

Tutorial No. 7

Title: Implementation of ArrayList

Batch:SY-IT(B3)**Roll No.:16010423076****Tutorial No.:7****Aim:** To implement ArrayList

Resources needed: Java SDK

Theory:**Basics of ArrayList**

ArrayList is a data structure in Java. It keeps properties of an array but govern by List data structure.

Purpose of ArrayList

To implement mutable array structure, this can expand or shrink as per the requirements.

It is also called resizable array which is defined in java.util package.

Syntax: Refer : https://www.w3schools.com/java/java_arraylist.asp

```
import java.util.ArrayList; // import the ArrayList class

ArrayList<String> cars = new ArrayList<String>(); // Create an
ArrayList object
```

Adding items to arrayList

```
import java.util.ArrayList;

public class Main {
    public static void main(String[] args) {
        ArrayList<String> cars = new ArrayList<String>();
        cars.add("Volvo");
        cars.add("BMW");
        cars.add("Ford");
        cars.add("Mazda");
        System.out.println(cars);
    }
}
```

Get() and Set() methods in arrayList

`cars.set(0, "Opel");` //set() method is used to refer to an index element and modify its value

`cars.get(0);` // To access an element from arrayList, use the get() method

Other methods in ArrayList includes:

1. `remove()` :- To remove an element from the arrayList
2. `clear()` :- To remove all elements from an arrayList
3. `size()` :- To indicate how many elements are there in the arrayList

Looping in arrayList

```
for (int i = 0; i < cars.size(); i++) {  
    System.out.println(cars.get(i));  
}
```

Sorting in arrayList

```
for (String i : cars) {  
    System.out.println(i);  
}
```

Wrapper classes used in Java

wrapper class: Integer.

For other primitive types, use: Boolean for boolean, Character for char, Double for double

Task:

Write a Java program to create an array list and add some product names (strings) to it. Make a menu driven program to perform following tasks

1. **Print the collection.**
2. **Insert an element into the array list at the first position.**
3. **Retrieve an element (at a specified index) from a given array list.**
4. **Update an arraylist element by the given element.**
5. **Remove the third element from an array list.**
6. **Search for an element in an array list.**
7. **Sort a given array list.**

Program and Result:

```
import java.util.ArrayList;  
  
import java.util.Collections;  
  
import java.util.Scanner;  
  
  
public class NamesList {
```

```
public static void main(String[] args) {  
    ArrayList<String> names = new ArrayList<>();  
    Scanner sc = new Scanner(System.in);  
  
    int choice;  
    String element;  
    int index;  
  
    names.add("Ritesh");  
    names.add("Dev");  
    names.add("Omkar");  
  
    do {  
        System.out.println("\nMenu:");  
        System.out.println("1.Print the collection");  
        System.out.println("2.Insert an element at the first position");  
        System.out.println("3.Retrieve an element at a specified index");  
        System.out.println("4.Update an element");  
        System.out.println("5.Remove the third element");  
        System.out.println("6.Search for an element");  
        System.out.println("7.Sort the array list");  
        System.out.println("8.Exit");  
        System.out.print("Enter your choice: ");  
        choice = sc.nextInt();  
        sc.nextLine();  
  
        switch (choice) {  
            case 1:
```

```
System.out.println("Names : "+ names);
```

```
break;
```

```
case 2:
```

```
System.out.print("Enter name to add at 1st position: ");
```

```
element = sc.nextLine();
```

```
names.add(0, element);
```

```
break;
```

```
case 3:
```

```
System.out.print("Enter index to retrieve: ");
```

```
index = sc.nextInt();
```

```
if (index >= 0 && index < names.size()) {
```

```
System.out.println("Name at index " + index + ": " + names.get(index));
```

```
} else {
```

```
System.out.println("Invalid index.");
```

```
}
```

```
break;
```

```
case 4:
```

```
System.out.print("Enter index to update: ");
```

```
index = sc.nextInt();
```

```
sc.nextLine();
```

```
if (index >= 0 && index < names.size()) {
```

```
System.out.print("Enter new name: ");
```

```
element = sc.nextLine();
```

```
names.set(index, element);
```

```
} else {
```

```
System.out.println("Invalid index.");
}
break;

case 5:
if (names.size() >= 3) {
names.remove(2);
System.out.println("Third element removed.");
}
else {
System.out.println("Less than 3 elements in the list.");
}
break;

case 6:
System.out.print("Enter name to search: ");
element = sc.nextLine();
if (names.contains(element)) {
System.out.println("Names found.");
}
else {
System.out.println("Names not found.");
}
break;

case 7:
Collections.sort(names);
System.out.println("Sorted list: " + names);
```

```
break;

case 8:
    System.out.println("Exiting...");
    break;
default:
    System.out.println("Invalid choice.");
}

} while (choice != 8);
sc.close();
}
}
```

Output

Clear

```
java -cp /tmp/DXVYeP7JpC/NamesList
```

Menu:

- 1.Print the collection
- 2.Insert an element at the first position
- 3.Retrieve an element at a specified index
- 4.Update an element
- 5.Remove the third element
- 6.Search for an element
- 7.Sort the array list
- 8.Exit

