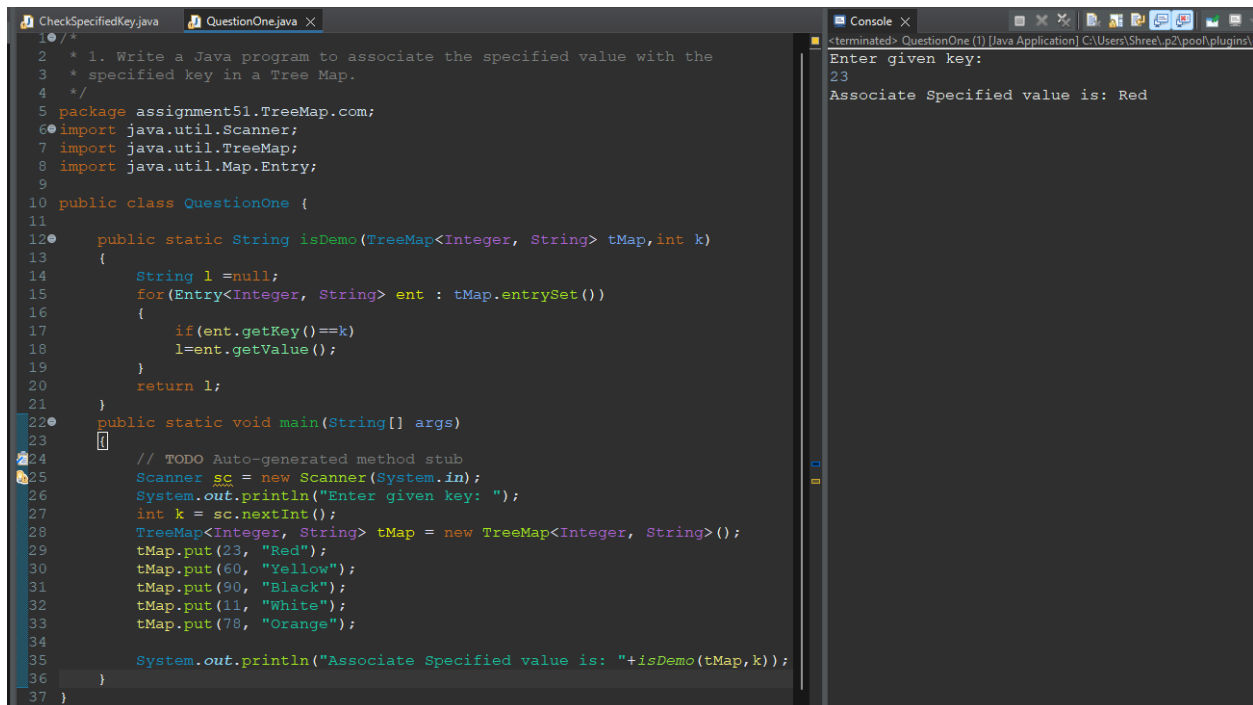


Assignment No:-51

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.

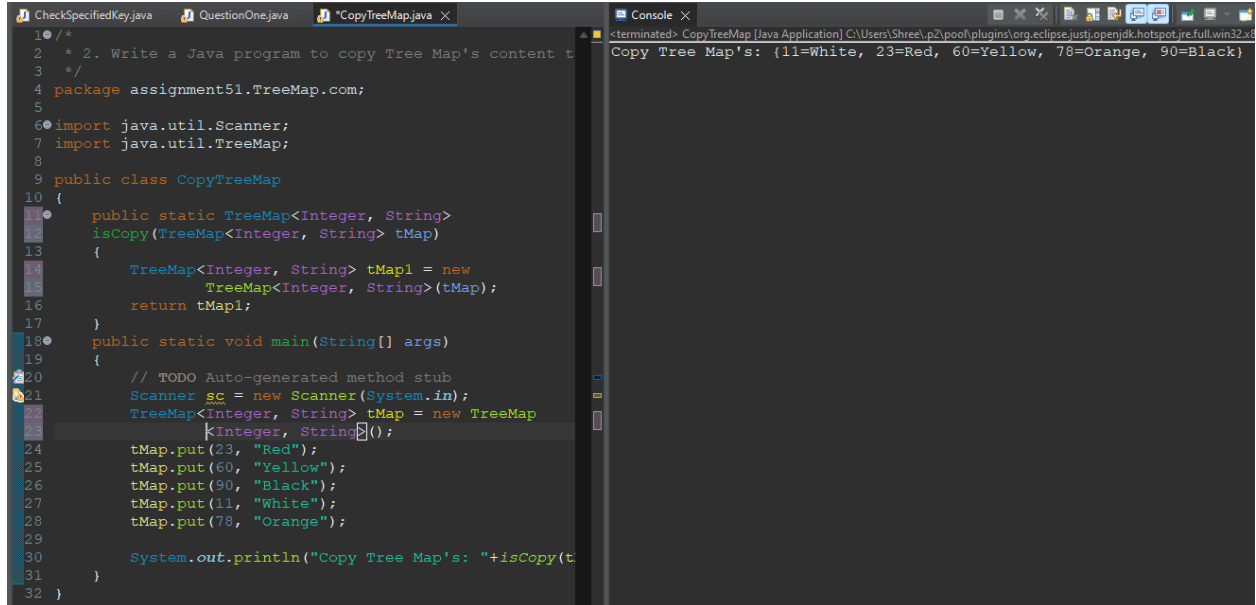


```
1  /*
2  * 1. Write a Java program to associate the specified value with the
3  * specified key in a Tree Map.
4  */
5  package assignment51.TreeMap.com;
6  import java.util.Scanner;
7  import java.util.TreeMap;
8  import java.util.Map.Entry;
9
10 public class QuestionOne {
11
12     public static String isDemo(TreeMap<Integer, String> tMap,int k)
13     {
14         String l =null;
15         for(Entry<Integer, String> ent : tMap.entrySet())
16         {
17             if(ent.getKey()==k)
18                 l=ent.getValue();
19         }
20         return l;
21     }
22     public static void main(String[] args)
23     {
24         // TODO Auto-generated method stub
25         Scanner sc = new Scanner(System.in);
26         System.out.println("Enter given key: ");
27         int k = sc.nextInt();
28         TreeMap<Integer, String> tMap = new TreeMap<Integer, String>();
29         tMap.put(23, "Red");
30         tMap.put(60, "Yellow");
31         tMap.put(90, "Black");
32         tMap.put(11, "White");
33         tMap.put(70, "Orange");
34
35         System.out.println("Associate Specified value is: "+isDemo(tMap,k));
36     }
37 }
```

Console

```
<terminated> QuestionOne (1) [Java Application] C:\Users\Shree\p2\pool\plugins\
Enter given key:
23
Associate Specified value is: Red
```

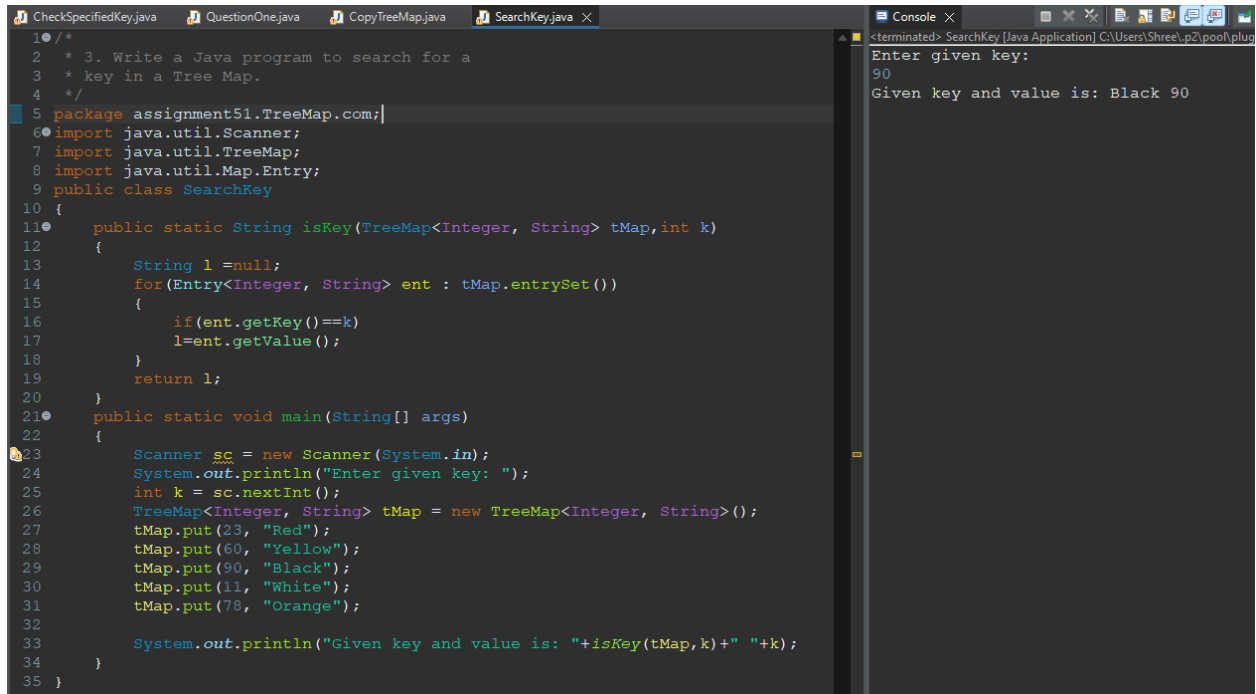
2. Write a Java program to copy Tree Map's content to another Tree Map.



```
1 /*
2  * 2. Write a Java program to copy Tree Map's content to another Tree Map.
3  */
4 package assignment51.TreeMap.com;
5
6 import java.util.Scanner;
7 import java.util.TreeMap;
8
9 public class CopyTreeMap
10 {
11     public static TreeMap<Integer, String>
12     isCopy(TreeMap<Integer, String> tMap)
13     {
14         TreeMap<Integer, String> tMap1 = new
15             TreeMap<Integer, String>(tMap);
16         return tMap1;
17     }
18     public static void main(String[] args)
19     {
20         // TODO Auto-generated method stub
21         Scanner sc = new Scanner(System.in);
22         TreeMap<Integer, String> tMap = new TreeMap
23             <Integer, String>();
24         tMap.put(23, "Red");
25         tMap.put(60, "Yellow");
26         tMap.put(90, "Black");
27         tMap.put(11, "White");
28         tMap.put(78, "Orange");
29
30         System.out.println("Copy Tree Map's: "+isCopy(t
31     )
32 }
```

