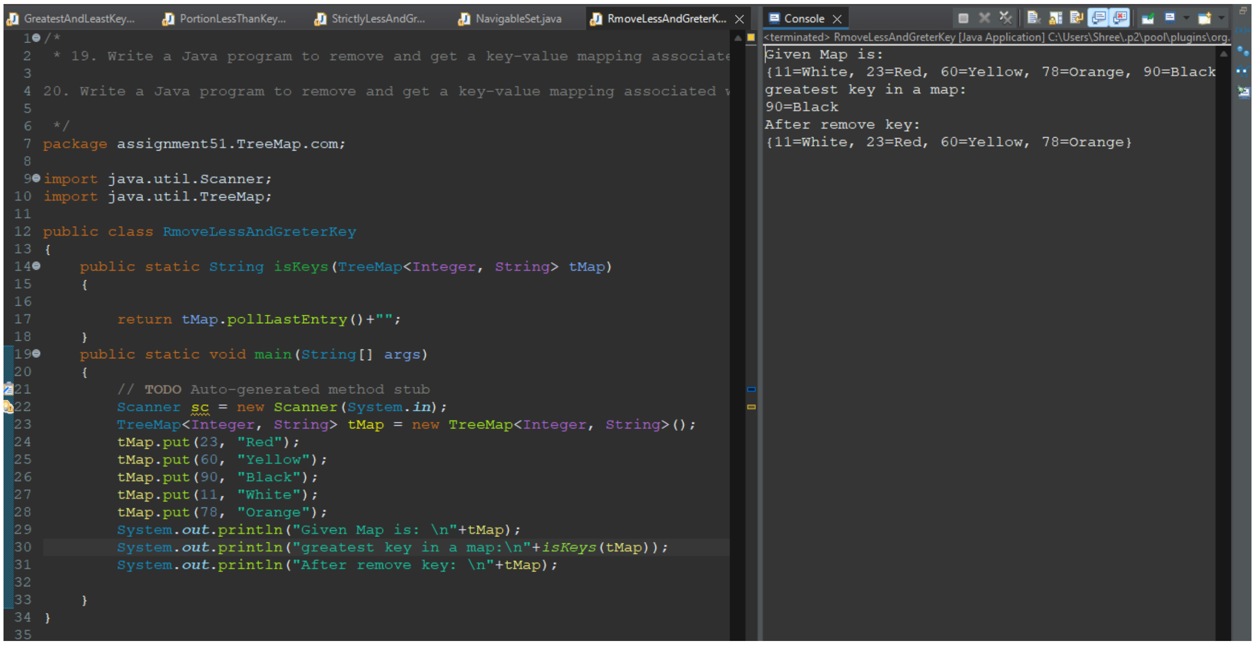
19. Write a Java program to remove and get a key-value mapping associated with the least key in a map.



```java
/*
 * 19. Write a Java program to remove and get a key-value mapping associate

20. Write a Java program to remove and get a key-value mapping associated w

 */
package assignment51.TreeMap.com;

import java.util.Scanner;
import java.util.TreeMap;

public class RmoveLessAndGreterKey
{
    public static String isKeys(TreeMap<Integer, String> tMap)
    {

        return tMap.pollFirstEntry()+"";
    }
    public static void main(String[] args)
    {
        // TODO Auto-generated method stub
        Scanner sc = new Scanner(System.in);
        TreeMap<Integer, String> tMap = new TreeMap<Integer, String>();
        tMap.put(23, "Red");
        tMap.put(60, "Yellow");
        tMap.put(90, "Black");
        tMap.put(11, "White");
        tMap.put(78, "Orange");
        System.out.println("Given Map is: \n"+tMap);
        System.out.println("least key in a map:\n"+isKeys(tMap));
        System.out.println("After remove key: \n"+tMap);

    }
}
```

```
<terminated> RmoveLessAndGreterKey [Java Application] C:\Users\Shree\.p2\pool\plugins\org.
