```java
/** This class represents a heap and all fns used in mainfn are defined this class
 * @Name:             Nitish
 * @Studentid:        7201791
 * @Assignment:       3
 */
import java.io.*;
import java.util.Scanner;

public class Drugheaps{
    int lngth =0;                           //length
    FileWriter wrt;                       // FileWriter Object
    Drug[] infoofdrugs;         // array infoofdrugs to store info of drugs as each information of each drug gets splitted into multiple string when tab comes



    /**This method is used  to restore heap-order.
     * @param indx- indx of infoofdrugs
     */
    public void trickleDown(int indx){

        Drug curr=infoofdrugs[indx];
        //keep on traversing until indx*2 is < lngth+1
        int ch;//child
        while (indx*2<lngth+1){
            ch=2*indx;
            if(ch!=lngth)
            {
                if( infoofdrugs[ch+1].drugBankID.compareTo(infoofdrugs[ch].drugBankID)<0){
                    ++ch;
                }
                else{
                  ////
                }
            }
            //comparing drugbankids
             if(infoofdrugs[ch].drugBankID.compareTo(curr.drugBankID)<0){
                infoofdrugs[indx]=infoofdrugs[ch];
            }
            else{
                break;//control out of while loop
            }
            //assigning value of child to index
            indx=ch;
        }
        infoofdrugs[indx]=curr;
    }


    /** This method used to build heap whcih is further used to further used to perform operations
     */
    public void heapBuild(){
        //casting to int
        int hb=(int) Math.floor(lngth /2.0);
        //trcikle down till floor value>0
        while(hb>0){
            trickleDown(hb);
            --hb;
        }
    }

    /** This method removes minimum value from heap basically in heap root is element that is removed bcoz in min heap is minimum value and all elements are > than root
     * @return The Drug Object with least value of DrugBankID
     */
    public Drug removeMinVal(){
        Drug val= infoofdrugs[1];                 // starting index
        infoofdrugs[1]= infoofdrugs[lngth];
        lngth--;
        if(0<lngth){
            trickleDown(1);
        }
        return val;
    }
    /** HEAP SORT
     * Heap sort algorithm basically since its a min heap so root is smallest and all elements are greater than root
     * first create heap and then delete element one by one
     */
    public void heapSort(){
        try{
            FileWriter writter=new FileWriter("dockedApprovedheapsort.tab");   //Filewriter object
            for(int k=1;k-1<lngth;++k)
            {
                Drug heap= removeMinVal();
                writter.write(heap.getGenericname()+"  "+heap.getSmiles()+"  "+heap.getDrugbankID()+"  "+heap.getUrl()+"  "+heap.getDruggroups()+"  "+heap.getScores()+"\n");
            }
            writter.close();      // stream close
        }
        catch (IOException exceptions){
            System.out.println("wrong");
        }
    }

    /**
     * readdata fn is used to read data from dockedapproved file and once data is read further operations are performed
     * @return type-void
     */
    public void readData() {
        try {
            int iterator = 0;    //iterartor
            File textdoc = new File("dockedApproved.tab");          //dockedapproved file passed as argument
            Scanner scn = new Scanner(textdoc);                          // Scanner Object
            String line = scn.nextLine();                                // for reading the line of dockedapproved

            while (scn.hasNext()) {
                line = scn.nextLine();
                iterator++;
            }
            scn.close();
            infoofdrugs = new Drug[iterator];
            textdoc = new File("dockedApproved.tab");
            String[] original;                                           //original to store different string values which will be splitted in separate strings
            scn = new Scanner(textdoc);
            line = scn.nextLine();
            for (int i = 1; i <=infoofdrugs.length-1; ++i) {
                original = scn.nextLine().split("\t");
                //trim strings basis of tab and store in each index of infofdrugs array
                infoofdrugs[i] = new Drug(original[0].trim(), original[1].trim(), original[2].trim(), original[3].trim(), original[4].trim(), Double.parseDouble(original[5].trim()));
                lngth++;
            }
        } catch (FileNotFoundException exceptions) {                     // file not found
            System.out.print("file donot found");
        }
    }
    /**This method performs  traversal on drugheap
```

```java
and result shown on inorder.tab
     * @param idx- indx of infoofdrugs
     */
    public void inordTraversal(int idx){
        try{
            //Base case if we trying to access invalid index
            wrt.write("");
            if(lngth<idx){
                return;
            }
            //recurive case first traverse left child and then right child
            inordTraversal(idx*2);    // processing left child of heap
            //WRITING IN FILE instance variables value by accessing index and then accessing fns
            wrt.write(infoofdrugs[idx].getGenericname()+"  "+ infoofdrugs[idx].getSmiles()+"  "+ infoofdrugs[idx].getDrugbankID()+"  "+ infoofdrugs[idx].getUrl()+"  "+ infoofdrugs[idx].getDruggroups()+"  "+ infoofdrugs[idx].getScores()+" "+"\n");
            System.out.println();
            inordTraversal(idx*2+1);  // processing right child of heap
        }
        catch( IOException exceptions){
            System.out.println("Something wrong");
        }
    }
    //constructor of drugheaps so that readdata() can read data from file
    // Constructor of class runs as soon as object of class is created
    public Drugheaps(){
        readData();
    }

}

```