**UCS1313: Object Oriented Programming Using Java Lab**

**2019-2020 Odd – III Semester**

**Assignment – 7 : Collections Framework**

**===============================================**

**===================**

Objective:

To use ADTs and algorithms from Java Collections Framework

------------------------------------------------------------------

1. Write a program to perform string operations using ArrayList. Write functions for the following

a. Append - add at end

b.Insert – add at particular index

c.Find the index of a particular element (Search)

d.Display the list

e.List all string starts with given letter

f.List of all string contains the Substring

g.Sort the elements in ArrayList

h.Remove a particular element

i.Replace one string with another string in ArrayList

j.Remove duplicate elements

import java.util.\*;

public class StringOp{

public static void main(String args[]){

ArrayList<String> arr=new ArrayList<String>();

arr.add("Hi");

arr.add("Hello");

System.out.println("Array is: "+arr);

arr.add(1,"Little");

System.out.println("Array is: "+arr);

String element=new String();

Scanner in=new Scanner(System.in);

element=in.nextLine();

System.out.println(element+" is found at "+arr.indexOf(element));

char chr;

chr=in.next().charAt(0);

ListIterator<String> itr=arr.listIterator();

while(itr.hasNext()){

element=itr.next();

char ch=element.charAt(0);

if(ch==chr)

System.out.println(itr.next());

}

Collections.sort(arr);

System.out.println("Sorted Array is: "+arr);

element=in.nextLine();

ListIterator<String> itr1=arr.listIterator();

while(itr1.hasNext()){

String str=itr1.next();

if(element.equals(str)){

arr.remove(arr.indexOf(str));

}

}

String s1=new String();

String s2=new String();

s1=in.nextLine();s2=in.nextLine();

arr.set(arr.indexOf(s1),s2);

LinkedHashSet<String> hash=new LinkedHashSet<>(arr);

ArrayList<String> arr1=new ArrayList<>(hash);

System.out.println("No duplicate array: "+arr1);

int flag=0;

String sub=new String();

sub=in.nextLine();

ListIterator<String> itr2=arr.listIterator();

while(itr2.hasNext()){

String str=itr2.next();

if(str.contains(sub)){

System.out.println(element+" ");

flag=1;

}

}

if(flag==0){

System.out.println("Not found");

}

}

}

/\*

Output:

Array is: [Hi, Hello]

Array is: [Hi, Little, Hello]

Hello

Hello is found at 2

L

Little

Sorted Array is: [Hello, Hi, Little]

Little

Hi

Hello

No duplicate array: [Hello]

ll

Hello

\*/

2.Write a program to get two integer ArrayList. Perform the following operations

a.Merge the two lists

b.Find Union of two lists

c.Find Intersection of two lists

d.Compare two lists

import java.util.\*;

public class ArrayFunc{

public static void main(String args[]){

ArrayList<Integer> arr1=new ArrayList<Integer>();

ArrayList<Integer> arr2=new ArrayList<Integer>();

int a1,a2;

Scanner in=new Scanner(System.in);

System.out.println("Enter array1");

for(int i=0;i<5;i++){

a1=in.nextInt();

arr1.add(a1);

}

System.out.println("Enter array2");

for(int i=0;i<5;i++){

a2=in.nextInt();

arr2.add(a2);

}

arr1.addAll(arr2);

System.out.println("Merged: "+arr1);

Set<Integer> s=new HashSet<Integer>();

s.addAll(arr1);s.addAll(arr2);

System.out.println("Union: "+s);

List<Integer> l=new ArrayList<Integer>();

for(Integer i: arr1){

if(arr1.contains(i))

l.add(i);

}

System.out.println("Intersection: "+l);

if(arr1.equals(arr2))

System.out.println("Equal");

else

System.out.println("Unequal");

}

}

/\*

Output:

Enter array1

10 24 54 61 32

Enter array2

21 10 61 50 90

Merged: [10, 24, 54, 61, 32, 21, 10, 61, 50, 90]