Console Output:
<terminated> CopyTreeMap [Java Application] C:\Users\Shree\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64.jre\bin\java.exe
Copy Tree Map's: {11=White, 23=Red, 60=Yellow, 78=Orange, 90=Black}

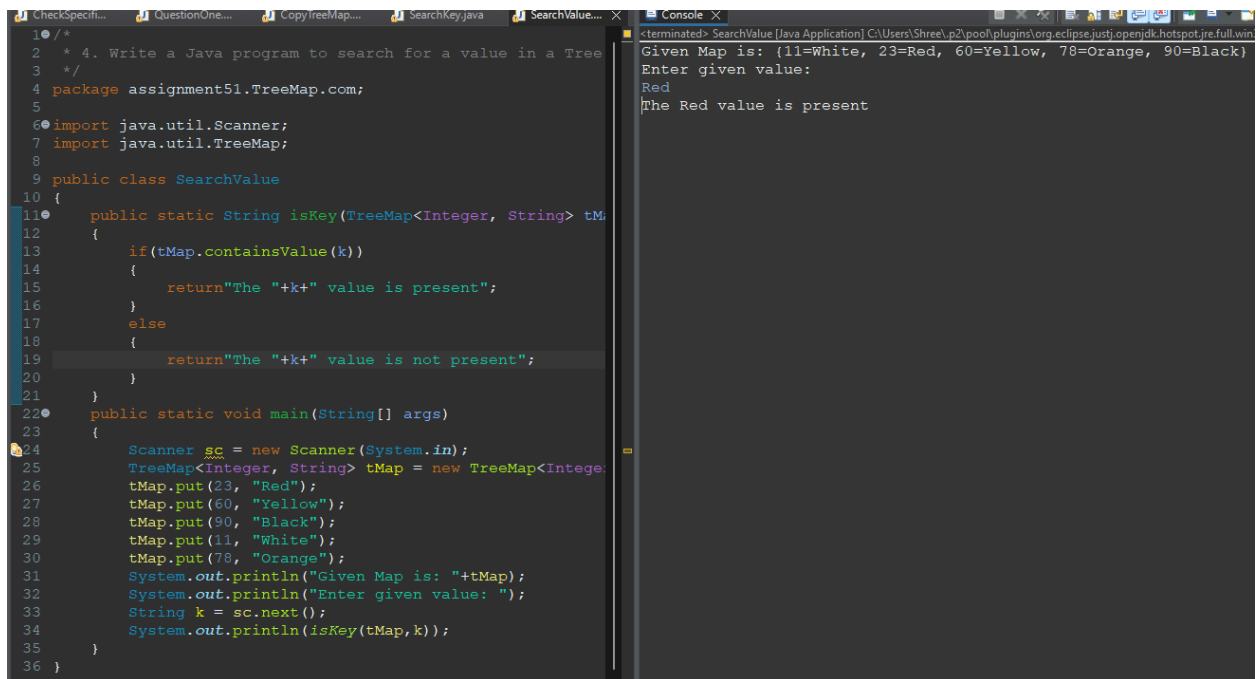
3. Write a Java program to search for a key in a Tree Map.



```
1 /*
2  * 3. Write a Java program to search for a
3  * key in a Tree Map.
4  */
5 package assignment51.TreeMap.com;
6 import java.util.Scanner;
7 import java.util.TreeMap;
8 import java.util.Map.Entry;
9 public class SearchKey
10 {
11     public static String isKey(TreeMap<Integer, String> tMap,int k)
12     {
13         String l =null;
14         for(Entry<Integer, String> ent : tMap.entrySet())
15         {
16             if(ent.getKey()==k)
17                 l=ent.getValue();
18         }
19         return l;
20     }
21     public static void main(String[] args)
22     {
23         Scanner sc = new Scanner(System.in);
24         System.out.println("Enter given key: ");
25         int k = sc.nextInt();
26         TreeMap<Integer, String> tMap = new TreeMap<Integer, String>();
27         tMap.put(23, "Red");
28         tMap.put(60, "Yellow");
29         tMap.put(90, "Black");
30         tMap.put(11, "White");
31         tMap.put(78, "Orange");
32
33         System.out.println("Given key and value is: "+isKey(tMap,k)+" "+k);
34     }
35 }
```

Console Output:
<terminated> SearchKey [Java Application] C:\Users\Shree\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64.jre\bin\java.exe
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.

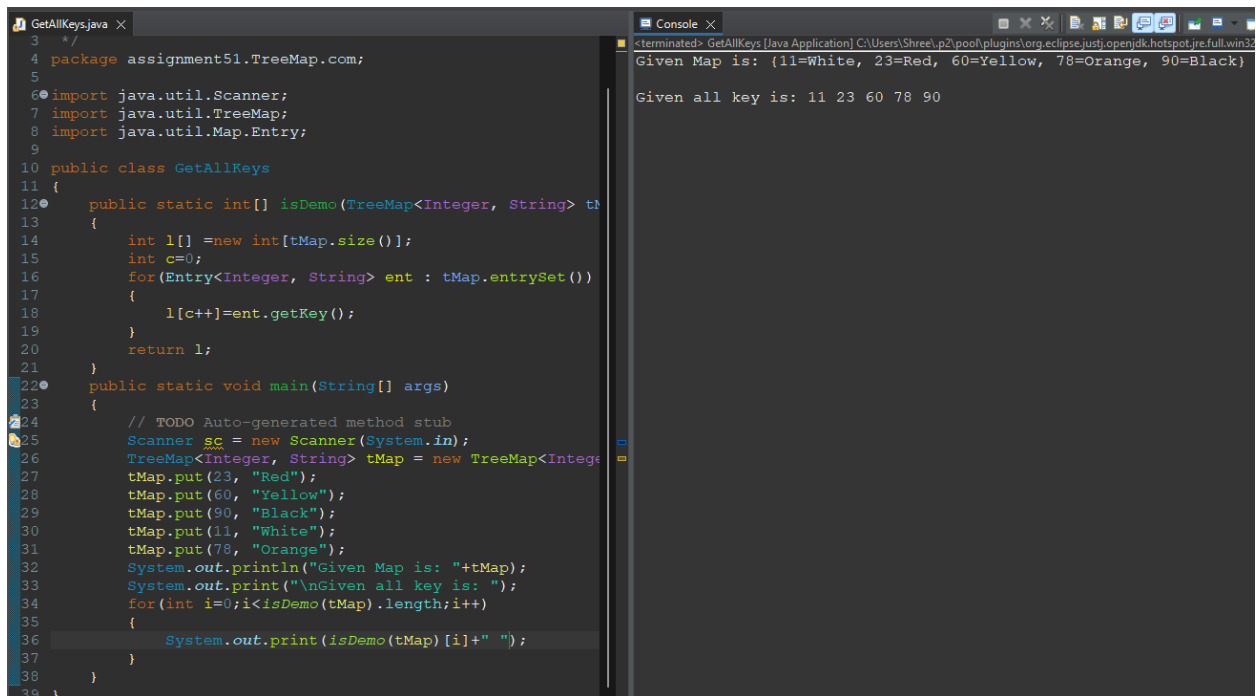


```
1 /*
2  * 4. Write a Java program to search for a value in a Tree
3  */
4 package assignment51.TreeMap.com;
5
6 import java.util.Scanner;
7 import java.util.TreeMap;
8
9 public class SearchValue
10 {
11     public static String isKey(TreeMap<Integer, String> tM
12     {
13         if(tMap.containsKey(k))
14         {
15             return "The "+k+" value is present";
16         }
17         else
18         {
19             return "The "+k+" value is not present";
20         }
21     }
22     public static void main(String[] args)
23     {
24         Scanner sc = new Scanner(System.in);
25         TreeMap<Integer, String> tMap = new TreeMap<Integer,
26         tMap.put(23, "Red");
27         tMap.put(60, "Yellow");
28         tMap.put(90, "Black");
29         tMap.put(11, "White");
30         tMap.put(78, "Orange");
31         System.out.println("Given Map is: "+tMap);
32         System.out.println("Enter given value: ");
33         String k = sc.next();
34         System.out.println(isKey(tMap,k));
35     }
36 }
```

