28 Oct 2020 Rinoy Kuriyakose R3 56

**Experiment: 12**

**Aim:**

|  |  |
| --- | --- |
|  | Write a Java program for the following:  A) Create a doubly linked list of elements.  B) Delete a given element from the above list.  C) Display the contents of the list after deletion. |

**Concept Used:**

Collection Framework.

**Algorithm:**

1. import java.util .\*

2. Initialise LinkedList<String> object list

3. Add members using list.add() method

4. Print list

5. Remove members using list.remove() method

6. Initialise Iterator object itr = list.iterator()

7. while(itr.hasNext()) //Displaying the list

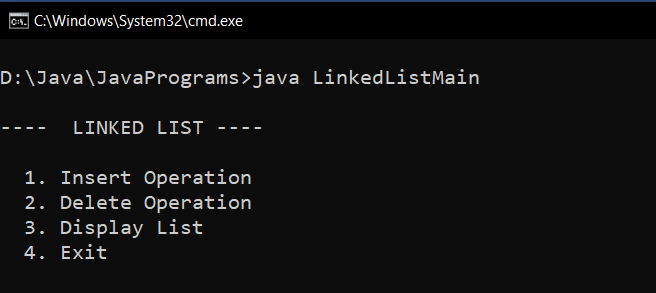
8. Print itr.next()

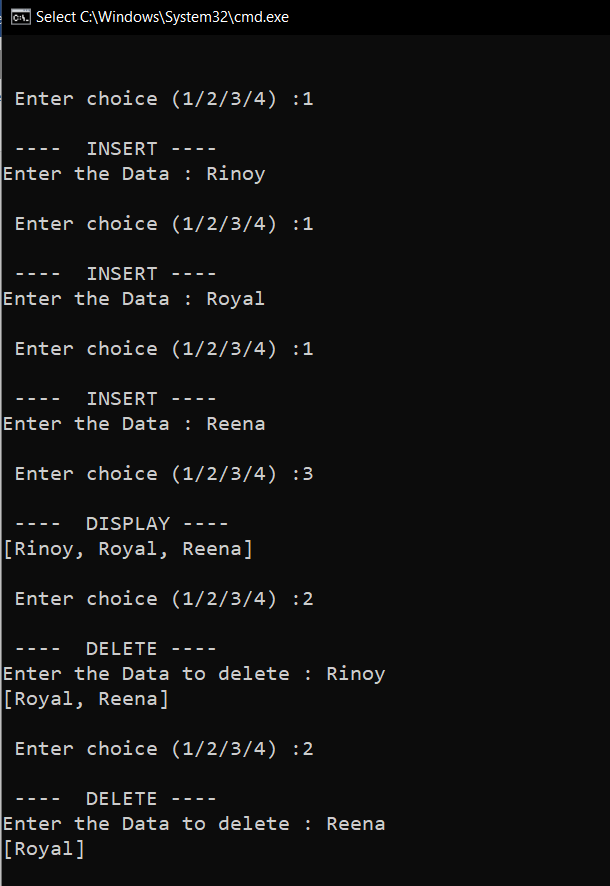
9. endwhile

**Program:**

import java.util.\*;  
import java.io.\*;  
  
class LinkedListMain {  
 public static void main(String[] args) throws IOException{  
 LinkedList<String> data = new LinkedList<>();  
 BufferedReader br=new BufferedReader(new InputStreamReader(System.in));  
 int ans=1;  
 System.out.println(“ \n---- LINKED LIST ---- \n”);  
 System.out.println(“ 1. Insert Operation “);  
 System.out.println(“ 2. Delete Operation “);  
 System.out.println(“ 3. Display List “);  
 System.out.println(“ 4. Exit \n”);  
 while(ans!=4) {  
 System.out.print(“\n Enter choice (1/2/3/4) :”);  
 ans=Integer.parseInt(br.readLine());  
 switch(ans){  
 case 1 : System.out.println(“\n ---- INSERT ---- “);  
 System.out.print(“Enter the Data : “);  
 data.add(br.readLine());  
 break;  
 case 2 : System.out.println(“\n ---- DELETE ----");  
 System.out.print(“Enter the Data to delete : “);  
 data.remove(br.readLine());  
 System.out.println(data);  
 break;  
 case 3 : System.out.println(“\n ---- DISPLAY ----");  
 System.out.println(data);  
 break;  
 case 4 : break;  
 default: System.out.println(“Invalid Choice”);  
 }  
  
 }  
 }  
}

**Output:**

****



**Result:**

A doubly linked list is implemented using Collections framework.