Given Map is:
{11=White, 23=Red, 60=Yellow, 78=Orange, 90=Black
least key in a map:
11=White
After remove key:
{23=Red, 60=Yellow, 78=Orange, 90=Black}
```

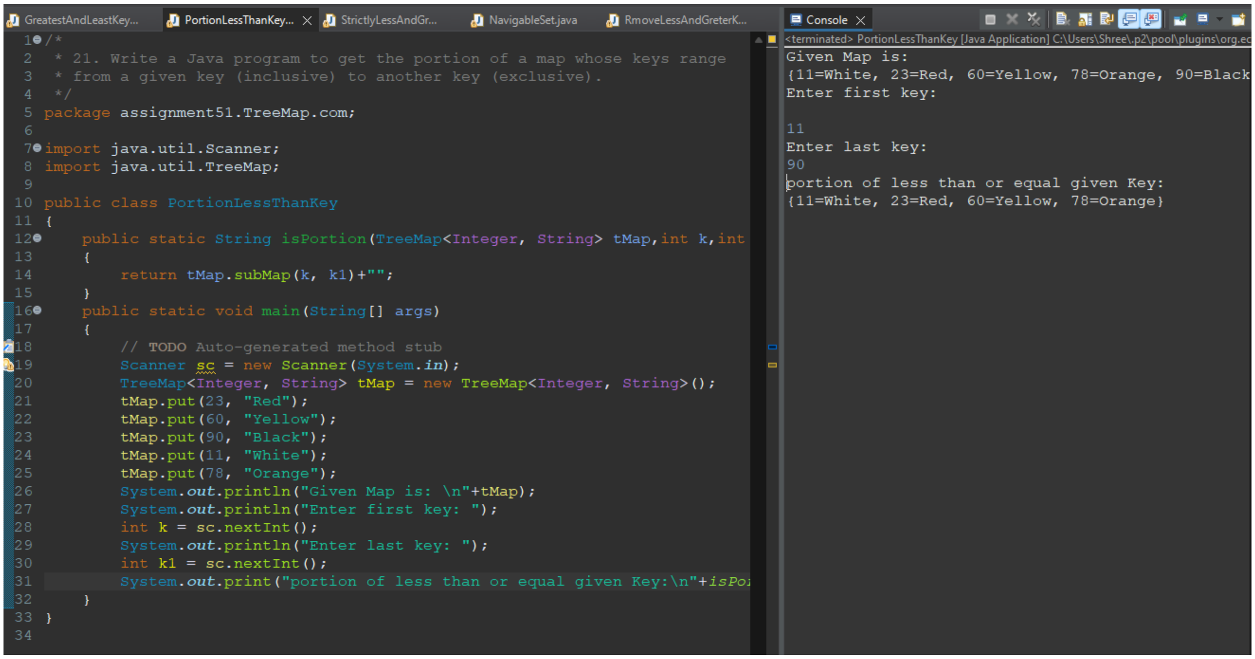
20. Write a Java program to remove and get a key-value mapping associated with the greatest key in this map.



```java
/*
 * 19. Write a Java program to remove and get a key-value mapping associate
 
20. Write a Java program to remove and get a key-value mapping associated w

 */
package assignment51.TreeMap.com;

import java.util.Scanner;
import java.util.TreeMap;

public class RmoveLessAndGreterKey
{
    public static String isKeys(TreeMap<Integer, String> tMap)
    {

        return tMap.pollLastEntry()+"";
    }
    public static void main(String[] args)
    {
        // TODO Auto-generated method stub
        Scanner sc = new Scanner(System.in);
        TreeMap<Integer, String> tMap = new TreeMap<Integer, String>();
        tMap.put(23, "Red");
        tMap.put(60, "Yellow");
        tMap.put(90, "Black");
        tMap.put(11, "White");
        tMap.put(78, "Orange");
        System.out.println("Given Map is: \n"+tMap);
        System.out.println("greatest key in a map:\n"+isKeys(tMap));
        System.out.println("After remove key: \n"+tMap);

    }
}
```

```
Given Map is:
{11=White, 23=Red, 60=Yellow, 78=Orange, 90=Black
greatest key in a map:
90=Black
After remove key:
{11=White, 23=Red, 60=Yellow, 78=Orange}
```

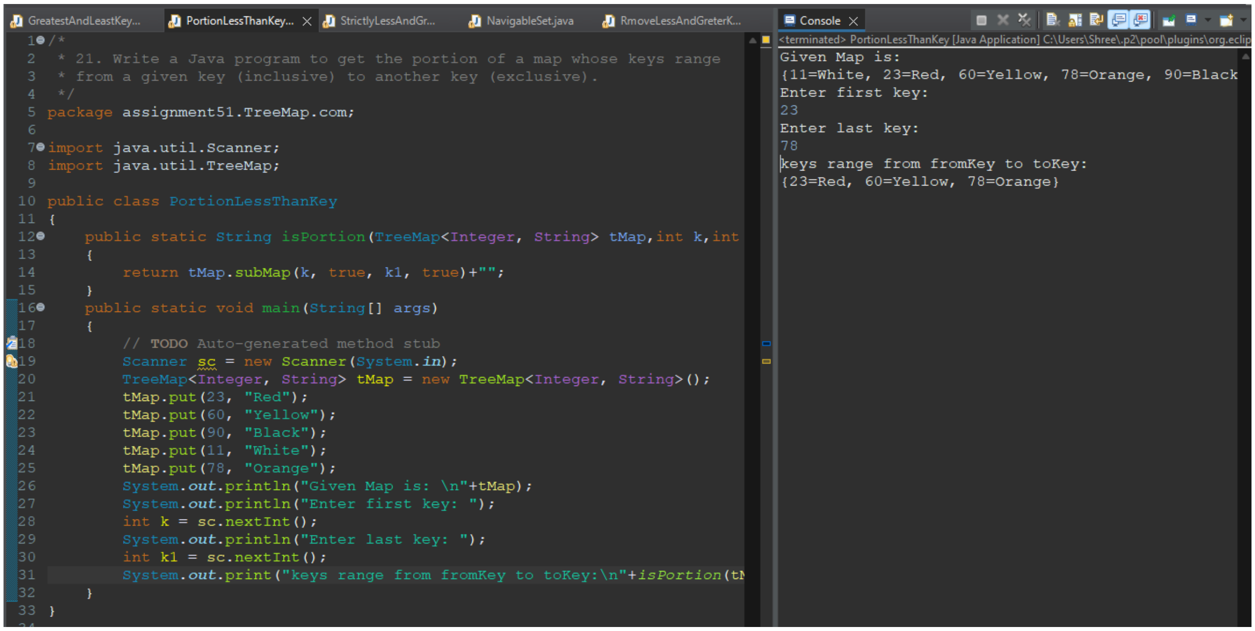
21. Write a Java program to get the portion of a map whose keys range from a given key (inclusive) to another key (exclusive).



```java
/*
 * 21. Write a Java program to get the portion of a map whose keys range
 * from a given key (inclusive) to another key (exclusive).
 */
package assignment51.TreeMap.com;

import java.util.Scanner;
import java.util.TreeMap;

public class PortionLessThanKey
{
    public static String isPortion(TreeMap<Integer, String> tMap,int k,int
    {
        return tMap.subMap(k, k1)+"";
    }
    public static void main(String[] args)
    {
        // TODO Auto-generated method stub
        Scanner sc = new Scanner(System.in);