Console X

```
<terminated> SearchValue [Java Application] C:\Users\Shree\p2\poo\plugins\org.eclipse.justi.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.



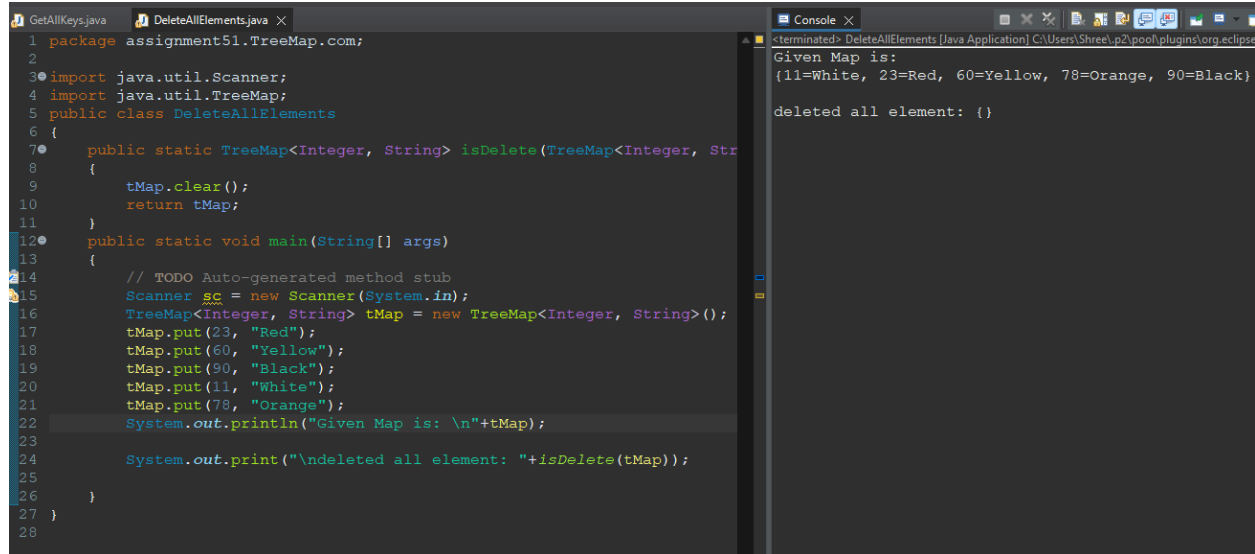
```
1 GetAllKeys.java
2 /*
3  */
4 package assignment51.TreeMap.com;
5
6 import java.util.Scanner;
7 import java.util.TreeMap;
8 import java.util.Map.Entry;
9
10 public class GetAllKeys
11 {
12     public static int[] isDemo(TreeMap<Integer, String> tM
13     {
14         int l[] = new int[tMap.size()];
15         int c=0;
16         for(Entry<Integer, String> ent : tMap.entrySet())
17         {
18             l[c++] = ent.getKey();
19         }
20         return l;
21     }
22     public static void main(String[] args)
23     {
24         // TODO Auto-generated method stub
25         Scanner sc = new Scanner(System.in);
26         TreeMap<Integer, String> tMap = new TreeMap<Integer,
27         tMap.put(23, "Red");
28         tMap.put(60, "Yellow");
29         tMap.put(90, "Black");
30         tMap.put(11, "White");
31         tMap.put(78, "Orange");
32         System.out.println("Given Map is: "+tMap);
33         System.out.print("\nGiven all key is: ");
34         for(int i=0; i<isDemo(tMap).length; i++)
35         {
36             System.out.print(isDemo(tMap)[i]+" ");
37         }
38     }
39 }
```

Console X

```
<terminated> GetAllKeys [Java Application] C:\Users\Shree\p2\poo\plugins\org.eclipse.justi.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.



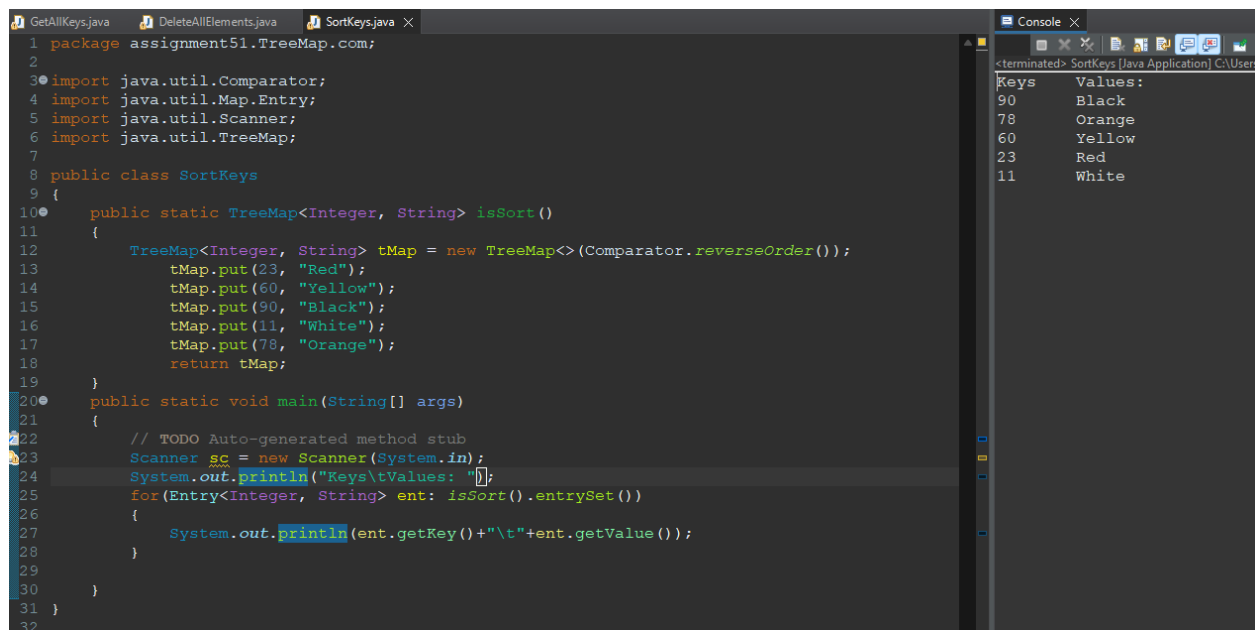
```
1 package assignment51.TreeMap.com;
2
3 import java.util.Scanner;
4 import java.util.TreeMap;
5 public class DeleteAllElements
6 {
7     public static TreeMap<Integer, String> isDelete(TreeMap<Integer, Str
8     {
9         tMap.clear();
10        return tMap;
11    }
12    public static void main(String[] args)
13    {
14        // TODO Auto-generated method stub
15        Scanner sc = new Scanner(System.in);
16        TreeMap<Integer, String> tMap = new TreeMap<Integer, String>();
17        tMap.put(23, "Red");
18        tMap.put(60, "Yellow");
19        tMap.put(90, "Black");
20        tMap.put(11, "White");
21        tMap.put(78, "Orange");
22        System.out.println("Given Map is: \n"+tMap);
23
24        System.out.print("\ndeleted all element: "+isDelete(tMap));
25    }
26 }
27
28
```

