Ex. 5 **OPERATIONS ON ARRAY LIST**

Date: 16-08-2024

AIM:

To create a Java program to perform string operations using an ArrayList.

ALGORITHM:

1. Create an instance of the methods class and a Scanner for user input.
2. Display the menu and prompt the user for a choice.
3. Based on the input, perform the required operation on the ArrayList.

PROGRAM:

```
package Lab5;
import java.util.*;
public class methods {
   ArrayList<String> list = new ArrayList<String>();

   // Method to Append a String
   public void append(String value) {
      list.add(value);
   }

   // Method to Insert a String at Any Valid Position
   public int insertAtPosition(String value, int position) {
      if (position > list.size()) {
         return 0;
      }
      list.add(position, value);
      return 1;
   }

   // Method to search the String
   public boolean search(String value) {
      return list.contains(value);
   }
```

```
    // Method to Display All Elements Starting in the Character
    public void displayElementsStartingWith(char value) {
        for (String i : list) {
            if (i.charAt(0) == value) {
                System.out.println(i);
            }
        }
    }

    // Method to Sort Elements in Ascending Order
    public void ascendingSort() {
        Collections.sort(list);
    }

    // Method to Sort Elements in Descending Order
    public void descendingSort() {
        Collections.sort(list, Collections.reverseOrder());
    }
    // Method to Display all Elements
    public void displayElements() {
        for (String i : list) {
            System.out.println(i);
        }
    }
}
```

```
// Main Class
package Lab5;

import java.util.*;

public class lab5 {
    // Main Method
    public static void main(String[] args) {
        methods operations = new methods();
        Scanner input = new Scanner(System.in);
        String val;
        boolean loopController = true;

        while (loopController) {
            System.out.println("Hello!");
            System.out.println("1. Append");
            System.out.println("2. Insert at Particular Index");
            System.out.println("3. Search");
            System.out.println("4. List All Strings Starting With The
Given Letter");
            System.out.println("5. Sort");
            System.out.println("6. Display all Elements");
            System.out.println("7. Exit");
            int choice = input.nextInt();
```

```
            switch (choice) {
                case 1:
                    System.out.println("Enter the String to Append:");
                    val = input.next();
                    operations.append(val);
                    System.out.println("Insertion Successful.");
                    break;

                case 2:
                    System.out.println("Enter the String to Add:");
                    val = input.next();
                    System.out.println("Enter the Position (Zero Index
Based):");
                    int position = input.nextInt();
                    int flag = operations.insertAtPosition(val, position);
                    if (flag == 0) {
                        System.out.println("Error while Inserting Value.");
                    } else {
                        System.out.println("Insertion Successful.");
                    }
                    break;

                case 3:
                    System.out.println("Enter the String to Search:");
                    val = input.next();
```

```
    if (operations.search(val))
        System.out.println("String Found in List.");
    else
        System.out.println("String Not Found in List");
    break;

case 4:
    System.out.println("Enter the Character to Search:");
    char character = input.next().charAt(0);
    operations.displayElementsStartingWith(character);
    break;

case 5:
    System.out.println("Enter:\n1. Ascending Order");
    System.out.println("2. Descending Order");
    int orderChoice = input.nextInt();
    switch (orderChoice) {
        case 1:
            operations.ascendingSort();
            System.out.println("Ascending Sort Successful.");
            break;

        case 2:
            operations.descendingSort();
            System.out.println("Descending Sort Successful.");
```

