Ex-no:8 Name: M.Rohith
Date:15-12-2022 3122 21 5001 085

SSN College of Engineering

Department of Computer Science and Engineering

UCS2313 – Object Oriented Programming Lab

II Year CSE - B Section (III Semester)

Academic Year 2022-23

Batch: 2021- 2025

Faculty Incharge :S.Rajalakshmi

Exercise - 8 – Collection Framework

Objective:

1. To perform operations on ArrayList and LinkedList in Java

Sample Learning Outcome:

1. Usage of ArrayList and LinkedList in Java

2. Perform operations using Collection Framework

3. Perform merge, union, intersection and comparison on ArrayLists

Best Practices:

1. Class Diagram usage

2. Naming convention – for file names, variables

3. Comment usage at proper places

4. Prompt messages during reading input and displaying output

5. Incremental program development

6. Modularity

7. All possible test cases in output

Ex-no:8
Date:15-12-2022

Name: M.Rohith
3122 21 5001 085

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**

Program code:

```
import java.util.*;

class List
{
    ArrayList<String> alist;
    List()
    {
        this.alist=new ArrayList<String>();
    }
    void appendList(String data) {
        alist.add(data);
    }
    void insert(String data, int pos) {
        alist.add(pos, data);
    }
    void searchIndex(String element) {
        System.out.println("Index : " + alist.indexOf(element));
    }

    void display() {
        System.out.println(alist);
    }
```

Name: M.Rohith
3122 21 5001 085

```
    void startsWithLetter(String c) {
        for (String s : alist) {
            if (s.startsWith(c)) {
                System.out.println(s);
            }
        }
    }

    void containSubstring(String substring) {
        for (String s : alist) {
            if (s.contains(substring))
                System.out.println(s);
        }
    }
    void sortList() {
        Collections.sort(alist);
    }

    void removeElement(String s) {
        alist.remove(s);
    }

    void replaceString(String string1, String string2) {
        for (String s : alist) {
            if (s.equals(string1)) {
                s=string2;
            }
        }
    }

    void removeDuplicate() {
        for (int i=0; i<alist.size();i++) {
            for (int j = i + 1; j < alist.size(); j++) {
                if (alist.get(i).equals(alist.get(j))) {
                    alist.remove(j);
                }
            }
        }
    }

}
```

Ex-no:8
Date:15-12-2022

Name: M.Rohith
3122 21 5001 085

```
class StringArrayListDemo {
    public static void main(String args[]) {
        Scanner scanner = new Scanner(System.in);
        List object = new List();
        object.appendList("Ford");
        object.appendList("Chevrolet");
        object.appendList("MarutiSuzuki");
        object.appendList("Toyota");
        int choice = 0;
        System.out.println("1.Append - add at end\n2.Insert - add at
particular index\n3.Search - Find the index of a particular
element\n4.Display the list\n5.List all strings start with given
letter\n6.List all strings contains the substring\n7.Sort the
elements in ArrayList\n8.Remove a particular element\n9.Replace one
string with another string in ArrayList\n10.Remove duplicate
elements\n");

        while (choice != -1) {
            System.out.println("\nEnter the choice: ");
            choice = scanner.nextInt();
            switch (choice) {
                case 1:
                    System.out.print("Enter string to add at end
:");
                    object.appendList(scanner.next());
                    object.display();
                    break;
                case 2:
                    System.out.print("Enter string to insert :");
                    String s = scanner.next();
                    System.out.print("Enter the position :");
                    int pos = scanner.nextInt();
                    object.insert(s,pos);
                    object.display();
                    break;
                case 3:
                    System.out.print("Enter string to search :");
                    s = scanner.next();
                    object.searchIndex(s);
                    object.display();
                    break;
```

Name: M.Rohith
3122 21 5001 085

```
                case 4:
                    object.display();
                    break;
                case 5:
                    System.out.print("Enter starting letter of the
strings :");
                    s = scanner.next();
                    object.startsWithLetter(s);
                    object.display();
                    break;
                case 6:
                    System.out.print("Enter substring :");
                    s = scanner.next();
                    object.containSubstring(s);
                    break;
                case 7:
                    object.sortList();
                    object.display();
                    break;
                case 8:
                    System.out.print("Enter string to remove :");
                    s = scanner.next();
                    object.removeElement(s);
                    object.display();
                    break;
                case 9:
                    System.out.print("Enter string that needs to be
replaced :");
                    s = scanner.next();
                    System.out.print("Enter new string:");
                    String newstring = scanner.next();
                    object.replaceString(s, newstring);
                    object.display();
                    break;

                case 10:
                    object.removeDuplicate();
                    object.display();
            }
        }
    }
}
```