Console Output:

```
<terminated> DeleteAllElements [Java Application] C:\Users\Shree\p2\poo\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.

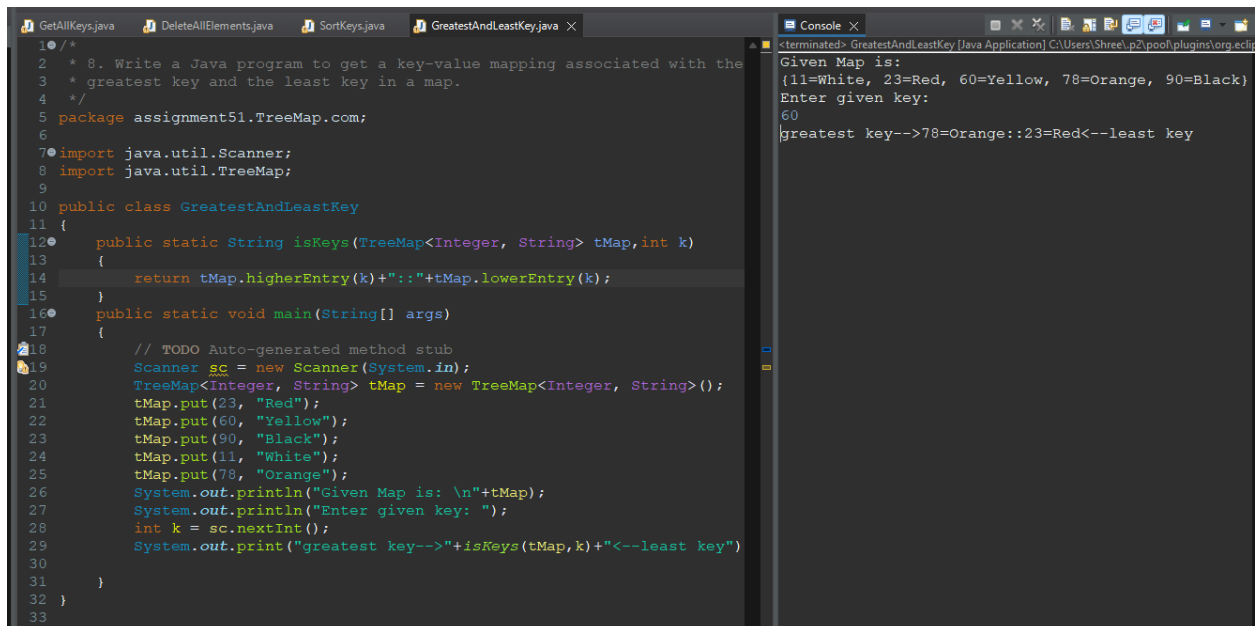


```
1 package assignment51.TreeMap.com;
2
3 import java.util.Comparator;
4 import java.util.Map.Entry;
5 import java.util.Scanner;
6 import java.util.TreeMap;
7
8 public class SortKeys
9 {
10    public static TreeMap<Integer, String> isSort()
11    {
12        TreeMap<Integer, String> tMap = new TreeMap<>(Comparator.reverseOrder());
13        tMap.put(23, "Red");
14        tMap.put(60, "Yellow");
15        tMap.put(90, "Black");
16        tMap.put(11, "White");
17        tMap.put(78, "Orange");
18        return tMap;
19    }
20    public static void main(String[] args)
21    {
22        // TODO Auto-generated method stub
23        Scanner sc = new Scanner(System.in);
24        System.out.println("Keys\tValues: ");
25        for(Entry<Integer, String> ent: isSort().entrySet())
26        {
27            System.out.println(ent.getKey()+"\t"+ent.getValue());
28        }
29    }
30 }
31
32
```

Console Output:

```
<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.

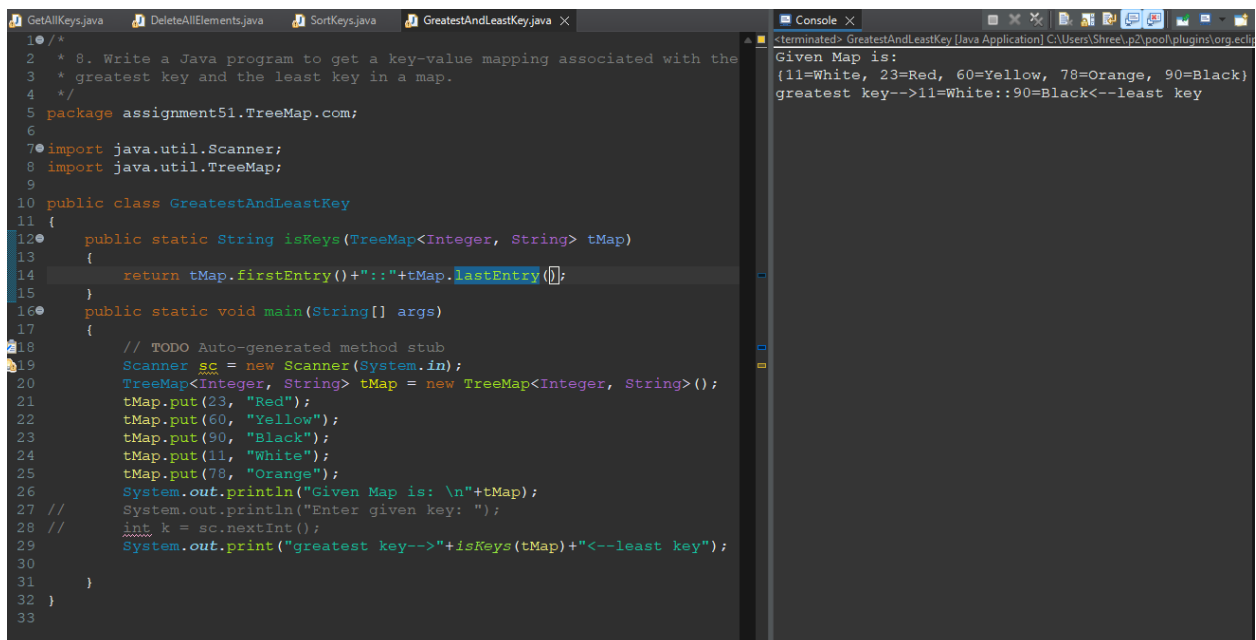


```
1 /*
2  * 8. Write a Java program to get a key-value mapping associated with the
3  * greatest key and the least key in a map.
4  */
5 package assignment51.TreeMap.com;
6
7 import java.util.Scanner;
8 import java.util.TreeMap;
9
10 public class GreatestAndLeastKey
11 {
12     public static String isKeys(TreeMap<Integer, String> tMap,int k)
13     {
14         return tMap.higherEntry(k)+":"+tMap.lowerEntry(k);
15     }
16     public static void main(String[] args)
17     {
18         // TODO Auto-generated method stub
19         Scanner sc = new Scanner(System.in);
20         TreeMap<Integer, String> tMap = new TreeMap<Integer, String>();
21         tMap.put(23, "Red");
22         tMap.put(60, "Yellow");
23         tMap.put(90, "Black");
24         tMap.put(11, "White");
25         tMap.put(78, "Orange");
26         System.out.println("Given Map is: \n"+tMap);
27         System.out.println("Enter given key: ");
28         int k = sc.nextInt();
29         System.out.print("greatest key-->"+isKeys(tMap,k)+"<--least key")
30     }
31 }
32
33
```

Console Output:

```
<terminated> GreatestAndLeastKey [Java Application] C:\Users\Shree.p2\pool\plugins\org.eclipse
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.

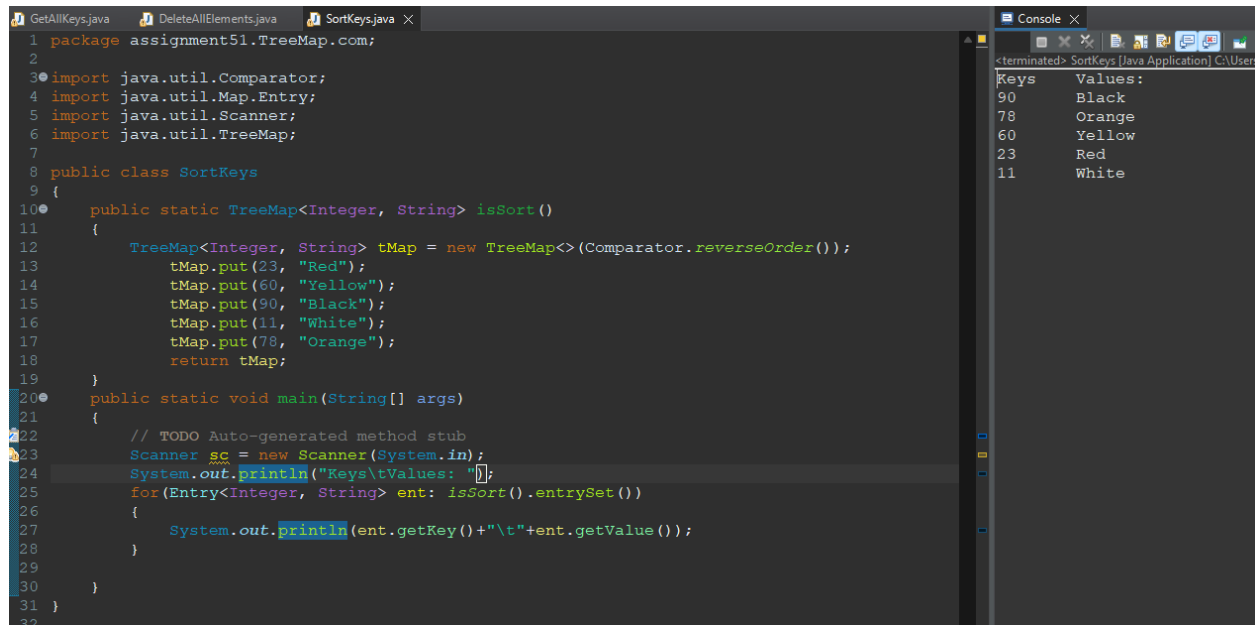


```
1 /*
2  * 8. Write a Java program to get a key-value mapping associated with the
3  * greatest key and the least key in a map.
4  */
5 package assignment51.TreeMap.com;
6
7 import java.util.Scanner;
8 import java.util.TreeMap;
9
10 public class GreatestAndLeastKey
11 {
12     public static String isKeys(TreeMap<Integer, String> tMap)
13     {
14         return tMap.firstEntry()+":"+tMap.lastEntry();
15     }
16     public static void main(String[] args)
17     {
18         // TODO Auto-generated method stub
19         Scanner sc = new Scanner(System.in);
20         TreeMap<Integer, String> tMap = new TreeMap<Integer, String>();
21         tMap.put(23, "Red");
22         tMap.put(60, "Yellow");
23         tMap.put(90, "Black");
24         tMap.put(11, "White");
25         tMap.put(78, "Orange");
26         System.out.println("Given Map is: \n"+tMap);
27         // System.out.println("Enter given key: ");
28         // int k = sc.nextInt();
29         System.out.print("greatest key-->"+isKeys(tMap)+"<--least key");
30     }
31 }
32
33
```

Console Output:

```
<terminated> GreatestAndLeastKey [Java Application] C:\Users\Shree.p2\pool\plugins\org.eclipse
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.

