

Day 33

Q. Write a program to add, retrieve and remove the element from the ArrayList.

Answer:

An ArrayList class is a dynamic data structure, which can add and remove the data dynamically. It allows random access of the elements.

Below is the example of ArrayList. It performs addition, retrieval and deletion of elements from the ArrayList. We create generic list which accepts only String type value.

```
import java.util.*;
class ArrayListDemo
{
    public static void main(String[] args)
    {
        ArrayList<String> al = new ArrayList<String>();
        System.out.println("Size of ArrayList: "+al.size());
        //Adding the elements
        al.add("Java");
        al.add("JDBC");
        System.out.println("Elements of first ArrayList: "+al);

        ArrayList<String> al2 = new ArrayList<String>();
        al2.add("EJB");
        al2.add("Struts");
        //Adding the both array
        al2.addAll(al);
        System.out.println("Elements of second ArrayList: "+al2);
        //remove the element
        al2.remove("EJB");
        System.out.println("Elements of ArrayList after deletion: "+al2);
        System.out.println("Size of ArrayList: "+al2.size());

        //Retriving 2nd index element
        System.out.println("The element at 2nd index is: "+al2.get(2));
    }
}
```

Q. Write a program to implement LinkedList and perform different operations on it.

Answer:

In Java LinkedList class having the doubly linked list to store the elements. It extends the AbstractList class and implements List and Deque interfaces. It can store duplicate elements in the list.

In this example of linked list, we perform add, remove and retrieve the elements from the list.

```

import java.util.*;
public class LinkedListDemo
{
    public static void main(String args[])
    {
        // create a linked list
        LinkedList<String> ll = new LinkedList<String>();
        System.out.println("Initial size of LinkedList: "+ll.size());

        // add elements to the linked list
        ll.add("Pune");
        ll.add("Mumbai");
        ll.add("Delhi");
        ll.add("Noida");
        System.out.println("Element of LinkedList: " + ll);
        System.out.println("Size of LinkedList after addition: "+ll.size());

        // remove elements from the linked list
        ll.remove(2);
        System.out.println("Element of LinkedList after deletion: " + ll);
        System.out.println("Size of the LinkedList after deletion: "+ll.size());

        // remove first and last elements
        ll.removeFirst();
        ll.removeLast();
        System.out.println("ll after deleting first and last: " + ll);
        System.out.println("Size of the LinkedList after deletion: "+ll.size());

        //Retriving First and Last element
        System.out.println("Fisrt element of LinkedList: "+ll.getFirst());
        System.out.println("Last element of LinkedList: "+ll.getLast());

        Iterator<String> itr = ll.iterator();
        while(itr.hasNext())
        {
            System.out.print(itr.next()+" , ");
        }
    }
}

```

Q. Write a program to sort and reverse the LinkedList elements.

Answer:

In Java there are also some build function to sort the elements. In this example, we create the SortList class, which is used to compare the elements and sort them in decreasing order.

```

class student
{
    private String name;

```

```

private int regNumber;
public student(String n, int reg)
{
    this.name = n;
    this.regNumber = reg;
}
public String getName()
{
    return name;
}
public void setName(String name)
{
    this.name = name;
}
public int getReg()
{
    return regNumber;
}
public void setReg(int regNumber)
{
    this.regNumber = regNumber;
}
public String toString()
{
    return "Student Name: "+this.name+", Reg Number: "+this.regNumber;
}
}

```

```

import java.util.*;
class StudCompare implements Comparator<student>
{
    @Override
    public int compare(student stud1, student stud2)
    {
        if (stud1.getReg() < stud2.getReg())
        {
            return 1;
        }
        return -1;
    }
}
class SortList
{
    public static void main(String[] args)
    {
        LinkedList<student> list = new LinkedList<student>();
        list.add(new student("John", 1001));
        list.add(new student("Raju", 1005));
        list.add(new student("Sham", 1000));
        list.add(new student("Ravi", 1002));
        Collections.sort(list, new StudCompare());
        System.out.println("Sorted list based on registration number\n");
    }
}

```

```
        System.out.println("**=====**=====**");
        for (student stud: list )
        {
            System.out.println(stud);
        }
        System.out.println("**=====**=====**\n");
    }
}
```