Union: [32, 50, 21, 54, 24, 10, 90, 61]

Intersection: [10, 24, 54, 61, 32, 21, 10, 61, 50, 90]

Unequal

\*/

3.Using Collection framework, create a doubly linked list of integers and perform the following

operations.

a. Insert element on both sides

b. Delete element on both sides

c. Insert an element at a particular position

d. Delete a particular element

e. Search for a particular element

f. Display list in forward order and backward order

g. Sort the elements in LinkedList

h. Replace one element in the list with another list

i. Remove duplicate elements

import java.util.\*;

public class DLL{

public static void main(String args[]){

LinkedList<Integer> dll=new LinkedList<Integer>();

Scanner in=new Scanner(System.in);

int option;

do{

System.out.println(" Do what?");

System.out.println(" 1.Insert on both sides");

System.out.println(" 2.Delete on both sides");

System.out.println(" 3.Insert at index");

System.out.println(" 4.Delete element ");

System.out.println(" 5.Search element");

System.out.println(" 6.Display in forward and reverse order");

System.out.println(" 7.Sort");

System.out.println(" 8.Replace element with list");

System.out.println(" 9.Remove duplicates" );

option=in.nextInt();

if(option==1){

int x;

System.out.print("Enter element to insert at front");

x=in.nextInt();

dll.addFirst(x);

System.out.print("Enter element to insert at end");

x=in.nextInt();

dll.add(x);

}

else if(option==2){

dll.removeFirst();

dll.removeLast();

}

else if(option==3){

int x,index;

System.out.print("Enter element to insert");

x=in.nextInt();

System.out.print("Enter index to insert at");

index=in.nextInt();

dll.add(index,x);

}

else if(option==4){

int x;

System.out.print("Enter element to delete");

x=in.nextInt();

dll.remove(dll.indexOf(x));

}

else if(option==5){

int x;

System.out.print("Enter element to search for");

x=in.nextInt();

System.out.println("Found at index "+dll.indexOf(x));

}

else if(option==6){

System.out.println("Forward order: "+dll);

Collections.reverse(dll);

System.out.println("Reverse order: "+dll);

Collections.reverse(dll);

}

else if(option==7){

Collections.sort(dll);

System.out.println("Sorted list: "+dll);

}

else if(option==8){

int x,y;

System.out.print("Enter element to replace ");

x=in.nextInt();

System.out.print("Enter replacement ");

y=in.nextInt();

dll.set(dll.indexOf(x),y);

}

else if(option==9){

LinkedList<Integer> noDup=new LinkedList<Integer>();

for(int i=0;i<dll.size();i++){

if(!noDup.contains(dll.get(i)))

noDup.add(dll.get(i));

}

dll=noDup;

System.out.println("Non Duplicate "+dll);

}

else if(option!=0)

System.out.println("Invalid choice ");

else;

}while(option!=0);

}

}

/\*

Output:

Do what?

1.Insert on both sides

2.Delete on both sides

3.Insert at index

4.Delete element

5.Search element

6.Display in forward and reverse order

7.Sort

8.Replace element with list

9.Remove duplicates

1

Enter element to insert at front5

Enter element to insert at end10

Do what?

1.Insert on both sides

2.Delete on both sides

3.Insert at index

4.Delete element

5.Search element

6.Display in forward and reverse order

7.Sort

8.Replace element with list

9.Remove duplicates

1

Enter element to insert at front6

Enter element to insert at end11

Do what?

1.Insert on both sides

2.Delete on both sides

3.Insert at index

4.Delete element

5.Search element

6.Display in forward and reverse order

7.Sort

8.Replace element with list

9.Remove duplicates

1

Enter element to insert at front3

Enter element to insert at end56

Do what?

1.Insert on both sides

2.Delete on both sides

3.Insert at index

4.Delete element

5.Search element

6.Display in forward and reverse order

7.Sort

8.Replace element with list

9.Remove duplicates

1

Enter element to insert at front13

Enter element to insert at end14

Do what?