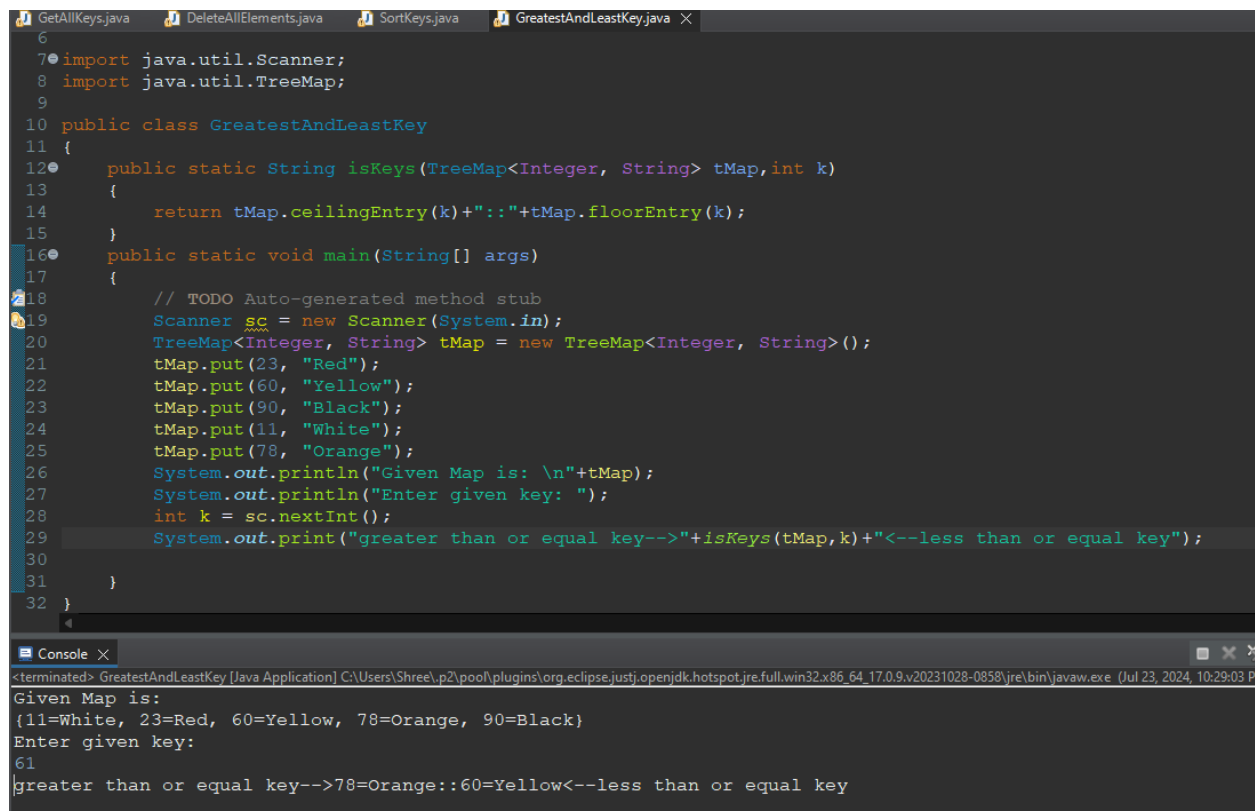


```
1 package assignment51.TreeMap.com;
2
3 import java.util.Comparator;
4 import java.util.Map.Entry;
5 import java.util.Scanner;
6 import java.util.TreeMap;
7
8 public class SortKeys
9 {
10     public static TreeMap<Integer, String> isSort()
11     {
12         TreeMap<Integer, String> tMap = new TreeMap<>(Comparator.reverseOrder());
13         tMap.put(23, "Red");
14         tMap.put(60, "Yellow");
15         tMap.put(90, "Black");
16         tMap.put(11, "White");
17         tMap.put(78, "Orange");
18         return tMap;
19     }
20     public static void main(String[] args)
21     {
22         // TODO Auto-generated method stub
23         Scanner sc = new Scanner(System.in);
24         System.out.println("Keys\tValues: ");
25         for(Entry<Integer, String> ent: isSort().entrySet())
26         {
27             System.out.println(ent.getKey()+"\t"+ent.getValue());
28         }
29     }
30 }
31
32
```

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.

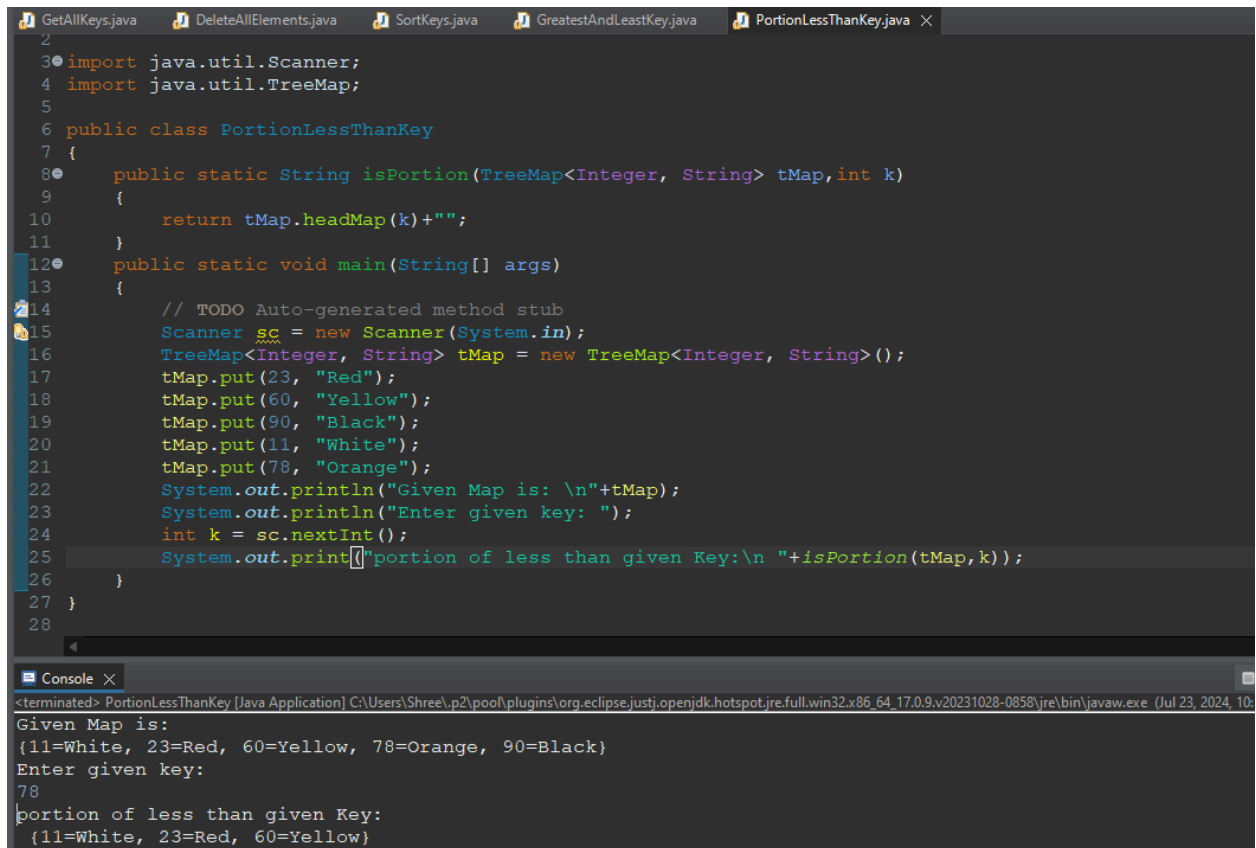


```
6
7 import java.util.Scanner;
8 import java.util.TreeMap;
9
10 public class GreatestAndLeastKey
11 {
12     public static String isKeys(TreeMap<Integer, String> tMap,int k)
13     {
14         return tMap.ceilingEntry(k)+"::"+tMap.floorEntry(k);
15     }
16     public static void main(String[] args)
17     {
18         // TODO Auto-generated method stub
19         Scanner sc = new Scanner(System.in);
20         TreeMap<Integer, String> tMap = new TreeMap<Integer, String>();
21         tMap.put(23, "Red");
22         tMap.put(60, "Yellow");
23         tMap.put(90, "Black");
24         tMap.put(11, "White");
25         tMap.put(78, "Orange");
26         System.out.println("Given Map is: \n"+tMap);
27         System.out.println("Enter given key: ");
28         int k = sc.nextInt();
29         System.out.print("greater than or equal key-->"+isKeys(tMap,k)+"<--less than or equal key");
30     }
31 }
32
```

Console Output:

```
<terminated> GreatestAndLeastKey [Java Application] C:\Users\Shree\p2\pool\plugins\org.eclipse.justi.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.



The screenshot shows an IDE with a Java file named `PortionLessThanKey.java`. The code defines a `PortionLessThanKey` class with a static method `isPortion` and a `main` method. The `main` method creates a `TreeMap` with the following entries: (23, "Red"), (60, "Yellow"), (90, "Black"), (11, "White"), and (78, "Orange"). It prompts the user to enter a key, and the user enters 78. The program then prints the portion of the map with keys less than 78, which is {11=White, 23=Red, 60=Yellow}.