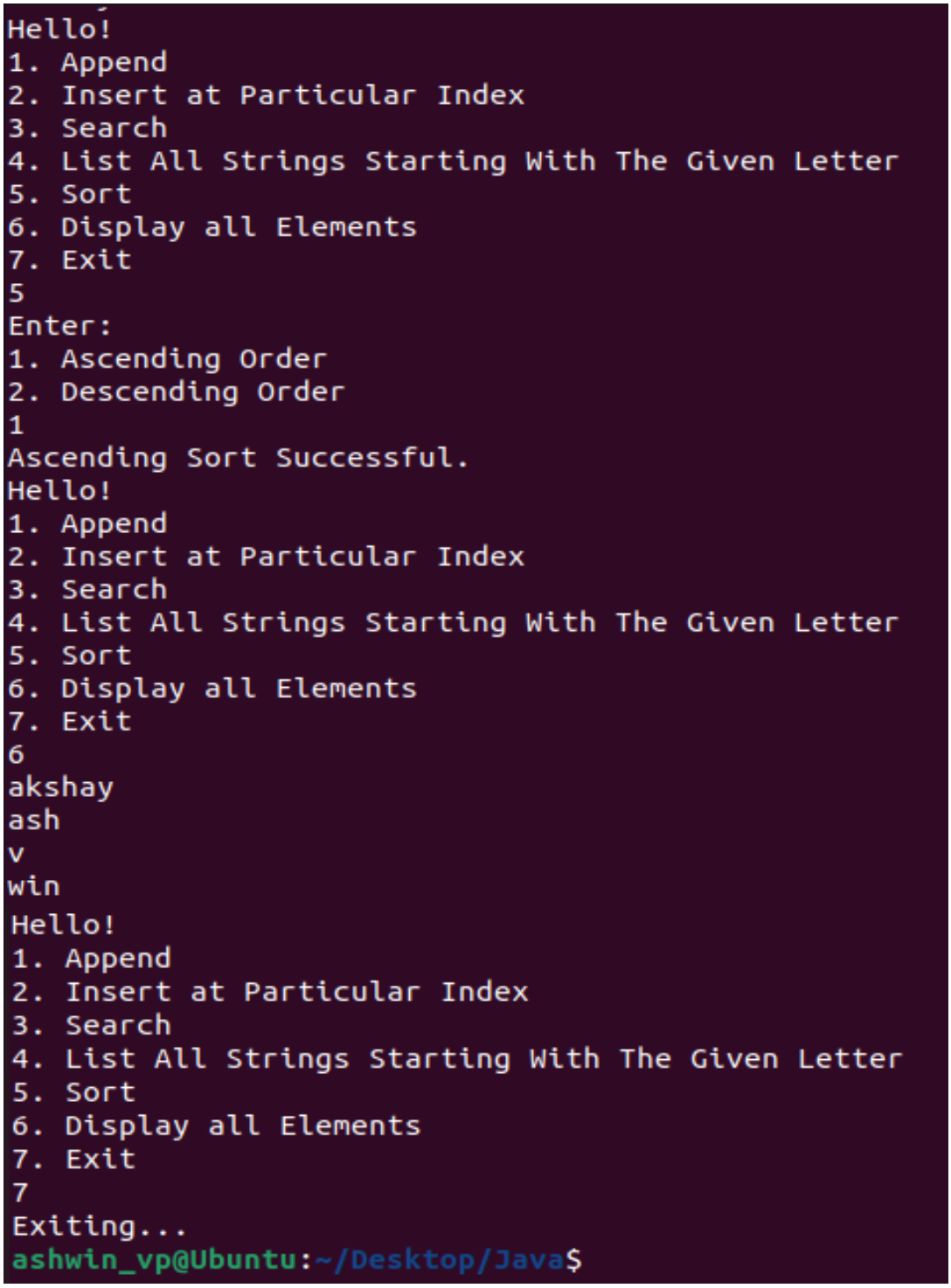
```
                    break;

                default:
                    System.out.println("Invalid Input..");
                    break;
            }
            break;

        case 6:
            operations.displayElements();
            break;

        case 7:
            System.out.println("Exiting...");
            loopController = false;
            break;

        default:
            System.out.println("Invalid Input..");
            break;
    }
  }
 }
}
```

OUTPUT:

```
ashwin_vp@Ubuntu:~$ cd Desktop
ashwin_vp@Ubuntu:~/Desktop$ javac Java/Lab5/*.java
ashwin_vp@Ubuntu:~/Desktop$ cd Java
ashwin_vp@Ubuntu:~/Desktop/Java$ java Lab5.lab5
Hello!
1. Append
2. Insert at Particular Index
3. Search
4. List All Strings Starting With The Given Letter
5. Sort
6. Display all Elements
7. Exit
1
Enter the String to Append:
ash
Insertion Successful.
Hello!
1. Append
2. Insert at Particular Index
3. Search
4. List All Strings Starting With The Given Letter
5. Sort
6. Display all Elements
7. Exit
1
Enter the String to Append:
win
Insertion Successful.
Hello!
1. Append
2. Insert at Particular Index
3. Search
4. List All Strings Starting With The Given Letter
5. Sort
6. Display all Elements
7. Exit
6
ash
win
Hello!
```

```
Hello!
1. Append
2. Insert at Particular Index
3. Search
4. List All Strings Starting With The Given Letter
5. Sort
6. Display all Elements
7. Exit
3
Enter the String to Search:
ash
String Found in List.
Hello!
1. Append
2. Insert at Particular Index
3. Search
4. List All Strings Starting With The Given Letter
5. Sort
6. Display all Elements
7. Exit
3
Enter the String to Search:
ashwin
String Not Found in List
Hello!
1. Append
2. Insert at Particular Index
3. Search
4. List All Strings Starting With The Given Letter
5. Sort
6. Display all Elements
7. Exit
4
Enter the Character to Search:
a
ash
akshay
```

```
Hello!
1. Append
2. Insert at Particular Index
3. Search
4. List All Strings Starting With The Given Letter
5. Sort
6. Display all Elements
7. Exit
5
Enter:
1. Ascending Order
2. Descending Order
2
Descending Sort Successful.
Hello!
1. Append
2. Insert at Particular Index
3. Search
4. List All Strings Starting With The Given Letter
5. Sort
6. Display all Elements
7. Exit
6
win
v
ash
akshay
Hello!
1. Append
2. Insert at Particular Index
3. Search
4. List All Strings Starting With The Given Letter
5. Sort
6. Display all Elements
7. Exit
8
Invalid Input..
```



RESULT:

Thus, a Java Application to perform operations on an ArrayList has be created.