1.Insert on both sides

2.Delete on both sides

3.Insert at index

4.Delete element

5.Search element

6.Display in forward and reverse order

7.Sort

8.Replace element with list

9.Remove duplicates

6

Forward order: [13, 3, 6, 5, 10, 11, 56, 14]

Reverse order: [14, 56, 11, 10, 5, 6, 3, 13]

Do what?

1.Insert on both sides

2.Delete on both sides

3.Insert at index

4.Delete element

5.Search element

6.Display in forward and reverse order

7.Sort

8.Replace element with list

9.Remove duplicates

2

Do what?

1.Insert on both sides

2.Delete on both sides

3.Insert at index

4.Delete element

5.Search element

6.Display in forward and reverse order

7.Sort

8.Replace element with list

9.Remove duplicates

6

Forward order: [3, 6, 5, 10, 11, 56]

Reverse order: [56, 11, 10, 5, 6, 3]

Do what?

1.Insert on both sides

2.Delete on both sides

3.Insert at index

4.Delete element

5.Search element

6.Display in forward and reverse order

7.Sort

8.Replace element with list

9.Remove duplicates

3

Enter element to insert15

Enter index to insert at3

Do what?

1.Insert on both sides

2.Delete on both sides

3.Insert at index

4.Delete element

5.Search element

6.Display in forward and reverse order

7.Sort

8.Replace element with list

9.Remove duplicates

6

Forward order: [3, 6, 5, 15, 10, 11, 56]

Reverse order: [56, 11, 10, 15, 5, 6, 3]

Do what?

1.Insert on both sides

2.Delete on both sides

3.Insert at index

4.Delete element

5.Search element

6.Display in forward and reverse order

7.Sort

8.Replace element with list

9.Remove duplicates

4

Enter element to delete10

Do what?

1.Insert on both sides

2.Delete on both sides

3.Insert at index

4.Delete element

5.Search element

6.Display in forward and reverse order

7.Sort

8.Replace element with list

9.Remove duplicates

6

Forward order: [3, 6, 5, 15, 11, 56]

Reverse order: [56, 11, 15, 5, 6, 3]

Do what?

1.Insert on both sides

2.Delete on both sides

3.Insert at index

4.Delete element

5.Search element

6.Display in forward and reverse order

7.Sort

8.Replace element with list

9.Remove duplicates

5

Enter element to search for15

Found at index 3

Do what?

1.Insert on both sides

2.Delete on both sides

3.Insert at index

4.Delete element

5.Search element

6.Display in forward and reverse order

7.Sort

8.Replace element with list

9.Remove duplicates

7

Sorted list: [3, 5, 6, 11, 15, 56]

Do what?

1.Insert on both sides

2.Delete on both sides

3.Insert at index

4.Delete element

5.Search element

6.Display in forward and reverse order

7.Sort

8.Replace element with list

9.Remove duplicates

8

Enter element to replace 5

Enter replacement 10

Do what?

1.Insert on both sides

2.Delete on both sides

3.Insert at index

4.Delete element

5.Search element

6.Display in forward and reverse order

7.Sort

8.Replace element with list

9.Remove duplicates

6

Forward order: [3, 10, 6, 11, 15, 56]

Reverse order: [56, 15, 11, 6, 10, 3]

Do what?

1.Insert on both sides

2.Delete on both sides

3.Insert at index

4.Delete element

5.Search element

6.Display in forward and reverse order

7.Sort

8.Replace element with list

9.Remove duplicates

8

Enter element to replace 10

Enter replacement 11

Do what?

1.Insert on both sides

2.Delete on both sides

3.Insert at index

4.Delete element

5.Search element

6.Display in forward and reverse order

7.Sort

8.Replace element with list

9.Remove duplicates

9

Non Duplicate [3, 11, 6, 15, 56]

Do what?

1.Insert on both sides

2.Delete on both sides

3.Insert at index

4.Delete element

5.Search element

6.Display in forward and reverse order

7.Sort

8.Replace element with list

9.Remove duplicates

0

\*/