**LAB # 02**

**Linear Array Implementation**

**Object**

Implementing linear array and associated methods.

**Theory**

## **Linear Array**

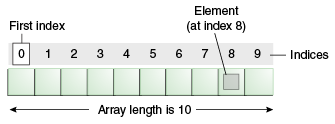
List of finite number N of homogenous data elements (i.e. data elements of same type) or as an ADT any data type such that

* The elements of the array are referenced respectively by an index set consisting of N consecutive number
* The elements of the array are stored respectively in successive memory location

## **Arrays in java**

Static Array (Single dimension)

* The length of an array is established when the array is created. After creation, its length is fixed.



## **Example**

int[] anArray;

// allocates memory for 10 integers

anArray = new int[10];

## **LinearArray Class**

|  |  |
| --- | --- |
| Class Name : LinearArray | Super Class: Object |
| <Object > LArray []  Int N |  |
| Responsibilities |  |
| LinearArray()  Create an empty larray with the default size 10, if initial capacity is not mentioned | [**void clear()**](http://www.tutorialspoint.com/java/util/arraylist_clear.htm)  Remove all the elements from the array |
| LinearArray(int initialCapacity)  Create an empty larray according to the initial capacity | [**int size()**](http://www.tutorialspoint.com/java/util/arraylist_size.htm)  Returns the number of elements in array |
| [**int search(item)**](http://www.tutorialspoint.com/java/util/arraylist_clear.htm)  Searches element in the array and returns index if found , else returns -1 | [**E getElement(K)**](http://www.tutorialspoint.com/java/util/arraylist_get.htm)  Returns the element at the specified position in the array |
| [**void insert(item)**](http://www.tutorialspoint.com/java/util/arraylist_add.htm)  Add elements at the end of the array if Index is not mentioned | [**Item delete(item)**](http://www.tutorialspoint.com/java/util/arraylist_remove.htm)  Searches the element and removes it at the specified position in the array and returns the element |
| [**void insert(K, item)**](http://www.tutorialspoint.com/java/util/arraylist_add_index.htm)  Add elements at the specific Index number | [**bol isFull()**](http://www.tutorialspoint.com/java/util/arraylist_size.htm)  Returns TRUE if array is full else returns FALSE |
| [**Item delete(K)**](http://www.tutorialspoint.com/java/util/arraylist_remove.htm)  Removes the element at the specified positionin the array and returns the element | [**bol isEmpty()**](http://www.tutorialspoint.com/java/util/arraylist_size.htm)  Returns TRUE if array is empty else returns FALSE |

## **Search Algorithm**

1. [Initialize Index to 0] Set LOC := 0

2. [Search for the ITEM till the end of the array or till the ITEM is found]

Repeat while LOC < =N OR LARRAY[LOC] :≠ ITEM

If LARRAY[ITLOC] := ITEM Then Return LOC

Else LOC := LOC + 1

[End of If Structure]

[End of loop]

1. [Unsuccessful Search] Set LOC := -1 And Return LOC

4. Exit

**Insert Algorithm**:

1. [Initialize Counter] Set J:=N
2. Repeat Steps 3 and 4 while J≥K
3. [Move Jth element downward]

Set LArray[J+1] := LArray[J]

1. [Decrease counter] Set J:=J-1

[End of Step 2 loop]

1. [Insert element] Set LArray[K]:=ITEM
2. [Reset N] N:= N+1
3. Return

**Delete Algorithm:**

1. Set ITEM:=LArray[K]
2. Repeat for J:=K to N-J
3. [Move Jth element upward] Set LA[J] := LA[J+1]

[End of Step 2 loop]

4. [Reset N] N:= N-1

5. Return item

**Task**

1. Write a program to implement all basic operations in linear array (specified in linear array class).write a demo class to show the working of array.
2. Add a method in the class that takes array and merge it with the existing one.
3. Add a method in the same class that splits the existing array into two. The method should search a key in array and if found splits the array from that index of the key