```
2
3 import java.util.Scanner;
4 import java.util.TreeMap;
5
6 public class PortionLessThanKey
7 {
8     public static String isPortion(TreeMap<Integer, String> tMap, int k)
9     {
10         return tMap.headMap(k).toString();
11     }
12     public static void main(String[] args)
13     {
14         // TODO Auto-generated method stub
15         Scanner sc = new Scanner(System.in);
16         TreeMap<Integer, String> tMap = new TreeMap<Integer, String>();
17         tMap.put(23, "Red");
18         tMap.put(60, "Yellow");
19         tMap.put(90, "Black");
20         tMap.put(11, "White");
21         tMap.put(78, "Orange");
22         System.out.println("Given Map is: \n"+tMap);
23         System.out.println("Enter given key: ");
24         int k = sc.nextInt();
25         System.out.print("portion of less than given Key:\n "+isPortion(tMap,k));
26     }
27 }
28
```

Console Output:

```
<terminated> PortionLessThanKey [Java Application] C:\Users\Shree\p2\pool\plugins\org.eclipse.justi.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.

```

1  GetAllKeys.java  DeleteAllElements.java  SortKeys.java  GreatestAndLeastKey.java  PortionLessThanKey.java
2
3  import java.util.Scanner;
4  import java.util.TreeMap;
5
6  public class PortionLessThanKey
7  {
8      public static String isPortion(TreeMap<Integer, String> tMap,int k)
9      {
10         return tMap.headMap(k, true)+" ";
11     }
12     public static void main(String[] args)
13     {
14         // TODO Auto-generated method stub
15         Scanner sc = new Scanner(System.in);
16         TreeMap<Integer, String> tMap = new TreeMap<Integer, String>();
17         tMap.put(23, "Red");
18         tMap.put(60, "Yellow");
19         tMap.put(90, "Black");
20         tMap.put(11, "White");
21         tMap.put(78, "Orange");
22         System.out.println("Given Map is: \n"+tMap);
23         System.out.println("Enter given key: ");
24         int k = sc.nextInt();
25         System.out.print("portion of less than or equal given Key:\n "+isPortion(tMap,k));
26     }
27 }
28
29 Console
30 <terminated> PortionLessThanKey [Java Application] C:\Users\Shree\p2\poo\plugins\org.eclipse.justi.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
31 Given Map is:
32 {11=White, 23=Red, 60=Yellow, 78=Orange, 90=Black}
33 Enter given key:
34 60
35 portion of less than or equal given Key:
36 {11=White, 23=Red, 60=Yellow}

```

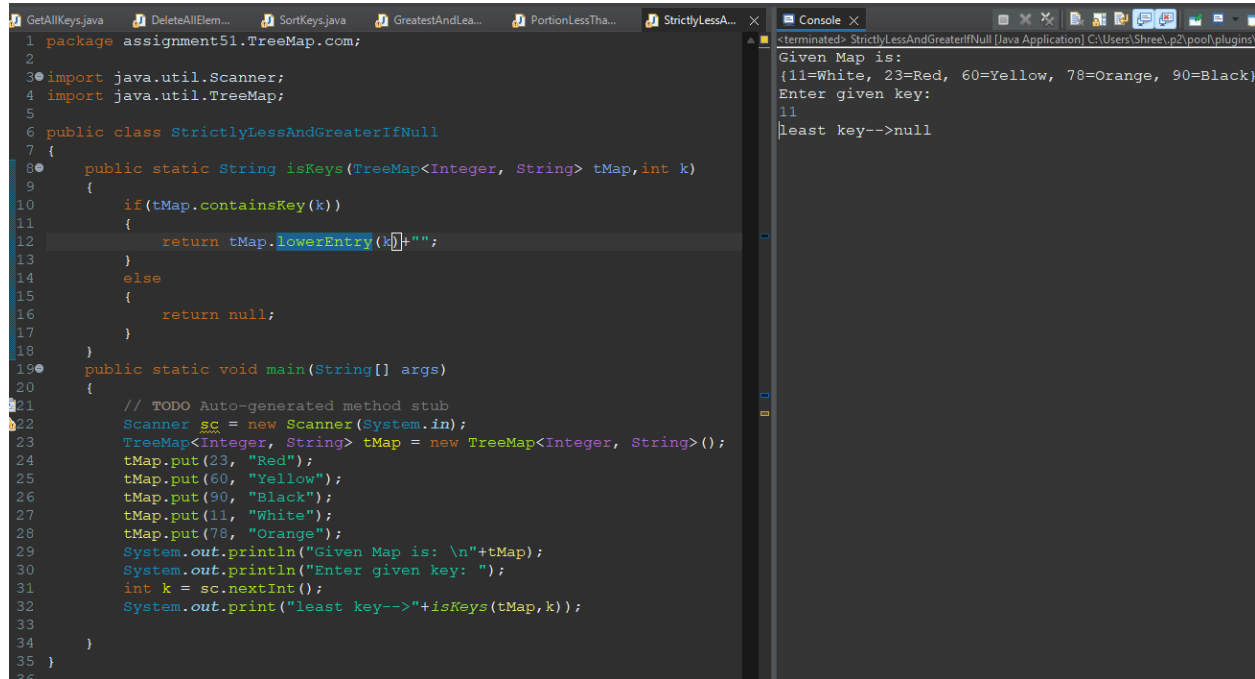
15. Write a Java program to get the least key strictly greater than the given key. Return null if there is no such key.

```

1  GetAllKeys.java  DeleteAllElem...  SortKeys.java  GreatestAndLea...  PortionLessTha...  StrictlyLessA...
2  package assignment51.TreeMap.com;
3
4  import java.util.Scanner;
5  import java.util.TreeMap;
6
7  public class StrictlyLessAndGreaterIfNull
8  {
9      public static String isKeys(TreeMap<Integer, String> tMap,int k)
10     {
11         if(tMap.containsKey(k))
12         {
13             return tMap.higherEntry(k)+" ";
14         }
15         else
16         {
17             return null;
18         }
19     }
20     public static void main(String[] args)
21     {
22         // TODO Auto-generated method stub
23         Scanner sc = new Scanner(System.in);
24         TreeMap<Integer, String> tMap = new TreeMap<Integer, String>();
25         tMap.put(23, "Red");
26         tMap.put(60, "Yellow");
27         tMap.put(90, "Black");
28         tMap.put(11, "White");
29         tMap.put(78, "Orange");
30         System.out.println("Given Map is: \n"+tMap);
31         System.out.println("Enter given key: ");
32         int k = sc.nextInt();
33         System.out.print("greatest key-->"+isKeys(tMap,k));
34     }
35 }
36
37 Console
38 <terminated> StrictlyLessAndGreaterIfNull [Java Application] C:\Users\Shree\p2\poo\plugins\o
39 Given Map is:
40 {11=White, 23=Red, 60=Yellow, 78=Orange, 90=Black}
41 Enter given key:
42 24
43 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.

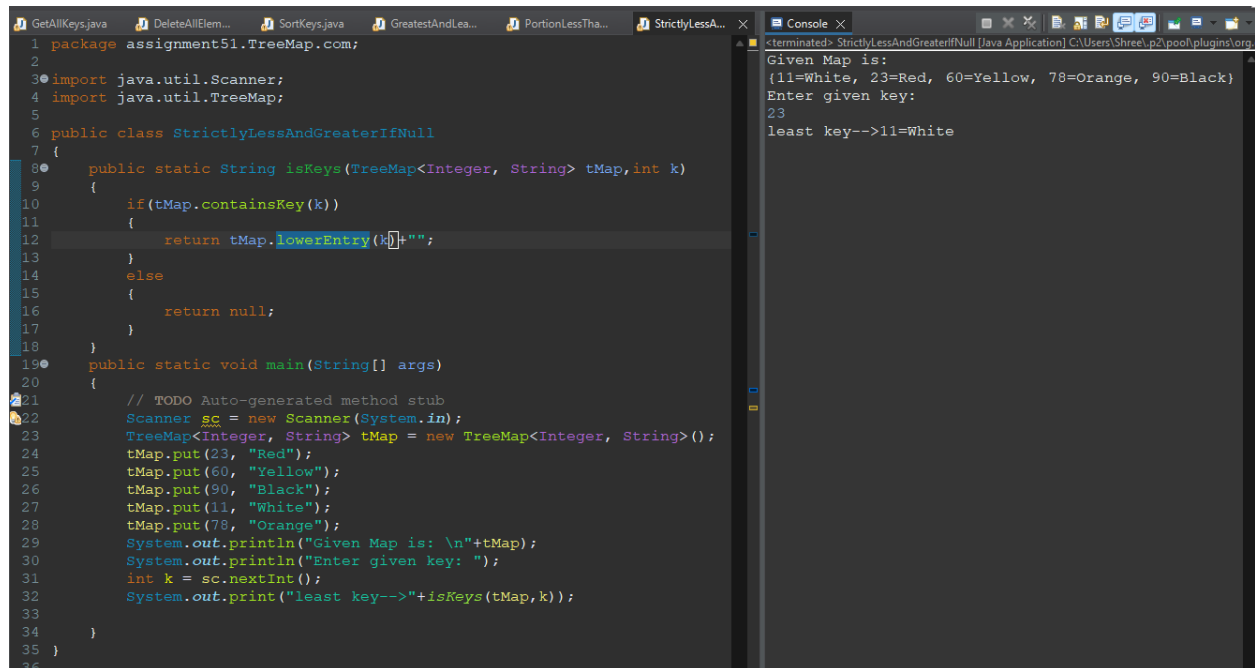


