Treemap Class in Java

Submitted by: Prashant Dagar
Enroll. No. : 190020203031
Course : B. Tech. – CSE

Submitted to: Ms. Neha Kapoor

Code:

```
import java.util.*;
import java.lang.String;
public class Treemp {
   public static void main(String args[]){
       Scanner abc = new Scanner(System.in);
       TreeMap<Integer,String> t1 = new TreeMap<Integer,
String>();
       TreeMap<Integer,String> t2 = new TreeMap<Integer,
String>();
       int w=1;
       while(w==1){
```

```
        System.out.println("Note: It is important to make a new
TreeMap and enter some values to see the effects of most of
the methods");
        System.out.println("Enter the value you want to know
more about:\n1.Constructors \n2.Methods");
        int ch = abc.nextInt();
        switch(ch){
            case 1:{
                System.out.println("The constructors are as follows:
\nInput a value you want to know more about \n
1.TreeMap()\n 2. TreeMap(Comparator comp)\n 3.
TreeMap(Map m)\n 4. TreeMap(SortedMap sm) ");
                int c= abc.nextInt();
                switch(c){
                    case 1:{
                        System.out.println("This constructor creates a
empty news TreeMap.");
                        TreeMap<Integer, String> t = new
TreeMap<Integer, String>();
                        break;
                    }
                    case 2:{
                        System.out.println("This constructor constructs
an empty tree-based map that will be sorted using the
Comparator comp.");
                        break;

                    }
                    case 3:{
                        System.out.println("This constructor initializes
a tree map with the entries from m, which will be sorted using
the natural order of the keys.");
                        break;
```

```
                }
                case 4:{
                    System.out.println("This constructor initializes
a tree map with the entries from the SortedMap sm, which
will be sorted in the same order as sm.");
                    break;
                }
                default:System.out.println("Wrong choice
entered");

              }
             break;
           }
           case 2:{
              System.out.println("Enter the choice from the menu:
\n 1. Object put(Object key, Object value)\n 2. void clear()\n
3. Object clone() \n 4. boolean containsKey(Object Key) \n 5.
boolean containsValue(Object value)\n 6. Set entrySet() \n 7.
Object firstKey() \n 8. Object get(Object key)\n 9. SetkeySet()
\n 10. Object lastKey() \n 11. void putAll(Map map) \n
12.Object remove(Object key) \n 13. int size() \n 14.
SortedMap subMap(Object fromKey, Object toKey)\n 15.
SortedMap tailMap(Object fromKey) \n 16. Collection
values()\n 17. SortedMap headMap(Object toKey)\n 18.
Object firstEntry()");
              int a = abc.nextInt();
              switch(a){
                    case 1:{
                    int b=1;
                      while(b==1)
                        {
                        System.out.println("Enter the key:");
                        int k = abc.nextInt();
```

```
                    abc.nextLine();
                    System.out.println("Enter the string:");
                    String n = abc.nextLine();
                    t1.put(k, n);
                    System.out.println("Do you want to neter
more key and values??\n Press Y for yes, any athor key for
NO");
                    char op= abc.next().charAt(0);
                    if(op!='Y'){
                       b=0;
                    }
                    }
                    break;


                }
                case 2:{
                    System.out.println("This removes all the
mappings from the TreeMap");
                    t1.clear();
                    break;
                }
                case 3:{
                    System.out.println("This returns a copy of
TreeMap instance like this:");
                     System.out.println(t1.clone());
                     break;
                }
                case 4:{
                    System.out.println("This method returns true if
the map ocntains a mapping for the given key.\n Enter a
key:");
                    int ab = abc.nextInt();
```

```
                System.out.println(t1.containsKey(ab));
                break;
            }
            case 5:{
                System.out.println("this method returns true if
the map contains a the given value.\n Enter a value:");
                String e =abc.nextLine();
                System.out.println(t1.containsValue(e));
                break;
            }
            case 6:{
                System.out.println("this method returns a set
view of the mappings contained in this map.");
                System.out.println(t1.entrySet());
                break;
            }
            case 7:{
                System.out.println("This method returns the
first (lowest) key currently in this sorted map.");
                System.out.println(t1.firstKey());
                break;
            }
            case 8:{
                System.out.println("This method returns the
value specified to a given key.\n Enter a key:");
                int r = abc.nextInt();
                System.out.println("The associated value to it
is:");
                System.out.println(t1.get(r));
                break;
            }
            case 9:{
```

```
                System.out.println("This method returns a Set
view of the keys contained in this map.");
                System.out.println(t1.keySet());
                break;
            }
            case 10:{
                System.out.println(" this method returns the
last (highest) key currently in this sorted map.");
                System.out.println(t1.lastKey());
                break;
            }
            case 11:{
                System.out.println("Copies all of the mappings
from the specified map to this map.");
                t1.putAll(t2);
                System.out.println("Second map after copying
is: ");
                System.out.println(t2.clone());
                break;
            }
            case 12:{
                System.out.println("This method removes the
mapping for this key from this TreeMap if present.\nEnter the
key :");
                int g=abc.nextInt();
                t1.remove(g);
                System.out.println("The map after removing
the key is: " + t1.clone());
                break;
            }
            case 13:{
                System.out.println("Returns the number of
key-value mappings in this map.");
```

```
                    System.out.println(t1.size());
                    break;
                }
                case 14:{
                    System.out.println("Returns a view of the
portion of this map whose keys range from fromKey,
inclusive, to toKey, exclusive.\n Enter first key:");
                    int u = abc.nextInt();
                    System.out.println("enter the second key");
                    int y=abc.nextInt();
                    SortedMap<Integer,String> s =
(SortedMap<Integer,String>)t1;
                    System.out.println("The elements between the
keys are:"+  s.subMap(u,y));
                    break;
                }
                case 15:{
                    System.out.println("Returns a view of the
portion of this map whose keys are greater than or equal to
fromKey.\n Enter first key");
                    int q = abc.nextInt();
                    SortedMap<Integer,String> s2 =
(SortedMap<Integer,String>)t1;
                    System.out.println("The elements between the
keys are:"+  s2.tailMap(q));
                    break;

                }
                case 16:{
                   System.out.println("Returns a collection view
of the values contained in this map.");
                    System.out.println(t1.values());
                    break;
```

```
                }
                case 17:{
                    System.out.println("Returns a view of the
portion of this map whose keys are strictly less than
toKey.\nEnter the key:");
                    int o=abc.nextInt();
                    SortedMap<Integer,String> s2 =
(SortedMap<Integer,String>)t1;
                    System.out.println("The elements between the
keys are:"+  s2.headMap(o));
                    break;
                }
                case 18:{
                    System.out.println("This method gives the first
key (lowest) in the mapping:");
                    System.out.println("The first key here is: "+
t1.firstEntry());
                    break;
                }

            }
            break;
        }


        }
        System.out.println("Press 1 to go to main menu, any
other key to exit ");
        int ar= abc.nextInt();
        if(ar!=1){
          w=0;
        }
        }
```

```
            abc.close();
        }

    }
```