Ex-no:8 Name: M.Rohith
Date:15-12-2022 3122 21 5001 085

Output:

```
PS C:\Rohith\Backup\Desktop\SEM 3\OOP-Java\Java programs\Lab programs\Exercise-8> javac Program1.java
PS C:\Rohith\Backup\Desktop\SEM 3\OOP-Java\Java programs\Lab programs\Exercise-8> java StringArrayListDemo
1.Append - add at end
2.Insert - add at particular index
3.Search - Find the index of a particular element
4.Display the list
5.List all strings start with given letter
6.List all strings contains the substring
7.Sort the elements in ArrayList
8.Remove a particular element
9.Replace one string with another string in ArrayList
10.Remove duplicate elements


Enter the choice:
1
Enter string to add at end :India
[Ford, Chevrolet, MarutiSuzuki, Toyota, India]

Enter the choice:
2
Enter string to insert :Honda
Enter the position :2
[Ford, Chevrolet, Honda, MarutiSuzuki, Toyota, India]

Enter the choice:
3
Enter string to search :Ford
Index : 0
[Ford, Chevrolet, Honda, MarutiSuzuki, Toyota, India]

Enter the choice:
4
[Ford, Chevrolet, Honda, MarutiSuzuki, Toyota, India]

Enter the choice:
5
Enter starting letter of the strings :F
Ford
[Ford, Chevrolet, Honda, MarutiSuzuki, Toyota, India]
```

```
Enter the choice:
6
Enter substring :Suzuki
MarutiSuzuki

Enter the choice:
7
[Chevrolet, Ford, Honda, India, MarutiSuzuki, Toyota]

Enter the choice:
8
Enter string to remove :Chevrolet
[Ford, Honda, India, MarutiSuzuki, Toyota]

Enter the choice:
9
Enter string that needs to be replaced :India
Enter new string:Mahindra
[Ford, Honda, India, MarutiSuzuki, Toyota]

Enter the choice:
10
[Ford, Honda, India, MarutiSuzuki, Toyota]

Enter the choice:
-1
PS C:\Rohith\Backup\Desktop\SEM 3\OOP-Java\Java programs\Lab programs\Exercise-8>
```

Ex-no:8 Name: M.Rohith
Date:15-12-2022 3122 21 5001 085

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**

Program code:

```
import java.util.*;

public class Program2 {
    public static void main(String[] args) {
        // Create two lists of integers
        ArrayList<Integer> alist1 = new ArrayList<Integer>();
        ArrayList<Integer> alist2 = new ArrayList<Integer>();

        // Add some elements to the first list
        alist1.add(10);
        alist1.add(20);
        alist1.add(30);
        alist1.add(40);

        // Add some elements to the second list
        alist2.add(30);
        alist2.add(40);
        alist2.add(50);
        alist2.add(60);

        // Merge the two lists
        ArrayList<Integer> merge = new ArrayList<Integer>(alist1);
        merge.addAll(alist2);
        System.out.println("Merged list: " + merge);

        // Find the union of the two lists
        ArrayList<Integer> union = new ArrayList<Integer>(alist1);
        union.addAll(alist2);
        System.out.println("Union of two lists: " + union);

        // Find the intersection of the two lists
        ArrayList<Integer> intersection = new
ArrayList<Integer>(alist1);
```

Name: M.Rohith
3122 21 5001 085

```
        intersection.retainAll(alist2);
        System.out.println("Intersection of two lists: " +
intersection);

        // Compare the two lists
        boolean isequal = alist1.equals(alist2);
        System.out.println("Lists are equal: " + isequal);
    }
}
```

Output:

```
PS C:\Rohith\Backup\Desktop\SEM 3\OOP-Java\Java programs\Lab programs\Exercise-8> javac Program2.java
PS C:\Rohith\Backup\Desktop\SEM 3\OOP-Java\Java programs\Lab programs\Exercise-8> java Program2
Merged list: [10, 20, 30, 40, 30, 40, 50, 60]
Union of two lists: [10, 20, 30, 40, 30, 40, 50, 60]
Intersection of two lists: [30, 40]
Lists are equal: false
PS C:\Rohith\Backup\Desktop\SEM 3\OOP-Java\Java programs\Lab programs\Exercise-8>
```