        TreeMap<Integer, String> tMap = new TreeMap<Integer, String>();
        tMap.put(23, "Red");
        tMap.put(60, "Yellow");
        tMap.put(90, "Black");
        tMap.put(11, "White");
        tMap.put(78, "Orange");
        System.out.println("Given Map is: \n"+tMap);
        System.out.println("Enter first key: ");
        int k = sc.nextInt();
        System.out.println("Enter last key: ");
        int k1 = sc.nextInt();
        System.out.print("portion of less than or equal given Key:\n"+isPor
    }
}
```

```
Given Map is:
{11=White, 23=Red, 60=Yellow, 78=Orange, 90=Black
Enter first key:

11
Enter last key:
90
portion of less than or equal given Key:
{11=White, 23=Red, 60=Yellow, 78=Orange}
```

22. Write a Java program to get the portion of a map whose keys range from a given key to another key.



```java
/*
 * 21. Write a Java program to get the portion of a map whose keys range
 * from a given key (inclusive) to another key (exclusive).
 */
package assignment51.TreeMap.com;

import java.util.Scanner;
import java.util.TreeMap;

public class PortionLessThanKey
{
    public static String isPortion(TreeMap<Integer, String> tMap,int k,int
    {
        return tMap.subMap(k, true, k1, true)+"";
    }
    public static void main(String[] args)
    {
        // TODO Auto-generated method stub
        Scanner sc = new Scanner(System.in);
        TreeMap<Integer, String> tMap = new TreeMap<Integer, String>();
        tMap.put(23, "Red");
        tMap.put(60, "Yellow");
        tMap.put(90, "Black");
        tMap.put(11, "White");
        tMap.put(78, "Orange");
        System.out.println("Given Map is: \n"+tMap);
        System.out.println("Enter first key: ");
        int k = sc.nextInt();
        System.out.println("Enter last key: ");
        int k1 = sc.nextInt();
        System.out.print("keys range from fromKey to toKey:\n"+isPortion(tM
    }
}
```

```
<terminated> PortionLessThanKey [Java Application] C:\Users\Shree\.p2\pool\plugins\org.eclip
Given Map is:
{11=White, 23=Red, 60=Yellow, 78=Orange, 90=Black
Enter first key:
23
Enter last key:
78
keys range from fromKey to toKey:
{23=Red, 60=Yellow, 78=Orange}
```