```
1 package assignment51.TreeMap.com;
2
3 import java.util.Scanner;
4 import java.util.TreeMap;
5
6 public class StrictlyLessAndGreaterIfNull
7 {
8     public static String isKeys(TreeMap<Integer, String> tMap,int k)
9     {
10         if(tMap.containsKey(k))
11         {
12             return tMap.lowerEntry(k).toString();
13         }
14         else
15         {
16             return null;
17         }
18     }
19     public static void main(String[] args)
20     {
21         // TODO Auto-generated method stub
22         Scanner sc = new Scanner(System.in);
23         TreeMap<Integer, String> tMap = new TreeMap<Integer, String>();
24         tMap.put(23, "Red");
25         tMap.put(60, "Yellow");
26         tMap.put(90, "Black");
27         tMap.put(11, "White");
28         tMap.put(78, "Orange");
29         System.out.println("Given Map is: \n"+tMap);
30         System.out.println("Enter given key: ");
31         int k = sc.nextInt();
32         System.out.print("least key-->" + isKeys(tMap,k));
33     }
34 }
35
36
```

Console Output:

```
<terminated> StrictlyLessAndGreaterIfNull [Java Application] C:\Users\Shree\p2\poof\plugins\org
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.

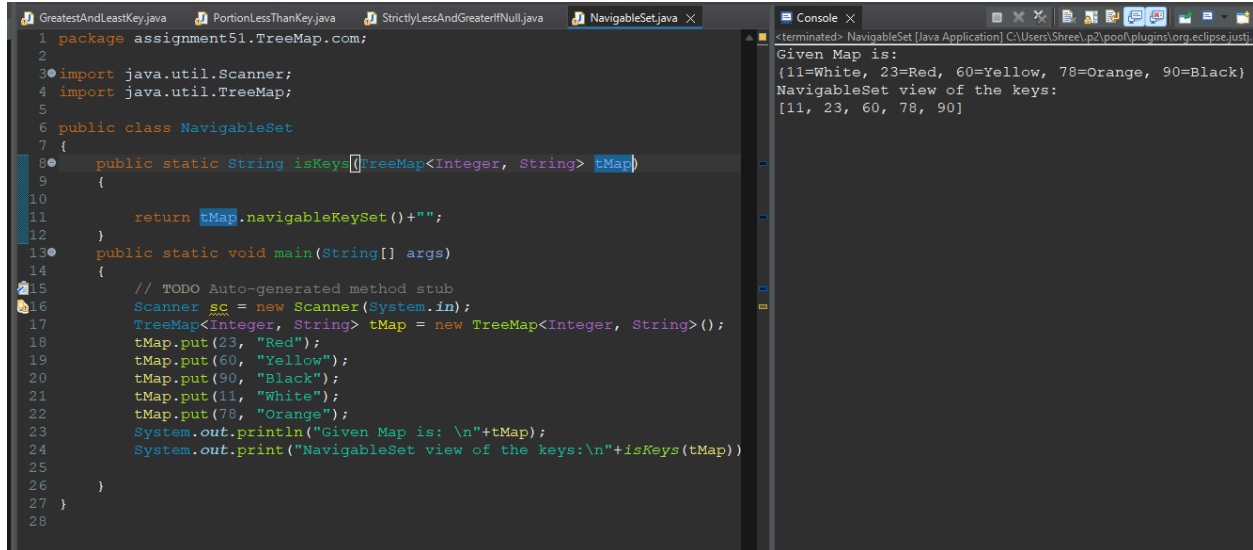


```
1 package assignment51.TreeMap.com;
2
3 import java.util.Scanner;
4 import java.util.TreeMap;
5
6 public class StrictlyLessAndGreaterIfNull
7 {
8     public static String isKeys(TreeMap<Integer, String> tMap,int k)
9     {
10         if(tMap.containsKey(k))
11         {
12             return tMap.lowerEntry(k).toString();
13         }
14         else
15         {
16             return null;
17         }
18     }
19     public static void main(String[] args)
20     {
21         // TODO Auto-generated method stub
22         Scanner sc = new Scanner(System.in);
23         TreeMap<Integer, String> tMap = new TreeMap<Integer, String>();
24         tMap.put(23, "Red");
25         tMap.put(60, "Yellow");
26         tMap.put(90, "Black");
27         tMap.put(11, "White");
28         tMap.put(78, "Orange");
29         System.out.println("Given Map is: \n"+tMap);
30         System.out.println("Enter given key: ");
31         int k = sc.nextInt();
32         System.out.print("least key-->" + isKeys(tMap,k));
33     }
34 }
35
36
```

Console Output:

```
<terminated> StrictlyLessAndGreaterIfNull [Java Application] C:\Users\Shree\p2\poof\plugins\org
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.

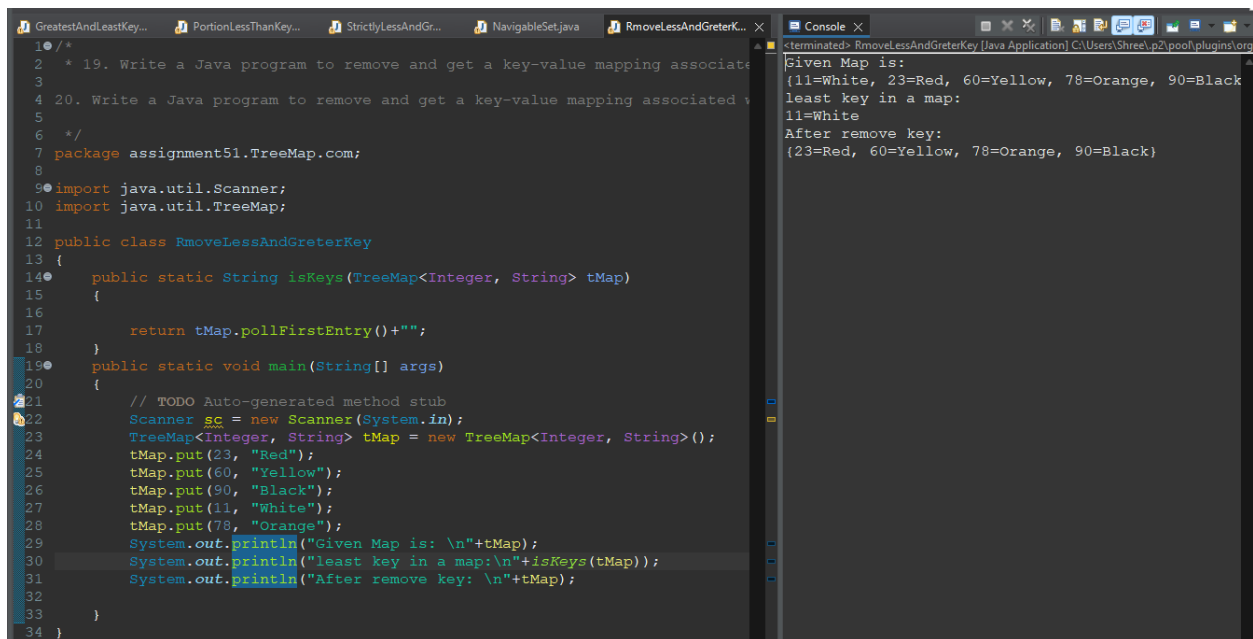