Enter your choice: 1

Names : [Ritesh, Dev, Omkar]

Menu:

- 1.Print the collection
- 2.Insert an element at the first position
- 3.Retrieve an element at a specified index
- 4.Update an element
- 5.Remove the third element
- 6.Search for an element
- 7.Sort the array list
- 8.Exit

Enter your choice: 2

Enter name to add at 1st position: Akshay

Menu:

- 1.Print the collection
- 2.Insert an element at the first position
- 3.Retrieve an element at a specified index
- 4.Update an element
- 5.Remove the third element
- 6.Search for an element
- 7.Sort the array list
- 8.Exit

Enter your choice: 3

Enter index to retrieve: 2

Name at index 2: Dev

Menu:

- 1.Print the collection
- 2.Insert an element at the first position
- 3.Retrieve an element at a specified index
- 4.Update an element
- 5.Remove the third element
- 6.Search for an element
- 7.Sort the array list
- 8.Exit

Enter your choice: 4

Enter index to update: 2

Enter new name: Adit

Menu:

- 1.Print the collection
- 2.Insert an element at the first position
- 3.Retrieve an element at a specified index
- 4.Update an element
- 5.Remove the third element
- 6.Search for an element
- 7.Sort the array list
- 8.Exit

Enter your choice: 8

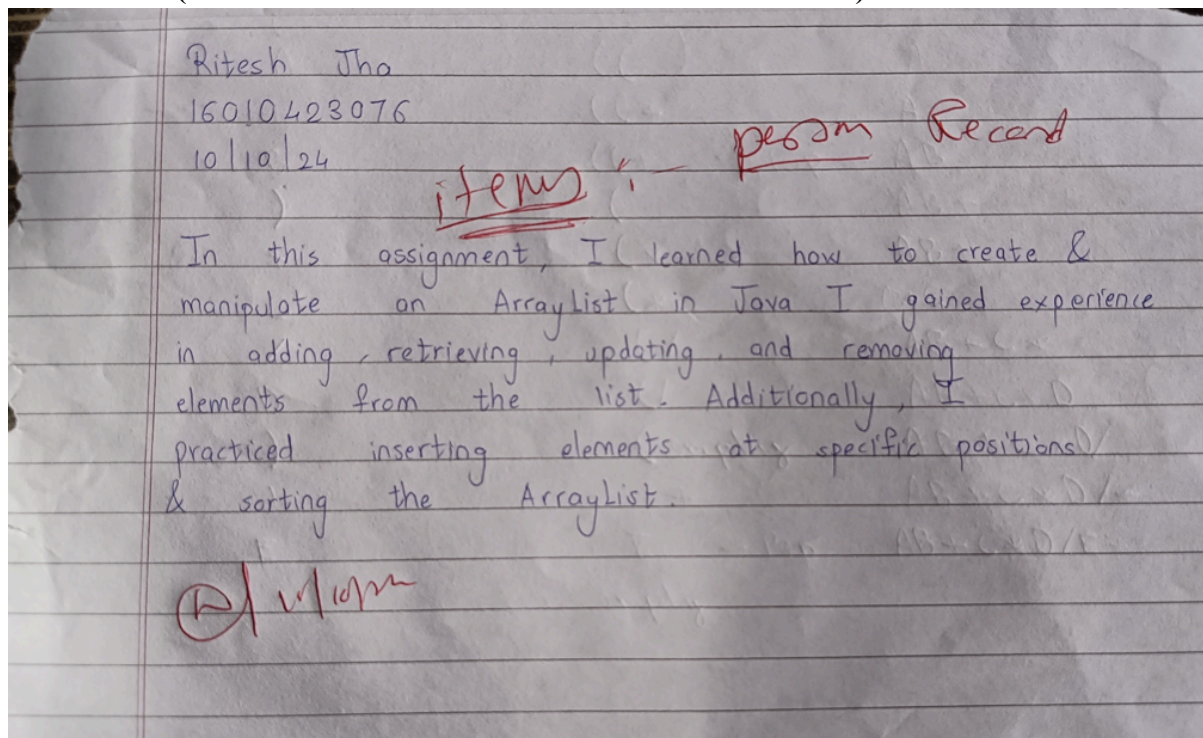
Exiting...

=== Code Execution Successful ===

Outcomes:

CO4 : Illustrate the use of collection classes, functional programming and GUI programming with java.

Conclusion: (Conclusion to be based on the outcomes achieved)



Grade: AA / AB / BB / BC / CC / CD /DD

Signature of faculty in-charge with date

References Books

1. Herbert Schildt; JAVA The Complete Reference; Seventh Edition, Tata McGraw- Hill Publishing Company Limited 2007.
2. Java 7 Programming - Black Book : Kogent Learning Solutions Inc.
3. Sachin Malhotra, Saurabh Chaudhary "Programming in Java", Oxford University Press, 2010
4. Jaime Nino, Frederick A. Hosch, 'An introduction to Programming and Object Oriented Design using Java', Wiley Student Edition.