

## SortedArray.java

```
////////////////////////////////////
//www.karamian.com
//Program SortedArray
////////////////////////////////////

import java.io.*;
import java.util.*;

//*****
// Class SortedArray
//*****

class SortedArray
{
    protected int top;
    public BufferedReader stdin;
    protected int sortedArray[];
    protected final int ARRAYSIZE=10;
    protected int position;
    protected int insertNum;
    protected int newNum;
    protected boolean emptyArr;
    protected int midpoint;

    //=====
    // The SortedArray constructor.
    //=====

    public SortedArray () throws IOException
    {
        top=- 1;
        stdin=new BufferedReader(new InputStreamReader(System.in));
        sortedArray=new int[ARRAYSIZE];
        sortedArray[0]=0;
        position=0;
        insertNum=0;
        newNum=0;
        emptyArr=false;
    } // end SortedArray constructor

    //=====
    // The appropriatePosition method.
    //=====

    protected int appropriatePosition(int x, int n) throws IOException
    {
        int j=(x- 1);
        if(x==0)
        {
            position=0;
        }
        if(x>0)
        {
            if(n>sortedArray[j])
            {
                position=x;
            }
            if(n<=sortedArray[j])
            {
                int l=0;
                for(int i=0; i<x; i++)
                {
                    if(n>=sortedArray[i])

```

## SortedArray.java

```
        {
            l++;
        }
        position=l;
    }
}
return position;
}

} // end appropriatePosition method.

//=====
// The full method.
//=====

protected void full(int x) throws IOException
{
    int j=(sortedArray.length+5);
    int tempArray[]=new int[j];
    if(x>=sortedArray.length-1)
    {
        for(int i=0; i<sortedArray.length; i++)
            tempArray[i]=sortedArray[i];
        sortedArray=tempArray;
    }
}

} // end full method.

//=====
// The insert method.
//=====

protected int insert() throws IOException
{
    System.out.print("Enter number to insert: ");
    newNum=Integer.parseInt(stdIn.readLine());
    top++;
    full(top);
    appropriatePosition(top, newNum);
    int i=position;
    int a=0;
    int b=(position+1);
    int c;
    if (position>=top)
    {
        sortedArray[top]=newNum;
    }
    if (position<top)
    {
        while(i!=top)
        {
            c=sortedArray[position];
            sortedArray[position]=sortedArray[b];
            sortedArray[b]=c;
            b++;
            i++;
        }
        sortedArray[position]=newNum;
    }
    return top;
}

} // end insert method.
```

## SortedArray.java

```
//=====
// The find method.
//=====

protected int find(int k, int l, int r) throws IOException
{
    midpoint=(l+r)/2;
    if(l>r)
    {
        midpoint=-1;
        return midpoint;
    }
    else if(k==sortedArray[midpoint])
    {
        return midpoint;
    }
    else if(k>sortedArray[midpoint])
    {
        return find(k, midpoint+1, r);
    }
    else
    {
        return find(k, l, midpoint-1);
    }
}

} // end find method.

//=====
// The empty method.
//=====

protected boolean empty(int t) throws IOException
{
    if(t>=0)
        emptyArr=false;
    else
        emptyArr=true;
    return emptyArr;
}

} // end empty method.

//=====
// The delete method.
//=====

protected int delete() throws IOException
{
    System.out.print("Enter number to delete: ");
    newNum=Integer.parseInt(stdIn.readLine());
    empty(top);
    if(emptyArr==true)
    {
        System.out.println("Sorry, there is no data entered currently.");
    }
    else
    {
        find(newNum, 0, top);
        if(midpoint== -1)
        {
            System.out.println("Number not found.");
        }
        else
        {
            for(int i=midpoint; i!=top; i++)
                Page 3
```

SortedArray.java

```
        {
            sortedArray[i]=sortedArray[i+1];
        }
        top--;
    }
}
return top;
} // end delete method.

//=====
// The clear method.
//=====

protected int clear() throws IOException
{
    System.out.println("All data cleared.");
    top-- 1;
    return top;
} // end clear method.

//=====
// The print method.
//=====

public void print()
{
    for(int i=0; i<=top; i++)
        System.out.println("Position "+ (i+1)+ " : "+ "\t"+ sortedArray[i]);
} // end print method.
} //end of Class SortedArray.
```