```
1 package assignment51.TreeMap.com;
2
3 import java.util.Scanner;
4 import java.util.TreeMap;
5
6 public class NavigableSet
7 {
8     public static String isKeys(TreeMap<Integer, String> tMap)
9     {
10
11         return tMap.navigableKeySet()+"";
12     }
13     public static void main(String[] args)
14     {
15         // TODO Auto-generated method stub
16         Scanner sc = new Scanner(System.in);
17         TreeMap<Integer, String> tMap = new TreeMap<Integer, String>();
18         tMap.put(23, "Red");
19         tMap.put(60, "Yellow");
20         tMap.put(90, "Black");
21         tMap.put(11, "White");
22         tMap.put(78, "Orange");
23         System.out.println("Given Map is: \n"+tMap);
24         System.out.print("NavigableSet view of the keys:\n"+isKeys(tMap))
25     }
26 }
27
28
```

Console Output:

```
<terminated> NavigableSet [Java Application] C:\Users\Shree\p2\plugins\org.eclipse.justi...
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.



```
1 /*
2  * 19. Write a Java program to remove and get a key-value mapping associat
3
4  * 20. Write a Java program to remove and get a key-value mapping associat
5
6  */
7 package assignment51.TreeMap.com;
8
9 import java.util.Scanner;
10 import java.util.TreeMap;
11
12 public class RemoveLessAndGreterKey
13 {
14     public static String isKeys(TreeMap<Integer, String> tMap)
15     {
16
17         return tMap.pollFirstEntry()+"";
18     }
19     public static void main(String[] args)
20     {
21         // TODO Auto-generated method stub
22         Scanner sc = new Scanner(System.in);
23         TreeMap<Integer, String> tMap = new TreeMap<Integer, String>();
24         tMap.put(23, "Red");
25         tMap.put(60, "Yellow");
26         tMap.put(90, "Black");
27         tMap.put(11, "White");
28         tMap.put(78, "Orange");
29         System.out.println("Given Map is: \n"+tMap);
30         System.out.println("least key in a map:\n"+isKeys(tMap));
31         System.out.println("After remove key: \n"+tMap);
32     }
33 }
34
```

Console Output:

```
<terminated> RemoveLessAndGreterKey [Java Application] C:\Users\Shree\p2\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.

```
1 /*
2  * 19. Write a Java program to remove and get a key-value mapping associated
3  *
4  * 20. Write a Java program to remove and get a key-value mapping associated
5  *
6  */
7 package assignment51.TreeMap.com;
8
9 import java.util.Scanner;
10 import java.util.TreeMap;
11
12 public class RmoveLessAndGreterKey
13 {
14     public static String isKeys(TreeMap<Integer, String> tMap)
15     {
16
17         return tMap.pollLastEntry()+"";
18     }
19     public static void main(String[] args)
20     {
21         // TODO Auto-generated method stub
22         Scanner sc = new Scanner(System.in);
23         TreeMap<Integer, String> tMap = new TreeMap<Integer, String>();
24         tMap.put(23, "Red");
25         tMap.put(60, "Yellow");
26         tMap.put(90, "Black");
27         tMap.put(11, "White");
28         tMap.put(78, "Orange");
29         System.out.println("Given Map is: \n"+tMap);
30         System.out.println("greatest key in a map:\n"+isKeys(tMap));
31         System.out.println("After remove key: \n"+tMap);
32     }
33 }
34
35
```

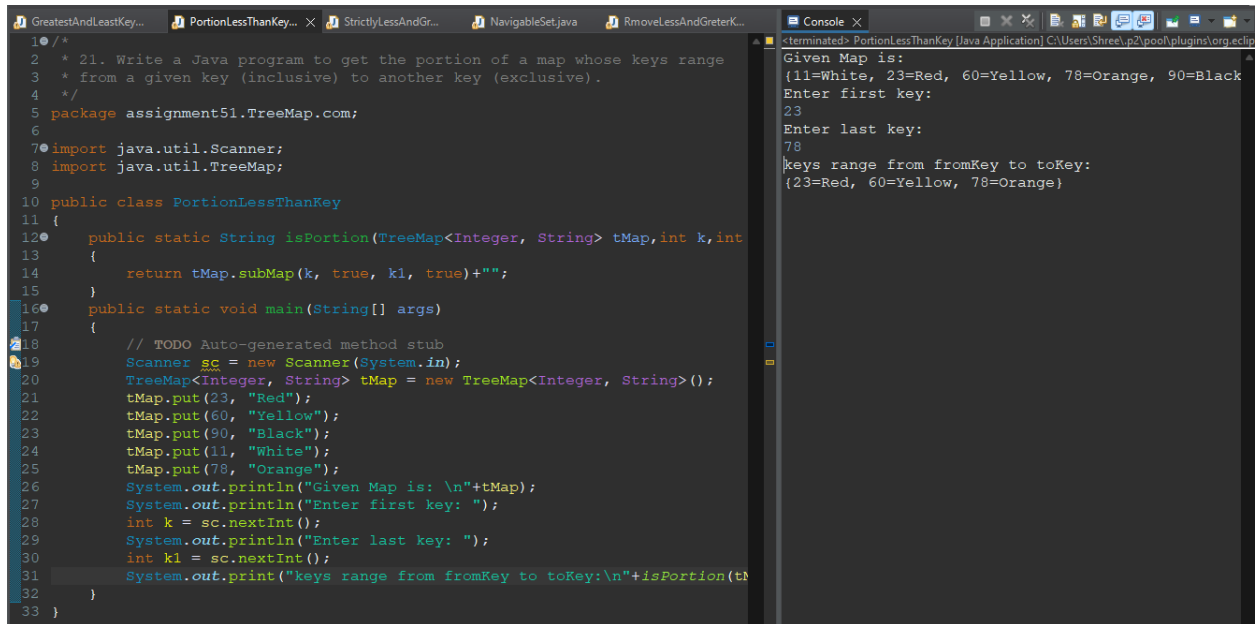
```
<terminated> RmoveLessAndGreterKey [Java Application] C:\Users\Shree\p2\pooh\plugins\org...
Given Map is:
{11=White, 23=Red, 60=Yellow, 78=Orange, 90=Black}
greatest key in a map:
90=Black
After remove key:
{11=White, 23=Red, 60=Yellow, 78=Orange}
```

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

```
1 /*
2  * 21. Write a Java program to get the portion of a map whose keys range
3  * from a given key (inclusive) to another key (exclusive).
4  */
5 package assignment51.TreeMap.com;
6
7 import java.util.Scanner;
8 import java.util.TreeMap;
9
10 public class PortionLessThanKey
11 {
12     public static String isPortion(TreeMap<Integer, String> tMap,int k,int
13     {
14         return tMap.subMap(k, k1)+"";
15     }
16     public static void main(String[] args)
17     {
18         // TODO Auto-generated method stub
19         Scanner sc = new Scanner(System.in);
20         TreeMap<Integer, String> tMap = new TreeMap<Integer, String>();
21         tMap.put(23, "Red");
22         tMap.put(60, "Yellow");
23         tMap.put(90, "Black");
24         tMap.put(11, "White");
25         tMap.put(78, "Orange");
26         System.out.println("Given Map is: \n"+tMap);
27         System.out.println("Enter first key: ");
28         int k = sc.nextInt();
29         System.out.println("Enter last key: ");
30         int k1 = sc.nextInt();
31         System.out.print("portion of less than or equal given Key:\n"+isPortion(tMap,k,k1));
32     }
33 }
34
```

```
<terminated> PortionLessThanKey [Java Application] C:\Users\Shree\p2\pooh\plugins\org...
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.



The screenshot shows an Eclipse IDE with a Java project. The editor displays a file named `PortionLessThanKey.java` with the following code:

```
1  /*
2   * 21. Write a Java program to get the portion of a map whose keys range
3   * from a given key (inclusive) to another key (exclusive).
4   */
5   package assignment51.TreeMap.com;
6
7   import java.util.Scanner;
8   import java.util.TreeMap;
9
10  public class PortionLessThanKey
11  {
12      public static String isPortion(TreeMap<Integer, String> tMap, int k, int
13      {
14          return tMap.subMap(k, true, k1, true) + "";
15      }
16      public static void main(String[] args)
17      {
18          // TODO Auto-generated method stub
19          Scanner sc = new Scanner(System.in);
20          TreeMap<Integer, String> tMap = new TreeMap<Integer, String>();
21          tMap.put(23, "Red");
22          tMap.put(60, "Yellow");
23          tMap.put(90, "Black");
24          tMap.put(11, "White");
25          tMap.put(78, "Orange");
26          System.out.println("Given Map is: \n" + tMap);
27          System.out.println("Enter first key: ");
28          int k = sc.nextInt();
29          System.out.println("Enter last key: ");
30          int k1 = sc.nextInt();
31          System.out.print("keys range from fromKey to toKey:\n" + isPortion(t
32      }
33  }
```

The console output shows the execution of the program:

```
<terminated> PortionLessThanKey [Java Application] C:\Users\Shree.p2\poo\plugins\org.eclipse
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}
```