Ex-no:8 Name: M.Rohith
Date:15-12-2022 3122 21 5001 085

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**

Program code:

```
import java.util.*;
import java.util.stream.Collectors;

public class Program3 {
    public static void main(String[] args) {
        // Create a new doubly linked list of integers
        LinkedList<Integer> alist = new LinkedList<>();
        alist.add(10);
        alist.add(20);
        alist.add(30);
        alist.add(40);
        alist.add(50);
        alist.add(60);
        System.out.println("List at the begin: " + alist);

        // Insert some elements on both sides of the list
        System.out.println("Adding element 70 and element 80 to
first and last respectively");

        alist.addFirst(70);
        alist.addLast(80);

        System.out.println("List after adding elements at first and
last: " + alist);
```

Name: M.Rohith
3122 21 5001 085

```
        // Delete some elements from both sides of the list
        System.out.println("Removing the first and last element");

        alist.removeFirst();
        alist.removeLast();
        System.out.println("List after removing first and last: " +
alist);

        // Insert an element at a particular position in the list
        System.out.println("Inserting element 90 in the third
index");
        alist.add(3, 90);
        System.out.println("After adding element 90 in 3rd index: "
+ alist);

        // Delete a particular element from the list
        System.out.println("Deleting element 70 in the
DoublyLinkedlist");
        alist.remove(3);
        System.out.println("After removing 70: " + alist);

        // Search for a particular element in the list
        System.out.println("Searching for element 20 in the
DoublyLinkedlist and printing its index");

        int ind = alist.indexOf(20);
        System.out.println("Element 20 is found at index: " + ind);

        // Display the list in forward and backward order
        System.out.println("Printing in Forward and backward
order");

        System.out.println("Forward order: " + alist);
        System.out.println("Backward order: " + reverse(alist));

        // Sort the elements in the list
        System.out.println("Soring the DoublyLinkedlist ");
        alist.sort(null);
        System.out.println("Sorted list: " + alist);

        // Replace an element in the list with another element
```

Name: M.Rohith
3122 21 5001 085

```
        System.out.println("Replacing element at index 3 in the
DoubblyLinkedlist with 100");
        alist.set(3, 100);
        System.out.println("List with replacement: " + alist);

        // Remove duplicate elements from the list
        System.out.println("Removing the duplicate elements in the
DoublyLinkedlist");

        alist = (LinkedList<Integer>) removeDuplicate(alist);
        System.out.println("List without duplicates: " + alist);
    }

    // Helper method for reversing the elements in a list
    public static <T> LinkedList<T> reverse(LinkedList<T> alist) {
        LinkedList<T> reversed = new LinkedList<T>();
        for (int i = alist.size() - 1; i >= 0; i--) {
            reversed.add(alist.get(i));
        }
        return reversed;
    }

    public static <T> LinkedList<T> removeDuplicate(LinkedList<T>
alist) {
        for (int i = 0; i < alist.size(); i++) {
            for (int j = i + 1; j < alist.size(); j++) {
                if (alist.get(i).equals(alist.get(j))) {
                    alist.remove(j);
                }
            }
        }
        return alist;
    }
}
```

Output:

Ex-no:8 Name: M.Rohith
Date:15-12-2022 3122 21 5001 085

```
PS C:\Rohith\Backup\Desktop\SEM 3\OOP-Java\Java programs\Lab programs\Exercise-8> javac Program3.java
PS C:\Rohith\Backup\Desktop\SEM 3\OOP-Java\Java programs\Lab programs\Exercise-8> java Program3
List at the begin: [10, 20, 30, 40, 50, 60]
Adding element 70 and element 80 to first and last respectively
List after adding elements at first and last: [70, 10, 20, 30, 40, 50, 60, 80]
Removing the first and last element
List after removing first and last: [10, 20, 30, 40, 50, 60]
Inserting element 90 in the third index
After adding element 90 in 3rd index: [10, 20, 30, 90, 40, 50, 60]
Deleting element 70 in the DoublyLinkedlist
After removing 70: [10, 20, 30, 40, 50, 60]
Searching for element 20 in the DoublyLinkedlist and printing its index
Element 20 is found at index: 1
Printing in Forward and backward order
Forward order: [10, 20, 30, 40, 50, 60]
Backward order: [60, 50, 40, 30, 20, 10]
Soring the DoublyLinkedlist
Sorted list: [10, 20, 30, 40, 50, 60]
Replacing element at index 3 in the DoubblyLinkedlist with 100
List with replacement: [10, 20, 30, 100, 50, 60]
Removing the duplicate elements in the DoublyLinkedlist
List without duplicates: [10, 20, 30, 100, 50, 60]
PS C:\Rohith\Backup\Desktop\SEM 3\OOP-Java\Java programs\Lab programs\Exercise-8>
```

Result:

Java Program using collections framework has been implemented and executed successfully.