28 Oct 2020 Rinoy Kuriyakose R3 56

Experiment: 13

**Aim:**

Write a Java program that implements the binary search algorithm.

**Concept Used:**

Scanner, Binary Search algorithm.

**Algorithm:**

Algorithm BinarySearch

1. import java.io.\*

2. Read array size,

3. Initialise array arr of size n

4. Read the elements

5. first = 0, last = n –1

6. while (first <= last)

7. mid = (first + last) / 2

8. if(search < arr[mid])

9. last = mid –1

10. else if (search > arr[mid])

11. first = mid + 1

12. else

13. Print “Element found”

14. endif

15. endwhile

16. if(first>last)

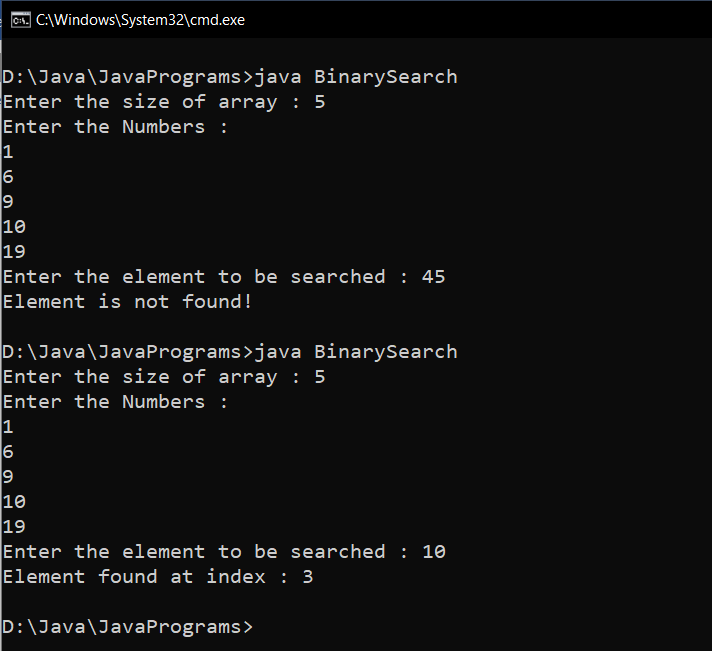
17. print “Element Not Found”

18. endIf

**Program:**

import java.io.\*;  
class BinarySearch{  
 public static void main(String args[]) throws IOException {  
 BufferedReader br=new BufferedReader(new InputStreamReader(System.in));  
 int[] arr = new int[50];  
 int size,search;  
 System.out.print("Enter the size of array : ");  
 size=Integer.parseInt(br.readLine());  
 System.out.println("Enter the Numbers : ");  
 for(int i=0;i<size;i++){  
 arr[i]=Integer.parseInt(br.readLine());  
 }  
 System.out.print("Enter the element to be searched : ");  
 search = Integer.parseInt(br.readLine());  
 int first=0;  
 int last=size-1;  
 int mid = (first + last)/2;  
 while( first <= last ){  
 if ( arr[mid] < search){  
 first = mid + 1;  
 }else if ( arr[mid] == search){  
 System.out.println("Element found at index : " + mid);  
 break;  
 }else{  
 last = mid-1;  
 }  
 mid =(first + last)/2;  
 }  
 if ( first > last ){  
 System.out.println("Element is not found!");  
 }  
 }  
}

**Output:**



**Result:**

Binary Search is performed on an array of integers.